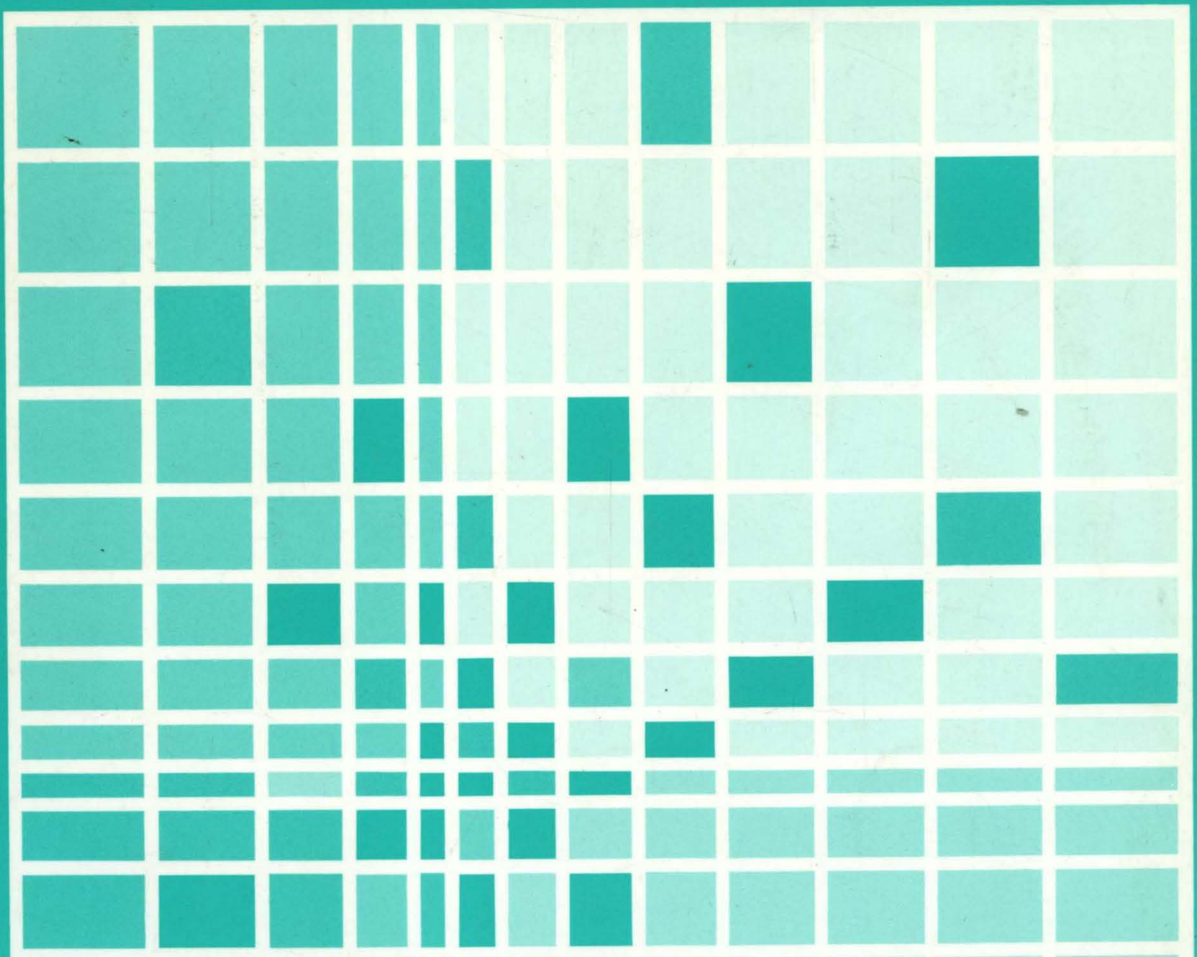
SHARP

MICROCOMPUTERS DATA BOOK



MICROCOMPUTERS DATA BOOK

Preface

As we become more and more an information-oriented society, microcomputers have come to play a major role in numerous areas of computer application. As computer-related services grow ever more sophisticated and diverse, we are faced with a growing demand for microcomputers using most advanced technology.

To keep pace with this rapid progress, we at SHARP will continue to direct our efforts at understanding the crucial trends of the moment in this area and supply our customers with products that truly meet their needs in short to contribute to a better life for all of us in this age of expanding technology.

SHARP has developed a wide range of 4-bit, 8-bit and 16-bit microcomputer units which have numerous areas of computer-related applications from home and consumer appliances to office and industrial equipment.

This databook has been especially compiled for the use of our customers. Listed here is the entire range of microcomputer products developed and manufactured by SHARP, with detailed explanations of their many functions and outstanding features. We hope that you find this book useful in determining which SHARP products are best suited to your needs. Please contact us directly if you have any further questions.

Notice

Specifications contained in this databook are current as of the publication dated September, 1990.

SHARP reserves the right to make changes in the circuitry or specifications described herein at any time without notice in order to improve design or reliability. The system configuration examples described herein are just intended for LSI evaluation; the external circuit configuration, constants and other related conditions must be studied for application to an actual system. The information in this databook has been carefully checked to be accurate, however, SHARP makes no warranty for any errors which may appear in this document. Contact SHARP to obtain the latest version of device specifications before placing your order.

SHARP makes no representations that circuitry described herein is free from infringement of patent or other rights of third parties which may result from its use. No license is granted by implication under any patent rights or other rights of SHARP CORPORATION.

This is a newly revised 1990/91 Microcomputers Databook which can be used in place of the former editions.

Contents

Sharp's IC Data Book Family

- MICRO-COMPUTERS
  - 4-bit Single-chip Microcomputers
  - 8-bit Single-chip Microcomputers
  - Development Support Tools
  - 8-bit Microprocessors/Peripherals
  - 16-bit Microprocessors
- MEMORY
  - Dynamic RAMs
  - Static RAMs
  - EPROMs/OTPROMs
  - Mask Programmable ROMs
  - Dual Port RAMs
  - FIFO Memories
- MOS
  - Gate arrays/Standard cells
  - Display drivers, Telecommunications
  - MODEMs, CCDs/CCD peripherals
  - ICs for audio/visual equipment
  - Voice/Melody generators, ICs for clock, etc.
- BIPOLAR
  - Operational amplifiers/Comparators
  - Transistor arrays, Voltage regulators
  - A/D, D/A converters, Bus interfaces
  - ICs for audio/visual equipment
  - CCD peripherals, ICs for telephone, etc.

General Information

Alphanumeric Index

Product Lineup

■ 4-bit Single-chip Microcomputers

(1) Controller Series

ROM (bit)	RAM (bit)	Model No.	I/O	Application
0.5K	32×4	SM590	15	Compact system
0.7K	32×4	SM595	15	Compact system
1K	56×4	SM591	15	Compact system
	80×4	SM550	36	General-purpose
2K	128×4	SM551	48	General-purpose
4K	192×4	SM578	52	Built-in A/D
	256×4	SM552	48	General-purpose
6K	256×4	SM579	52	Built-in A/D
	320×4	SM5E4	68	Multi-I/O
		LU5E4POP	68	Built-in OTPROM
8K	256×4	SM5J6	52	Built-in A/D

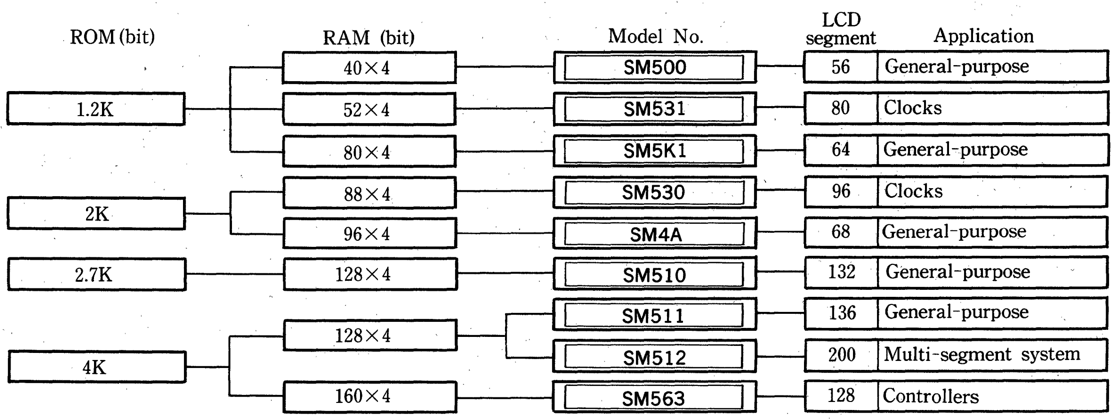
(2) VFD Driver Series

			VFD segment	
8K	256×4	SM5J5	160 (16×10)	Controllers

Model No.	Memory(bit)		Port				A/D conver-sion	Instruction cycle (μs)	Current consumption		OSC	Supply voltage (V)	Operating temp. (°C)	Package	Evaluation board	Ramarks	Page
	ROM	RAM	I	O	I/O	Total			Operating (mA)	Standby (μA)							
SM590	508×8	32×4	—	—	15 (MAX)	15	—	1	1	1	Ceramic Resistor	2.5 to 5.5	−10 to 70	16DIP 18DIP 20DIP 18MFP *3	LU590H2A		26
SM595	762×8	32×4															
SM591	1016×8	56×4															
SM550	1024×8	80×4	4	8	24	36	—	1.6	1	50	Resistor	2.7 to 5.5	−20 to 70	48QFP	LU550H2A	SIO (8-bit)	36
SM551	2048×8	128×4	4	16	28	48	—	1.6	1	50	Rasistor	2.7 to 5.5	−20 to 70	60QFP			
SM552	4096×8	256×4												64SDIP			
SM578	4064×9	192×4	9	2	41	52	8-bit 20 pins	2	1.6	1	Caramic	2.7 to 5.5	−10 to 70	64QFP 64SDIP	LU578H2A	SIO (8-bit)	49
SM579	6096×9	256×4															
SM5E4	6144×8	320×4	4	16	48	68	—	1.6	1	50	Resistor	2.7 to 5.5	−10 to 70	80QFP	LU5E3H2	SIO (8-bit)	65
LU5E4P0P	6144×8 *1	320×4							15	120		4.5 to 5.5					75
SM5J6	8192×9	256×4	9	12	31	52	8-bit 10 pins	2	5	10	Caramic	2.7 to 5.5	−10 to 80	64QFP 64SDIP	LU5J5H2	SIO (8-bit)	83
SM5J5	8192×9	256×4	9	12	31	52	8-bit 10 pins	2.5	3 *2	—	Caramic Resistor	4.5 to 5.5	−10 to 70	64QFP 64SDIP	LU5J5H2	SIO (8-bit) Medium power output −40V	

*1 OTPROM
*2 V_{DSP} open
*3 Applicable to SM595, SM591

(3) LCD Driver Series



Model No.	Memory(bit)		Port				LCDdrive		Instruction cycle (μs)	Current consumption		OSC	Supply voltage (V)	Operating temp. (℃)	Package	Evaluation board	Remarks	Page
	ROM	RAM	I	O	I/O	Total	Segment	Duty ratio		Operating (μA)	Standby (μA)							
SM530	2016×8 256×6	64×4 24×4	8	58	—	66	48×2	1/2 duty 1/2 bias	91.6	12	1.5	Crystal	1.5	−10 to 60	80QFP	LU530H2A	Built-in melody generator	100
SM531	1260×8 128×7	32×4 20×4	6	42	—	48	40×2	1/2 duty 1/2 bias	91.6	10	1.5	Crystal	1.5	−10 to 60	60QFP	LU530H2A	Built-in melody generator	110
SM500	1197×8	40×4	6	26	8*2	40	28×2 (MAX)	1/2 duty 1/2 bias	61	20	3	Crystal	3	−20 to 70	48QFP	LU500H2A		119
SM5K1	1280×8	64×4 16×4	6	25	8	39	16×4	1/4 duty 1/3 bias	5	1000 (5V)	5	*3	2.4 to 5.5	−20 to 70	48QFP 42SDIP	LU5K1H2A	LED direct drive 1/3 duty	128
SM4A	2268×8	96×4	6	40	4	50	34×2	1/2 duty 1/2 bias	61	50	10	Crystal	3	−5 to 55	60QFP	LU041H2A		141
SM510	2772×8	96×4 32×4	6	47	—	53	33×4	1/4 duty 1/3 bias	61	40	15	Crystal	3	0 to 50	60QFP	LU510H2A		149
SM511	4032×8 256×6	96×4 32×4	6	47	—	53	34×4	1/4 duty 1/3 bias	61	45	15	Crystal	3	0 to 50	60QFP	*1	Built-in melody generator	159
SM512	4032×8 256×6	80×4 48×4	6	63	—	69	50×4	1/4 duty 1/3 bias	61	50	20	Crystal	3	0 to 50	80QFP	*1	Built-in melody generator	
SM563	4096×8	128×4 32×4	4	21	11 +15*2	51	32×4	1/4 duty 1/3 bias	2 (5V)	400 (3V)	8 (3V)	Resistor	2.7 to 5.5	−20 to 70	64QFP	LU563H2A	SIO (8-bits)	172

ROM {Upper: Program ROM / Lower: Melody ROM}
RAM {Upper: Data RAM / Lower: Display RAM}

*1 Emulation by a bread board
*2 Available for segment output
*3 Crystal or ceramic oscillator

■ 8-bit Single-chip Microcomputers

(1) Controller Series (Z8® Family/SM Series)

ROM (byte)	RAM (bit)	Model No.	Process	Application
ROMless	124×8	LH0881/A	NMOS	Controllers
		LU800V1/AV1	CMOS	Controllers
ROMless	236×8	LU805BV2	NMOS	Controllers
2K	128×8	LH0801/A	NMOS	Controllers
4K	124×8	LH0811/A	CMOS	Controllers
		SM803/A	CMOS	Controllers
8K	256×8	SM805	CMOS	Contorllers

(2) ASSP Series

ROM (byte)	RAM (bit)	Model No.	Process	Application
10K	256×8	SM8202/SM8203	CMOS	VCRs
12K	512×8	SM8320	CMOS	Inverter air conditioners

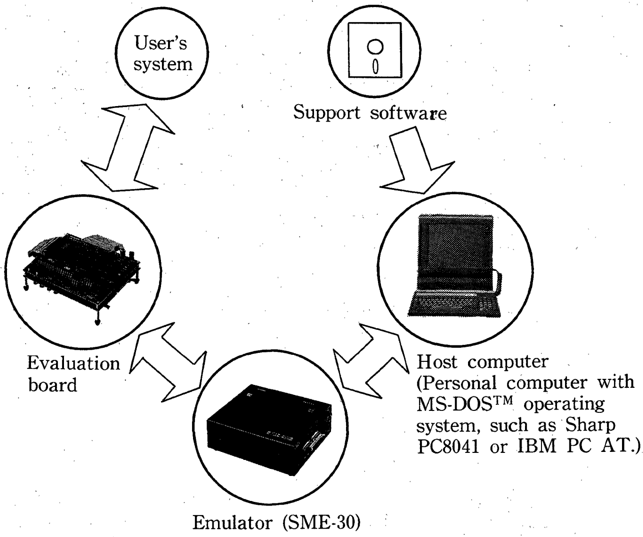
Model No.	Process	Memory(bit)		External memory (bit)	Parl				Sub-routine	Inter-rupt	In-struc-tion sel	Instruction cycle (μS)	Current consumption		OSC	Supply voltage (V)	Operating temp. (℃)	Package	Evaluation board (ICE)	Remarks	Page
		ROM	RAM		I	O	I/O	Total					Operating (mA)	Slandby (μA)							
LH0881/A	NMOS	—	124×8	128K	4	4	8	16	Uses RAM area	6	231	1.5/1	180	—	Crystal	4.5 to 5.25	0 to 70	40DIP 44QFJ	LH80H321	Built-in full duplex UART Z8	203
LH0801/A	NMOS	2048×8	124×8	124K	4	4	24	32													188
LH0811/A	NMOS	4096×8	124×8	120K																	
LU800V1/AV1	CMOS	—	124×8	128K	4	4	8	16	Uses RAM area	6	233	1.5/1	15	0.3	Crystal	4.5 to 5.5	0 to 70	440FP 40DIP 44QFJ	*1 LH80H321	Built-in full duplex UART CMOSZ8	220
LU805BV2	CMOS	—	236×8	128K	4	4	8	16		6	233	0.75	15	0.3							
SM803/A	CMOS	4096×8	124×8	120K	4	4	24	32		6	233	1.5/1	15	0.3							207
SM805/A	CMOS	8192×8	236×8	112K																	
SM8202	CMOS	10240×8	256×8	—	8	16	24	48	Uses RAM area	10	64	0.8	10	—	Crystal	4.5 to 5.5	−20 to 70	64SDIP	LU8200H7	Servo controller	226
SM8203																				Increased function	
SM8320	CMOS	12288×8	256×4	—	8	6	40	54	Uses RAM area	8	81	1	—	—	Crystal Ceramic	4.5 to 5.5	−20 to 70	64SDIP 64QFP	—	*2, *3	243

*1 There is a slight difference in the I/O characteristics between the LH8DH321 and the SM800 series.
*2 Built-in PWM generator circuits
*3 Built-in A/D, SIO

Support Tools for 4-bit Single-chip Microcomputers

(1) 4-bit Single-chip Microcomputer Development Support System

The software program for 4-bit single-chip microcomputers can be developed through a simple system composed of a personal computer running on an MS-DOS^{TM} operating system serving as a host computer, and a debugging unit that includes an emulator and an evaluation board. Sharp also offers a highperformance SM emulator (SME-30).


User's system
Support software
Evaluation board
Host computer (Personal computer with MS-DOS^{TM} operating system, such as Sharp PC8041 or IBM PC AT.)
Emulator (SME-30)

MS-DOS^{TM} is a trademark of Microsoft Corporation.

(2) 4-bit Single-chip Microcomputer Development Support Tools (SME-30 System)

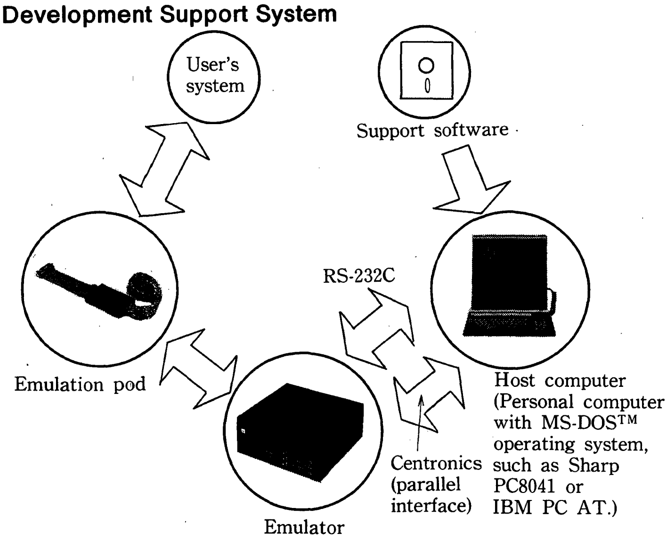
SME-30 System	SM series	Evaluation board	Evaluation card	Piggy-back	Page
● Target microcomputers: 4-bit single-chip microcomputers ● Emulator: SME-30 (LU4DH300) ● Evaluation board ● Host computer ● Optional software: Cross-assembler Emulator software PROM programmer	SM4A	**LU041H2**	LU041H4	—	254
	SM500	**LU500H2A**	LU500H4A	—	
	SM510	**LU510H2A**	LU510H4A	—	
	SM511/512	Emulation by a bread board	Emulation by a bread board	—	—
	SM530/531	**LU530H2A**	LU530H4A	—	254
	SM550/551/552	**LU550H2A**	LU550H4A	LU550H6	
	SM563	**LU563H2**	LU563H4	—	
	SM578/579	**LU578H2A**	LU578H4A	LU578H6	
	SM590/591/595	**LU590H2A**	LU590H4A	LU590H6	
	SM5E4	**LU5E3H2**	LU5E3H4	—	
	SM5J5	**LU5J5H2**	LU5J5H4A	LU5J5H6	

Support Tools for 8-bit Single-chip Microcomputers

(1) 8-bit Single-chip Microcomputer Development Support System

The software program for 8-bit single-chip microcomputers can be developed through a system consisting of a personal computer (MS-DOS™) serving as a host and in-circuit emulator tailored to each model.

MS-DOS™ is a trademark of Microsoft Corporation.


User's system
Support software
RS-232C
Emulation pod
Host computer (Personal computer with MS-DOS™ operating system, such as Sharp PC8041 or IBM PC AT.)
Centronics (parallel interface)
Emulator

(2) 8-bit Single-chip Microcomputer Development Support Tools

Model No.	Piggy-back	In-circuit emulator	Page
SM8202	LU8203H6	LU8200H7+LU8202H4	263
SM8203		LU8200H7+LU8203H4	

* The SM82 ICE (LU8200H7) with applicable emulation pods (LU820XH4) will meet each model of the SM82 series.

Support tool	Features	Page
SM82 In-circuit emulator (LU8200H7)	• 64K bytes of emulation memory • RS232C interface with the host • Instruction cycle time count • Line assembler and reverse assembler • Centronics interface • Coverage function	263

(3) 8-bit Microprocessors (Z80® Family)

Process	Product	Model No.	Function	Clock frequency (MHz)				Power consumption (mW)MAX.	Supply voltage (V)	Package	Page
				2.5	4	6	8				
CMOS	CPU	LH5080/M	Central Processing Unit	●				138	5±10%	40DIP/44QFP	268
		LH5080A/AM			●					40DIP/44QFP	
	PIO	LH5081/M	Parallel I/O Unit	●				44	5±10%	40DIP/44QFP	274
		LH5081A/AM			●					40DIP/44QFP	
		LH5081B				●				40DIP	
	CTC	LH5082/M	Counter/Timer Circuit	●				44	5±10%	28DIP/44QFP	279
		LH5082A/AM			●					28DIP/44QFP	
		LH5082B				●				28DIP	
NMOS	CPU	LH0080/M/U	Central Processing Unit	●				1050	5±5%	40DIP/44QFP/44QFJ	284
		LH0080A/AM/AU			●					40DIP/44QFP/44QFJ	
		LH0080B/BU				●				40DIP/44QFJ	
		LH0080E					●			40DIP	
	PIO	LH0081/M/U	Parallel I/O Unit	●				525	5±5%	40DIP/44QFP/44QFJ	309
		LH0081A/AM/AU			●					40DIP/44QFP/44QFJ	
		LH0081B/BU				●				40DIP/44QFJ	
		LH0081E					●			40DIP	
	CTC	LH0082/M/U	Counter/Timer Circuit	●				630	5±5%	28DIP/44QFP/44QFJ	319
		LH0082A/AM/AU			●					28DIP/44QFP/44QFJ	
		LH0082B/BU				●				28DIP/44QFJ	
		LH0082E					●			28DIP	
	DMA	LH0083	Direct Memory Access	●				1050	5±5%	40DIP	329
		LH0083A			●					40DIP	
	SIO	LH0084/85/86	Serial I/O Unit	●				525	5±5%	40DIP	344
		LH0087M/U		●						40QFP/44QFJ	
		LH0084A/85A/86A			●					40DIP	
		LH0087AM/AU			●					40QFP/44QFJ	
		LH0084B/85B/86B				●				40DIP	
		LH0087BU				●				44QFJ	
	SCC*1	LH8530P/U	Serial Communications Controller		●			1313	5±5%	40DIP/44QFJ	354
		LH8530AP/AU				●		1470		40DIP/44QFJ	

*1 Z8500™ family

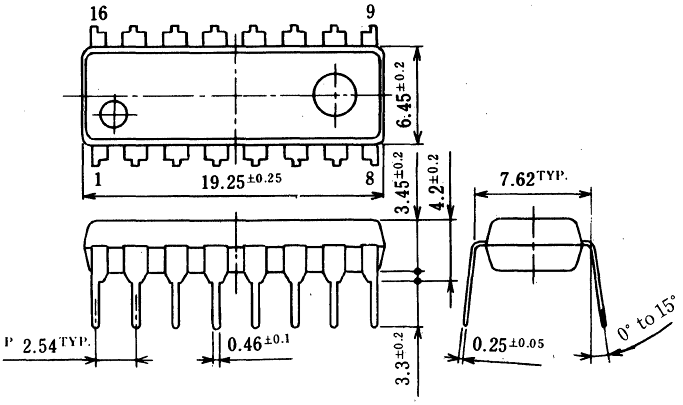
(4) 16-bit Microprocessors (V Series)

Process	Product	Model No.	Function	Clock frequency (MHz)		Power consumption (mW) MAX.	Supply voltage (V)	Package	Page
				5	8				
CMOS	V20*1	LH70108-5	16-bit Microprocessors (V Series)	●		420	5±5%	40DIP	370
		LH70108-8			●			40DIP	
	V30*1	LH70116-5		●				40DIP	414
		LH70116-8			●			40DIP	

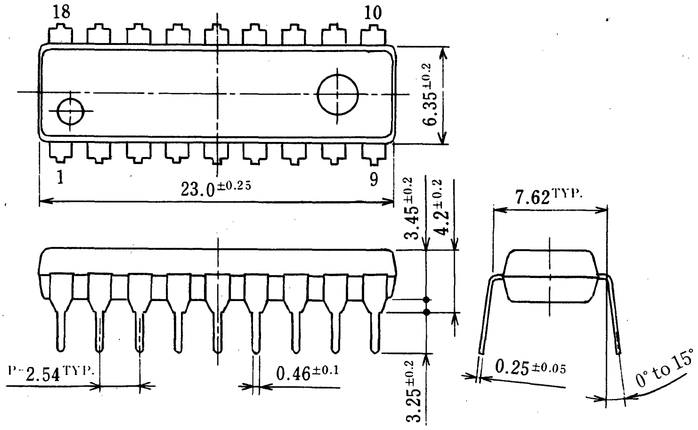
*1 V20/V30 is a trademark of NEC Corporation.

Package Outline

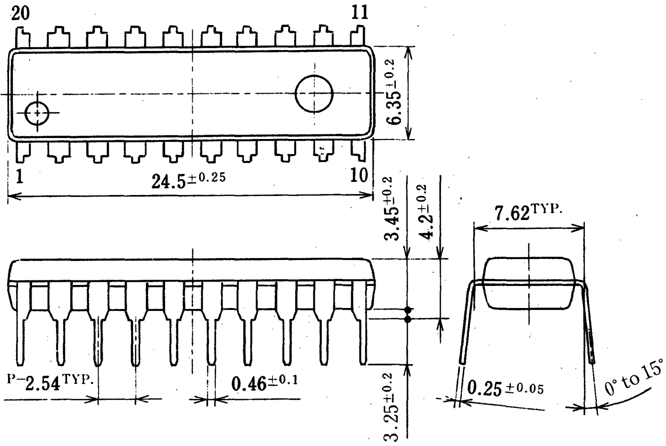
16DIP (DIP16-P-300)



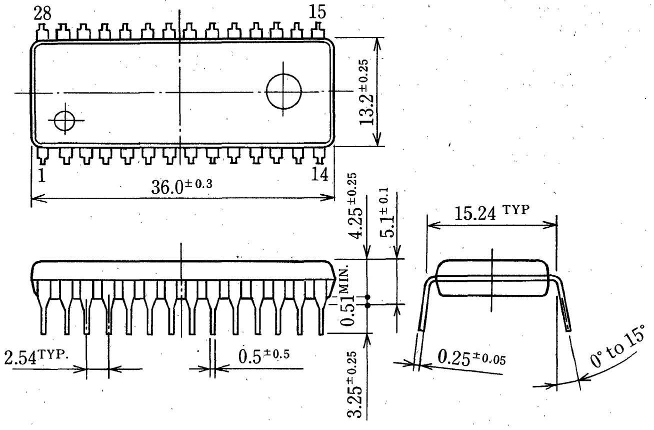
18DIP (DIP18-P-300)



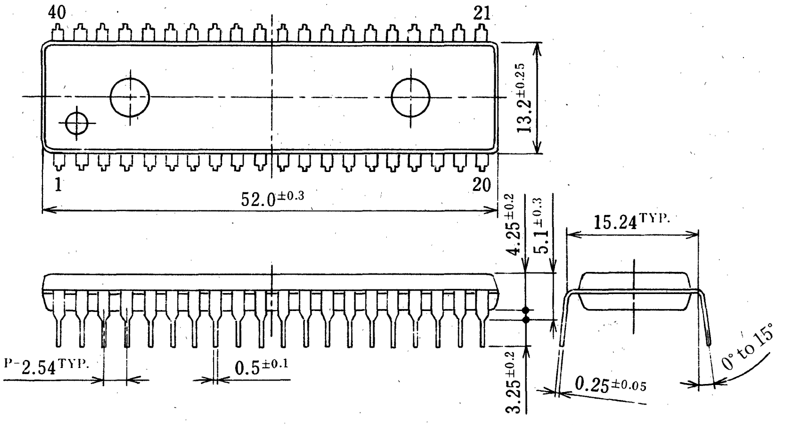
20DIP (DIP20-P-300)



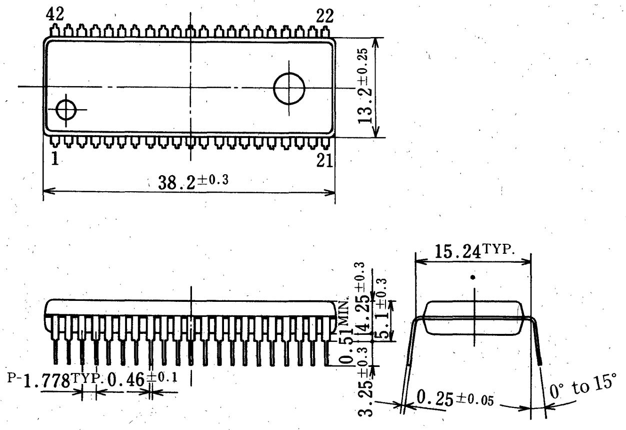
28DIP (DIP28-P-600)



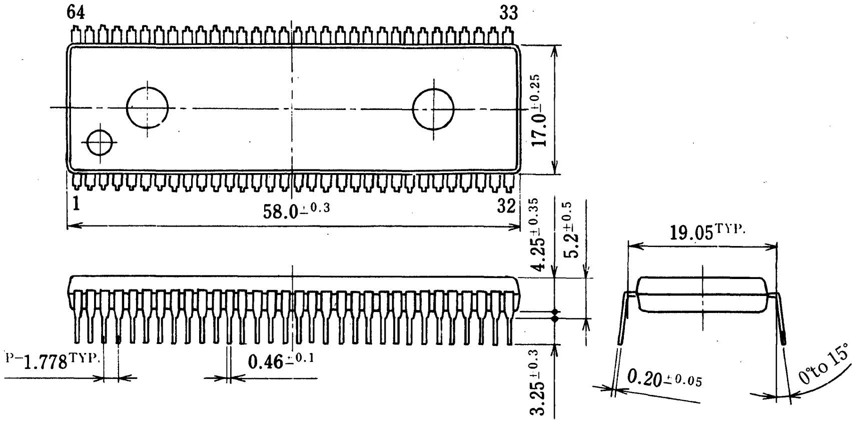
40DIP (DIP40-P-600)



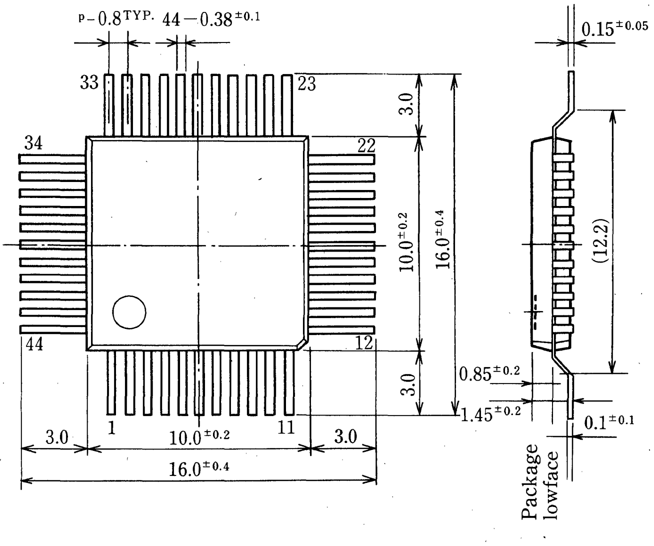
42SDIP (SDIP42-P-600)



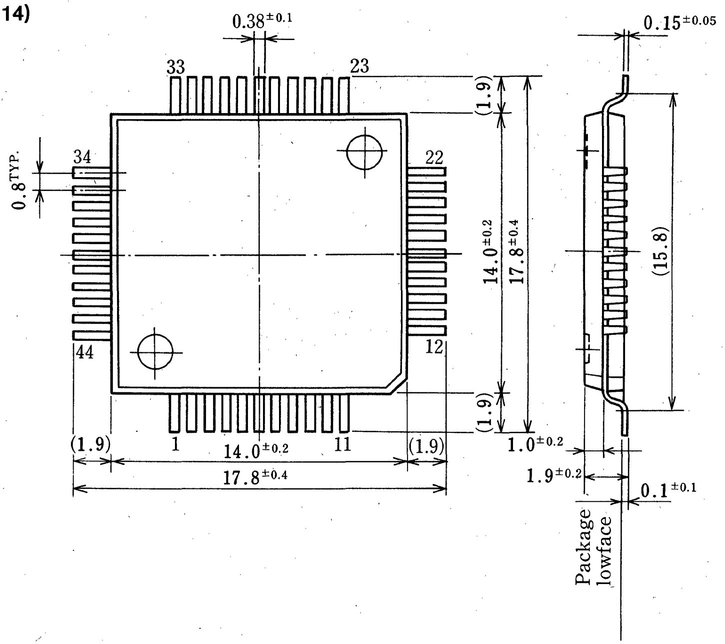
64SDIP (SDIP64-P-750)



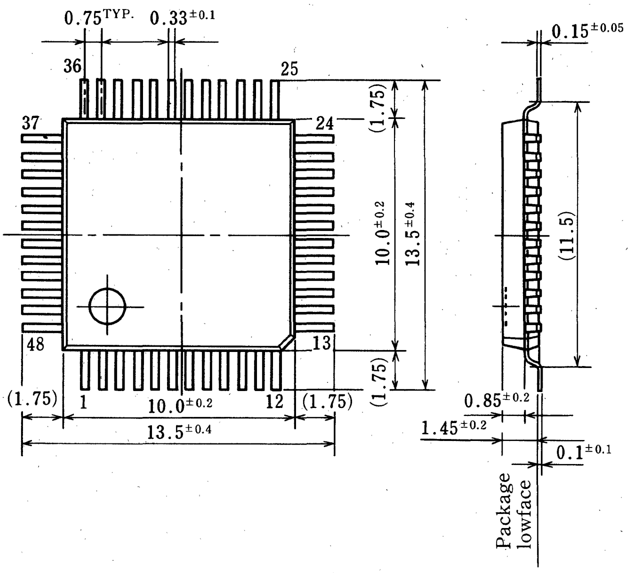
44QFP (QFP44-P-1010A)



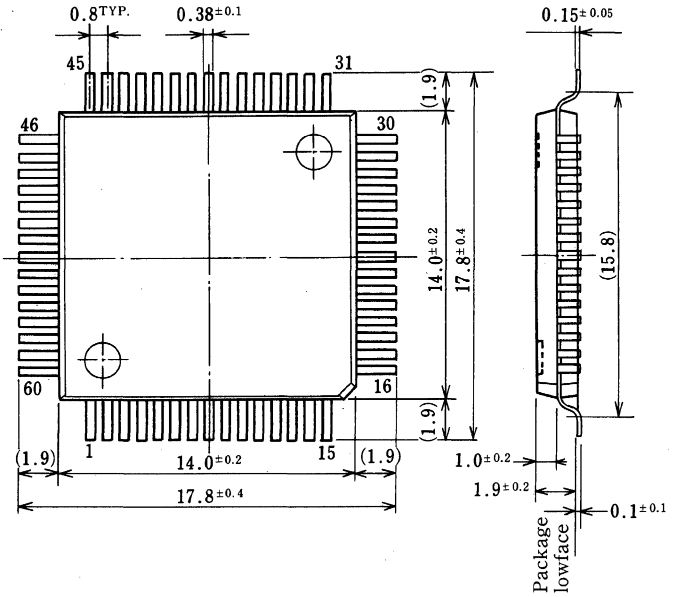
44QFP (QFP44-P-1414)



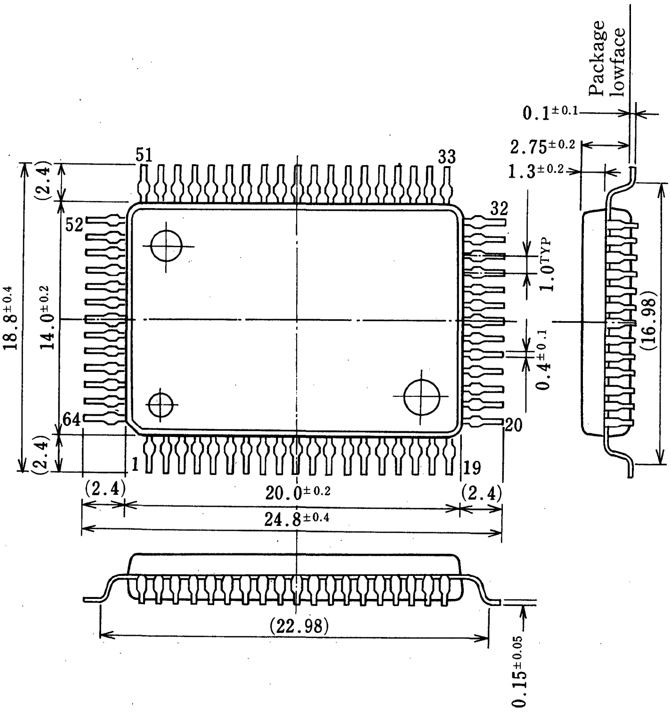
48QFP (QFP48-P-1010)



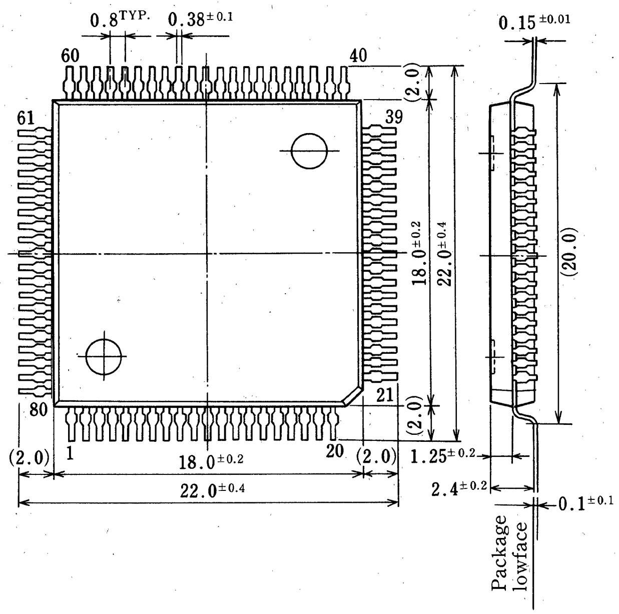
60QFP (QFP60-P-1414)



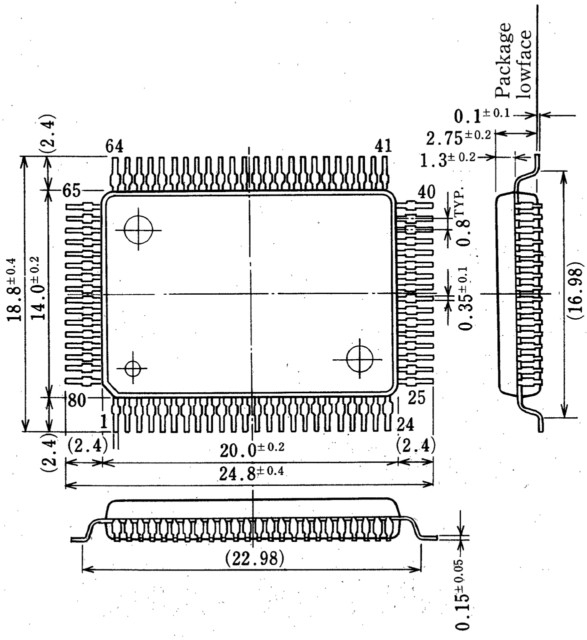
64QFP (QFP64-P-1420)



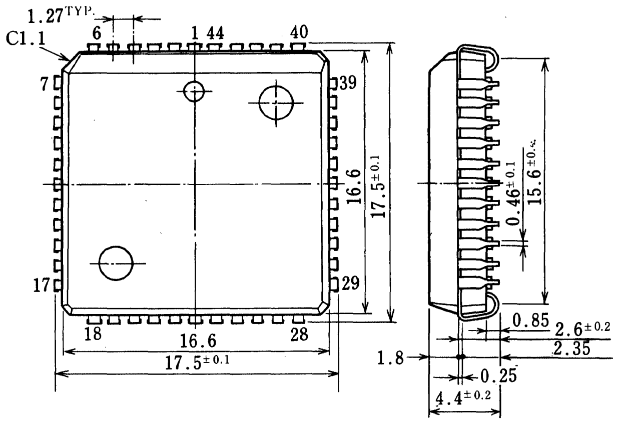
80QFP (QFP80-P-1818)



80QFP (QFP80-P-1420)



44QFJ (QFJ44-P-S650)



Quality Assurance

1. Quality Assurance System

Sharp develops and manufactures a wide range of consumer and industrial-use semiconductor products.

In recent years, the applications of ICs have expanded significantly, into fields where extremely high levels of quality are critical.

In response, Sharp has implemented a total quality assurance system that encompasses the entire production process from planning to after-sales service. This system ensures that reliability is a priority in the planning and manufacturing stages, and guarantees product quality through rigorous reliabirity testing. We will introduce a part of this system here.

Sharp's quality and reliability assurance activities are based on the following guidelines:

(1) All personnel should participate in quality assurance by continually cultivating a higher level of quality awareness.

(2) In the developmental stage of new products, create designs that consider reliability in every respect.

(3) In addtion to quality control in all manufacturing processes, all working environments, materials, equipment, and measuring devices should be carefully monitored to ensure quality and reliability from the very begining of the process.

(4) Confirm long-term reliability and obtain a thorough understanding of practical limits through reliabilty tesing.

(5) Continually work to improve quality through application of data from process inspections, reliabilty testing, and market surveys.

2. Quality and Reliability Contol in New Product Development

The development of new products begins with a thorough understanding of the product specifications and quality that will satisfy the purpose for which the product is intended and with developmental planning that carefully considers pricing, quantity, the time of introduction to the market and the target reliability.

In the design stage, reliability is designed into the product based on test data, process capability, and field data, and experimental models are made. These trial products are referred to as TS (technical samples), and are evaluated primarily for their ability to function and their perfomance.

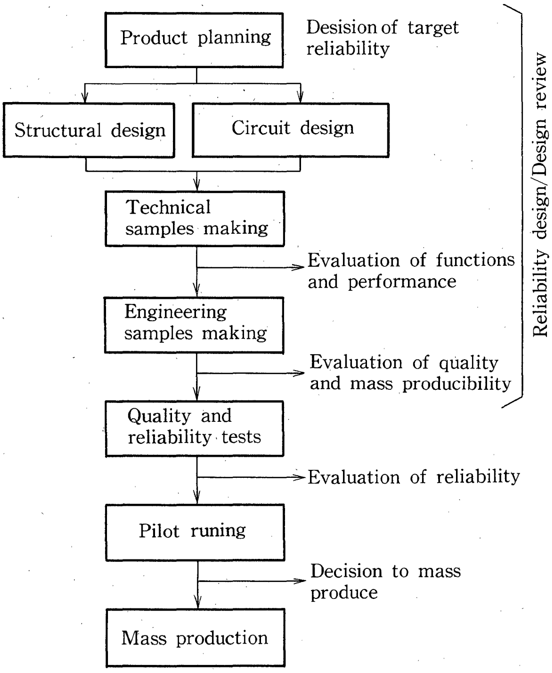

Product planning
Desision of target reliability
Structural design
Circuit design
Technical samples making
Evaluation of functions and performance
Engineering samples making
Evaluation of quality and mass producibility
Quality and reliability tests
Evaluation of reliability
Pilot runing
Decision to mass produce
Mass production
Reliability design/Design review

Fig. 1 New Product Development Steps

Next, ES (engineering samples) are made and evaluated to detemine whether the functions, performance and quality aimed for in the disign stage can be guaranteed under the existing manufacturing conditions. These ES are also evaluated in quality and reliability tests to determine whether their long-term reliability can be guaranteed.

At the final stage, the availability for mass-production will be deliberated based upon the evaluation result of TS and ES. After transition to the massproduction step, pilot production will be performed to confirm the quality and reliability obtained on the way of designing, and variations in the process. It will be judged whether or not mass-production is available according to the result.

DR (Desing Review) is performed to prevent from faulty operation and to enhance the functions, usability, quality and reliability, upon completion of structural design, software design, circuit design, TS/ES evaluation and reliability tests. Fig. 1 shows the steps in the development of new products.

Process	Control items	C
Silicon wafers ▽ Purchase of material		
□ Inspection upon receipt	Appearance, dimensions, specific resistance	Eliminate items with incorrect dimensions, scratches, and crystal defects and assure resistance values.
○ Oxidation		
□ Oxidation inspection	Appearance, film thickness	Confirm the absence of pin holes and assure firm thickness.
◇ On line QC	Surface cleanliness	Check the cleanliness of surfaces.
○ Photolithography		
□ Visual inspection	Development, etching	Check the suitability of development and etching.
◇ On line QC	Wire width	Control the wire width.
○ Ion implantation		
□ Chip electrical inspection	Electrical characteristics	Elminate items with unsuitable electrical characteristics.
○ Dicing		
○ Breakage screening		
▽ □ Frame → □ Die inspection	Appearance	Confirm the absence of breakage and chips.
○ Die bonding		
▽ □ Gold wire → □ Die bonding inspection	Appearance, bond strength	Check quality of die bond.
○ Wire bonding		
□ Wire bonding inspection	Appearance, tensile strength	Check position and shape of bond and assure sufficient tensile strength.
▽ □ Mold resin → ○ Sealing/molding Monitoring	Heat, time, pressure Wire bond	Assure original shape Assure wire shape
○ Stabilized baking		
○ Lead surface finishing	Ingredient, temperature, pollution	Assure finishing quality
□ Finishing inspection Monitoring	Thickness, uniformity (soldering conditions) Plate making, plate thickness	Assure plating quality
○ Approval	Temperature, time, marking material	Assure marking quality
○ Lead cutting	Mold sharpness damage-proof	Irregular stress to resin,
↓ Forming	Mold dimensions	Dimensions

Fig. 2 Example of process quality control

3. Quality and Reliability Control in Mass Production

(1) Quality Control of Materials

The quality and reliability of a product is affected by its component materials as well as the manufacturing processes and conditions.

The quality control of purchased component materials is basically ensured by a material supplier, based upon the quality control system between SHARP and a supplier as follows.

- Selection of suppliers prior to the placement of purchase order.
- Material qualification upon receipt of new materials. (Evaluation of device quality and reliability used with new materials.)
- Regular quality meeting based upon quality information when massproduction between both parties.

The incoming inspection may be performed according to the inspection standard based upon approved specifications.

(2) Control of Manufacturing Environment

Environmental conditions in the manufacturing process—such as temperature, humidity and dust—significantly affect the finished quality of semiconductor products.

Temperature is especially critical in maintaining the accuracy of the measurements of electrical characteristics and the accuracy of various devices. Humidity control is important for the prevention of moisture penetration into a device and the prevention of static electricity. Temperature and humidity are thus strictly maintained at constant levels.

A dust-free environment is vital in the manufacture of refined semiconductor circuits, as dust can be the critical determining factor in their quality and reliability. Thus, cleanliness of everything from air conditioning equipment to work benches to work clothes and office items is carefully controlled.

Sharp is also concerned about creating an environment conductive to error-free high-precision work, and so provides background music and interior colors appropriate for specific tasks.

(3) Control of Manufacturing Equipment and Measuring Devices

Tremendous technological innovations and progress has been made in integrated circuits and in the processes and equipment by which they are produced.

To achieve even higher levels of product uniformity and quality, Sharp is continually furthering the automation of its processes, strictly managing the maintenance of its manufacturing equipment, and carefully monitoring the accuracy of all measuring devices through daily and periodic inspections.

The productive control is systematized based upon TPM (Total Productive Maintenance). Sharp is cultivating experts in productive maintenance through a self-maintenance, a planning maintenance, a repair maintenance.

The measuring device is controlled with the regular proof by an officially authorized constitution based upon national standard, in order to keep high precision.

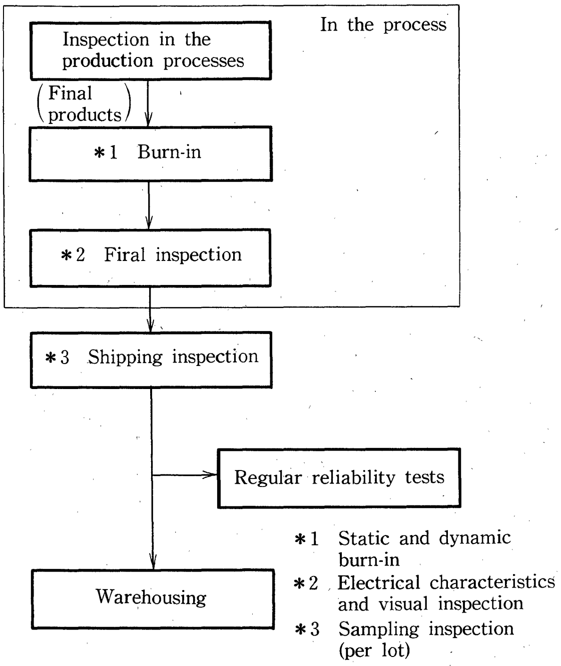

In the process
Inspection in the production processes
(Final products)
*1 Burn-in
*2 Firal inspection
*3 Shipping inspection
Regular reliability tests
Warehousing
*1 Static and dynamic burn-in
*2 Electrical characteristics and visual inspection
*3 Sampling inspection (per lot)

Fig. 3 Product inspection system

(4) Process Quality Control and Product inspections

Based on the fundamental concept of ensuring quality and reliability throughout the manufacturing process, we check at each stage to determine whether the prescribed characteristics are being obtained and to prevent defective items from going on to the next stage. We do this thorough strict monitoring, inspection of all items, sampling inspections, and other standardized methods of management.

We perform a final inspection of all finished products as well as further quality assurance inspections through sampling to fully ensure quality.

Detects found in these inspections are promptly reported to the design and production sections, and improvements made in the processes to upgrade our uniform quality capabilities.

Fig. 3 shows the product inspection system.

Fig. 4 shows an example of process quality contol.

(5) Reliability Assurance

To guarantee the long-term reliability of our products, we periodically sample products and subject them to reliability testing such as life tests and environmental tests.

These tests are long-term reliability tests and the obtained data will be given to the related sections.

The inspections and tests for quality assurance are made to maintain and enhance the quality as well as to predict the reliability of products in the market. Thus assures quality and reliability of products from many aspect.

Table 1 Reliability test for memory products

Type	Test item	Test condition	Test objectives
Life tests	High temperature storage	Ta=150℃	Evaluate resistance to high temperature in long-term storage.
	High temperature operation	Ta=125℃ or 150℃ power supply voltage (MAX.)	Evaluate resistance to long-term high temperature and electrical stress.
	High temperature, humidity storage	①85℃ 85% RH ②Pressure cooker test, 121℃ 100%RH, 15PSIG	Evaluate resistance to high temperature and humidity in long-term storage.
	High temperature, humidity bias	85℃ 85% RH Power supply voltage (MAX.)	Evaluate resistance to long-term high temperature, humidity and electrical stress.
	Low temperature storage	Ta=−65℃	Evaluate resistance to low temperature in long-term storage.
Thermal environmental tests	Temperature cycling	Tstg(MAX.)-Tstg(MIN.) −65℃ to 150℃ air	Evaluate resistance to sudden extreme temperature changes.
	Thermal shock	Tstg(MAX.)-Tstg(MIN.) −65℃ to 150℃ liq	Evaluate resistance to sudden extreme temperature changes.
	Resistance to solder heat	260℃ 10s	Evaluate resistance to thermal stress during soldering
Mechanical environment tests	Mechanical shock	1,500G, 0.5 ms ±X, ±Y, ±Z	Evaluate structural and mechanical resistance to strong shocks.
	Variable-frequency vibration	20G, 100 to 2,000 Hz, X, Y, Z	Evaluate resistance to vibration during transport and use.
	Constant accelration	20,000G±X, ±Y, ±Z	Evaluate resistance to constant acceleration.
	Lead fatigue	Lead pull: holds fixed load for 10 seconds Lead bend: bend once 90 in forward and reverse directions (Load is determined based on pin shape and the surface area of pins section.)	Evaluate resistance to mechanical stress applied to pins.
	Hermecity	Test for minute leaks using helium gas and large leaks using foaming.	Evaluate hermetic sealing.
	Salt atmosphere	Spray 5% salt solution at Ta=35℃ for 24 hours	Evaluate resistance to corrosion in salt spray environment.
	Solderibility	230℃ for 5 seconds (with flux)	Evaluate solderability of pins.

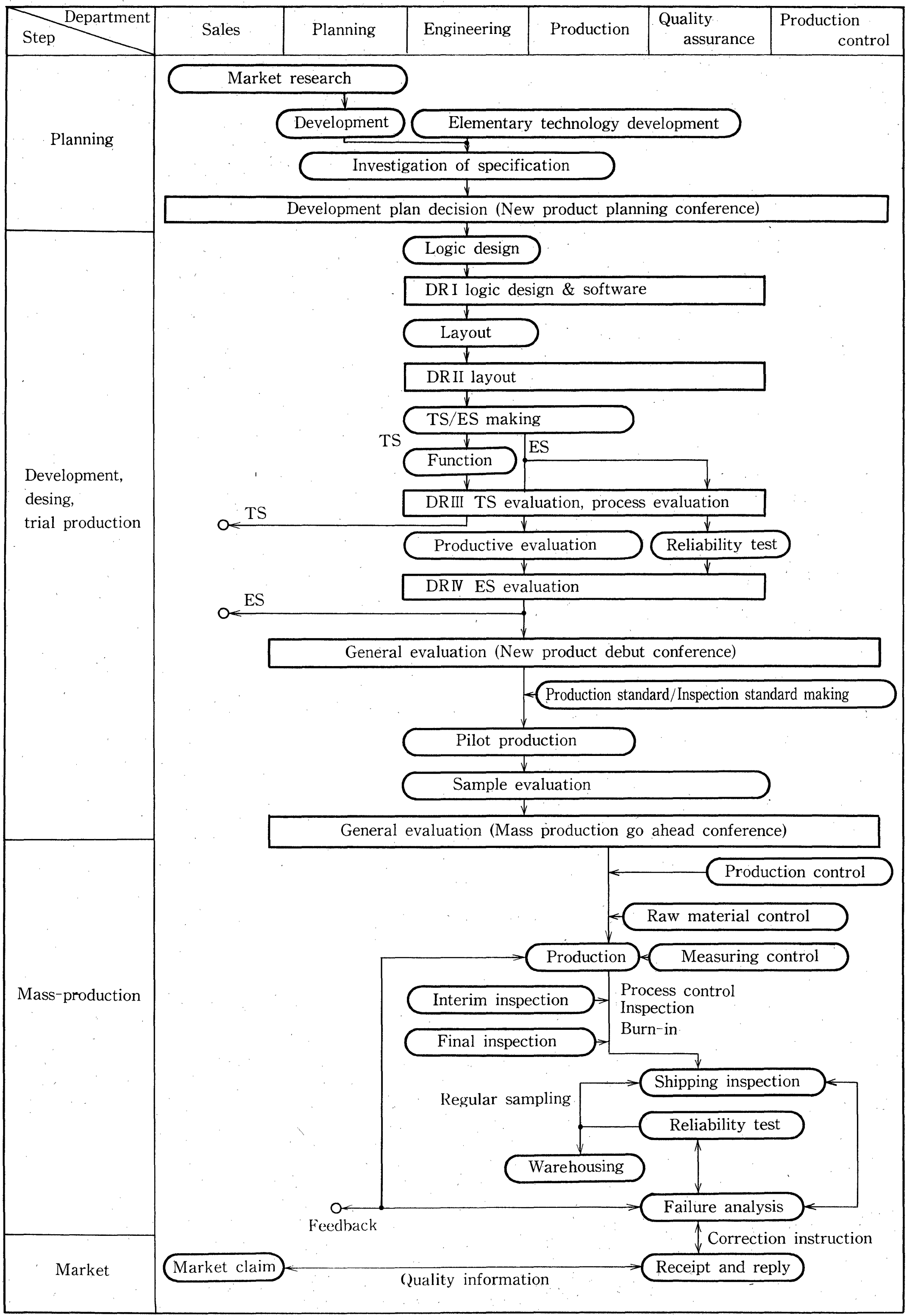


Fig. 4 Quality assurance system

4. Reliability Tests

In addition to determining the extent to which product reliability can be assured, the objectives of reliability testing include getting an understanding of design limitations and the catastrophic failure mode, and prediction reliability in the field.

The major categories of reliability testing are life tests, thermal environmental tests, and mechanical environment tests. The standardized test methods used are those prescribed by official standards or associaitions such as the International Electronics Commission (IEC), and the U. S. Military Specifications (MIL). Sharp standardizes all specifications to conform with these standerds.

Table 1 shows a representative reliability test.

5. After-sales Service

If a product malfunction after shipment, we have the customer return the product for detailed analysis. We also obtain complete information concerning conditions of use, frequency of occurrence, and symptoms.

When the cause has been determined, we report findings concerning the design, manufacturing process, or method of use to the departments concerned for preventive action against recurrence of the malfunction. We then submit a report to the customer.

This process of tracking the performance of our products in actual use is an extermely effective way to enchance product reliabilty. We direct a lot of energy forwards its full implementation.

Fig. 5 shows the quality information flowchart, and Fig. 6 shows the procedures used in their analysis.

6. Handling Precautions

All of the semiconductor products listed in this data book were manufactured based on exacting designs and under compreshensive quality control. However, to take full advantage of the features offered and assure the products' long-life service, please refer to this manual to help in designing systems that make best use of their capabilities.

(1) Maximum Ratings

It is generally known that the failure rate of semiconductor products increases as the tempera-

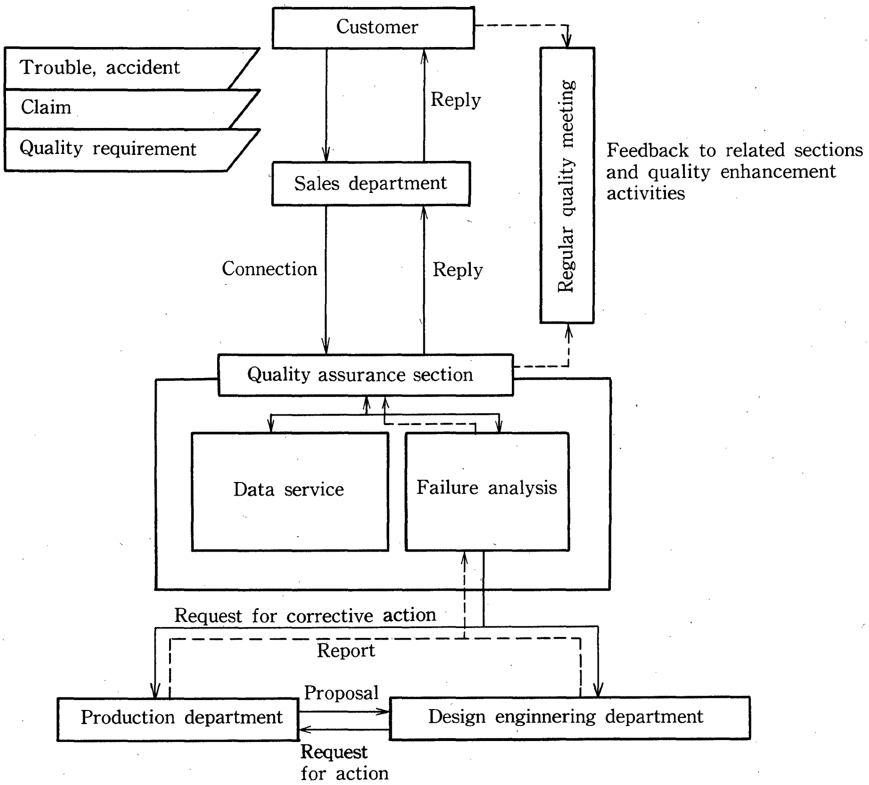


Fig. 5 Routes through which malfunctions outside the companiy are handlid

ture increases. It is necessary, of course, that the ambient temperature be within the maximum rated temperature. Further it is desirable from the standpoint of reliability that the ambient temperature be lowered as much as possible. The voltage, current, and electric power used are also factors that significantly influence the life of semiconductor products. Voltage or current that exceeds the rated level may damaged, the semiconductor product; even if applied only momentarily and the unit continues to operate properly, excessive voltage or current will likely increase the failure rate.

Therefore, in actual circuit design, it is important that the semiconductor products used have a certain degree of 1allowance with respect to the voltage, current and temperature conditions under which they will be used. The greater this allowance, the fewer the failures that will occur.

To keep failures to a minimum, the circuit should be designed so that under all conditions to absolute maximum, the ratings are not exceeded even momentarily and so that the maximum values for any two or more items are not achieved simultaneously. In addition, remember that the circuit functions of semiconductor products are guaranteed within the operating temperature range (Topr) of the absolute maximum ratings, but that storage temperature (Tstg) is the range in a nonoperating condition.

(2) Transportation and Storage

It is recommended to store semiconductor products under circumstances of normal temperature (5 to 30℃ and normal humidity (45 to 75%RH)).

The products in moisture-proof package should be stored under circumstances of 5 to 30℃ and less than 70%RH, and they should be mounted in systems immediately after unpacking.

During shipping and storage, keep semiconductor products in the packaging they were delivered in to prevent damage due to static electricity. If removed from their packaging, the terminals must be shortcircuited with a conductive material or the entire units wrapped in aluminium foil. Also remember that nylon and plastic containers build up electrostatic charges easily and so should not be used for storage or transportation.

Mechanical vibration and shock also be kept to a minimum.

(3) Assembly

When attached to printed circuit boards, semiconductor products are removed from a conductive container, so electrical equipment, work benches and operators must be grounded to protect the products from static electricity. It is good to use grounded metal plating on the surfaces of work benches. Grounding metal rings and watch bands is a convenient method for grounding operators. The grounding of operators is required to prevent electric shock due to current leaks from electrical equipment, so it must be performed through a resistance of 1 MΩ.

Working attire made of synthetic fabrics should be avoided in favor of fabrics such as cotton that do not easily genarate static electricity.

Keeping the relative humidity in working areas around 50% will also help to prevent the generation of static electricity.

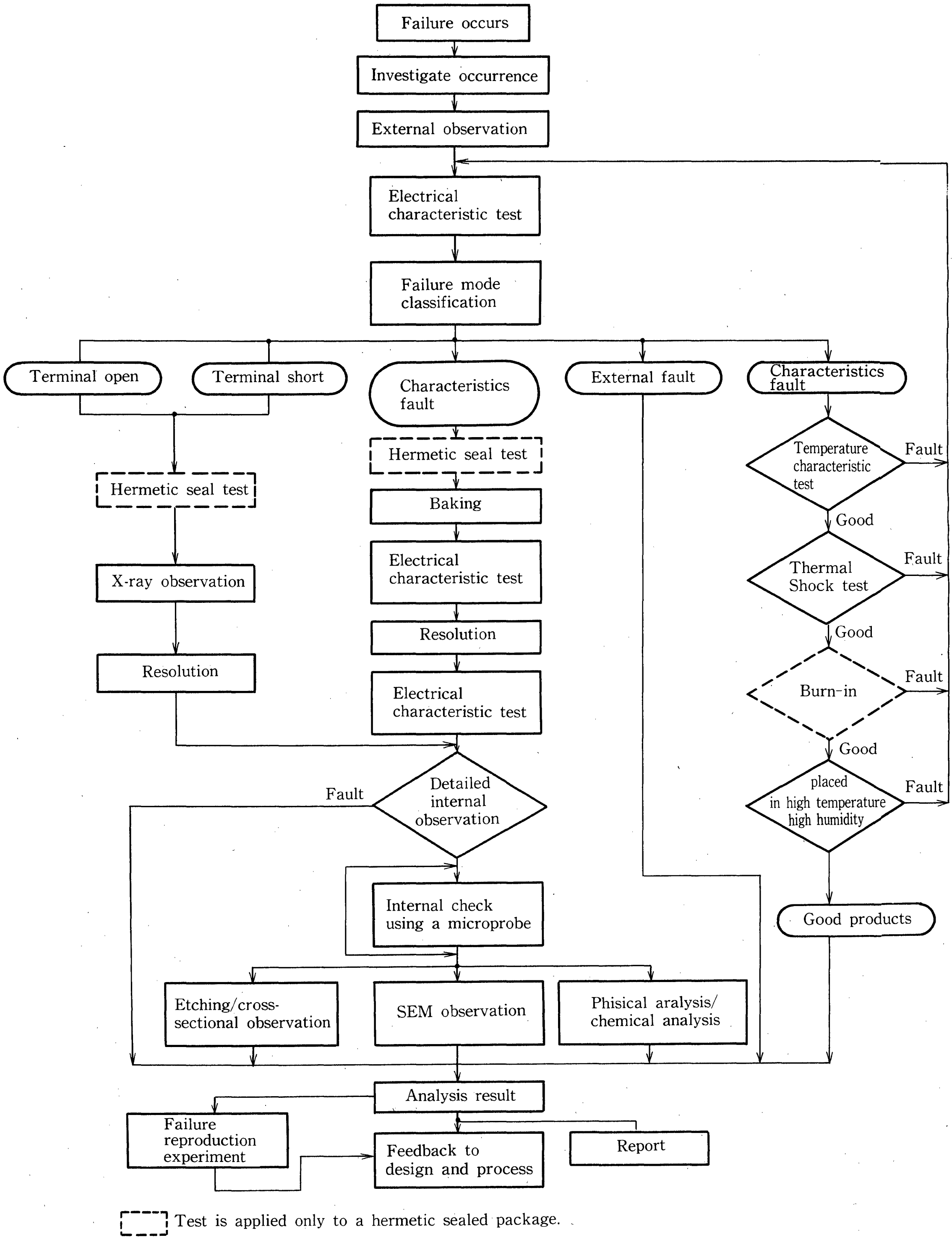
Current leakage from electrical equipment is not desirable from the standpoint of safety. All equipment should therefore be checked periodically for current leakage.

When forming the lead wires of semiconductor products to be mounted, forceps or a similar tool that will prevent stress from being applied to the base of the wires should be used.

To prevent the input terminals of semiconductor products on completed printed circuit boards from becoming open during storage or transport, the terminals of the circuit board should be shortcircuited or the entire circuit board itself should be wrapped in aluminium foil.

Table 2

Bonding method	Temperature and time	Test position
Infrared reflow	Peak temp. 240℃or less. 230℃ or more within 15 sec. Heating speed: 1 to 4℃/sec.	Surface IC package
Flow dipping	245℃ or less, Within 3 sec./cycle Within 5 sec. in total	Solder bathe
VPS	215℃ or less, 200℃ or less/within 40 sec.	Steam
Hand soldering	260℃ or less, within 10 sec.	IC outer lead



Failure occurs
Investigate occurrence
External observation
Electrical characteristic test
Failure mode classification
Terminal open
Terminal short
Characteristics fault
External fault
Characteristics fault
Hermetic seal test
X-ray observation
Resolution
Hermetic seal test
Baking
Electrical characteristic test
Resolution
Electrical characteristic test
Temperature characteristic test
Fault
Good
Thermal Shock test
Fault
Good
Burn-in
Fault
Good
placed in high temperature high humidity
Fault
Good products
Fault
Detailed internal observation
Internal check using a microprobe
Etching/cross-sectional observation
SEM observation
Phisical aralysis/ chemical analysis
Analysis result
Failure reproduction experiment
Feedback to design and process
Report
Test is applied only to a hermetic sealed package.

Fig. 6 Failure analysis procedure

(4) Soldering and Cleaning

When a semiconductor product are solder-bonded, specify the best conditions according to the Table 2. If using a soldering iron, use one with no leakage from the soldering tip. An A class soldering iron with an insulation resistance of less than 10 MΩ is recommended. When using a solder bath, it should be grounded to prevent its having an unstable electric potential.

Using a strongly acidic or alkaline flux for soldering can cause corrosion of the lead wires. A resin flux is ideal for this type of soldering.

To assure the reliability of a system, removal of the flux used in soldering is generally required.

To prevent stress of semiconductor products and circuit board when using ultrasonic cleaning, a cleaning method must be used that will shadow the main unit from the vibrator and specify the best conditions according to the cleaning conditions as below.

Ultrasonic output: 25W/l or less
Cleaning time: 1 min. or less in total
Cleaning fluid temp.: 15 to 40℃

(5) Adjustment and Tests

When the set is to be adjusted and tested upon completion of the printed circuit board, the printed circuit board must be checked to ensure that there are no solder bridges or cracks before the power is turned on. Also, if the market rated valtage and current are to be used, it is wise to use a current limiter.

Whenever a printed circuit board is to be removed or mounted or mounted on a socket, the power must be turned off.

When testing with a probe, care must taken to assure that the probe does to come in contact with other signals or the power supply. If the test location has been decided beforehand, it is wise to set up a specially designed test pin for testing.

When testing in high and low temperatures, the constant temperature bath must be grounded and measures taken to protect the set inside the bath from static electricity.

4-bit Single-chip Microcomputers

SM590/SM591/SM595

4-Bit Microcomputer (Controller for Low Power Systems)

■ Description

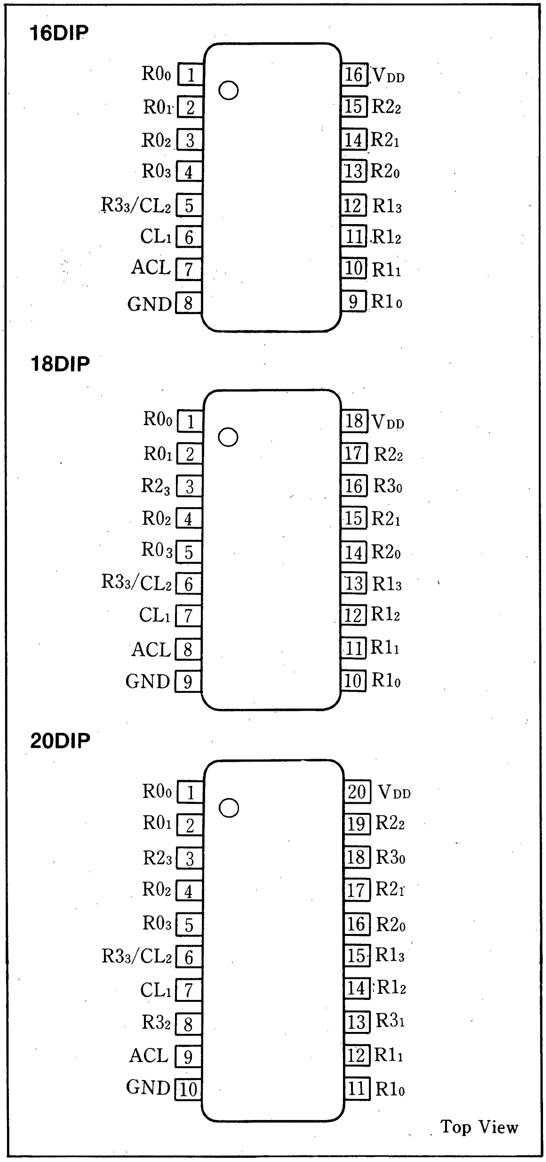
The SM590/SM591/SM592 is a CMOS 4-bit microcomputer which integrates a 762×8-bit ROM, a 41 instruction set, a 4-level subroutine stack, a 15 I/O port (for 20-pin DIP), and a standby function in a single chip.

Operated from 1 μs instruction cycle with low power consumption, this microcomputer is applicable to replacement of a compact controller circuit or any circuits consisting of conventional standard ICs.

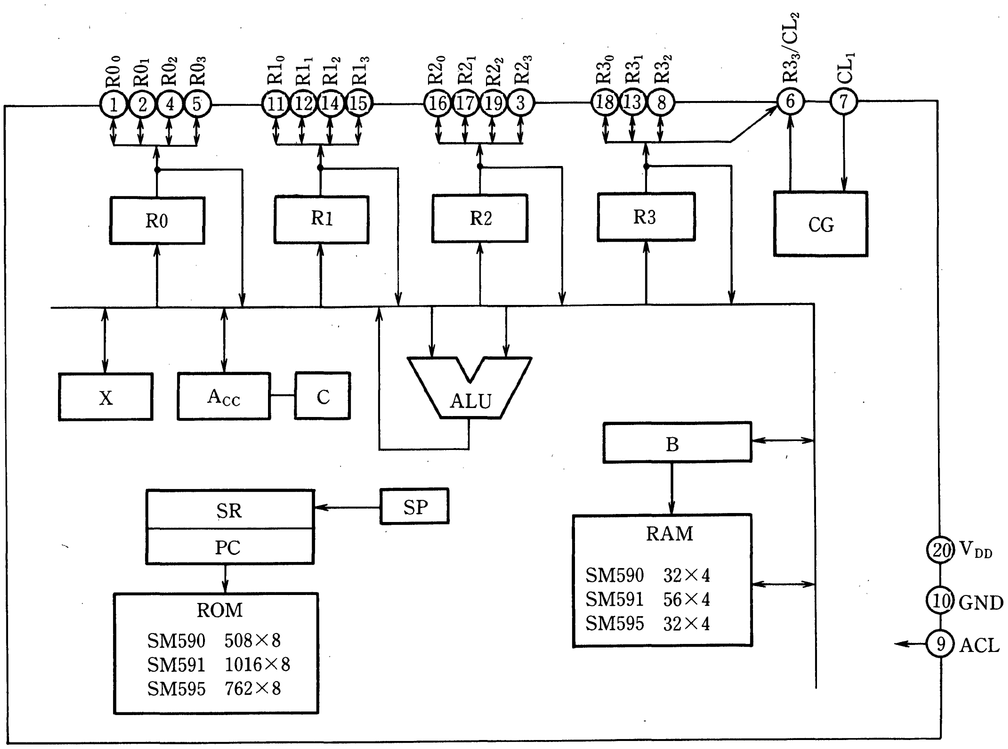
■ Features

1. CMOS process
2. ROM capacity
   SM590: 508×8 bits
   SM591: 1016×8 bits
   SM595: 762×8 bits
3. RAM capacity
   SM590: 32×4 bits
   SM591: 56×4 bits
   SM595: 32×4 bits
4. Instruction set 41
5. Subroutine nesting 4 levels
6. Input/Output ports
   11 bits (16DIP)
   13 bits (18DIP)
   15 bits (20DIP)
7. Output current (10 bits MAX.)
   SM590/SM591: 10mA (MAX.)
   SM595: 7mA (MAX.)
8. Clock oscillator
   - Ceramic oscillator
   - Resistor
   - External clock
9. Standby mode
10. Power supply (2.5 to 5.5V)
11. Instruction cycle
    V_{DD}=3V: 4 μs (MIN.)
    V_{DD}=5V: 1 μs (MIN.)
12. 16-pin DIP (DIP16-P-300)
    18-pin DIP (DIP18-P-300)
    20-pin DIP (DIP20-P-300)

■ Pin Connections



Block Diagram



Symbol description

Acc	: Accumulators	PC	: Program counter
ALU	: Arithmetic logic unit	R0-R3	: Registers
B	: RAM address register	SP	: Stack pointer
C	: Carry F/F	SR	: Stack register
CG	: Clock generator	X	: Temporary register

Note: Pin numbers apply to 20-pin DIP only.

Pin Description

Pin name	I/O	Circuit type	Function	Note
$R0_0$-$R0_3$	I/O	Pull down	Input/Output ports	1
$R1_0$-$R1_3$	I/O	Pull down	Input/Output ports	1
$R2_0$-$R2_3$	I/O	Pull down	Input/Output ports	2
$R3_0$-$R3_2$	I/O	Pull down	Input/Output ports	1
ACL	I	Pull down	Auto clear	
CL_1			System clock oscillation	
$R3_3/CL_2$	O		Output/system clock oscillation	3
V_{DD}			Power supply for logic circuit	
GND			Ground	

Note 1: Open drain I/O or CMOS outputs selectable with a mask option.
Note 2: Open drain I/O is selectable with a mask option.
Note 3: An external clock should be applied when the $R3_3$ output port is selected with a mask option.

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	−0.3 to +7.5	V
Input voltage	V_I	−0.3 to V_{DD}+0.3	V
Output voltage	V_O	−0.3 to V_{DD}+0.3	V
Source output current sum	ΣI_{OH}	120	mA
Sync output current sum	ΣI_{OL}	20	mA
Operating temperature	Topr	−10 to +70	℃
Storage temperature	Tstg	−55 to +150	℃

Recommended Operating Conditions

(Ta=−10 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD}		2.5		5.5	V
Instruction cycle	t_{SYS}	V_{DD}=3V±0.5V	4		50	μs
		V_{DD}=5V±0.5V	1		50	

Electrical Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−10 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			$0.7V_{DD}$		V_{DD}	V	1
	V_{IL1}			0		$0.3V_{DD}$	V	
	V_{IH2}			$V_{DD}-0.5$		V_{DD}	V	2
	V_{IL2}			0		0.5	V	
	V_{IL3}	V_{DD}=5V±10%	SM590/SM591	0.7	1.4	2.1	V	3
		V_{DD}=3V	SM595			0.5		
		V_{DD}=5V	SM595			0.7		
	ΔV_I	V_{DD}=5V±10%	SM590/SM591	1.1	2.0	3.1	V	
	V_{IH3}	V_{DD}=3V	SM595	2.8			V	
		V_{DD}=5V	SM595	4.6				
Input current	I_{IH1}	$V_{IN}=V_{DD}$	V_{DD}=3V±10%	15	70	200	μA	1
			V_{DD}=5V±10%	70	250	750		
	I_{IH2}	$V_{IN}=V_{DD}$	V_{DD}=3V±10%		7	20	μA	4
			V_{DD}=5V±10%		20	60		
Current consumption	I_{A1}	t_{SYS}=2μs	V_{DD}=5V±10%		1	3	mA	5
	I_{A2}	t_{SYS}=10μs	V_{DD}=3V±10%		100	200	μA	
			V_{DD}=5V±10%		200	500		
	I_{ST}	Standby mode			1	2	μA	
Output current	I_{OH1}	V_{DD}=5V±10%	SM590/SM591	10			mA	6
		$V_{OH}=V_{DD}-2V$	SM595	7				
		$V_{OH}=V_{DD}-0.5V$		1				
	I_{OL1}	V_{OL}=0.4V	V_{DD}=5V±10%	1.6			mA	
		CMOS output		0.8				
		V_{OL}=0.4V	V_{DD}=5V±10%	15			μA	
		Pull-down output		8				
	I_{OH2}	V_{DD}=5V±10%	SM590/SM591	4			mA	7
		$V_{OH}=V_{DD}-2V$	SM595	3				
		$V_{OH}=V_{DD}-0.5V$	SM590/SM591	0.5				
			SM595	0.4				
	I_{OL2}	V_{OL}=0.4V	V_{DD}=5V±10%	15			μA	
		Pull-down output		8				
	I_{OH3}	V_{DD}=5V±10%	SM590/SM591	4			mA	8
		$V_{OH}=V_{DD}-2V$	SM595	3				
		$V_{OH}=V_{DD}-0.5V$	SM590/SM591	0.5				
			SM595	0.4				
	I_{OL3}	V_{OL}=0.4V	V_{DD}=5V±10%	1.6			mA	
		CMOS output		0.8				
		V_{OL}=0.4V	V_{DD}=5V±10%	15			μA	
		Pull-down output		8				
	I_{OH4}	V_{DD}=5V±10%	SM590/SM591	10			mA	9, 10
		$V_{OH}=V_{DD}-2V$	SM595	7				
		$V_{OH}=V_{DD}-0.5V$		1				
		V_{DD}=5V±10%	SM590/SM591	3				9, 11
		$V_{OH}=V_{DD}-2V$	SM595	2				
		$V_{OH}=V_{DD}-0.5V$	SM590/SM591	0.4				
			SM595	0.3				
	I_{OL4}	V_{OL}=0.4V	V_{DD}=5V±10%	1.6			mA	9
		CMOS output		0.8				
		V_{OL}=0.4V	V_{DD}=5V±10%	15			μA	
		Pull-down output		8				

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Output current	I_{OH5}	$V_{DD}=5V\pm10\%$ $V_{OH}=V_{DD}-2V$	SM590/SM591	1			mA	12
			SM595	0.7				
		$V_{OH}=V_{DD}-0.5V$	SM590/SM591	0.15				
			SM595	0.1				
	I_{OL5}	$V_{OL}=0.4V$	$V_{DD}=5V\pm10\%$	0.6			mA	
		CMOS output		0.3				

Note 1: Applied to pins $R0_0$–$R0_3$, $R1_0$–$R1_3$, $R2_0$–$R2_3$, $R3_0$–$R3_3$.
Note 2: Applied to pins ACL and CL_1.
Note 3: Applied to pin $R2_2$. (When a standby clear signal is input.)
V_{IL3}: Oscillation start input voltage (No oscillation is occurred under this level.)
V_{IH3}: Systemclock start voltage (See Fig. 7)
ΔV_1: $V_{IH3}-V_{IL3}$
Note 4: Applied to pin ACL.
Note 5: No load condition.
Note 6: Applied to pins $R0_0$–$R0_3$, $R1_0$–$R1_3$, $R3_1$.
Note 7: Applied to pins $R2_0$–$R2_3$.
Note 8: Applied to pin $R3_0$.
Note 9: Applied to pin $R3_2$.
Note10: When the content of R latch is output from the pin $R3_2$.
Note11: When the clock input to the pin CL_1 is output from pin $R3_2$.
Note12: Applied to pin $CL_2/R3_3$.

Oscillator Circuits

CL_1 and CL_2 are the clock oscillator input and output ports respectively. The basic clock signal can be obtained by the ceramic oscillator and resistor. The external clock signal may also be provided. (See Fig. 1.)

For an external clock input, provide the external clock to the CL_1 pin. In this case, the CL_2 pin can be used as the output pin ($R3_3$ pin) with a mask option.

The internal system clock is equivalent to the basic clock supplied to the CL_1 pin divided by four.

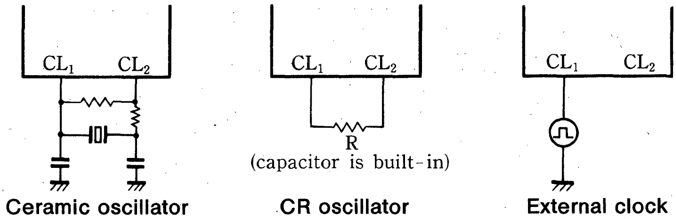
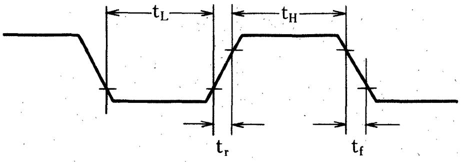


Fig. 1 Reference clock generator circuit

External Input Signal AC Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock rise time	t_r	$V_{DD}=2.5$ to $5.5V$			50	ns
Clock fall time	t_f	$V_{DD}=2.5$ to $5.5V$			50	
Clock pulse width	t_L	$V_{DD}=5V\pm0.5V$	0.08		6.3	μs
	t_H	$V_{DD}=3V\pm0.5V$	0.45		6.3	

Note: When external clock is input.



Pin Descriptions

(1) V_{DD}, GND (Power supply)

Apply 2.5 to 5.5V power supply to the V_{DD} pin with respect to GND pin which provides a reference level of the LSI.

(2) ACL (Reset pin)

The ACL pin is used to initialize the LSI. The LSI will be reset upon completion of two instruction cycles after ACL pin goes High. The ACL (reset) mode will be cleared upon completion of one instruction cycle after ACL pin goes LOW.

Connect a capacitor between ACL and V_{DD} to reset when power on. Two or more instruction cycles should be taken for the ACL input.

When a ceramic oscillator is used for a system clock, take a certain period of ACL time with the oscillation to be stabled.

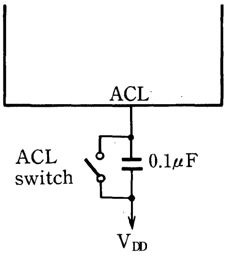


Fig. 2 ACL circuit

(3) $R3_i$-$R0_i$ (i=0 to 3) I/O pin

$R3_i$-$R0_i$ (i=0 to 3) pins may be used for both input and output, and a pull-down resistor is connected to the output buffer.

Data should be transferred between ports ($R3_i$-$R0_i$) specified by the BL and the accumulator (A_{CC}) or data memory by instructions.

When $R3_i$-$R0_i$ pins are used as inputs, reset the output latch and connect a pull-down resistor to the I/O pins.

Note: Upon completion of RTA instruction for $R3_2$ pin, the contents of an internal output latch are loaded into the accumulator A_{CC}.

The circuit type of $R3_i$-$R0_i$ can be used not only as a pull-down type but also as the following two types with a mask option.

Mask option I

When the R port is used as only output port with a large sink current, it can be replaced with the CMOS buffer.

Applicable pins:

$R0_0$-$R0_3$, $R1_0$-$R1_3$, $R3_0$-$R3_2$

Not applicable pins:

$R2_0$-$R2_3$

Mask option II

When the R port is used as only input port with a reduced current flowing into the pull-down resistor, it can be replaced with an open drain with a protective diode not to be pulled-down.

Note: The $CL_2/R3_3$ pin can be used as $R3_3$ output pin with a mask option, and the circuit type should be set to the CMOS buffer.

Hardware Configuration

(1) Program counter and stack

The ROM addresses can be specified by a program counter (PC).

The program counter (PC) consists of 10 bits including 1 bit (P_U) for the field specification, 2 bits (P_M) for the page specification and 7 bits (P_L) for the step specification.

The P_M for the page specification is a binary counter, and the P_L for the step specification is a polynomial counter (provided that it is inhibited for P_L=7F).

A 4-bit stack register enables 4 levels of subroutine nesting.

(2) Program memory (ROM)

The program memory (ROM) is used to store programs. See Fig. 3 and Fig. 4 for ROM configuration.

1 field has a configuration of 4pages×127 steps ×8 bits.

The SM590 has 508 bytes of ROM which consists of 1 field (0)×127 steps×4 pages.

The SM591 has 1016 bytes of ROM which consists of 2 fields (0 and 1)×127 steps×4 pages.

The SM595 has 762 bytes of ROM which consists of 1 field (0)×127 steps×4 pages+1 field (1)×127 steps×2 pages.

The ACL program starts at field 0, page 0 and step 0.

When the standby mode is cleared, execute the program at field 0, page 1 and step 0.

P_M \ P_U	Field 0	Field 1
Page 0	ACL start	
Page 1	Standby mode start	
Page 2		
Page 3		
	←SM590→	
	←SM591	→

Fig. 3 ROM configuration (SM590/SM591)

P_M \ P_U	Field 0	Field 1
Page 0	ACL start	
Page 1	Standby mode start	
Page 2		
Page 3		

Fig. 4 ROM configuration (SM595)

The TR instruction is used to jump within a page, while the TL (two-word) instruction is used to jump to any desired address. The TLS instruction executes a subroutine jump to any desired address.

(3) Data memory (RAM) and B register

The data memory (RAM) is used to store data. The RAM size of the SM590 and SM595 is 16×2 ×4 (128 bits), while that of the SM591 is 16×3.5 ×4 (244 bits).

Each file consists of a 16 word×4-bit configuration as shown in Fig. 5.

The RAM address is specified by a B register composed of 1-bit for the SM590/SM595 or 2 bits for the SM591 of B_M and 4 bits of B_L.

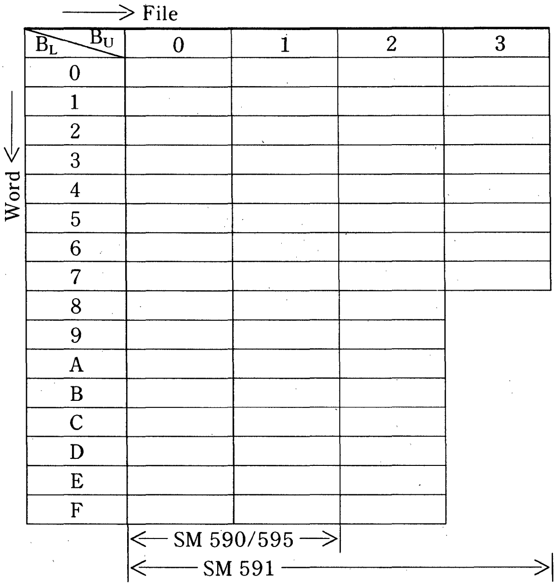


Fig. 5 RAM Configuration

(4) Accumulator (A_{CC}) and X register

The accumlator (A_{CC}) is a 4-bit register. It transfers data to I/O ports and performs operations in combination with an arithmetic and logic unit (ALU), a carry flag (C) and a RAM.

The X register is a 4-bit register used as a temporary register which transfers and compares data with the A_{CC}.

(5) Arithmetic and logic unit (ALU) and carry flag (C)

The arithmetic and logic unit (ALU) performs 4-bit parallel arithmetic operations. Executing the ADC and ADCS instructions shifts the carry of operations into the carry flag (C).

(6) Output latches (R [0], R [1] R [2], R[3])

The output latches consist of 16 bits. 11 bits for 16 pin package, 13 bits for 18 pin package and 15 bits for 20 pin package of the output latches are connected to external pins. The rest of the output latches not connected to external pins can be used as temporary registers.

The R output latches are specified by B_L.

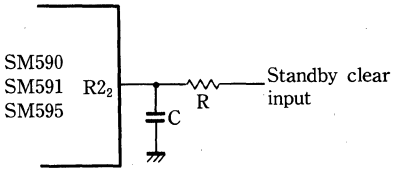


Fig. 6

(7) System clock generator circuit

The system clock generator circuit divides the basic clock supplied from the CL_1 pin, generates the system clock.

The circuit externally outputs the clock signals generated from clock oscillators (CL_1, CL_2) through $R3_2$ pin with a mask option.

This function enables to be synchronized with other LSIs.

Note that the instruction cycle time of 1 word instruction is equivalent to 1 cycle of system clocks.

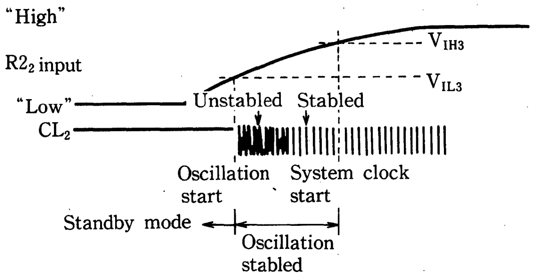


Fig. 7 Clock timing for a ceramic ascillator

(8) Standby function

Executing an instruction places the device in standby mode to reduce current consumption.

The oscillator and the system clock are iactivated in standby mode.

When the $R2_2$ accepts High level in standby mode, the standby mode is cleared and restarts program execution at field 0, page 1 and step 0.

If a ceramic oscillator is used as a clock generator, a delay circuit shown in Fig. 6 is required to obtain the clock oscillation time to be stabled.

Fig. 7 shows the timing in this case.

(9) Reset function (ACL)

Applying a High level signal to the ACL pin resets the carry flag (c) and the output latch, and the input pins are pulled-down.

Be sure not to apply High level to both $R2_0$ and $R2_1$ pins in the reset (ACL) mode.

Applying a Low level signal to the ACL pin starts execution of the program at field 0, page 0, step 0.

■ Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-FE	$P_L \leftarrow I_6$-I_0 (jump within a page)
TL xyz	78-7B	$P_U \leftarrow I_9$, $P_M \leftarrow I_8$, I_7
	00-FE	$P_L \leftarrow I_6$-I_0 (jump to any page)
TLS xyz	7C-7F	$SP \leftarrow SP+1$, $SR \leftarrow PC+2$
	00-FE	$P_U \leftarrow I_9$, $P_M \leftarrow I_8$, I_7, $P_L \leftarrow I_6$-I_0
RTN	4C	$SP \leftarrow SP-1$, $PC \leftarrow SR$
RTNS	4D	$SP \leftarrow SP-1$, $PC \leftarrow SR$, Skip

(2) Data transfer instructions

Mnemonic	Machine code	Operation
LAX x	30-3F	$A_{CC} \leftarrow I_3$-I_0 Skip if last instruction is LAX
LBLX x	20-2F	$B_L \leftarrow I_3$-I_0
LBMX x	74-77	$B_M \leftarrow I_1$, I_0
STR	4A	$M \leftarrow A_{CC}$
LDA	40	$A_{CC} \leftarrow M$
EXC	41	$M \longleftrightarrow A_{CC}$
EXCI	42	$M \longleftrightarrow A_{CC}$, $B_L \leftarrow B_L+1$ Skip if Carry=1
EXCD	43	$M \longleftrightarrow A_{CC}$, $B_L \leftarrow B_L-1$ Skip if Borrow=1
EXAX	5D	$A_{CC} \longleftrightarrow X$
ATX	5C	$X \leftarrow A_{CC}$
XBLA	57	$A_{CC} \longleftrightarrow B_L$
BLTA	56	$A_{CC} \leftarrow B_L$

(3) Arithmetic instructions

Mnemonic	Machine code	Operation
ADX x	00-0F	$A_{CC} \leftarrow A_{CC}+x$, Skip if Carry=1
ADD	70	$A_{CC} \leftarrow A_{CC}+M$
ADS	71	$A_{CC} \leftarrow A_{CC}+M$, Skip if Carry=1
ADC	72	$A_{CC} \leftarrow A_{CC}+M+C$, $C \leftarrow$ Carry
ADCS	73	$A_{CC} \leftarrow A_{CC}+M+C$, $C \leftarrow$ Carry Skip if Carry=1
COMA	44	$A_{CC} \leftarrow \overline{A_{CC}}$
INBL	52	$B_L \leftarrow B_L+1$, Skip if Carry=1
DEBL	53	$B_L \leftarrow B_L-1$, Skip if Borrow=1
INBM	50	$B_M \leftarrow B_M+1$
DEBM	51	$B_M \leftarrow M_M-1$

(4) Test instructions

Mnemonic	Machine code	Operation
TAX x	10-1F	Skip if $A_{CC}=x$
TBA x	64-67	Skip if $A_{CCX}=1$ (x=3 to 0)
TM x	60-63	Skip if $Mx=1$ (x=3 to 0)
TAM	45	Skip if $A_{CC}=M$
TC	54	Skip if $C=1$

(5) Bit manipulation instructions

Mnemonic	Machine code	Operation
SM x	6C-6F	$Mx \leftarrow 1$ (x=3 to 0)
RM x	68-6B	$Mx \leftarrow 0$ (x=3 to 0)
SC	49	$C \leftarrow 1$
RC	48	$C \leftarrow 0$

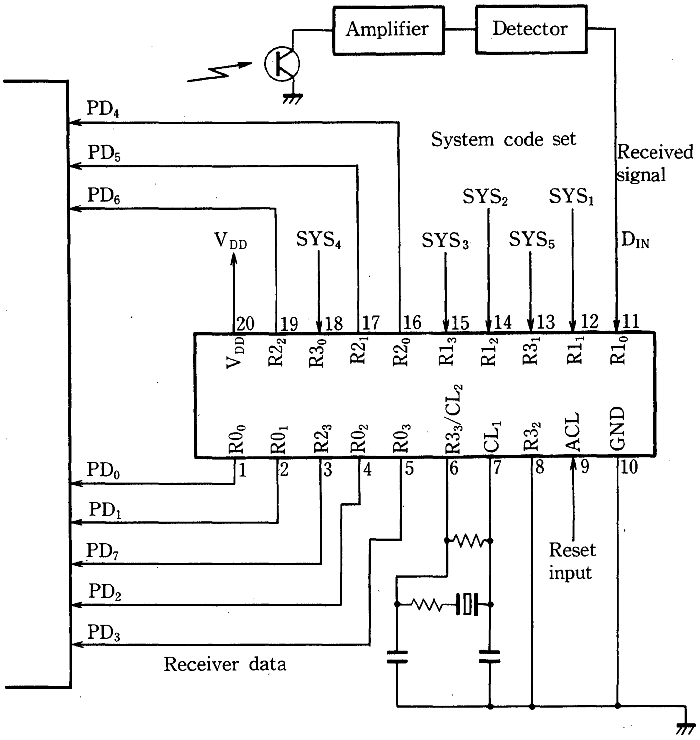
(6) I/O instructions

Mnemonic	Machine code	Operation
ATR	46	$R(B_L) \leftarrow A_{CC}$
MTR	47	$R(B_L) \leftarrow M$
RTA	55	$A_{CC} \leftarrow R(B_L)$

(7) Special instructions

Mnemonic	Machine code	Operation
NOP	00	No Operation
CCTRL	4B	Standby Mode

System Configuration Example



Remote control receiver

SM550/SM551/SM552 4-Bit Microcomputer (Controller)

■ Description

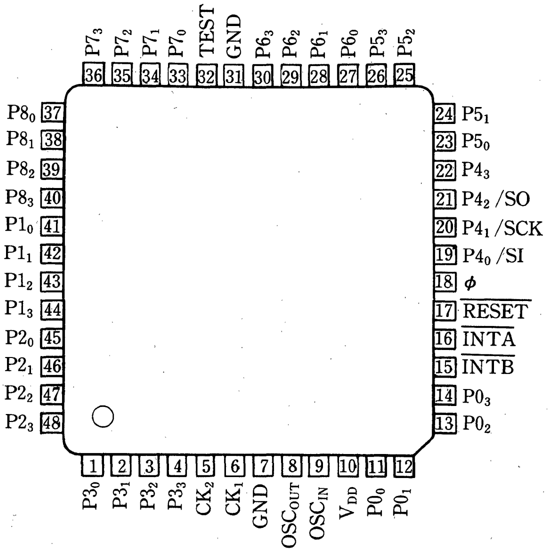
The SM550/SM551/SM552 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a ROM, a RAM, I/O ports, a serial interface, a timer/event counter in a single chip.

It provides five kinds of interrupt and a subroutine stack function using the RAM area, and accesses on a byte-by-byte basis.

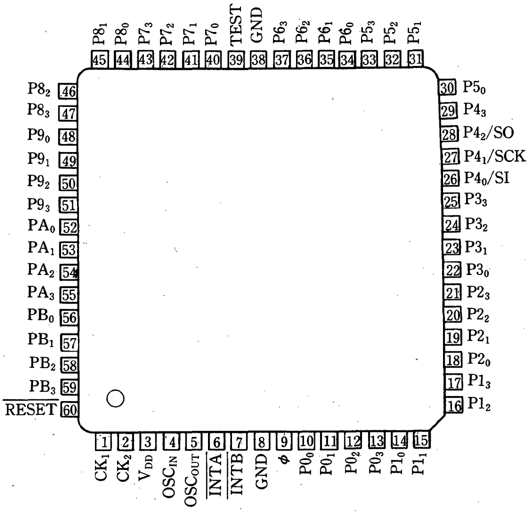
Operated from 3 to 5V single power supply with high speed, this microcomputers applicable to many applications from a battery back-up system to a high performance system.

■ Pin Connections

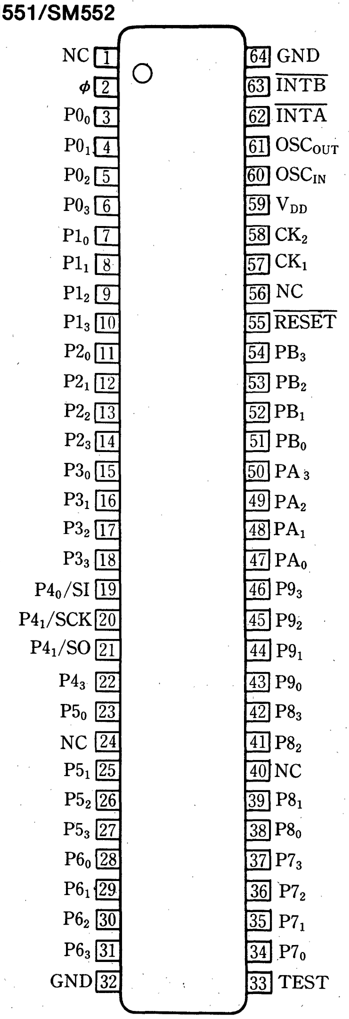
SM550



SM551/SM552



SM551/SM552

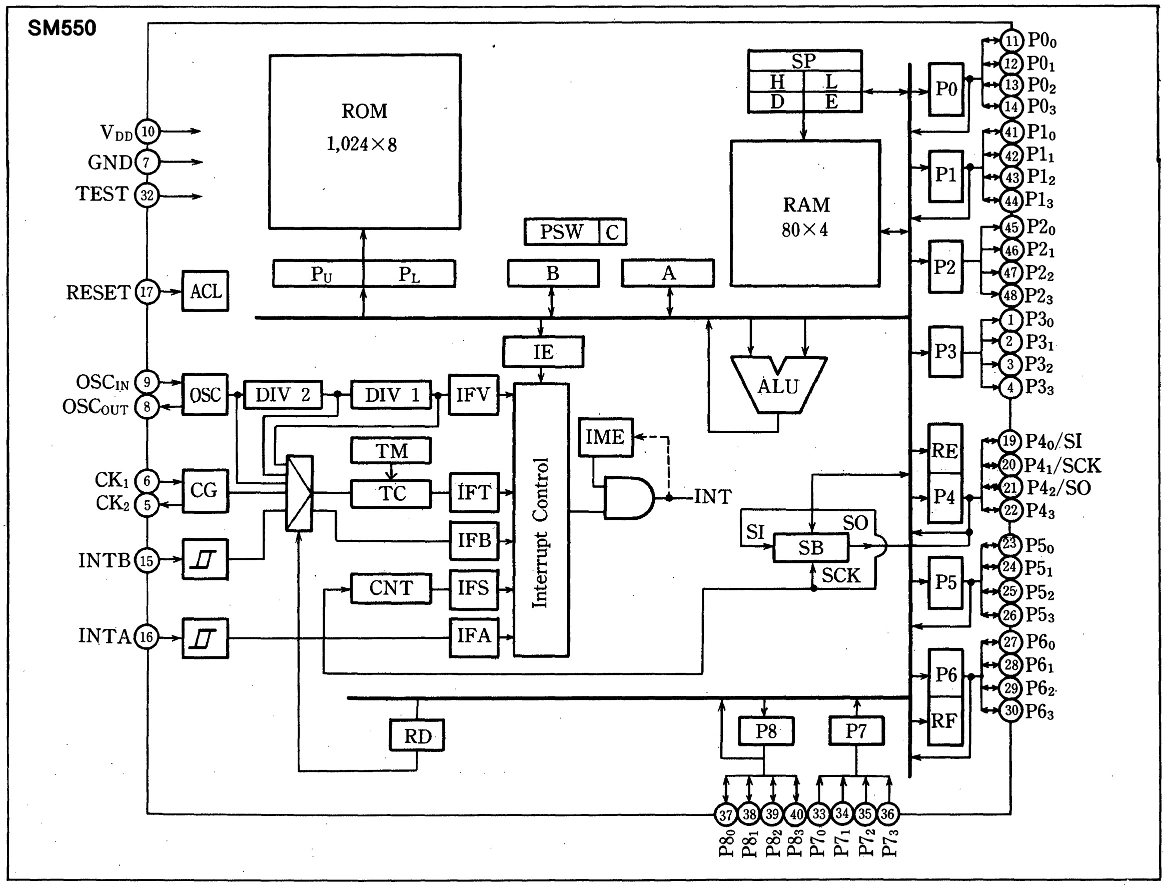


Top View

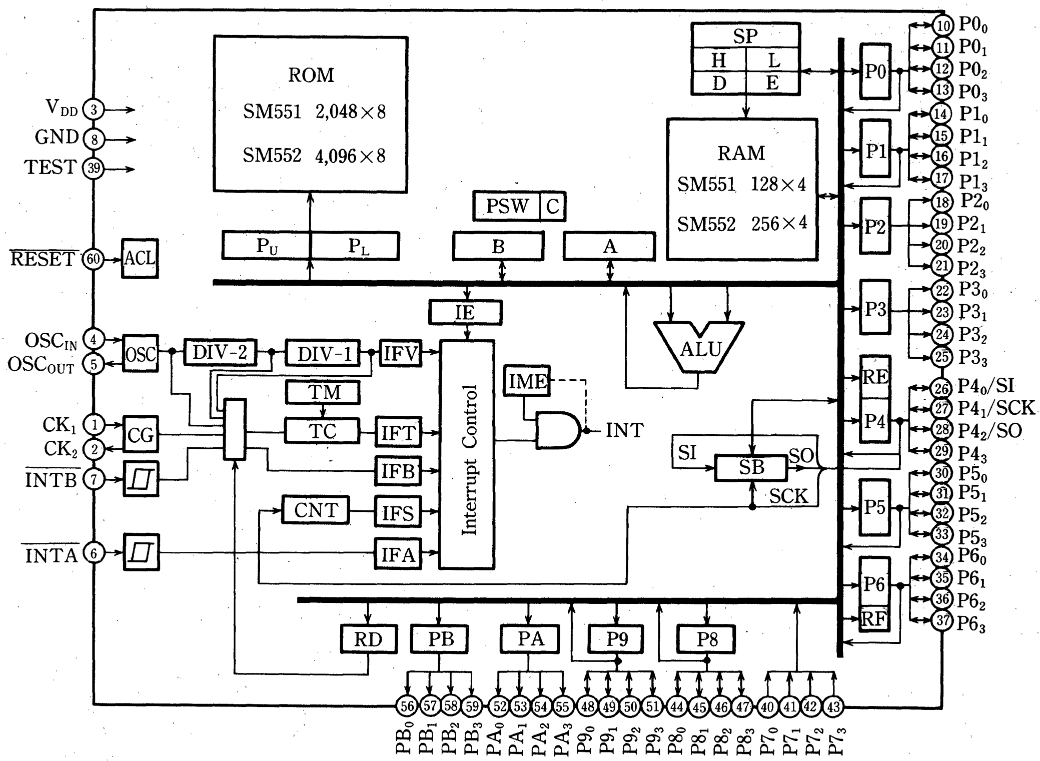
Features

1. CMOS process
2. ROM capacity
   SM550: 1,024×8 bits
   SM551: 2,048×8 bits
   SM552: 4,096×8 bits
3. RAM capacity
   SM550: 80×4 bits
   SM551: 128×4 bits
   SM552: 256×4 bits
4. Instruction set: 94
5. Subroutine stack: using RAM area
6. Instruction cycle:
   1.74 μs (MIN.) (V_{DD}=5V)
   5.3 μs (MIN.) (V_{DD}=3V)
7. Interrupts
   External interrupts: 2
   Internal interrupts: 3
8. Input/output ports
   SM550: I/O ports 24
   Input ports 4
   Output ports 8
   SM551/SM552: I/O ports 28
   Input ports 4
   Output ports 16
9. 8-bit serial I/O
10. Timer/counter: 1 set
11. On-chip crystal oscillator circuit and clock divider circuit
12. On-chip system clock oscillator
13. Standby function
14. Expandable external data ROM/RAM
15. Supply voltage: 2.7 to 5.5V
16. SM550: 48-pin QFP (QFP48-P-1010)
    SM551/SM552: 60-pin QFP
    (QFP60-P-1414)
    64-pin SDIP
    (SDIP64-P-750)

Block Diagram



SM551/SM552



Symbol description

A,B	: Accumulators	PU,PX	: Program counters
ACL	: Auto clear circuit	P0-PB	: Registers
ALU	: Arithmetic logic unit	PSW	: Program status word register
CG	: Clock generator	RD, RE, RF	: Mode registers
DIV	: Divider	SB	: Shift registers
H, L, D, E	: General-purpose registers	SP	: Stack pointer
IE	: Interrupt enable F/F	TC	: Count registers
IFT, IFA, IFS, IFB, IVV	: Interrupt requests	TM	: Module registers
IME	: Interrupt mask enable F/F		

Note: Pin numbers apply to a 60-pin QFP.

Pin Description

Symbol	I/O	Circuit type	Function	Note
$P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$, $P8_0$-$P8_3$	I/O	Pull-up (I)	Input/output ports	
$P9_0$-$P9_3$	I/O	Pull-up (I)	Input/output ports	1
$P2_0$-$P2_3$, $P3_0$-$P3_3$	O		Output ports	
PA_0-PA_3, PB_0-PB_3	O		Output ports	1
$P7_0$-$P7_3$	I	Pull-up	Input ports	
$\overline{INTA}$, $\overline{INTB}$	I	Pull-up	Interrupt input ports	
CK_1, CK_2			System clock CR oscillator	
OSC_{IN}, OSC_{OUT}			Crystal oscillator	
ϕ	O		Synchronous clock output	
V_{DD}, GND			Power supply	
TEST	I	Pull-down	Test input (normally connected to GND)	
$\overline{RESET}$	I	Pull-up	Reset input	

Note 1: Applied to the SM551/SM552.

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−0.3 to +7.5	V	1
Input voltage	V_{IN}	−0.3 to V_{DD}+0.3	V	1
Output voltage	V_{OUT}	−0.3 to V_{DD}+0.3	V	1
Output current	I_{OUT}	40	mA	2
Operating temperature	Topr	−20 to +70	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.
Note 2: Sum of current output from (or flowing into) output pin.

Recommended Operating Conditions

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Supply voltage	V_{DD}		2.7		5.5	V	
Crystal oscillator frequency	f_{OSC}			32.768		kHz	1
Basic clock oscillator frequency	f	V_{DD}=5V	0.25		2.3	MHz	2
		V_{DD}=3V	0.25		0.75		

Note 1: Oscillation starting time: within 10 seconds
Note 2: Degree of fluctuation frequency: ±30%
(Tolerance of voltage fluctuation: ±10%)

Electrical Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			$0.7V_{DD}$		V_{DD}	V	1
	V_{IL1}			0		$0.3V_{DD}$	V	
	V_{IH2}			$V_{DD}-0.5$		V_{DD}	V	2, 9
	V_{IL2}			0		0.5	V	
Input current	I_{IN}	$V_{IH}=0V$		2		200	μA	1, 9
			$V_{DD}=5.0V\pm10\%$	20		200		
Output current	I_{OH1}	$V_{OH}=V_{DD}-0.5V$		50			μA	3
	I_{OL1}	$V_{OL}=0.5V$		250			μA	
	I_{OH2}	$V_{OH}=V_{DD}-0.5V$		100			μA	4
	I_{OL2}	$V_{OL}=0.5V$		500			μA	
	I_{OH3}	$V_{OH}=V_{DD}-0.5V$		100			μA	5
			$V_{DD}=5.0V\pm10\%$	400				
	I_{OL3}	$V_{OL}=0.5V$		0.5			mA	
			$V_{DD}=5.0V\pm10\%$	1.6				
Current consumption	I_{OP}	f=0.5MHz, $V_{DD}=3.0V\pm10\%$			0.3	1.2	mA	6
		f=1MHz, $V_{DD}=5.0V\pm10\%$			1	4		
	I_{ST}	Standby current			1	5	μA	7
			$V_{DD}=3.0V\pm10\%$		12	40		8
			$V_{DD}=5.0V\pm10\%$		50	200		

Note 1: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$, $P8_0$-$P8_3$ (during input mode), $P7_0$-$P7_3$, $\overline{RESET}$.
Note 2: Applied to pins CK_1, OSC_{IN}, TEST.
Note 3: Applied to pin CK_2.
Note 4: Applied to pin ϕ.
Note 5: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$, $P8_0$-$P8_3$ (during output mode), $P2_0$-$P2_3$, $P3_0$-$P3_3$.
Note 6: No-load condition.
Note 7: No-load condition when crystal oscillation circuit is not operating. Connect OSC_{IN} pin to GND.
Note 8: No-load condition when crystal oscillation circuit is operating.
Note 9: Applied to pins $\overline{INTA}$, $\overline{INTB}$.

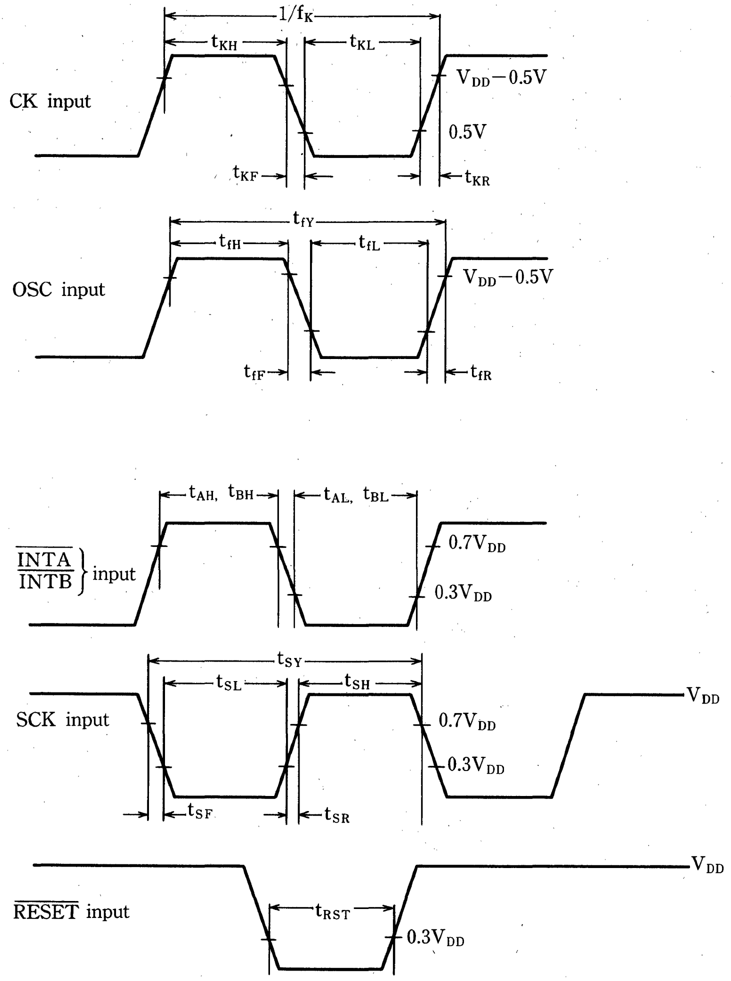
AC Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Reference clock oscillator frequency (CR oscillator)	f_{CR}	V_{DD}=5V±10%	1.7	2.0	2.3	MHz	1
		V_{DD}=3V±10%	0.5	0.75	1.0	MHz	2
		R=50kΩ±5%	0.5	0.75	1.0	MHz	
Reference clock input frequency (CK_1)	f_K	V_{DD}=5.0V±10%	0.25		2.3	MHz	
			0.25		1.0		
CK_1 input rise time	t_{KR}				100	ns	
CK_1 input fall time	t_{KF}				100	ns	
CK_1 input HIGH width	t_{KH}	V_{DD}=5.0V±10%	0.1			μs	
			0.4				
CK_1 input LOW width	t_{KL}	V_{DD}=5.0V±10%	0.1			μs	
			0.4				
Crystal oscillator frequency	f_{OSC}			32.768		kHz	
OSC_{OUT} input cycle time	t_{fY}		2			t_{CYC}	3
OSC_{OUT} iuput rise time	t_{fR}				500	ns	
OSC_{OUT} input fall time	t_{fF}				500	ns	
OSC_{OUT} input HIGH width	t_{fH}		1			t_{CYC}	3
OSC_{OUT} input LOW width	t_{fL}		1			t_{CYC}	3
$\overline{\text{INTA}}$ HIGH width	t_{AH}		2			t_{CYC}	3
$\overline{\text{INTA}}$ LOW width	t_{AL}		2			t_{CYC}	3
$\overline{\text{INTB}}$ HIGH width	t_{BH}		2			t_{CYC}	3
$\overline{\text{INTB}}$ LOW width	t_{BL}		2			t_{CYC}	3
SCK cycle time	t_{SY}		1			t_{CYC}	3
SCK HIGH width	t_{SH}		1/2			t_{CYC}	3
SCK LOW width	t_{SL}		1/2			t_{CYC}	3
SCK rise time	t_{SR}				500	ns	
SCK fall time	t_{SF}				500	ns	
$\overline{\text{RESET}}$ pulse LOW width	t_{RST}		300			ns	

Note 1: SM550: R=17kΩ±5%, SM551/SM552: R=10kΩ±5%
Note 2: SM550: R=50kΩ±5%, SM551/SM552: lR=33kΩ±5%
Note 3: Cycle time at one fouth of a reference clock frequency.

Timing Diagram



Hardware Configuration

(1) Program memory (ROM)

The on-chip ROM of the SM550/SM551/SM552 has a configuration of 16/32/64 pages× 64 steps×8 bits respectively, and stores programs and table data.

The program counter of the SM550/SM551/SM552 consists of a 4-bit/5-bit/6-bit page address counter P_U and a 6-bit binary counter P_L used to specify the steps within a page.

Fig. 1 shows the locations allocated in the on-chip ROM.

(2) Data memory (RAM)

Data memory of the SM550/SM551/SM552 has 80-word/128-word/160-word×4 bit configuration respectively.

Fig. 2 shows the RAM configuration.

(3) General-purpose registers (H, L, D, E)

Registers H and L are 4-bit general-purpose registers. They can transfer and compare data with the A_{CC} on 4-bit basis.

Registers D and E are 4-bit registers and can transfer data with the H and L registers on an 8-bit basis.

The H and L as well as the D and E registers can be combined into 8-bit register pairs, and can be used as pointers to data memory locations.

The L register can be incremented or decremented and is used to access I/O ports and mode registers.

(4) Clock divider (DIV)

The device contains a crystal oscillator and a 15-stage divider. A real-time clock can be provided by connecting an external crystal oscillator between the oscillator pins.

The on-chip divider is reset by an ACL operation or an IDIV instruction. The low-order 8 bits of the divider can be loaded into the B/A register pair by the LDDIV instruction.

When an external 32.768kHz crystal oscillator is used, the final state signal is set at a frequency of 1Hz.

(5) Timer/event counter (TC)

The timer/event counter consists of an 8-bit count register (TC) and an 8-bit modulo register (TM).

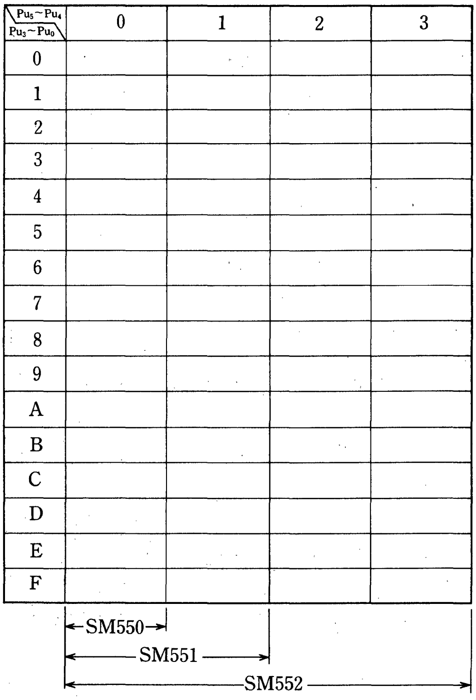
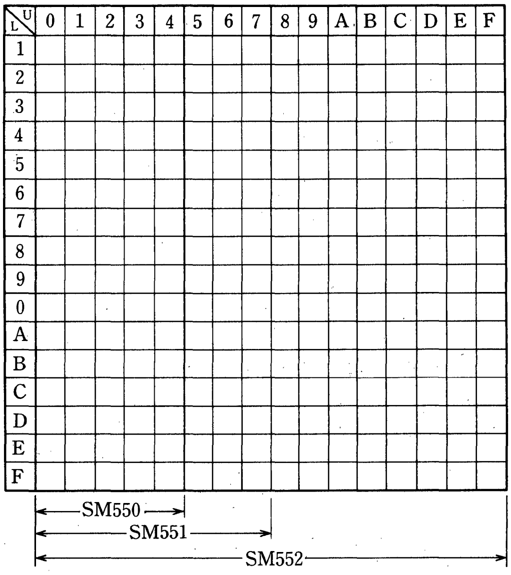


Fig. 1 ROM configuration



U: Upper L: Lower

Fig. 2 RAM configuration

Table 1 Interrupt request

Interrupt request		Int./Ext.	Priority	Interrupt routine start address
INTT	Timer/event counter interrupt	Int.	1	Page 1, Address 0
INTA	External signal INTA interrupt	Ext.	2	Address 2
INTS	Serial I/O interrupt	Int.	3	Address 4
INTB	External signal INTB and frame frequency interrupts	Ext.	4	Address 6
INTV	Divider overflow interrupt	Int.	5	Address 8

The count register is an 8-bit incremental binary counter. It is incremented by one at the falling edge of its count pulse (CP) input. If the count register overflows, the timer interrupt request flag IFT is set, and the contents of the modulo register (TM) are loaded into the count register. The contents of the count register can be loaded into the B/A register pair by the LDTC instruction.

(6) Serial interface (SIO)

The serial interface consists of an 8-bit shift register (SB) and a 3-bit counter, which is used to input and output the serial data.

In serial shift operations, the highest bit data of the shift register (SB) is output from the SO pin at the falling edge of the serial clock, and the data input from the SI pin is loaded into the lowest bit of the shift register.

When the internal clock is used, the serial operation stops with 8 clocks of serial shift operations which are output from the SCK pin.

(7) Interrupts

The interrupts can be selected within three kinds of internal interrupts and two kinds of external interrupts as shown in Table 1.

(8) I/O ports and mode registers (RD, RE, RF)

The device has I/O ports and three mode registers (RED, RE, RF). Data can be transferred between these ports and registers under instruction control or L register control.

- Ports P0, P1, P4, P5, P8 and P9* can be switched between input and output modes, 4 bits at a time.
- Ports P2, P3, PA* and PB* are 4-bit parallel output ports.
- Port P7 is a 4-bit parallel input port.
- Each bit of port P2 can be independently placed in input or output mode by setting the corresponding bit of mode register RF.
- Ports (P0, P1), (P2, P3), (P8, P9)*, and (PA, PB)* can be paired for use in data transfer on a byte-by-byte basis. However, port pairs (P2, P3) and (PA, PB)* are usable only for output.
- The mode registers RD, RE and RF are treated in much the same way as output ports.
- Each bit of port P2 can be set to the I/O modes (SI, SO and SCK) of a serial interface under program control.
- Pins P50 and P51 can output the OD and R/W signals respectively when an external memory is accessed. In those cases, these pins should be kept High in output mode.

*Applicable to the SM551 and SM552.

Every input port has pull-up resistors.

Pull-up resistors can be omitted and output ports can be designed to consist of open-drain transistors with a mask option.

(9) Standby mode

Executing the CEND instruction places the device in standby mode. To reduce power consumption, the system clock is inactivated.

Standby mode may be cleared with the Interrupt request or the RESET signal.

(10) Reset function (ACL)

Applying a Low level signal to the RESET pin resets the internal logic of the device, and starts execution of the program at address 0, page 0.

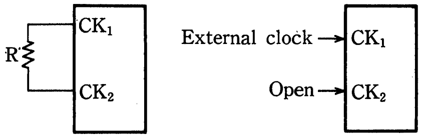
Once the device is reset, all I/O ports are placed in input mode to disable all interrupts. The mode registers RD, RE and RF are all cleared. The output ports P2, P3, PA* and PB* are all cleared to output "0". The device is also reset when it is powered up. The program starts (master clock period $\times 2^{14}$) clock periods after the reset signal is ineffected.

*Applicable to the SM551 and SM552.

(11) Master clock oscillator circuit

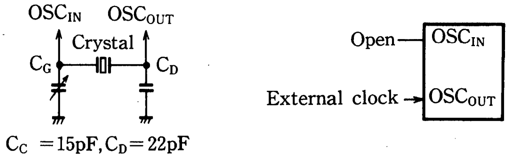
The master clock oscillator requires an external resistor across pins CK$_1$ and CK$_2$. Instead of using on-chip oscillator, an external clock may be applied to pin CK$_1$. In this case, pin CK$_2$ should be left open.

The system clock ϕ has a frequency of one fourth that of the clock applied to pin CK$_1$. When applying an external clock to pin OSC$_{OUT}$, the external clock frequency should be set at one eighth of the master clock frequency.



(a) CR oscillator (b) External clock

Fig. 3



(a) Crystal oscillator (b) External clock

Fig. 4

■ Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
STL	69	L←A
STH	68	H←A
EXHD	3F	H↔D L↔E
LIHL xy (2-byte)	3D 00-FF	H←x(I_7-I_4), L←y(I_3-I_0)

(2) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BF	P_L←x(I_5-I_0)
TL xy (2-byte)	E0-EF 00-FF	P_U←x(I_9-I_6) P_L←y(I_5-I_0)
TRS x	C0-DF	(SP-1), (SP-2), (SP-3)←PC SP←SP-4 P_U←0(SM550), P_U←10_H(SM551/SM552) P_L←x($I_4I_3I_2I_1I_0$0)
CALL xy (2-byte)	F0-FF 00-FF	(SP-1), (SP-2), (SP-3)←PC SP←SP-4, P_U←x(I_9-I_6) P_L←x($I_5I_4I_3I_2I_1I_0$)
JBA x (2-byte)	7F 30-3F	P_{U5}-P_{U2}←x(I_3-I_0) P_{U1}, P_{U0}, P_{L5}, P_{L4}←B, P_{L3}-P_{L0}←A
RTN	61	P_U, P_L←(SP+1), (SP+2), (SP+3), SP←SP+4
RTNS	62	P_U, P_L←(SP+1), (SP+2), (SP+3), SP←SP+4
RTNI	63	P_U, P_L←(SP+1), (SP+2) (SP+3), PSW←(SP), SP←SP+4, IME←1

(3) Data transfer instructions

Mnemonic	Machine code	Operation
EX pr	5C-5F	A↔(pr)
LDX adr (2-byte)	7D 00-FF	A←(adr)
STX adr (2-byte)	7E 00-FF	(adr)←A
EXX adr (2-byte)	7C 00-FF	A↔(adr)
LAX x	10-1F	A←x(I_3-I_0)

LIBA xy
(2-byte)
	3C
	00-FF
		B←x(I_7-I_4)
		A←y(I_3-I_0)

LBAT	60	B←ROM(P_{U5}-P_{U2}, B, $A)_H$
		A←ROM(P_{U5}-P_{U2}, B, $A)_L$

LDL	65	A←L
LD pr	54-57	A←(pr)
ST pr	58-5B	(pr)←A
EXH	6C	A←H
EXL	6D	A←L
EXB	6E	A←B
STB	6A	B←A
LDB	66	A←B
LDH	64	A←H
PSHBA	28	(SP-1) ←B, (SP←2)←A, SP←SP-2
PSHHL	29	(SP-1) ←B, (SP←2)←A, SP←SP-2

POPBA	38	B←(SP+1), A←(SP),
		SP←SP+2

POPHL	39	H←(SP+1), L←(SP),
		SP←SP+2

STSB	70	SB_H←B, SB_L←A
STSP	71	SP_H←B, SP_L←A
STTC	72	TC←TM
STTM	73	TM_H←B, TM_L←A
LDSB	74	B←SB_H, A←SB_L
LDSP	75	B←SP_H, A←SP_L
LDTC	76	B←TC_H, A←TC_L
LDDIV	77	B←DIV_H, A←DIV_L

(4) Arithmetic instructions

Mnemonic	Machine code	Operation
ADX x	00-0F	A←A+x(I_3-I_0), Skip if Cy=1
ADD	36	A←A+(HL)
ADDC	37	A←A+(HL)+C, C←Cy Skip if Cy=1
OR	31	A←A+(HL)
AND	32	A←A · (HL)
EOR	33	A←A⊕(HL)
ANDB	22	A←A·B
ORB	21	A←A+B
EORB	23	A←A⊕B
COMA	6F	A←$\overline{A}$
ROTR	25	C→A_3→A_2→A_1→A_0→C
ROTL	35	C←A_3←A_2←A_1←A_0←C
INCB	52	Skip if B=F, B←B+1
DECB	53	Skip if B=0, B←B−1
INCL	50	Skip if L=F, L←L+1
DECL	51	Skip if L=0, L←L−1
DECM adr	79 00-FF	Skip if (adr)=0, (adr)←(adr)−1
INCM adr	78 00-FF	Skip if (adr)=F, (adr)←(adr)+1

(5) Test instructions

Mnemonic	Machine code	Operation
TAM	30	Skip if A=(HL)
TAH	24	Skip if A=H
TAL	34	Skip if A=L
TAB	20	Skip if A=B
TC	2A	Skip if C=0
TM x	48-4B	Skip if (HL)x=1
TA x	4C-4F	Skip if Ax=1
TSTT	2B	Skip if IFT=1, IFT←0
TSTA	2C	Skip if IFA=1, IFA←0
TSTS	2D	Skip if IFS=1, IFS←0
TSTB	2E	Skip if IFB=1, IFB←0
TSTV	2F	Skip if IFV=1, IFV←0

(6) Bit manipulation instructions

Mnemonic	Machine code	Operation
SM x	40-43	(HL)x←1
RM x	44-47	(HL)x←0
RC	26	C←0
SC	27	C←1
RIME	3A	IME←0
SIME	3B	IME←1
DI x (2-byte)	7F C0-DF	IEF←IEF·x
EI x (2 byte)	7F E0-FF	IEF←IEF+x

(7) I/O instructions

Mnemonic	Machine code	Operation
IN	67	A←P(L)
OUT	6B	P(L), R(L)←A
INA x (2-byte)	7F A0-A9	A←P(x)
OUTA x (2-byte)	7F B0-BF	P(x), R(x)←A

INBA x	7F
	80; 82
		B←P(x+1)
		A←P(x)

OUTBA x (2-byte)	7F 90-93	P(x+1)←B P(x)←A
SP xy (2-byte)	7A 00-F6	P(y)←P(y)+x
RP xy (2-byte)	7B 00-F6	P(y)←P(y) x
READ (2-byte)	7F 60	A←P0 with O/D
WRIT (2-byte)	7F 70	P0←A with R/W

READB
(2-byte)
	7F
	61
		B←P1
		A←P0 with O/D

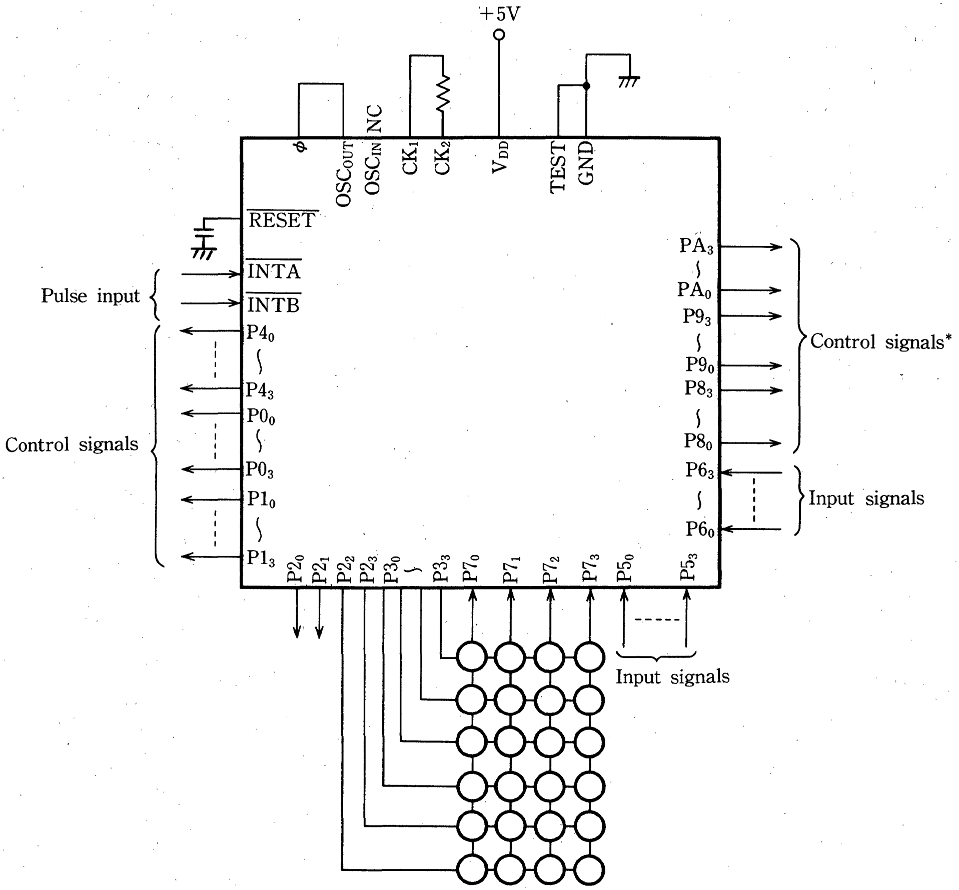
WRITB
(2-byte)
	7F
	71
		P1←B
		P0←A with R/W

(8) Special instructions

Mnemonic	Machine code	Operation
SIO	3E	Sperial I/O start
IDIV (2-byte)	7F 10	DIV←0
SKIP	00	No operation
CEND (2-byte)	7F 00	System clock stop

Note: The machine code consists of 8 bits including I_7, I_6, I_5, I_4, I_3,I_2, I_1 and I_0.

System Configuration Example (Mechanism controller)



＊Ports $P8_0$-$P8_3$ $P9_0$-$P9_3$, PA_0-PA_3 apply to the SM551 and SM552.

SM578/SM579

4-Bit Microcomputer (Controller with A/D Converter)

■ Description

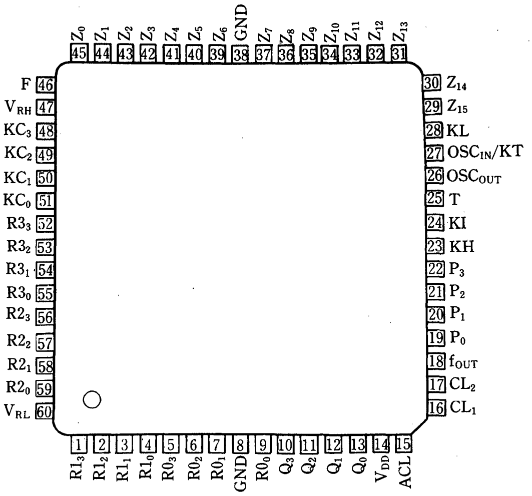
The SM578/SM579 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, interrupts, an A/D converter, a comparator, a counter/timer circuit, and a tone output function in a single chip.

An A/D conversion can be executed by one instruction with simple software, and provides a high speed processing. This feature enables to accept analog signals from sensors.

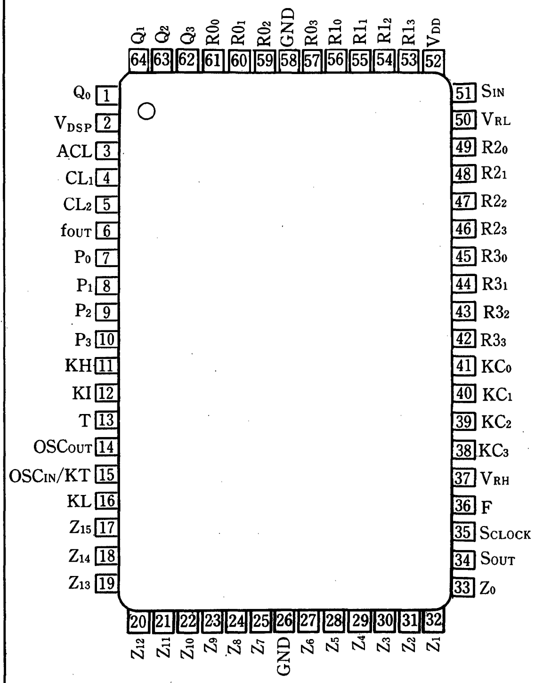
Provided with unique features of 52 I/O ports, a couple of programmable counter/timers, and many instruction sets, this microcomputer is applicable to many applications such as home appliances, office equipment, simple measuring instruments, and battery backup systems.

■ Pin Connections

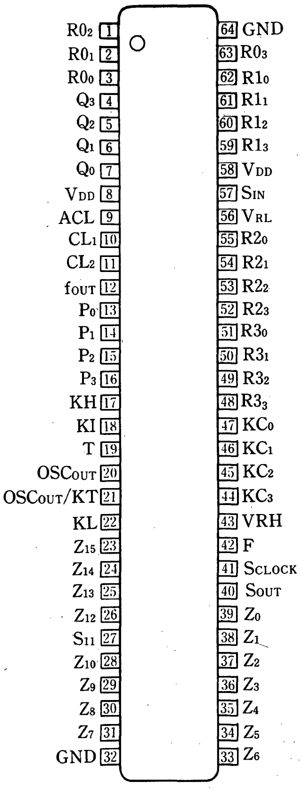
60QFP



64QFP



64SDIP



Top View

■ Features

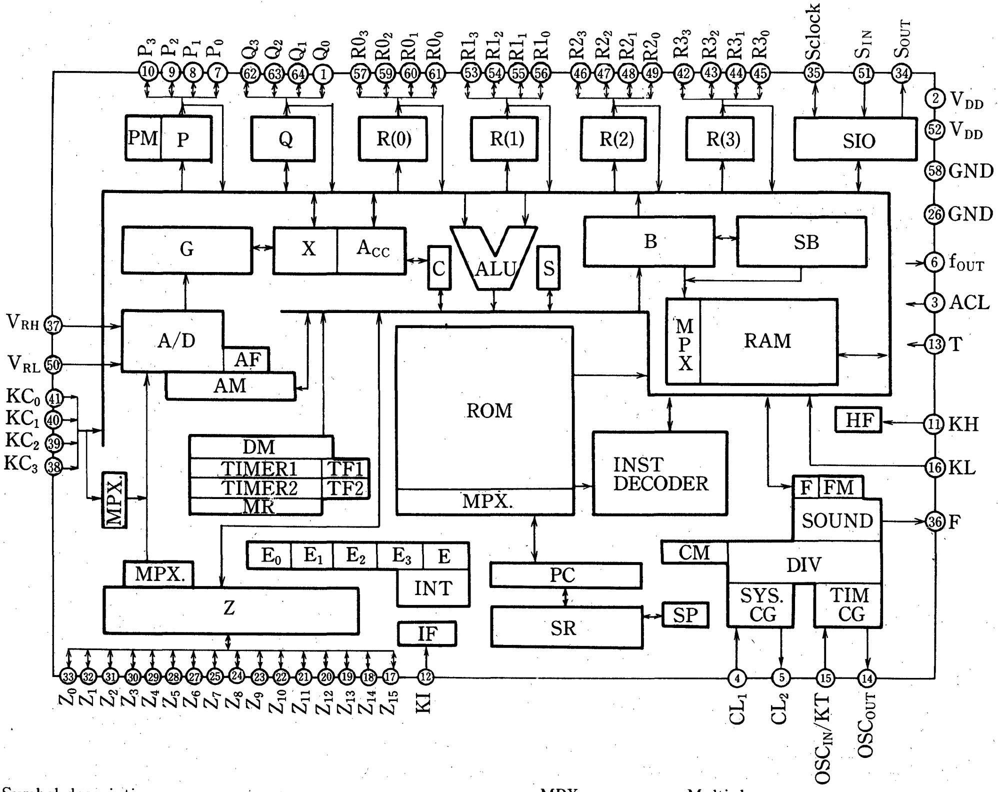
1. CMOS process
2. ROM capacity
   SM578: 4,064×9 bits
   SM579: 6,096×9 bits
3. RAM capacity
   SM578: 192×4 bits
   SM579: 256×4 bits
4. Instruction set
   SM578: 93
   SM579: 94
5. Subroutine nesting: 6 levels
6. Instruction cycle: 2 μs (MIN.)
7. Interrupts
   External interrupts: 2
   Internal interrupts: 3
8. Input/Output ports
   I/O ports: 41
   Input ports: 9
   Output ports: 2
9. 8-bit serial I/O
10. Counter/timer: 2 sets
11. A/D converter:
    8 bits (20 channels MAX.)
12. Standby function:
    2-stage system clocks

13 On-chip crystal/system clock
Oscillator circuits

14. Supply voltage: 2.7 to 5.5V
15. 60-pin QFP (QFP60-P-1414)*
    64-pin QFP (QFP64-P-1420)
    64-pin SDIP (SDIP64-P-750)

*Usable when serial I/O ports are not used.

■ Block Diagram



Symbol description

ALU	: Arithmetic logic unit	MPX.	: Multiplexer
X	: X register	A/D	: A/D converter and Comparator unit
B	: RAM address register	ACC	: Accumulator
C	: Carry F/F	SR	: Stack register
PC	: Program counter	R0, R1, R2, R3	: Latch
SP	: Stack pointer	DIV	: Divider
CG	: System clock generator		

Note: Pin numbers apply to a 64 pin QFP.

Clock Generator Circuit (preliminary constant)

Signal	I/O	Pin name
P_0-P_3, Q_0-Q_3	I/O	Input/Output ports (nibble unit)
$R0_0$-$R0_3$	I/O	Input/Output ports (nibble unit)
$R1_0$-$R1_3$	I/O	Input/Output ports (nibble unit)
$R2_0$-$R2_3$	I/O	Input/Output ports (nibble unit)
$R3_0$-$R3_3$	I/O	Input/Output ports (nibble unit)
Z_0-Z_{15}	I/O	Input/Output ports (Bit unit)
KC_0-KC_3	I	Input ports or analog input ports
KH, KL	I	Input ports
KI	I	Interrupt input port or input port
OSC_{IN}/KT	I	Timer clock input port or input port
OSC_{OUT}		Timer clock oscillator
F	O	Sound output port or output port
f_{OUT}	O	System synchronous signal output port
CL_1		Clock signal input port
CL_2		Clock signal oscillator
ACL	I	Auto clear input port
V_{RH}, V_{RL}		A/D conversion
V_{DD}		Power supply
GND		Ground
T	I	Test input port
S_{IN}	I	Serial I/O data input port*
S_{OUT}	O	Serial I/O data output port*
S_{CLOCK}	I/O	Serial I/O clock I/O port*

* Applicable only to 64-pin QFP and 64-pin DIP.

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	−0.3 to +7.5	V
Input voltage	V_{IN}	−0.3 to V_{DD} +0.3	V
Output voltage	V_{OUT}	−0.3 to V_{DD} +0.3	V
Output current*	I_{OUT}	30	mA
Operating temperature	Topr	−10 to +70	℃
Storage temperature	Tstg	−55 to +150	℃

*Source current from output pin or sum of sink current.

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	2.7 to 5.5	V
System oscillator frequency	f_{CL}	4 to 0.2 (V_{DD}=5V)	MH_Z
		2 to 0.2 (V_{DD}=3V)	
System clock frequency	f_S	500 to 50 (V_{DD}=5V)	kH_Z
		250 to 50 (V_{DD}=3V)	
Timer clock frequency	fosc	32.768 (TYP.)	kH_Z

■ Electrical Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−10 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			$0.7V_{DD}$		V_{DD}	V	1
	V_{IL1}			0		$0.3V_{DD}$		
	V_{IH2}			$V_{DD}-0.5$		V_{DD}	V	2
	V_{IL2}			0		0.5		
	V_{IH3}			$V_{DD}-0.4$		V_{DD}	V	3, 4
	V_{IL3}			0		0.4		
Input current	I_{IH1}	$V_{IH}=V_{DD}$	$V_{DD}=5.0V\pm10\%$	80	300	800	μA	5
			$V_{DD}=3.0V\pm10\%$	15	90	300		
	I_{IH2}	$V_{IH}=V_{DD}$	$V_{DD}=5.0V\pm10\%$	10	25	120	μA	6
			$V_{DD}=3.0V\pm10\%$	1	8	50		
	I_{IH3}	$V_{IH}=V_{DD}$	$V_{DD}=5.0V\pm10\%$	100	400	800	μA	3
			$V_{DD}=3.0V\pm10\%$	10	80	300		
	I_{IL}	$I_{IL}=0V$	$V_{DD}=5.0V\pm10\%$	10	100	200	μA	4
			$V_{DD}=3.0V\pm10\%$	2	15	60		
Output current	I_{OH1}	$V_{OH}=V_{DD}-0.5V$	$V_{DD}=5.0V\pm10\%$	1.0			mA	7, 8
			$V_{DD}=3.0V\pm10\%$	0.4				
	I_{OL1}	$V_{OL}=0.5V$	$V_{DD}=5.0V\pm10\%$	10			μA	7
				4				
	I_{OL2}	$V_{OL}=0.5V$	$V_{DD}=5.0V\pm10\%$	1.0			mA	8
				0.4				
	I_{OH3}	$V_{OH}=V_{DD}-0.5V$		100			μA	9
	I_{OL3}	$V_{OL}=0.5V$		100				
	I_{OH4}	$V_{OH}=V_{DD}-0.5V$	$V_{DD}=5.0V\pm10\%$	0.6			mA	10
				0.3				
	I_{OL4}	$V_{OL}=0.5V$	$V_{DD}=5.0V\pm10\%$	1.0				
				0.4				
ACL input pulse width	t_{ACL}		$V_{DD}=5.0V\pm10\%$	1			μs	16
			$V_{DD}=3.0V\pm10\%$					
Current consumption	I_A	$f_S=500kHz$	$V_{DD}=5.0V\pm10\%$		1.8		mA	11
		$f_S=100kHz$	$V_{DD}=5.0V\pm10\%$		0.8			
			$V_{DD}=3.0V\pm10\%$		0.5			
	I_{ST}		$V_{DD}=5.0V\pm10\%$		1		μA	12
			$V_{DD}=3.0V\pm10\%$		0.5			
		Off mode	$V_{DD}=5.0V\pm10\%$				μA	13
			$V_{DD}=3.0V\pm10\%$					
		Hold mode	$V_{DD}=5.0V\pm10\%$				μA	14
			$V_{DD}=3.0V\pm10\%$					
			$V_{DD}=3.0V\pm10\%$				μA	15

Note 1: Applied to pins KH, KL, KI, P_3-P_0, Q_3-Q_0, $R0_3$-$R0_0$, $R1_3$-$R1_0$, $R2_3$-$R2_0$, $R3_3$-$R3_0$, KC_3-KC_0

Note 2: Applied to pins Z_{15}-Z_0, CL_1, OSC_{IN}/KT, ACL

Note 3: Applied to pin S_{IN}

Note 4: Applied to pin S_{clock}

Note 5: Applied to pins Q_3-Q_0, Z_{15}-Z_0, $R0_3$-$R0_0$, $R1_3$-$R1_0$

Note 6: Applied to pin ACL

Note 7: Applied to pins Q_3-Q_0*, Z_{15}-Z_0*, $R0_3$-$R0_0$, $R1_3$-$R1_0$ (*If CMOS buffer is specified for mask option, note 8 applied to these pins.)

Note 8: Applied to pins P_3-P_0

Note 9: Applied to pin CL_2

Note 10: Applide to pins S_{clock}, S_{OUT}, F, f_{OUT}

Note 11: No-load condition

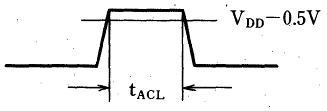
Note 12: When the OSC_{IN}/KT pin is connected to GND and in no-load condition. (The reference clock has a frequency of 4 times of fs.)

Note 13: When the timer clock crystal oscillation circuit and timer 1 are operating and in no-load condition. (The reference clock has a frequency of 4 times of fs.)

Note 14: When the OSC_{IN}/KT pin is connected to GND and in no-load condition, fs=100kHz (The reference clock has a frequensy of 4 times of fs.)

Note 15: When the OSC_{IN}/KT pin is connected to GND and in no-load condition, fs=500kHz (The reference clock has a frequency of 4 times of fs.)

Note 16: t_{ACL} is the ACL input pulse width required to cause ACL to operate when V_{DD} has completely risen.



Electrical Characteristics for A/D Conversion Block

(fs=100kHz, V_{DD}=5V, V_{RH}=4.608V, Ta=25℃)

Parameter	V_{RL} pin	MIN.	TYP.	MAX.	Unit
Non-linearity error	GND		±½	±1	LSB
	Open		±1	±1½	
Diffrential non-linerarity error	GND		±½	±1	LSB
	Open		±1	±1½	
Zero error	GND		±½	±1	LSB
	Open		±1	±1½	
Full-scale error	GND		±½	±1	LSB
	Open		±½	±1	
V_{RH} pin voltage	—		100	300	μA
Total error	GND		±1	±1½	LSB
	Open		±1½	±2	

Note: No quantizing tolerance (±½ LSB) should be specified.

(fs=100kHz, V_{DD}=3V, V_{RH}=3V, Ta=25℃)

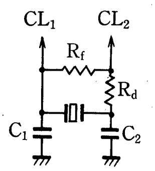
Parameter	V_{RL} pin	MIN.	TYP.	MAX.	Unit
Non-linearity error	GND		±1½	±2	LSB
	Open		±2	±2½	
Diffrential non-linerarity error	GND		±1	±1½	LSB
	Open		±1½	±2	
Zero error	GND		±½	±1	LSB
	Open		±1	±1½	
Full-scale error	GND		±½	±1	LSB
	Open		±½	±1	
V_{RH} pin voltage	—		60	200	μA
Total error	GND		±1½	±2	LSB
	Open		±2	±2½	

Note: No quantizing tolerance (±½ LSB) should be specified.

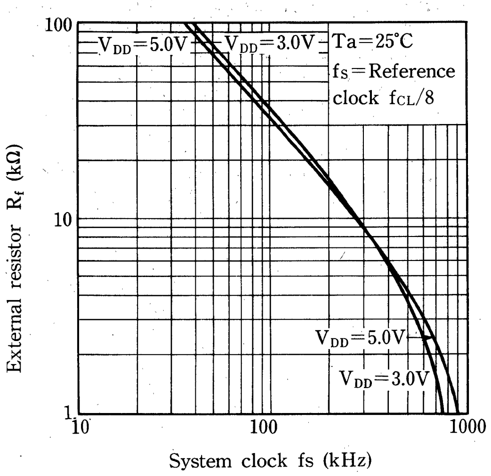
■ Clock Generator Circuit (preliminary constant)

(1) System clock generator circuit example 1

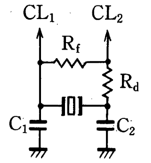
(a) 400kHz clock



Oscillator KBR-400B: KYOSERA
or CSB400P: MURATA
R_f=1MΩ
R_d=3.3kΩ
C_1=220pF
C_2=220pF

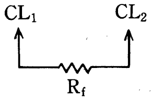


(b) 2MHz clock

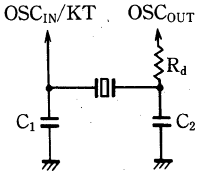


Oscillator KBR-2.0MS: KYOSERA
R_f=1MΩ
R_d=1kΩ
C_1=22pF
C_2=68pF

(2) System clock generator circuit example 2

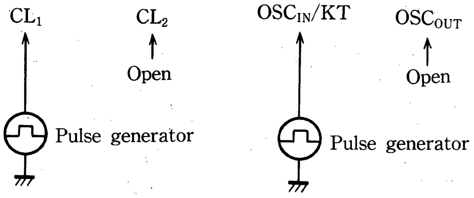


(3) Timer clock generator example



f_{OSC}=32.768kHz
R_d=56Ω
C_1=30pF, C_2=16pF
Crystal: 32.768kHz

(4) External clock input circuit



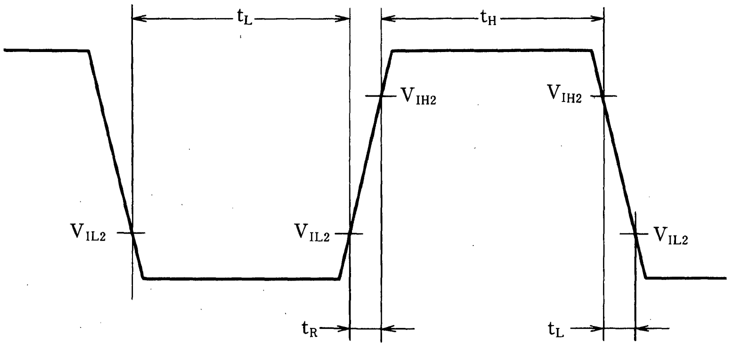
Note: The resistors, capacitors and crystal oscillators should be located as close to the LSI chip as possible to minimize the influence of stray capacitance.

AC Characteristics for External Clock Input Signal

(V_{DD}=2.7 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input rise time	t_R				100	ns	1
					200	ns	2
Input fall time	t_F				100	ns	2
					200	ns	2
Clock HIGH pulse	t_H	V_{DD}=5V±10%	0.125		2.5	μs	1
			0.25		2.5	μs	
		V_{DD}=5V±10%	4			μs	2
			4			μs	
Clock LOW pulse	t_L	V_{DD}=5V±10%	0.125		2.5	μs	1
			0.25		2.5	μs	
		V_{DD}=5V±10%	4			μs	2
			4			μs	

Note 1: Applied to CL_1 pin.
Note 2: Applied to OSK_{IN}/KT pin.



External clock timing

Pin Functions

(1) GND, V_{DD} (Power supply inputs)

The V_{DD} pin should be positive (2.7 to 5.5V) with respect to GND.

The GND pin is the resference power supply for the LSI.

(2) V_{RH}, V_{RL} (A/D converter inputs)

The V_{RL} pin is a GND pin for the A/D converter.

The V_{RH} pin provides the reference voltage V_{RH} for the A/D converter.

The current consumption and operating accuracy of the A/D converter must be changed according to the case where the V_{RL} pin is used to be left open or provide GND level.

(3) ACL (Reset input)

The ACL pin is used to reset the LSI.

The LSI should be reset with a transition of two instruction cycles after the rising edge of ACL.

Applying a Low level signal to the ACL pin starts execution of the program at field 0, page 0, step 0 after a transition of t_{ACL}.

The device is automatically reset when power on. But it is recommended to apply a capacitor between ACL pin and V_{DD} pin in order to prevent from external noise which affects the ACL circuit.

(4) KC_0-KC_3 (Analog inputs)

Executing the KCTA instruction transfers the KC input data to the accumulator A_{CC} through input buffers.

The KC input pin also provides analog input signals given to the A/D conversion block.

(5) KH, KI, OSC_{IN}/KT, KL (Inputs)

The KH and KI input pins are connected to the noise debounce circuit, and the KL and OSC_{IN}/KT input pins to input buffers.

The KL, OSC_{IN}/KT, KH and KI should be loaded into the A_0, A_1, A_2 and A_3 bits of the accumulator A_{CC} upon execution of KLTA instruction.

The noise debounce circuit does not accept the pulse input shorter than two instruction cycle width.

(6) Z_0-Z_{15} (Input/output)

The Z_0-Z_{15} can be controlled with the output latch F/F to be set or reset by instructions.

When used for the inputs, the Zi should be used with the outputs to be pulled down, and the input mode of Zi specified by lower 4 bits of B register B_L can be tested by instructions.

The Zi pin transfers analog signals into the compator of A/D converter.

The Zi pin transfers analog signals into the comparator of A/D converter.

(7) P_0-P_3 (Input/output)

The P_0-P_3 are three-state I/O pins.

Executing the ATP instruction transfers the accumulator A_{CC} to the output latch F/F which is loaded into the P_0-P_3.

The P_0-P_3 can be loaded into the A_{CC} upon execution of the PTA instruction. Then the P_0-P_3 remain high impedance.

(8) Q_0-Q_3 (Input/output)

Executing the ATQ instruction transfers the accumulator A_{CC} to the output latch F/F which is loaded into the Q_0-Q_3.

While, the Q_0-Q_3 can be loaded into the A_{CC} upon execution of the QTA instruction. Then, the Q_0-Q_3 should be used to reset the output latch F/F with the outputs to be pulled down.

(9) $R0_0$-$R0_3$, $R1_0$-$R1_3$, $R2_0$-$R2_3$, $R3_0$-$R3_3$ (Input/output)

Upon execution of the ATR instruction, the R0i-R3i outputs the accumulator A_{CC} specified by the lower 4 bits (BL) of the B register. While, executing the MTR instruction provides the RAM contents specified by the B register from the R0i-R3i.

The R3i-R0i are loaded into the A_{CC} by an RTA instruction. Then, the output port resets the output latch F/F to be pulled down.

8-bit data transfer can be performed in parallel among the R [1] i, R [0] i and A_{CC} or X register by the RTAX or AXTR instruction.

(10) F (Tone output)

The F output pin is used for a tone output as well as a general-purpose output.

(11) f_{OUT}

The f_{OUT} pin outputs the signal in synchronizing with the system clock f_S.

Note: When the I/O pins Z, Q, R are used for the outputs, the buffer with a pull down resistor can be replaced by the CMOS buffer with a mask option.

The output buffer with a pull-down resistor can also be structured by an open-drain transistor.

However, the $R0_3$-$R0_0$, $R1_3$-$R1_0$ and $R2_0$ can not be replaced by the COMS buffer.

Hardware Configuration

(1) Program counter and stack

The program counter (PC) is used to address a ROM location.

The PC consists of 12 bits (SM579: 13 bits) allocated 3 bits (P_U) (SM579: 4 bits) to the field specification of ROM, 2 bits (P_M) to the page specification, and 7 bits (P_L) to the step specification. The P_M is a binary counter and the P_L is a polynomial counter for the page specifications.

The SM578 is unable to use the area of the ET 1 (feilds 8 to 11).

The SR consists of 6 stages available for up to 6 levels of subroutine nesting.

(2) Program memory (ROM)

The ROM is used to store programs.

The SM578 has a 4096×9-bit ROM, and the SM579 has a 6096×9-bit. The ROM consists of 8 fields (SM579: 12 fields)×4 pages×127 steps.

When power on with the ACL to be reset, the program starts execution at field 0, page 0, step 0.

Fig. 1 shows the example of a jump to the ROM address by a ROM address instruction.

The TR instruction is used to jump within a page, and the TL instruction is used to jump to any address. A subroutine jump is executed by the TLS or TRS instruction.

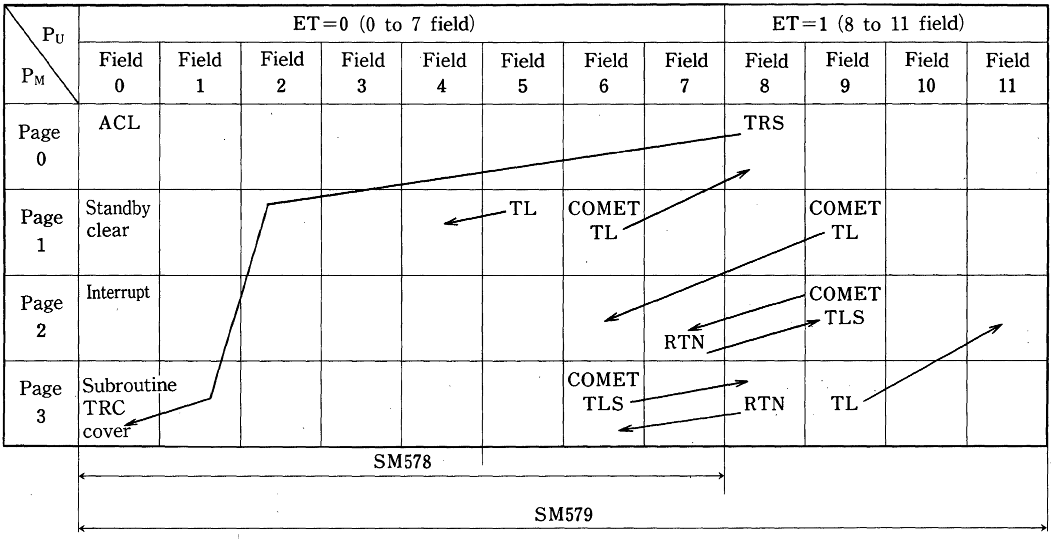
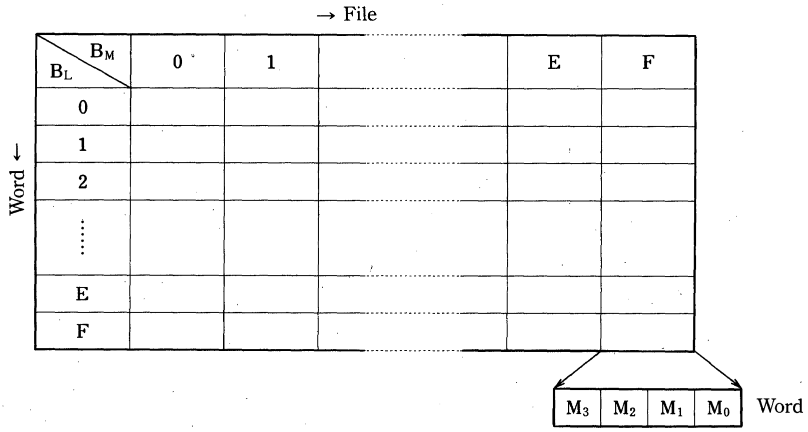


Fig. 1 ROM configuration



(The SM578 has a configuration allocated from 0 through B)

Fig. 2 RAM configuration

However, when the ET value may change due to a jump or subroutine jump on the SM579, execute the TL or TLS instruction following the COMET instruction.

(3) Data memory (RAM) and B register

The RAM is used to store data.

The SM578 has a 768-bit RAM organized as 12 ×16×4 which consists of 12 files as shown in Fig. 2. The SM579 has 16 files of RAM organized as 16×16×4. A file consists of 16 words×4 bits.

The RAM address is specified by a B register which consists of a 4-bit B_M for the file specification and a 4-bit B_L for the word specification.

(4) Accumulator A_{CC}, X and G registers

The accumulator A_{CC} is a 4-bit general-purpose register which transfers numerics and data. The A_{CC} can be decremented and shifted to the left in combination with the carry flag (C).Furthermore, the A_{CC} together with the arithmetic and logic unit (ALU), a carry flag (C) and RAM executes arithmetic operations.

It also transfers data to I/O ports.

The X register is a 4-bit register which can be used for a temporary register. It is incremented by instructions. It performs, in conjunction with the A_{CC}, logical sum and logical product.

An 8-bit parallel data of the A_{CC} and X register can be transferred to R [0] and R [1], a G register or a counter/timer.

On the other hand, each data on R0i and R1i, a G register or a countere timer can also be transferred to the A_{CC} and X register with an 8-bit parallel data.

The G register is an 8-bit register which is used for A/D conversion or comparison of analog signals.

(5) Arithmetic and logic unit (ALU), carry flag (C)

The arithmetic and logic unit (ALU) performs binary addition in conjunction with a RAM, a carry flag C and an accumulator A_{CC}.

The carry flag C latches the data incremented by the ADC or ADCS instruction.

(6) SB register

The SB register is an 8-bit register used for a save register.

(7) P, Q, R [3]-R [0], Z (Output latch registers)

Registers P, Q, R0, R2, R3, Z connect with the output latch F/F.

The accumulator A_{CC} can be transferred to registers P, Q, R [3]-R [0], and an 8-bit data of the A_{CC} and X register can be transferred, at the same time, to the output latch registers R [0] and R [1].

(8) System clock generator circuit

The system clock generator circuit generates a system clock of a base frequency input from the CL_1 pin divided by 4 or 8.

The system clock speed can be controlled by a program. If it is not required for high speed operation, the system clock can be switched to the low speed in order to save the power consumption. This function is also applicable to the case where the power supply is replaced by a battery backup power.

The system clock when reset is equivalent to the base frequency divided by 8.

The system clock fs is used to determine the instruction execution cycle, and the system clock cycle should be identical to the instruction execution cycle. However, the instruction execution cycle of a two word instruction should be two times as long as a one-word instruction.

(9) Counter/timer

A timer 1 and timer 2 are 8-bit counter/timers. The data incremented by a count up is latched into the flags TF1 and TF2 to be used for an interrupt request. Executing the TTF1 and TTF2 flags checks the flags TF1 and TF2.

- Timer 1: An 8-bit data of the A_{CC} and X register can be transferred to the timer 1. To the contrary, the timer 1 can be read out from the A_{CC} and X register.
- Timer 2: The timer 2 contains a modulo register (MR register). The contents of the modulo register (TM) are loaded into the timer 2 each time the register is incremented by one.
  An 8-bit data can be loaded into the MR register by instructions, and executing the next instruction cycle transfers the data to the timer 2 which can be read out from the A_{CC} and X register.

The count up pulses of a counter/timer include $(1/2)^9 f_S$, $(1/2)^3 f_S$, $(1/2)^6 f_T$ and f_T, under conditions of a system clock f_s and KT input pulse f_T, which can be selected by a program.

A carry output of one counter can be used for a count up pulse of the other counter, and it can be counted up by a TCTRL instruction.

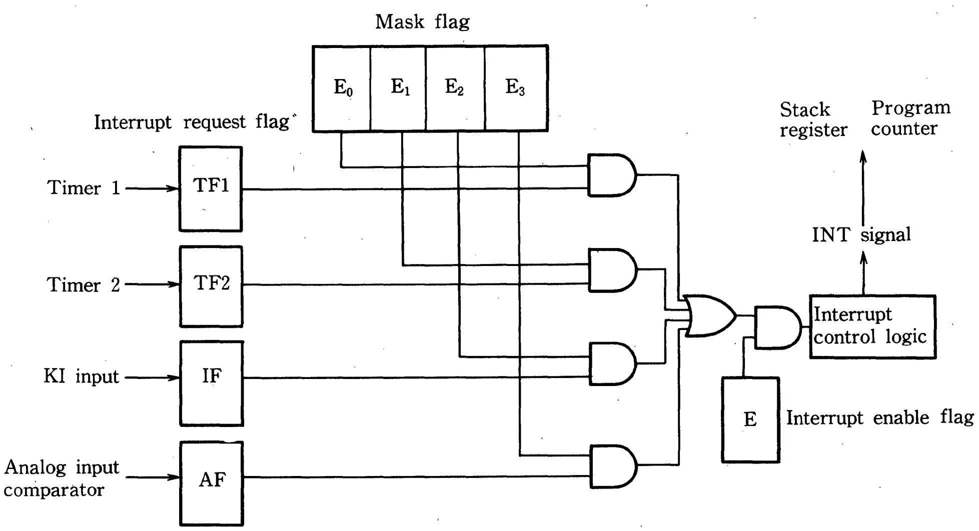


Fig. 3 Interrupt block

Table 1 Interrupt jump address

Interrupt request flag	Jump destination address			Priority
	Field	Page	Step (PL)	
Timer 1 carry (TF1)	0	2	0 (00)	1
Timer 2 carry (TF2)	0	2	2 (60)	2
KI input (IF)	0	2	4 (78)	3
Analog input comparator (AF)	0	2	6 (7E)	4

(10) Interrupt

A KI input, the timer 1 and timer 2 carry and an analog input are available for the interrupt request, and the interrupt request flags include the IF, TF1, TF2 and AF flags.

The interrupt block consists of the mask flags (E_3, E_2, E_1 and E_0), E flag and interrupt processing circuits.

(See Fig. 3)

Table 1 shows the jump address caused by an interrupt request.

(11) A/D converter

The A/D conversion block consists of an 8-bit D/A converter, a comparator, an AM flag and AF flag.

The KC and Z pins input the analog signals.

Executing the COMP instruction allows the A/D conversion and the large/small comparison automatically. (See Fig. 4.)

The result of A/D conversion is stored in the G register with the interval of 16 instruction cycles after the COMP instruction is executed.

The result of the large/small comparison is stored in the AF flag with the interval of 3 instruction cycles.

The G register is an 8-bit register which can be transferred to the A_{CC} and X register with the GTAX instruction.

The KC_0 pin can also be used for an external interrupt.

The D/A converter generates the voltage V_{REF} according to the contents of the G register.

Assuming that the "n" is placed in the G register as a result of A/D conversion, the analog input voltage may be regarded as a below expression.

$$\frac{256-n}{256} V_{RH} \ (n=0 \text{ to } 255)$$

*V_{RH} is a reference voltage supply from the V_{RH} pin.

When even more strict accuracy is required in the A/D conversion block, an external GND level should be applied to the V_{RL} pin.

The A/D conversion is executed by the comparison among a G register, a D/A converter and a comparator in order.

The large/small comparison is executed by the comparator output V_{REF} according to a G register value and the analog signal of the KC_0. The result of comparison is stored in the AF flag.

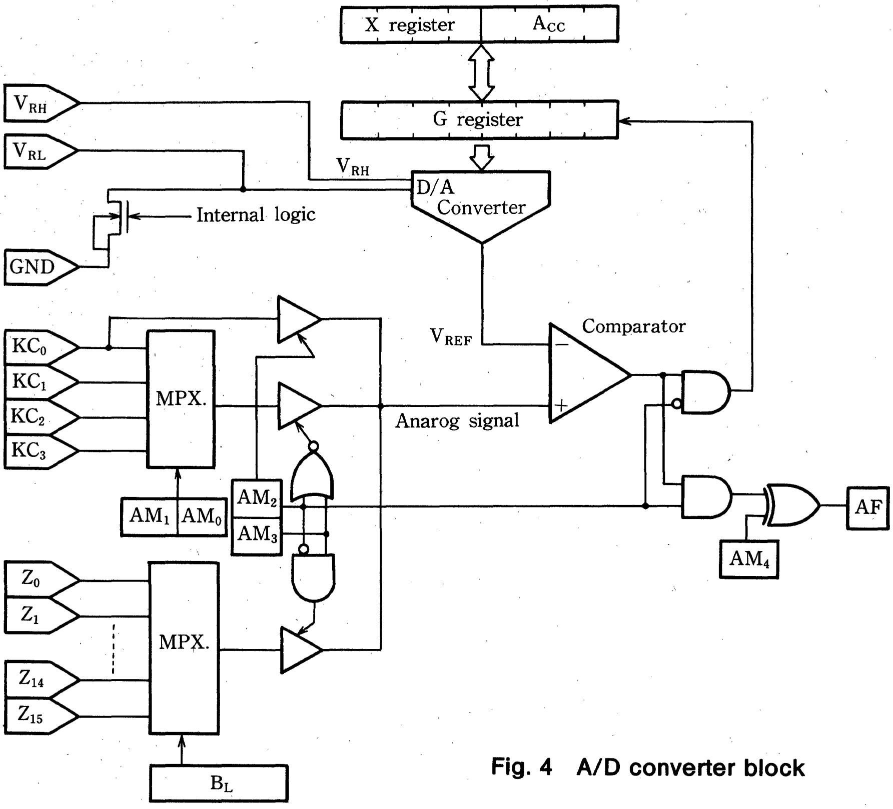


Fig. 4 A/D converter block

(12) Tone output block

The F pin outputs the frequency obtained from a count-up pulse generator circuit.

The pulse frequency can be selected among $(1/2)^5 f_S$, $(1/2)^6 f_S$, $(1/2)^2 f_T$ and $(1/2)^3 f_T$ by programs.

* The f_T is a timer clock frequency input from the OSC_{IN}/KT pin, and the f_S is a system clock frequency.

(13) Standby mode

To reduce power consumption, the device is placed in standby mode, and the program execution is inactivated.

The following two types of standby mode can be selected.

- **Off mode** In the off mode, the system clock generator circuits except for a counter/timer and a count-up pulse generator circuit are inactivated.
- **Hold mode** In the hold mode, the systems except for a system clock generator circuit, a counter/timer and a count-up pulse generator circuit are inactivated.

While in standby mode, if a KH input or an interrupt request from an unmasked KI, timer 1 or timer 2, the device exits standby mode and starts program execution.

(14) Reset function (ACL)

The device is reset with the interval of two instruction cycles from the rising edge of the ACL pin.

Immediately after the reset is cleared, the device starts execution of the program at the program counter 0.

In case the noise may harm the ACL operation, apply a capacitor between ACL pin and V_{DD} pin.

(15) Serial I/O

The serial I/O consists of an 8-bit shift register, a 3-bit counter and a 6-bit mode flag, which have the following features.

- Selectable either an 8-bit or a 4-bit transfer system
- Interrupt request available at the end of transfer
- Selectable transfer clock among a system clock, a timer 2 output or an external clock.
- Connectable to multiple chips.
- Usable in standby mode.
- An 8-bit shift register replaceable by the R/W register when the serial I/O is not used.

Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TR x	100-17F	Jump (within a page) $P_L \leftarrow I_6$-I_0
TL x	0F0-0F7	Jump
	000-1FF	$P_U \leftarrow I_{11}$-I_9, $P_M \leftarrow I_8$-I_7, P_L-I_6-I_0
MTPL	08A	Jump (within a page) ($P_L \leftarrow A_2$-A_0 M_3-M_0)
TRS x	180-1FF	CALL (Indirect address)
JUMP	000-1FF	$P_U \leftarrow 1$, $P_M \leftarrow I_8$, I_7, $P_L \leftarrow I_6$-I_0, if DI=1
TLS x	0F8-0FF	Call to subroutine
	000-1FF	$P_U \leftarrow I_{11}$-I_9, $P_M \leftarrow I_8$-I_7, $P_L \leftarrow I_6$-I_0
RTN	0C0	Return
RTNS	0C1	Return and skip
RTNI	0C2	Return from interrupt
COMET	08B	$ET \leftarrow \overline{ET}$

(2) Data transfer instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
LAX x	040-04F	Acc$\leftarrow I_3$-I_0, Skip if last instruction is LAX
WLAX x	087	$X \leftarrow I_7$-I_4, Acc$\leftarrow I_3$-I_0
	000-0FF	
LBMX x	0E0-0EF	$B_M \leftarrow I_3$-I_0
LBLX x	020-02F	$B_L \leftarrow I_3$-I_0
STXI x	050-05F	$M \leftarrow I_3$-I_0, $B_L \leftarrow B_L+1$, Skip if CY=1
EXCI x	070-077	$M \leftrightarrow$Acc, $B_M \leftarrow B_M \oplus I_2$-$I_0$ $B_L \leftarrow B_L+1$, Skip if CY=1
EXCD x	078-07F	$M \leftrightarrow$Acc, $B_M \leftarrow B_M \oplus I_2$-$I_0$ $B_L \leftarrow B_L+F_H$, Skip if CY=1
EXC x	068-06F	$M \leftrightarrow$Acc, $B_M \leftarrow B_M \oplus I_2$-$I_0$
LDA x	060-067	Acc$\leftrightarrow M$, $B_M \leftarrow B_M \oplus I_2$-$I_0$
STR	09E	$M \leftarrow$Acc
EXAX	0A6	Acc$\leftrightarrow X$
ATX	0AE	$X \leftarrow$Acc
GTAX	0BD	$X \leftarrow G_7$-G_4, Acc$\leftarrow G_3$-G_0
AXTG	08D	G_7-$G_4 \leftarrow X$, G_3-$G_0 \leftarrow$Acc
	0BD	
XBLA	0B3	$B_L \leftrightarrow$Acc
BLTA	0B1	Acc$\leftarrow B_L$
XBMA	0B2	$B_M \leftrightarrow$Acc
BMTA	0BA	Acc$\leftarrow B_M$
XBSB	084	B$\leftrightarrow$SB
BTSB	085	SB$\leftarrow$B
SAG	08D	$B_M \leftarrow 0$, only next step
SGL x	0C8-0CF	$B_M = I_2$-I_0, BL$=F_H$, only next step
ATIM	0B4	Ei$\leftarrow$Ai (i=3 to 0)

(3) Arithmetic instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
ADX x	000-00F	Acc←Acc+I_3-I_0, Skip if CY=1
ADA	09A	Acc←Acc+A_H
ADD	090	Acc←Acc+M
ADS	091	Acc←Acc+M, Skip if CY=1
ADC	092	Acc←Acc+M+C, C←CY
ADCS	093	Acc←Acc+M+C, C←CY, Skip if CY=1
ADBL	0BB	B_L←Acc+B_L
AND	0A1	Acc←Acc$\wedge$x
OR	0B0	Acc←Acc$\vee$x
COMA	086	Acc←$\overline{\mathrm{Acc}}$
ROT	09B	C←A_3←A_2←A_1←A_0←C
DECA	09F	Acc←Acc+F_H, Skip if CY=0
INCX	0A7	X←X+1, Skip if CY=1
INBL	0A3	B_L←B_L+1, Skip if CY=1
DEBL	0AB	B_L←B_L+F_H, Skip if CY=0
INBM	0A2	B_M←B_M+1, Skip if CY=1
DEBM	0AA	B_M←B_M+F_H, Skip if CY=0

(4) Test instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TAX x	010-01F	Skip if Acc=I_3-I_0
TBA x	0D4-0D7	Skip if Ai=1 (i=3 to 0)
TM x	0D0-0D3	Skip if Mi=1 (i=3 to 0)
TAM	096	Skip if Acc=M
TXM	0B6	Skip if X=M
TBLX x	030-03F	Skip if B_L=I_3-I_0
TC	0B8	Skip if C=1
TS	0B9	Skip if S=1
TIF	0C7	Skip and reset if IF=1
THAF	0C6	Skip and reset if HF=1 (AM_5=0) if AF=1 (AM_5=1)
TTF1	0C5	Skip and reset if TF_1=1
TTF2	0C4	Skip and reset if TF_2=1
TQZ	0A0	Skip if Q=0
TZ	080	SKip if Z [B_L]=1

(5) Bit manipulation instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
SM x	0DC-0DF	Mi←1 (i=3 to 0)
RM x	0D8-0DB	Mi←0 (i=3 to 0)
SC	099	C←1
RC	098	C←0
SS	0A9	S←1
RS	0A8	S←0
IE	095	E←1
ID	094	E←0

(6) I/O instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
ATQ	08E	Q←Acc
QTA	0BE	Acc←Q
ATP	08F	P←Acc, P_M←1
PTA	0A4	Acc←P, P_M←0
ATR	08C	R [B_L]←Acc
RTA	0C3	Acc←R [B_L]
AXTR	0AC	R [1]←X, R[0]←Acc
RTAX	0AD	X←R [1], Acc←R [0]
MTR	09C	R [B_L]←M
KCTA	0BC	Acc←KC
KITA	0BF	A_3←K_1, A_2←K_H, A_1←K_T, A_0←K_L
SZ	083	Z [B_L]←1
RZ	082	Z [B_L]←0
SF	089	F←1, FM_1←A_1, FM_0←A_0
RF	088	F←0

(7) Timer control instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TCTRL x	0AF	DM←I_7-I_0
	000-0FF	
STM1	097	TIMER1←X, Acc
LTM1	09D	X, Acc←TIMER1
STM2 x	0B7	TIMER2←MR, MR←I_7-I_0
	000-0FF	
LTM2	081	X, Acc←TIMER2

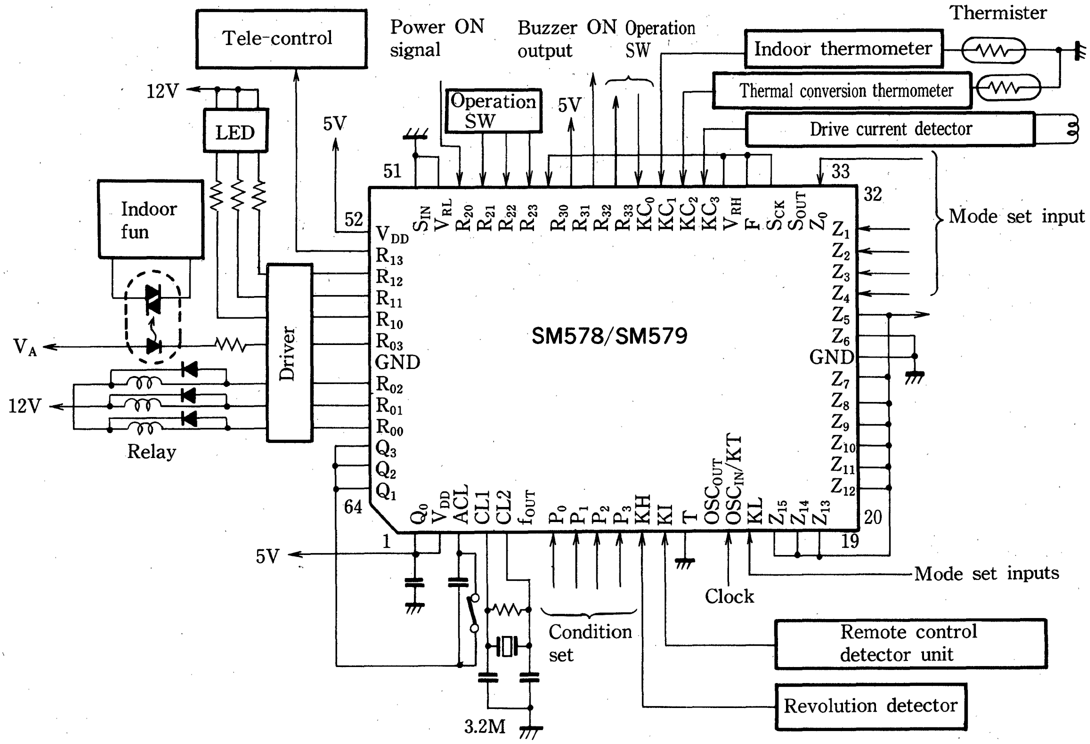
(8) A/D conversion instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
COMP	0A5	AM_5←$A_2 \wedge A_1$, A_4←$A_2 \wedge A_0$, AM_3←A_3 AM_2←A_2, AM_1←A_1, AM_0←A_0 A/D Conversion or comparison

(9) Standby instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
CCTRL	0B5	CM_2←A_2, CM_1←A_1, CM_0←A_0 Standby mode if A_3=0

System Configuration Example (Air conditioner)



SM5E4 4-Bit Microcomputer (Controller with Multi I/O Ports)

■ Description

The SM5E4 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a ROM, a RAM, I/Oports, a serial interface, a timer/event counter in a single chip.

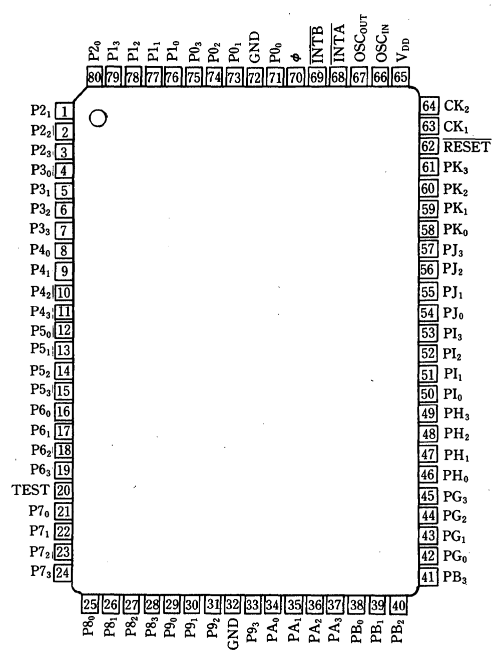
Provided with five kinds of interrupt and a subroutine stack function using the RAM area, it allows data transfer in byte unit.

Operated from 3 to 5V single power supply with high speed, this microcomputer is applicable to many applications from a battery back-up system to a high performance system. Especially, it is best suited to systems required for multiple control signals, due to equipped with 70 I/O pins.

■ Features

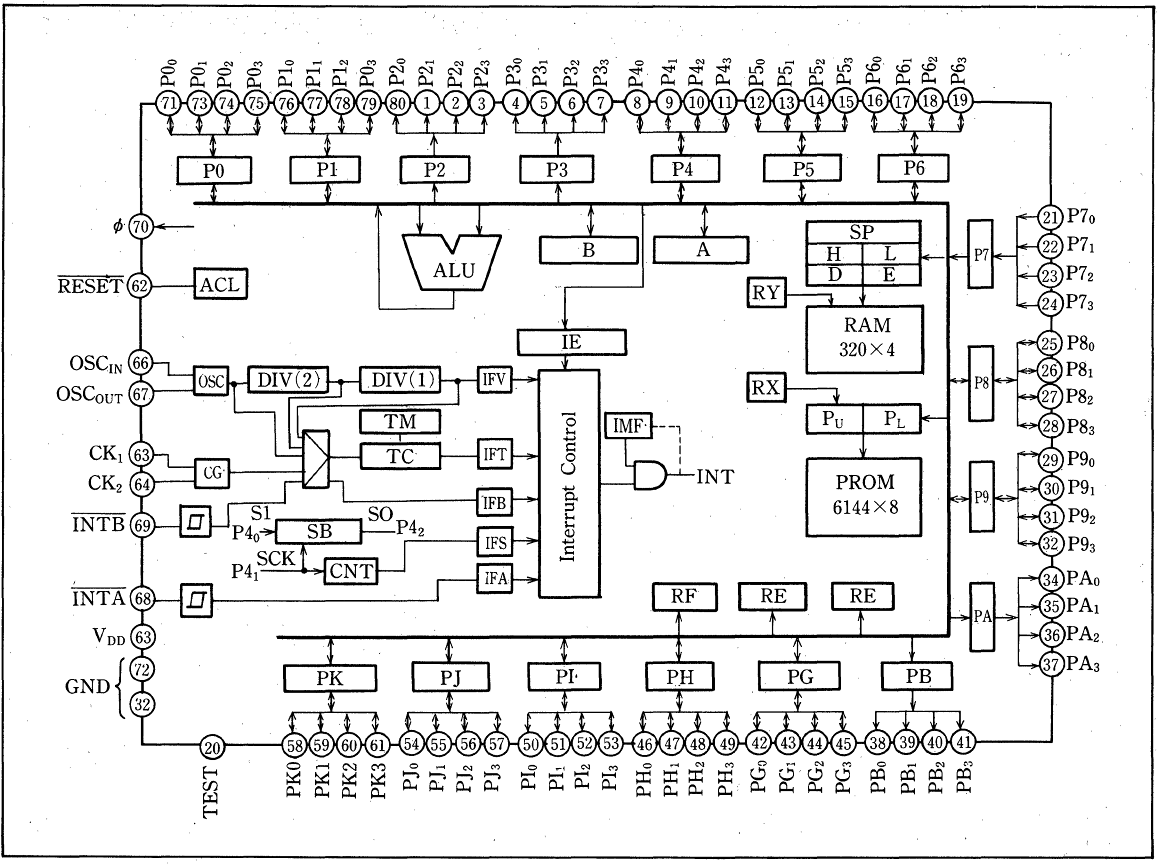
1. CMOS process
2. ROM capacity: 6,144×8 bits
3. RAM capacity: 320×4 bits
4. Instruction set: 98
5. Subroutine stack: using RAM area
6. Instruction cycle
   1.7 μs (MIN.) at 5V power supply
   3.1 μs (MIN.) at 3V power supply
7. Interrupts
   External interrupts: 2
   Internal interrupts: 3
8. Input/output ports
   I/O ports: 48
   Input ports: 6
   Output ports: 16
9. 8-bit serial I/O
10. 8-bit counter/timer: 1 set
11. Standby function
12. Expandable external data ROM/RAM
13. 8-bit parallel I/O
14. On-chip crystal oscillator and clock divider circuit
15. On chip system clock CR oscillator
16. Single power supply: 2.7 to 5.5V
17. 80-pin QFP (QFP80-P-1420)

■ Pin Connections



Top View

Block Diagram



Pin Discription

Symbol	I/O	Function
$P0_0$-$P0_3$, $P1_0$-$P1_3$ $P4_0$-$P4_3$, $P5_0$-$P5_3$ $P6_0$-$P6_3$, $P8_0$-$P8_3$ $P9_0$-$P9_3$, PG_0-PG_3 PH_0-PH_3, PI_0-PI_3 PJ_0-PJ_3, PK_0-PK_3	I/O	I/O ports
$P2_0$-$P2_3$, $P3_0$-$P3_3$ PA_0-PA_3, PB_0-PB_3	O	Output ports
$P7_0$-$P7_3$	I	Input ports
$\overline{INTA}$, $\overline{INTB}$	I	Interrupt input ports
CK_1, CK_2		System clock CR oscillator
OSC_{IN}, OSC_{OUT}		Crystal oscillator
φ	O	Synchronous clock output port
V_{DD}, GND		Power supply
TEST		Test (normally connected to GND)
$\overline{RESET}$	I	Reset input port

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−0.3 to +7.5	V	1
Input voltage	V_I	−0.3 to V_{DD}+0.3	V	
Output voltage	V_O	−0.3 to V_{DD}+0.3	V	
Output current	I_O	40	mA	2
Operating temperature	Topr	−20 to +70	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.
Note 2: Sum of current output from (or flowing into) output pin.

Recommended Operating Conditions

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Supply voltage	V_{DD}		2.7		5.5	V	
Crystal oscillation frequency	fosc			32.768		kHz	1
Reference clock oscillation frequency	f	V_{DD}=5V	1.7		2.3	MHz	
		V_{DD}=3V	0.7		1.3		

Note 1: Oscillation start time: within 10 seconds.

DC Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			0.7V_{DD}		V_{DD}	V	1
	V_{IL1}			0		0.3V_{DD}	V	
	V_{IH2}			V_{DD}−0.5		V_{DD}	V	2
	V_{IL2}			0		0.5	V	
Input current	I_{IH}	V_{IN}=0V	V_{PP}=5.0V±10%	20		200	µA	1
				2		200		
Output current	I_{OH1}	V_{OH}=V_{DD}−0.5V		50			µA	3
	I_{OL1}	V_{OL}=0.5V		250			µA	
	I_{OH2}	V_{OH}=V_{DD}−0.5V		100			µA	4
	I_{OL2}	V_{OL}=0.5V		500			µA	
	I_{OH3}	V_{OH}=V_{DD}−0.5V	V_{DD}=5.0V±10%	400			µA	5
				100				
	I_{OL3}	V_{OL}=0.5V	V_{DD}=5.0V±10%	1.6			mA	
				0.5				
Current consumption	I_{OP}	f=1MHz	V_{DD}=5.0V±10%		1		mA	6
		f=0.5MHz	V_{DD}=3.0V±10%		0.3			
	I_{ST}	Standby current	V_{DD}=5.0V±10%		50		µA	7
			V_{DD}=3.0V±10%		12			
					5			8

Note 1: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$.
$P8_0$-$P8_3$, $P9_0$-$P9_3$. (in input mode)
$P7_0$-$P7_3$, $\overline{RESET}$, PG_0-PG_3, PH_0-PH_3, PI_0-PI_3, PJ_0-PJ_3, PK_0-PK_3.
Note 2: Applied to pins CK_1, OSC_{IN}, $TEST_0$, INTA, INTB
Note 3: Applied to pin CK_2
Note 4: Applied to pin ϕ
Note 5: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$.
$P8_0$-$P8_3$, $P9_0$-$P9_3$. (In output mode)
$P2_0$-$P2_3$, $P3_0$-$P3_3$, PA_0-PA_3, PB_0-PB_3.
PG_0-PG_3, PH_0-PH_3, PI_0-PI_3, PJ_0-PJ_3, PK_0-PK_3.
Note 6: No-load condition
Note 7: When crystal oscillation circuit is activated under no load conditions.
Note 8: When crystal oscillation circuit is inactivated under no load conditions. OSC_{IN} pin should be connected to GND.

■ AC Characteristics

(1) Clock characteristics

(V_{DD}=2.7 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Reference clock oscillation frequency (CR oscillation)	f_{CR}	V_{DD}=5V±10%, R=12kΩ±5%	1.7	2.0	2.3	MHz	
		V_{DD}=3V±10%, R=39kΩ±5%	0.5	0.75	1.0		
		R=39kΩ±5%	0.5	0.75	1.0		
Reference clock input frequency (CK_1)	f_K	V_{DD}=5.0±10%	0.25		2.3	MHz	
			0.25		1.0		
CK_1 input rise time	t_{KR}				500	ns	
CK_1 input fall time	t_{KF}				500	ns	
CK_1 input high range	t_{KH}	V_{DD}=5.0V±10%	0.3			μs	
			0.6				
CK_1 input low range	t_{KL}	V_{DD}=5.0V±10%	0.3			μs	
			0.6				
OSC crystal oscillation frequency	f_{OSC}			32.768		kHz	
OSC_{OUT} input cycle time	t_{fY}		2			t_{CYC}	1
OSC_{OUT} input rise time	t_{fR}				500	ns	
OSC_{OUT} input fall time	t_{fF}				500	ns	
OSC_{OUT} input high range	t_{fH}		1			t_{CYC}	1
OSC_{OUT} input low range	t_{fL}		1			t_{CYC}	1

Note 1: t_{CYC}: Cycle time of one fourth the reference clock oscillation frequency.

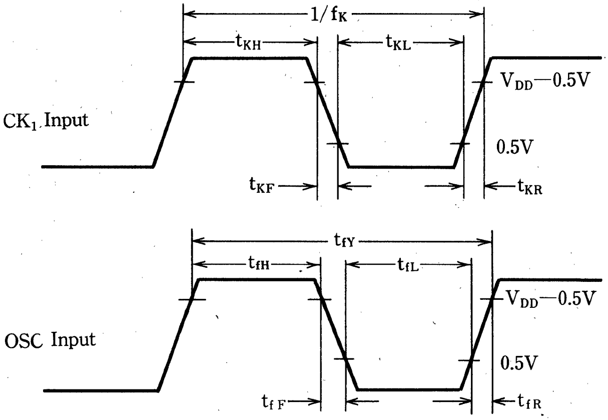


Fig. 1 Clock timing

(2) Interrupt input

(V_{DD}=2.7 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
$\overline{\text{INTA}}$ high range	t_{AH}		2			t_{CYC}	1
$\overline{\text{INTA}}$ low range	t_{AL}		2			t_{CYC}	
$\overline{\text{INTB}}$ high range	t_{BH}		2			t_{CYC}	
$\overline{\text{INTB}}$ low range	t_{BL}		2			t_{CYC}	

Note 1: t_{CYC}: Cycle time of one fourth the reference clock oscillation frequency.

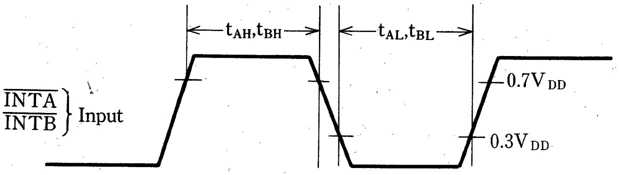


Fig. 2 Interrupt input timing

(3) External Serial Input Clock

(V_{DD}=2.7 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
SCK cycle time	t_{SY}	V_{DD}=5.0V±10%	1			t_{CYC}	1
			1				
SCK high range	t_{SH}	V_{DD}=5.0V±10%	1/2			t_{CYC}	
			1/2				
SCK low range	t_{SL}	V_{DD}=5.0V±10%	1/2			t_{CYC}	
			1/2				
SCK rise time	t_{SR}				500	ns	
SCK fall time	t_{SF}				500	ns	
Reset pulse width (low)	t_{RST}		300			ns	

Note 1: t_{CYC}: Cycle time of one fourth the reference clock oscillation frequency.

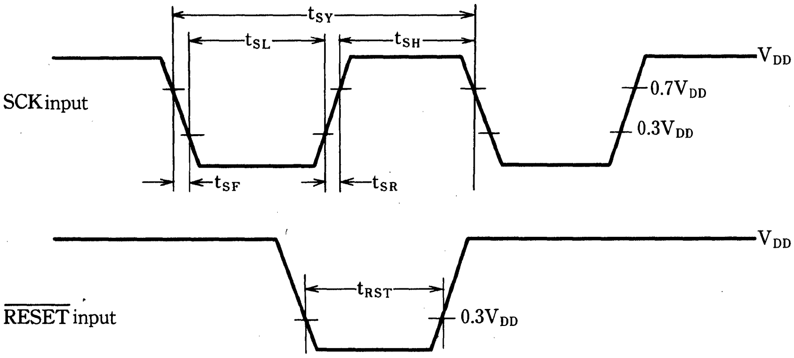


Fig. 3 External serial input clock timing

■ Hardware Configuration

(1) Program counter (PC) and stack

The program counter consists of a 7-bit page address register (P_U) and 6-bit binary counter (P_L) used to specify the steps within a page.

The stack pointer (SP) is a register which holds the starting address of the stack area of RAM space.

(2) Program memory (ROM)

The on-chip ROM has a configuration of 96 pages×64 steps×8 bits, and stores programs and table data. Fig. 4 shows the ROM configuration.

PU_6	0				1	
$PU_5 \sim PU_4$ / $PU_3 \sim PU_0$	0	1	2	3	0	1
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
A						
B						
C						
D						
E						
F						

Fig. 4 ROM configuration

(3) Data memory (RAM)

Data memory has a 320×4-bit configuration, and is used to store processing data and other information.

Data memory is also used as a subroutine stack. Fig. 5 shows the RAM configuration.

Upper / Lower	RY=0																RY=1			
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
A																				
B																				
C																				
D																				
E																				
F																				

Fig. 5 RAM configuration

Data memory is specified by a 9-bit address, and the RY is placed in the highest bit.

(4) General-purpose register

Registers H and L are 4-bit general-purpose registers. They can transfer and compare data with the A_{CC} on a 4-bit basis. Registers D and E are 4bit registers and can transfer data with the H and L registers on an 8-bit basis.

(5) Clock divider and IFV flag

The SM5E4 contains a crystal oscillator and a 15-stage divider. A real-time clock can be provided by connecting an external crystal oscillator between the oscillator pins.

(6) Timer/event counter

The timer/event counter consists of an 8-bit count register (TC) and an 8-bit modulo register (TM).

The count register (TC) is an 8-bit incremental binary counter. It is incremented by one at the falling edge of its count pulse (CP) input. If the count register overflows, the timer interrupt request flag IFT is set, and the contents of the modulo register (TM) are loaded into the count register.

(7) Serial interface

The serial interface consists of an 8-bit shift register (SB) and a 3-bit counter, which is used to input and output the serial data.

The input and output of serial data is controlled by the serial clock which can be selected with either an internal clock (system clock) or an external clock.

(8) Interrupts

The interrupts include three kinds of internal interrupts and two kinds of external interrupts (see Table 1).

(9) I/O ports and mode register

The SM5E4 has seventeen 4-bit ports (P0-PB, PG-PK), and three mode registers (RD, RE, RF). Data can be transferred between these ports and registers under direct instruction control or indirect L register control.

Ports P0, P1, P4, P5, P8, P9, PG, PH, PI, PJ and PK can be placed in input or output mode, 4-bits at a time.

Ports P2, P3, PA and PB are 4-bit parallel output ports. Port P7 is a 4-bit parallel input port.

Each bit of port P6 can be independently placed

Table 1 Interrupt request

Interrupt request		Int./Ext.	Priority	Interrupt routine start address
INTT	Timer/event counter interrupt	Int.	1	Page 1, Address 0
INTA	External signal INTA interrupt	Ext.	2	Address 2
INTS	Serial I/O interrupt	Int.	3	Address 4
INTB	External signal INTB or frame frequency interrupt	Ext.	4	Address 6
INTV	Divider overflow interrupt	Int.	5	Address 8

in input or output mode by setting or resetting the corresponding bit of mode register RF as follows:

RFi=0: Pin P6i is an input pin.
RFi=1: Pin P6i is an output pin.
(i=0, 1, 2, 3)

Ports (P0, P1), (P2, P3), (P8, P9), and (PA, PB) can be paired for use in data transfer on a byte-by-byte basis. However, port pairs (P2, P3) and (PA, PB) are usable only for output.

(10) Standby mode

Executing the CEND instruction places the device in standby mode. To reduce power consumption, the system clock is inactivated. Standby mode may be cleared with the interrupt request or the $\overline{\text{RESET}}$signal.

(11) Reset function (ACL: Auto Clear)

Applying a Low level signal to the $\overline{\text{RESET}}$pin resets the internal logic of the device.

When the device is reset, it is placed in the following initial state:

- The program starts execution at address 0 and page 0.
- I/O ports are placed in the input mode, mode registers RD, RE and RF are cleared, and all output ports (P2, P3, PA, PB) are cleared to 0.
- All interrupt flags are reset to disable all interrupts.
- The contents of RX and RY are cleared to 0.

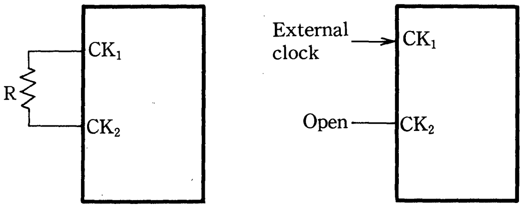
The reset feature is also activated when the power is turned on.

The program starts after the reset condition is cleared upon completion of the master clock period $\times 2^{14}$ clocks.

(12) Clock generator

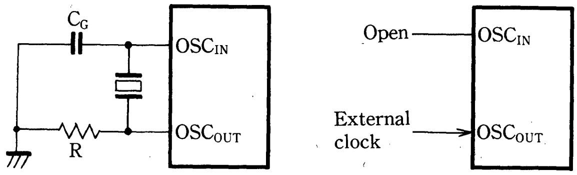
The system clock generator requires an external resistor across pins CK_1 and CK_2. Instead of using on-chip oscillator, an external clock may be applied to pin CK_1. In this case, pin CK_2 should be left open (see Fig. 6).

The real-time clock generator used for the divider circuit is shown in Fig. 7 (a). The system clock ϕ has a frequency of one fourth that of the clock applied to pin CK_1. When applying an external clock to pin OSC_{OUT}, the master clock frequency should be set at more than 8 times that of the external clock.



(a) CR oscillator (b) Extemal clock

Fig. 6 Clock oscillator circuit



(a) Crystal oscillator (b) External clock

Fig. 7 Crystal oscillator circuit

■ Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BF	PL←x(I_5-I_0)

TL xy
(2-byte)
	E0-EF
	00-FF
		PU←x(I_{11}-I_6)
		PL←y(I_5-I_0),PU_6←RX

TRS x	C0-DF	(SP−1), (SP−2), (SP−3),
		(SP−4)←PC
		SP←SP−4
		PU←$(10000)_2$
		PL←x($I_4I_3I_2I_1I_0$0)

CALL xy
(2-byte)
	F0-FF
	00-FF
		(SP−1), (SP−2), (SP−3),
		(SP−4)←PC
		SP←SP−4, PU←x(I_{11}-I_6)
		PL←y(I_5-I_0),PU_6←RX

JBA x (2-byte)	7F 30-3F	PU_5-PU_2←x(I_3-I_0), PU_1, PU_0, PL_5, PL_4←B, PL_3-PL_0←A,PU_6←0
RTN	61	PU, PL←SP, (SP+1), (SP+2), (SP+3) SP←SP+4
RTNS	62	PU, PL←SP, (SP+1), (SP+2), (SP+3) SP←SP+4, skip

RTNI	63	PU, PL, PSW←SP, (SP+1),
		(SP+2), (SP+3)
		SP←SP+4
		IME←1

(2) RAM address instructions

Mnemonic	Machine code	Operation
STL	69	L←A
STH	68	H←A
EXHD	3F	H⟷D L⟷E
LIHL xy (2-byte)	3D 00-FF	H←x(I_7-I_4), L←y(I_3-I_0)

(3) Data transfer instructions

Mnemonic	Machine code	Operation
EX pr	5C-5F	A⟷(pr)
LDX adr (2-byte)	7D 00-FF	A←(adr)
STX adr (2-byte)	7E 00-FF	(adr)←A
EXX adr (2-byte)	7C 00-FF	A←(adr)
LAX x	10-1F	A←x(I_3-I_0)

LIBA xy
(2-byte)
	3C
	00-FF
		B←x(I_7-I_4)
		A←y(I_3-I_0)

LBAT	60	B←ROM (Pu_6-Pu_2, B, A) H
		A←ROM (Pu_6-Pu_2, B, A) L

LDL	65	A←L
LD pr	54-57	A←(pr)
ST pr	58-5B	(pr)←A
EXH	6C	A↔H
EXL	6D	A↔L
EXB	6E	A↔B
STB	6A	B←A
LDB	66	A←B
LDH	64	A←H
PSHBA	28	(SP−1)←B, (SP−2)←A, SP←SP−2
PSHHL	29	(SP−1)←H, (SP−2)←L SP←SP−2

POPBA	38	B←(SP+1), A←(SP),
		SP←SP+2

POPHL	39	H←(SP+1), L←(SP)
		SP←SP+2

STSB	70	SB_H←B, SB_L←A
STSP	71	SP_H←B, SP_L←A
STTC	72	TC←TM
STTM	73	TM_H←B, TM_L←A
LDSB	74	B←SB_H, A←SB_L
LDSP	75	B←SP_H, A←SP_L
LDTC	76	B←TC_H, A←TC_L
LDDIV	77	B←DIV_H, A←DIV_L

(4) Arithmetic instructions

Mnemonic	Machine code	Operation
ADX x	00-0F	A←A+x(I_3-I_0), Skip if Cy=1
ADD	36	A←A+(HL)
ADDC	37	A←A+(HL)+C, C←Cy Skip if Cy=1
OR	31	A←A+(HL)
AND	32	A←A · (HL)
EOR	33	A←A⊕(HL)
ANDB	22	A←A · B
ORB	21	A←A+B
EORB	23	A←A⊕B
COMA	6F	A←$\overline{A}$
ROTR	25	C←A_0←A_1←A_2←A_3←C
ROTL	35	C←A_3←A_2←A_1←A_0←C
INCB	52	Skip if B=F_H, B←B+1
DECB	53	Skip if B=0, B←B−1
INCL	50	Skip if L=F_H, L←L+1
DECL	51	Skip if L=0, L←L−1
DECM adr	79 00-FF	Skip if (adr)=0, (adr)←(adr)−1
INCM adr	78 00-FF	Skip if (adr)=F_H, (adr)←(adr)+1

(5) Test instructions

Mnemonic	Machine code	Operation
TAM	30	Skip if A=(HL)
TAH	24	Skip if A=H
TAL	34	Skip if A=L
TAB	20	Skip if A=B
TC	2A	Skip if C=0
TM x	48-4B	Skip if (HL)x=1
TA x	4C-4F	Skip if Ax=1
TSTT	2B	Skip IFT=1, IFT←0
TSTA	2C	Skip if IFA=1, IFA←0
TSTS	2D	Skip if IFS=1, IFS←0
TSTB	2E	Skip if IFB=1, IFB←0
TSTV	2F	Skip if IFV=1, IFV←0

(6) Bit manipulation instructions

Mnemonic	Machine code	Operation
SM x	40-43	(HL)x←1
RM x	44-47	(HL)x←0
RC	26	C←0
SC	27	C←1
RIME	3A	IME←0
SIME	3B	IME←1
DI x (2-byte)	7F C0-DF	IEF←IEF · $\bar{x}$
EI x (2-byte)	7F E0-FF	IEF+x

(7) I/O instructions

Mnemonic	Machine code	Operation
IN	67	A←P(L)
OUT	6B	P(L), R(L)←A
INA x (2-byte)	7F A0-A9	A←P(x)
OUTA x (2-byte)	7F B0-BF	P(x), R(x)←A

INBA x	7F
	80 : 82
		B←P(x+1)
		A←P(x)

OUTBA x (2-byte)	7F 90-93	P(x+1)←B P(x)←A
SP xy (2-byte)	7A 00-F6	P(y)←P(y)+x
RP xy (2-byte)	7B 00-F6	P(y)←P(y) · x
READ (2-byte)	7F 60	A←P0
WRIT (2-byte)	7F 70	P0←A

READB
(2-byte)
	7F
	61
		B←P1
		A←P0

WRITB
(2-byte)
	7F
	71
		P1←B
		P0←A

INX x	7F 88-8C	A←P'(x)
OUTX x	7F 98-9C	P'(x)←A

Note: P'(x)=PG, PH, PI, PJ and PK

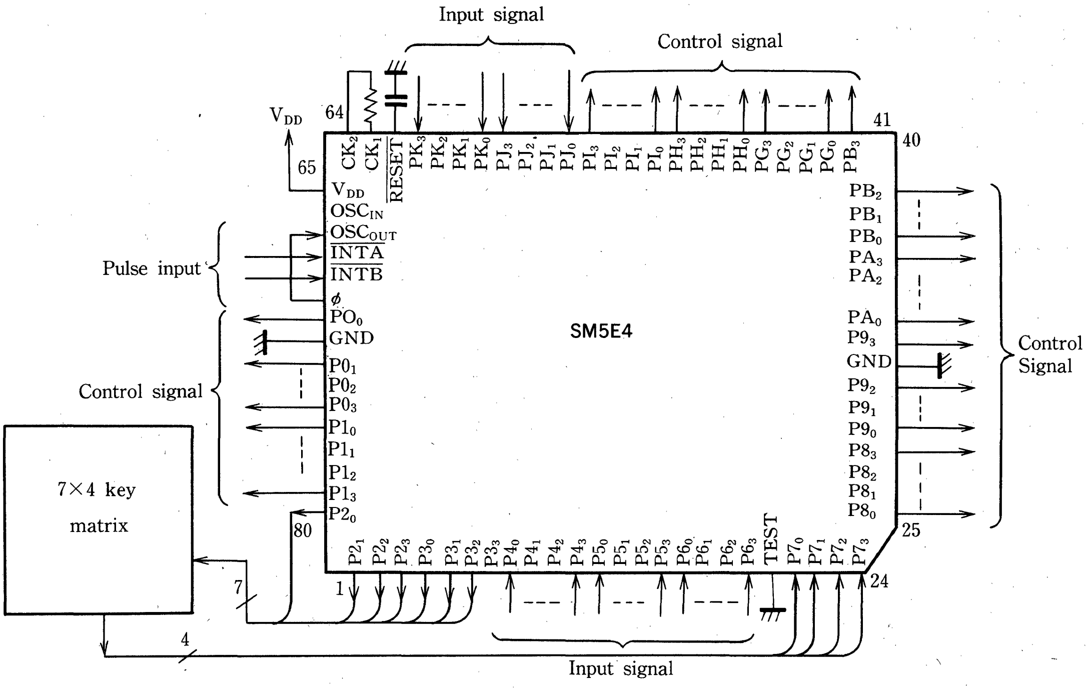
(8) Special instructions

Mnemonic	Machine code	Operation
SIO	3E	Serial I/O start
IDIV (2-byte)	7F 10	DIV←0
SKIP	00	No operation
CEND (2-byte)	7F 00	Stop CR oscillator and system clock
HALT (2-byte)	7F 01	Stop only system clock

LAP xy
(2-byte)
	7F
	20-25
		RX←x(I_0)
		RY←y(I_2)

Note: The machine code consists of 8-bits including I_7, I_6, I_5, I_4, I_3, I_2, I_1 and I_0

■ System Configuration Example



LU5E4P0P 4-Bit Microcomputer (Controller with On-Chip OTPROM)

Description

The LU5E4P0P is a CMOS 4-bit microcomputer which integrates a 6,144×8-bit OTPROM (One Time PROM), a 68 I/O port, a serial I/O, and a timer/counter in a single chip.

Provided with five kinds of interrupt and a subroutine stack function using the RAM area, it allows a data transfer in byte unit.

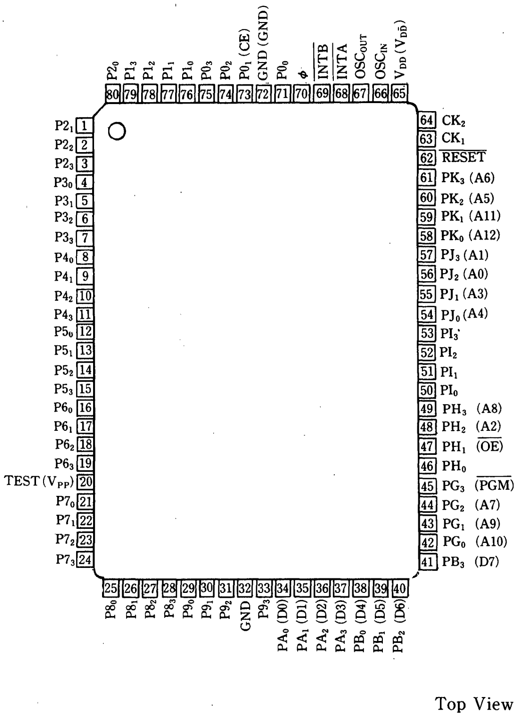
Operated from 5V single power supply with high speed, this microcomputer is applicable to many applications from a hand-held system to a high performance system.

The differences between the LU5E4P0P and SM5E4 are the supply voltage range and the current consumption.

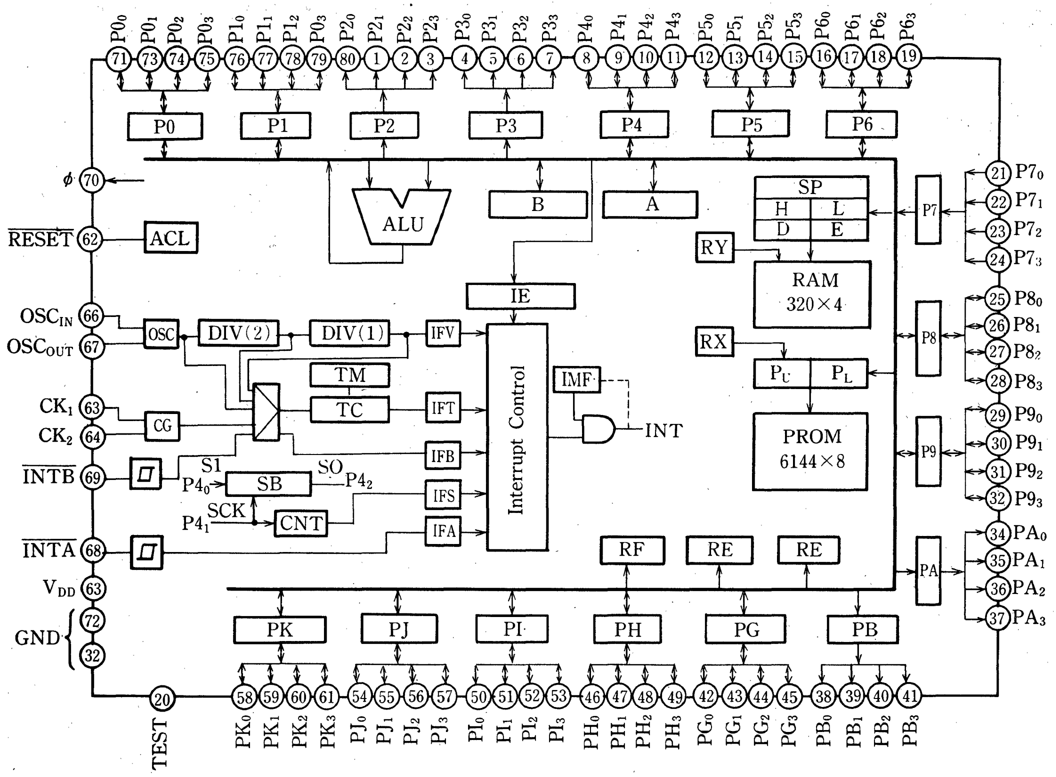
Features

1. CMOS process
2. OTPROM capacity: 6,144×8 bits
3. RAM capacity: 320×4 bits
4. Instruction set: 98
5. Subroutine stack: using RAM area
6. Instruction cycle: 1.6 μs (MIN.)
7. Interrupts
   External interrupts: 2
   Internal interrupts: 3
8. Input/output ports
   I/O ports: 48
   Input ports: 4
   Output ports: 16
9. Timer/counter: 1 set
10. On-chip crystal oscillator and CR oscillator circuits
11. Standby function
12. Expandable external data ROM/RAM
13. 8-bit parallel I/O
14. 80-pin QFP (QFP80-P-1420)

Pin Connections



Block Diagram



*Pin numbers apply to the case where the device is used in the microcomputer mode.
The address, data and control pins are used for switching with some pins described above.

Pin Description

(1) Microcomputer mode

Symbol	I/O	Function
$P0_0$-$P0_3$, $P1_0$-$P1_3$ $P4_0$-$P4_3$, $P5_0$-$P5_3$ $P6_0$-$P6_3$, $P8_0$-$P8_3$ $P9_0$-$P9_3$, PG_0-PG_3 PH_0-PH_3, PI_0-PI_3 PJ_0-PJ_3, PK_0-PK_3	I/O	I/O ports
$P2_0$-$P2_3$,$P3_0$-$P3_3$ PA_0-PA_3, PB_0-PB_3	O	Output ports
$P7_0$-$P7_3$	I	Input ports
$\overline{INTA}$, $\overline{INTB}$	I	Interrupt input ports
CK_1, CK_2		System clock CR oscillator
OSC_{IN}, OSC_{OUT}		Crystal oscillator
ϕ	O	Synchronous clock output port
V_{DD}, GND		Power supply
TEST		Test (normally connected to GND)
$\overline{RESET}$	I	Reset input port

(2) PROM program mode

Symbol	I/O	Function
A_0-A_{12}	I	Address input ports
D_0-D_7	I/O	Data I/O ports
$\overline{CE}$	I	Chip enable input port
$\overline{OE}$	I	Output enable input port
$\overline{PGM}$	I	Program enable input port
V_{PP}		Program power supply
V_{DD}, GND		Power supply

■ Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−0.3 to +7.5	V	1
Input voltage	V_I	−0.3 to V_{DD}+0.3	V	
Output voltage	V_O	−0.3 to V_{DD}+0.3	V	
Output current	I_O	40	mA	2
Operating temperature	Topr	−20 to +70	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note: 1 The maximum applicable voltage on any pin with respect to GND.
Note: 2 Sum of current output from (or flowing into) output pin.

■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	+4.5 to +5.5	V	
Crystal oscillation frequency	f_{OSC}	32.768 (TYP.)	kHz	1
Reference clock oscillation frequency	f	0.25 to 2.5	MHz	2

Note 1: Oscillation start time: within 10 seconds.
Note 2: Degree of fluctuation frequency: ±30% (tolerance of current/voltage fluctuation to be within±10%)

■ DC Characteristics

(V_{DD}=4.5 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		0.7V_{DD}		V_{DD}	V	1
	V_{IL1}		0		0.3V_{DD}	V	
	V_{IH2}		V_{DD}−0.5		V_{DD}	V	2
	V_{IL2}		0		0.5	V	
Input current	I_{IH}	V_{IN}=0V	20		200	µA	1
Output current	I_{OH1}	V_{OH}=V_{DD}−0.5V	50			µA	3
	I_{OL1}	V_{OL}=0.5V	250			µA	
	I_{OH2}	V_{OH}=V_{DD}−0.5V	100			µA	4
	I_{OL2}	V_{OL}=0.5V	500			µA	
	I_{OH3}	V_{OH}=V_{DD}−0.5V	400			µA	5
	I_{OL3}	V_{OL}=0.5V	1.6			mA	
Current consumption	I_{OP}	f=1MHz		15		mA	6
	I_{ST}	Standby current		120		µA	7

Note 1: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$.
$P8_0$-$P8_3$, $P9_0$-$P9_3$,. (In input mode)
$P7_0$-$P7_3$, $\overline{RESET}$, PG_0-PG_3, PH_0-PH_3, PI_0-PI_3, PJ_0-PJ_3, PK_0-PK_3
Note 2: Applied to pins CK_1, OSC_{IN}, $TEST_0$, $\overline{INTA}$, $\overline{INTB}$
Note 3: Applied to pin CK_2
Note 4: Applied to pin ϕ
Note 5: Applied to pins $P0_0$-$P0_3$, $P1_0$-$P1_3$, $P4_0$-$P4_3$, $P5_0$-$P5_3$, $P6_0$-$P6_3$.
$P8_0$-$P8_3$, $P9_0$-$P9_3$,. (In output mode)
$P2_0$-$P2_3$, $P3_0$-$P3_3$, PA_0-PA_3, PB_0-PB_3.
PG_0-PG_3, PH_0-PH_3, PI_0-PI_3, PJ_0-PJ_3,PK_0-PK_3
Note 6: No-load condition
Note 7: When crystal oscillation circuit is inactivated under no load conditions. OSC_{IN} pin should be connected to GND.

■ AC Characteristics

(1) Clock characteristics

(V_{DD}=4.5 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Reference clock oscillation frequency (CR oscillation)	f_{CR}	V_{DD}=5V±10%, R=12kΩ±5%	1.7	2.0	2.3	MHz	
		R=39KΩ±5%	0.5	0.75	1.0		
Reference clock input frequency (CK_1)	f_K		0.25		2.3	MHz	
CK_1 input rise time	t_{KR}				500	ns	
CK_1 input fall time	t_{KF}				500	ns	
CK_1 input high range	t_{KH}		0.3			μs	
CK_1 input low range	t_{KL}		0.3			μs	
OSC crystal oscillation frequency	f_{OSC}			32.768		kHz	
OSC_{OUT} input cycle time	t_{fY}		2			t_{CYC}	1
OSC_{OUT} input rise time	t_{fR}				500	ns	
OSC_{OUT} input fall time	t_{fF}				500	ns	
OSC_{OUT} input high range	t_{fH}		1			t_{CYC}	1
OSC_{OUT} input low range	t_{fL}		1			t_{CYC}	1

Note 1: Cycle time of one fourth the reference clock oscillation frequency.

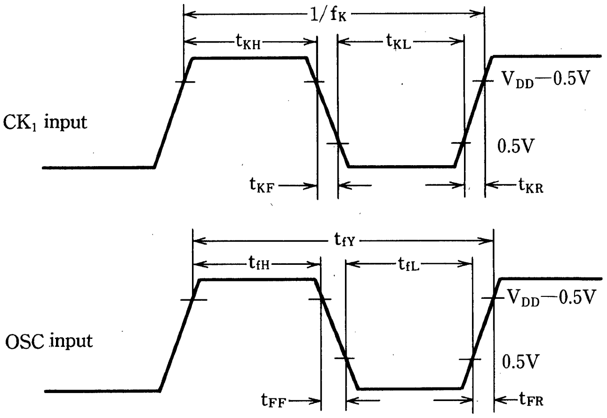


Fig. 1 Clock timing

(2) Interrupt input

(V_{DD}=4.5 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
$\overline{\text{INTA}}$ high range	t_{AH}		2			t_{CYC}	1
$\overline{\text{INTA}}$ low range	t_{AL}		2			t_{CYC}	
$\overline{\text{INTA}}$ high range	t_{RH}		2			t_{CYC}	
$\overline{\text{INTA}}$ low range	t_{RL}		2			t_{CYC}	

Note 1: t_{CYC}: Cycle time of one fourth the reference clock oscillation frequency.

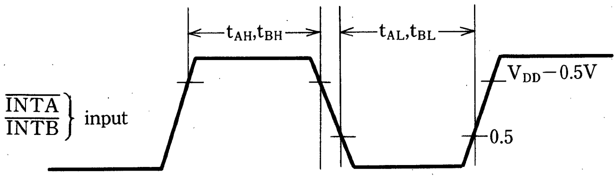


Fig. 2 Interrupt input timing

(3) External serial input clock

(V_{DD}=4.5 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
SCK cycle time	t_{SY}		1			t_{CYC}	1
SCK high range	t_{SH}		1/2			t_{CYC}	
SCK low range	t_{SL}		1/2			t_{CYC}	
SCK rise time	I_{SR}				500	ns	
SCK fall time	t_{SF}				500	ns	
Reset pulse width (low)	t_{RST}		300			ns	

Note 1: t_{CYC}: Cycle time of one fourth the reference clock oscillation frequency.

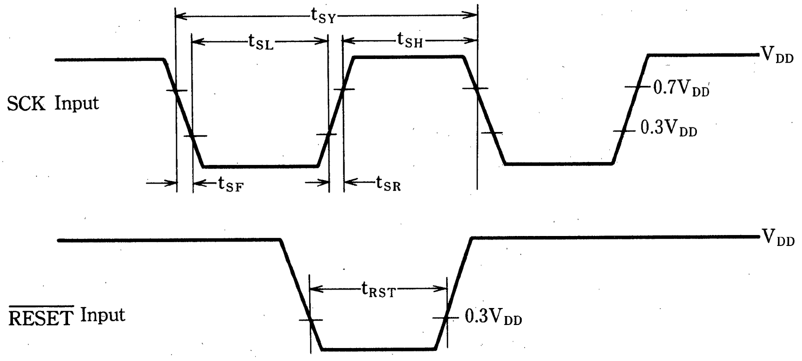
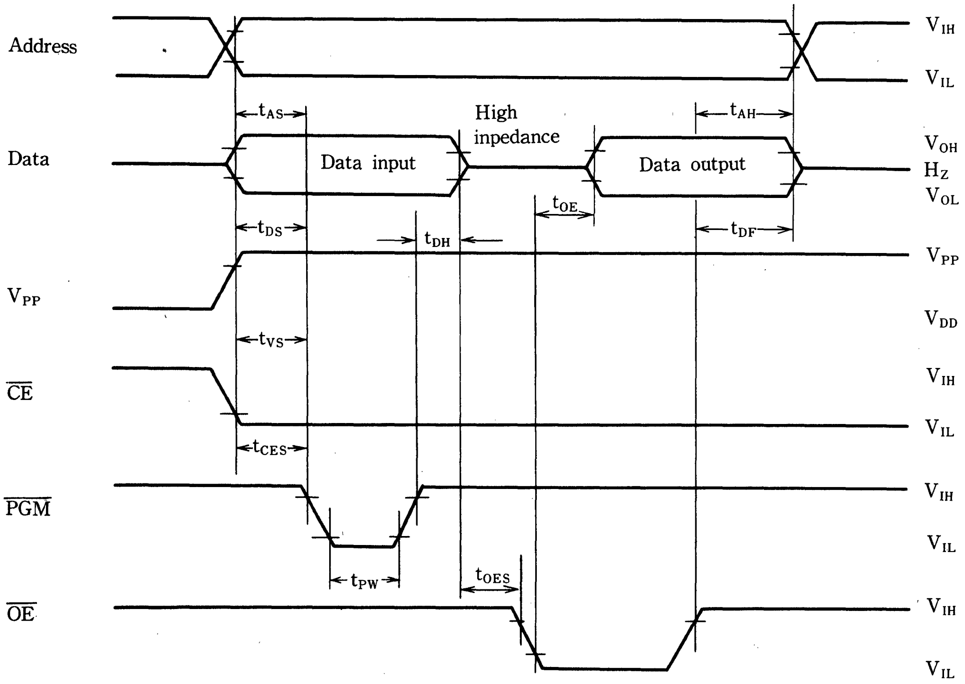


Fig. 3 External serial input clock timing

(4) PROM programming

(V_{DD}=4.75 to 6.25V, V_{PP}=12.0 to 13.0V, Ta=25℃±5℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Address setup time	t_{AS}	$\overline{PGM}$−Address	2			μs
$\overline{CE}$ setup time	t_{CES}	$\overline{PGM}$−$\overline{CE}$	2			μs
$\overline{OE}$ setup time	t_{OES}	Data−$\overline{OE}$	2			μs
Data setup time	t_{DS}	$\overline{PGM}$−Data	2			μs
Address hold time	t_{AH}	$\overline{OE}$−Address	0			μs
Data hold time	t_{DH}	$\overline{PGM}$−Data	2			μs
Output disable time	t_{DF}		0		150	ns
Output enable time	t_{OE}				150	ns
V_{PP} setup time	t_{VS}	$\overline{PGM}$−V_{PP}	2			μs
$\overline{PGM}$ pulse width	t_{PW}		1		55	ms



Hardware Configuration

The hardware configuration of the LU5E4P0P is the same as that of the SM5E4, except for an on-chip program memory of an OTPROM for the LU5E4P0P, and a ROM for the SM5E4, Refer to the SM5E4 for the hardware configuration.

PROM Programming

When data is written into an on-chip OTPROM, apply the conversion socket adapter (LR0E82) to the commercial EPROM writers.

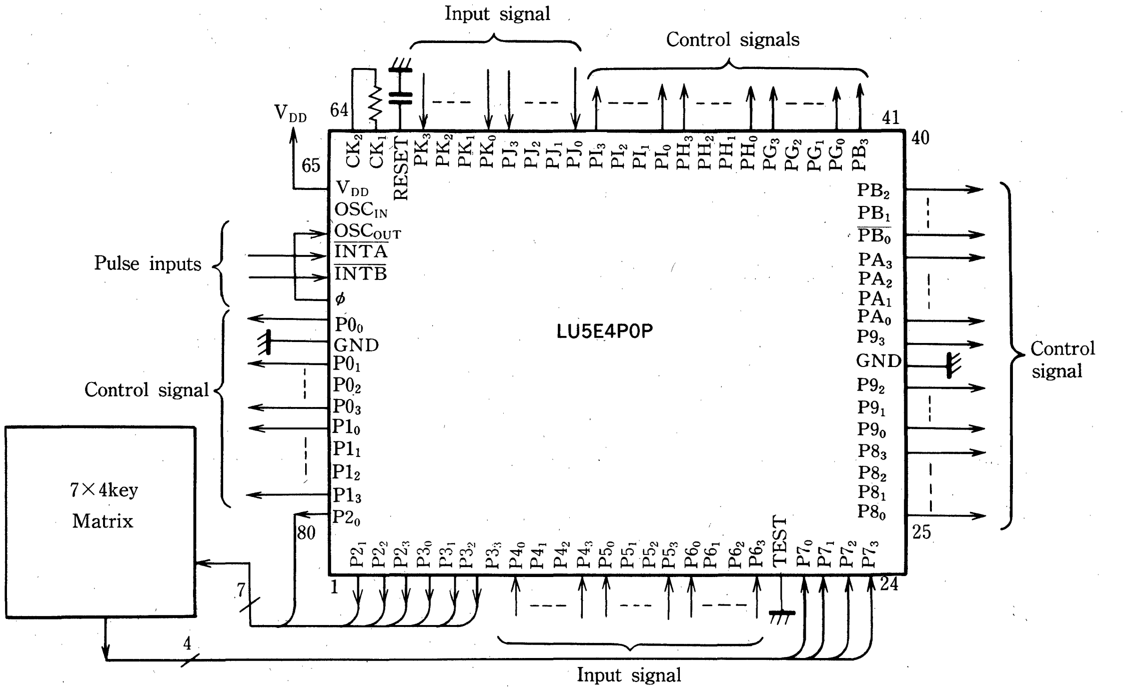
Use the EPROM writer which allows the LH5764 mode set, and eliminates or clears the electric signature mode.

See Pin Connections for the signals in parentheses used PROM programming.

Instruction set

See the SM5E4 for the instruction set which is the same as that of the LU5E4P0P.

System Configuration Example


Input signal
Control signals
V$_{DD}$
64
65
41
40
25
24
80
1
7
4
CK$_2$ CK$_1$ RESET PK$_3$ PK$_2$ PK$_1$ PK$_0$ PJ$_3$ PJ$_2$ PJ$_1$ PJ$_0$ PI$_3$ PI$_2$ PI$_1$ PI$_0$ PH$_3$ PH$_2$ PH$_1$ PH$_0$ PG$_3$ PG$_2$ PG$_1$ PG$_0$ PB$_3$
V$_{DD}$
OSC$_{IN}$
OSC$_{OUT}$
INTA
INTB
ϕ
P0$_0$
GND
P0$_1$
P0$_2$
P0$_3$
P1$_0$
P1$_1$
P1$_2$
P1$_3$
P2$_0$
PB$_2$
PB$_1$
PB$_0$
PA$_3$
PA$_2$
PA$_1$
PA$_0$
P9$_3$
GND
P9$_2$
P9$_1$
P9$_0$
P8$_3$
P8$_2$
P8$_1$
P8$_0$
P2$_1$ P2$_2$ P2$_3$ P3$_0$ P3$_1$ P3$_2$ P3$_3$ P4$_0$ P4$_1$ P4$_2$ P4$_3$ P5$_0$ P5$_1$ P5$_2$ P5$_3$ P6$_0$ P6$_1$ P6$_2$ P6$_3$ TEST P7$_0$ P7$_1$ P7$_2$ P7$_3$
LU5E4P0P
Pulse inputs
Control signal
7×4key Matrix
Control signal
Input signal

SM5J5/SM5J6

4-Bit Microcomputer (VFD Driver)

4-Bit Microcomputer (Controller with A/D Converter)

■ Description

The SM5J5/SM5J6 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, interrupts, an A/D converter, a comparator, a counter/timer circuit, and a sound output function in a single chip.

An A/D conversion can be executed by one instruction with simple software, and provides a high speed processing. This feature enables to accept analog signals from sensors.

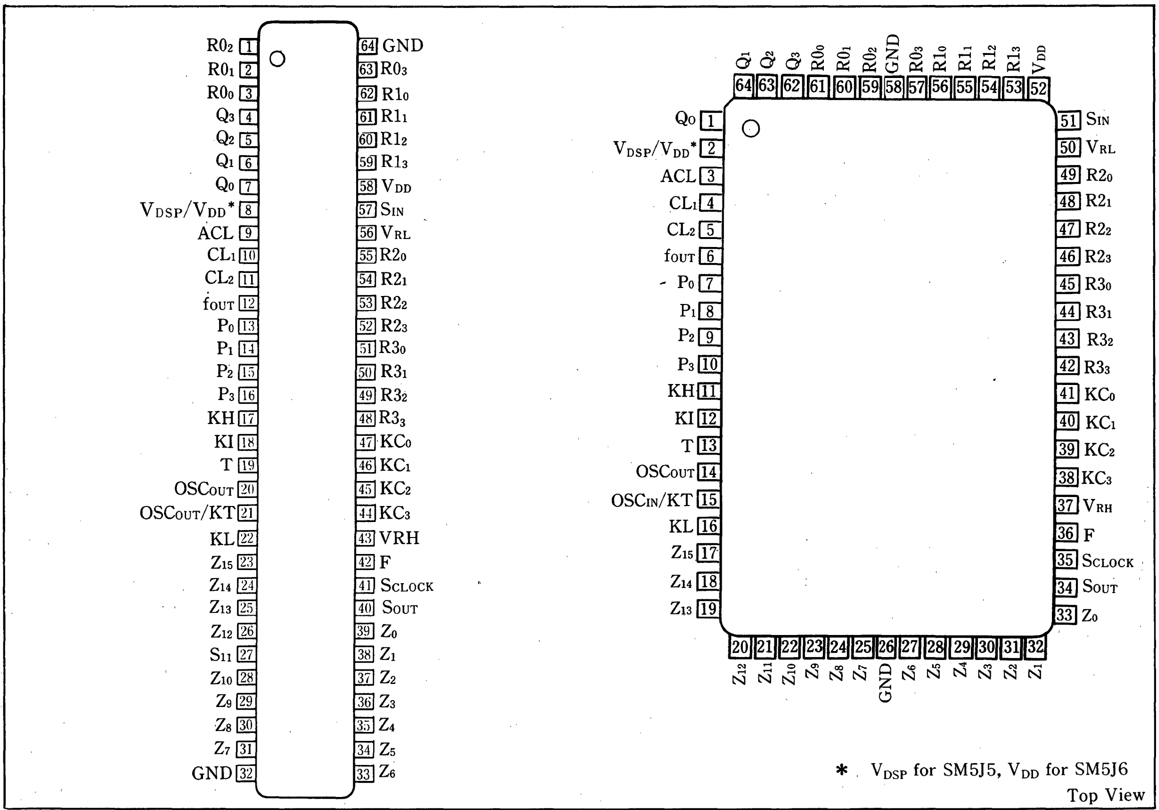
Provided with unique features of 52 I/O ports, a couple of programmable counter/timers, and interrupt functions, this microcomputer is suitable for controlling functions required for a timer set.

The SM5J5 directly drives a fluorescent display tube, and the SM5J6 provides two modes of standby function for low power operations.

■ Features

1. CMOS process
2. ROM capacity: 8,192×9 bits
3. RAM capacity: 256×4 bits
4. Instruction set: 94
5. Subroutine nesting: 6 levels
6. Instruction cycle
   SM5J5: 2.5 μs (MIN.)
   SM5J6: 2 μs (MIN.)
7. Interrupts
   External interrupts: 2
   Internal interrupts: 3
8. Input/output ports
   I/O ports: 31
   Input ports: 9
   Output ports: 12
9. 8-bit serial I/O

■ Pin Connections

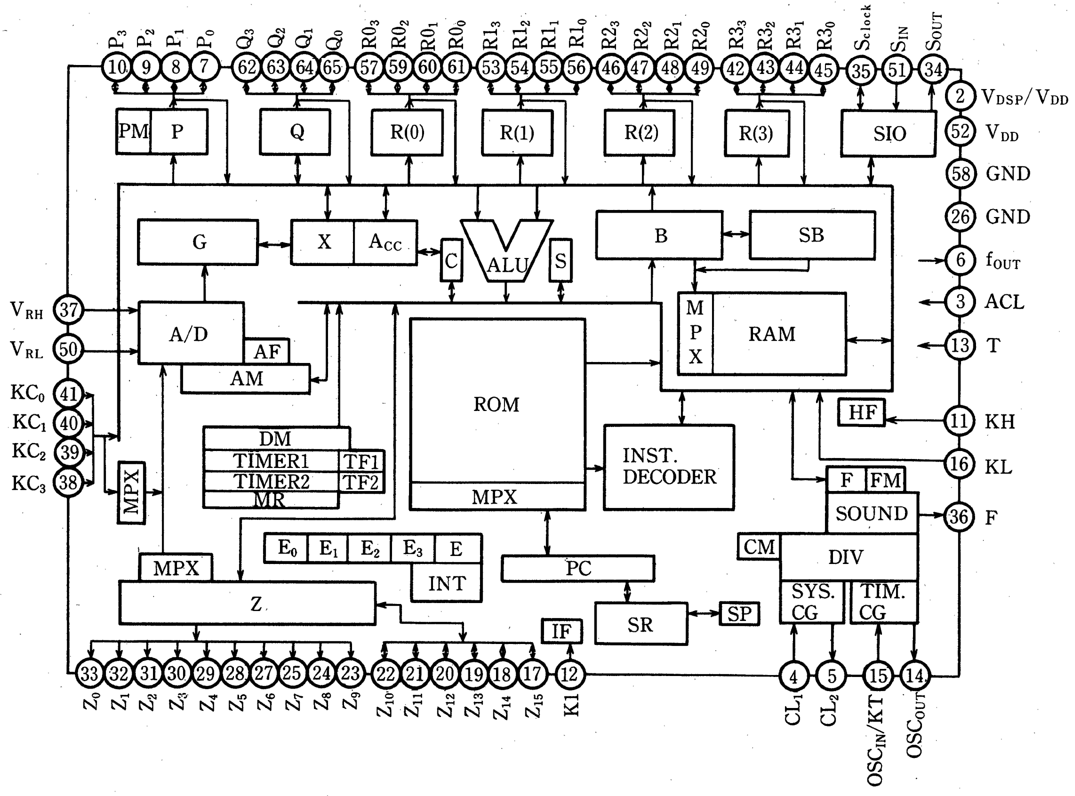


* V_{DSP} for SM5J5, V_{DD} for SM5J6

Top View

10. A/D converter:
    8 bits (10-channels MAX.)
11. Counter/timer: 2 sets
12. Standby function
    SM5J6: 2-stage system clocks
13. High voltage output: −40V
    SM5J5: 16-segment, 10-digit
14. Supply voltage
    SM5J5: 4.5 to 5.5V
    SM5J6: 2.7 to 5.5V
15. 64-pin SDIP (SDIP64-P-750)
    64-pin QFP (QFP64-P-1420)

Block Diagram



Symbol description

ALU	: Arithmetic logic unit	Acc	: Accumulator
X	: X register	SR	: Stack register
B	: RAM address register	R0, R1, R2, R3	: Latch
C	: Carry F/F	DIV	: Divider
PC	: Program counter		
SP	: Stack pointer		
CG	: System clock generator		
MPX.	: Multiplexer		
A/D	: A/D converter and comparator unit		

Note: Pin numbers apply to a 64-pin QFP.

Pin Description

Symbol	I/O	Function	Note
P_0-P_3, Q_0-Q_3	I/O	Input/output ports (Nibble unit)	
$R0_0$-$R0_3$	I/O	Input/output ports (Nibble unit)	1
$R1_0$-$R1_3$	I/O	Input/output ports (Nibble unit)	1
$R2_0$-$R2_3$	I/O	Input/output ports (Nibble unit)	1
$R3_0$-$R3_3$	I/O	Input/output ports (Nibble unit)	1
Z_0-Z_9	O	Output ports (Bit unit)	1
Z_{10}-Z_{15}	I/O	Input/output ports (Bit unit)	
KC_0-KC_3	I	Input ports or analog input ports	
KH, KL	I	Input ports	
KI	I	Interrupt input port or input port	
OSC_{IN}/KT	I	Timer clock input port or input port	
OSC_{OUT}		Timer clock oscillator	
F	O	Sound output port or output port	
f_{OUT}	O	System sync. signal output port	
CL_1		Clock signal input port	
CL_2		Clock signal oscillator	
ACL	I	Auto clear input port	
V_{RH}, V_{RL}		A/D converter	
V_{DD}		Power supply	
GND		Ground	
T	I	Test input port	
S_{IN}	I	Serial I/O data input port	
S_{OUT}	O	Serial I/O data output port	
S_{CLOCK}	I/O	Serial I/O clock port	2
V_{DSP}/V_{DD}		Power supply	3

Note 1: SM5J5, −40V high voltage
Note 2: Input port in the ACL
Note 3: SM5J5, −30V (TYP.)
SM5J6, +5V (TYP.)

Absolute Maximum Ratings

Parameter	Symbol	Applicable model	Rating	Unit	Note
Supply voltage	V_{DD}	SM5J5/5J6	-0.3 to $+7.5$	V	1
	V_{OSP}	SM5J5	$V_{DD}-40$ to $V_{DD}+0.3$		1
Input voltage	V_I	SM5J5/5J6	-0.3 to $V_{DD}+0.3$		1, 2
	V_{ID}	SM5J5	$V_{DD}-40$ to $V_{DD}+0.3$		1, 3
Output voltage	V_O	SM5J5/5J6	-0.3 to $V_{DD}+0.3$		1, 4
	V_{OD}	SM5J5	$V_{DD}-40$ to $V_{DD}+0.3$		1, 5
Output HIGH voltage	I_{OH}	SM5J5	-40	mA	6, 7
		SM5J6	-20		
		SM5J5	-12		8
		SM5J6	-10		
		SM5J5/5J6	-4		9
		SM5J5/5J6	-2.5		10
		SM5J5/5J6	-2		11
		SM5J5	-80		12
		SM5J6	-60		
		SM5J5/5J6	-20		13
Output LOW voltage	I_{OL}	SM5J5/5J6	4.0	mA	14
		SM5J5/5J6	400	μA	15
		SM5J5/5J6	2.0	mA	11
		SM5J5/5J6	25		16

Note 1: Referenced to GND.
Note 2: Applied to all input ports except for the case where the R (0)-R (3), Z_0-Z_9 of the SM5J5 are used as high voltage input ports.
Note 3: Applied to pins R (0)-R (3), Z_0-Z_9 which are used as high voltage input ports.
Note 4: Applied to all output ports except for the case where the R (0)-R (3), Z_0-Z_9 of the SM5J5 are used as high voltage input ports.
Note 5: Applied to pins R (0)-R (3), Z_0-Z_9 which are used as high voltage outputs.
Note 6: Applied to the case where only one port of Z_0-Z_9 is output.
Note 7: Applied to the case where the duty ratio during High level output is less than 1/7 (cycle: 10ms).
Note 8: Applied to the case where only one port of R (0)-R (3) is output.
Note 9: Applied to the case where only one port of P, Q, Z_{10}-Z_{15} is output.
Note 10: Applied to the case where only one port of F, f_{OUT}, S_{CLOCK} is output.
Note 11: Applied to CL_2 pin.
Note 12: Applied to the sum of R (0)-R (3), Z_0-Z_9.
Note 13: Applied to the output ports except for ports mentioned in note 10.
Note 14: Applied to the case where only one port of Q, Z_{10}-Z_{15} (CMOS output), P is output.
Note 15: Applied to the case where only one port of Q, Z_{10}-Z_{15} (with a pull-down resistor) is output.
Note 16: Applied to the sum of all ports.

*R (0)-R (3): $R0_0$-$R0_3$, $R1_0$-$R1_3$, $R2_0$-$R2_3$, $R3_0$-$R3_3$.

Recommended Operating Conditions

(1) SM5J5

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		+4.5 to +5.5V	V
System clock frequency	f_S		312.5	kHz

(2) SM5J6

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		+2.7 to +5.5	V
System clock frequency	f_S	V_{DD}=4.5 to 5.5V	50 to 500	kHz
		V_{DD}=2.7 to 3.3V	50 to 250	

DC Characteristics

SM5J5 (V_{DD}=4.5V to 5.5V, Ta=−10 to +80℃)
SM5J6 (V_{DD}=2.7V to 5.5V, Ta=−10 to +80℃)

Parameter	Symbol	Condition	Applicable model	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		SM5J5/5J6	$0.8V_{DD}$		V_{DD}	V	1
	V_{IL1}		SM5J5/5J6	0		$0.2V_{DD}$		
	V_{IH2}		SM5J5/5J6	$V_{DD}-0.5$		V_{DD}		2
	V_{IL2}		SM5J5/5J6	0		0.5		
	V_{IH3}		SM5J5/5J6	$V_{DD}-0.4$		V_{DD}		3
	V_{IL3}		SM5J5/5J6	0		0.4		
Input current	I_{IH}	$V_{IN}=V_{DD}$	SM5J5/5J6		50		μA	4
	I_{IL}	$V_{IL}=0V$	SM5J5/5J6		−45			5
Input leakage current	I_{LK}		SM5J5/5J6			10		6
Output current	I_{OH1}	$V_{OH}=V_{DD}-0.5V$	SM5J5			−0.8	mA	7
		$V_{OH}=V_{DD}-0.5V$	SM5J6			−0.4		
	I_{OL1}	$V_{OL}=0.5V$	SM5J5/5J6	10			μA	8
	I_{OH2}	$V_{OH}=V_{DD}-0.5V$	SM5J5/5J6			−100		9
	I_{OL2}	$V_{OL}=0.5V$	SM5J5	−0.8			mA	10
		$V_{OL}=0.5V$	SM5J6	−0.4				
	I_{OH3}	$V_{OH}=V_{DD}-0.5V$	SM5J5			−0.5		11
		$V_{OH}=V_{DD}-0.5V$	SM5J6			−0.4		
	I_{OL3}	$V_{OL}=0.5V$	SM5J5/5J6		30		μA	12
	I_{OH4}	$V_{OH}=V_{DD}-2.0V$	SM5J5			−15	mA	13
		$V_{OH}=V_{DD}-2.0V$	SM5J6			−7		
	I_{OL4}	$V_{OL}=0.5V$	SM5J5/5J6			−100	μA	9
	I_{OH5}	$V_{OH}=V_{DD}-2.0V$	SM5J5			−7	mA	14
			SM5J6			−4		
Output voltage	V_{OL}	$V_{DD}=5.5V$, $V_{DSP}=-30V$	SM5J5			−28	V	15
Internal resistance	R_1	$V_{DD}=5.0V$	SM5J5	10	30	60	kΩ	10
		$V_{DD}=5.0V$	SM5J6	5	25	50		
	R_2	$V_O=V_{DD}$	SM5J5	40	200	500		16
		$V_O=V_{DD}$	SM5J6	30	100	200		
Current consumption	I_{ST}	Standby mode	SM5J6			10	μA	17
	I_{DD}	f_{CL}=500kHz, V_{DD}=5.0V operation	SM5J6		5	10	mA	18
	I_{DD1}	f_{CL}=312.5kHz, V_{DSP}: open	SM5J5		5	8		17
	I_{DD2}	f_{CL}=312.5kHz, V_{DSP}: −30V	SM5J5		15	30		19

Note 1: Applied to pins KH, KL, KI, P_0-P_3, Q_0-Q_3, KC_0-KC_3.
Note 2: Applied to pins Z_{10}-Z_{15}, CL_1, OSC_{IN}/KT, ACL, R (0)-R (3).
Note 3: Applied to S_{IN}, S_{CLOCK}.
Note 4: Applied to ACL pin.
Note 5: Applied to S_{CLOCK} pin.
Note 6: Applied to pins Q_0-Q_3, Z_{10}-Z_{15} without any pull-down resistors, or KH, KL, KI, KC_0-KC_3, P_0-P_4, S_{IN}, S_{CLOCK}, OSC_{IN}/KT, ACL.
Note 7: Applied to pins P_0-P_3, Q_0-Q_3, Z_{10}-Z_{15}.
Note 8: Applied to the case where Q_0-Q_3, Z_{10}-Z_{15} are used as open drain outputs.
Note 9: Applied to CL_2 pin.
Note 10: Applied to pins Q_0-Q_3, Z_{10}-Z_{15} used as CMOS outputs, or P_0-P_3, F, f_{OUT}, S_{OUT}, S_{CLOCK}.
Note 11: Applied to pins F, f_{OUT}, S_{OUT}, S_{CLOCK}.
Note 12: Applied to pins Q_0-Q_3, Z_{10}-Z_{15} with pull-down resistors.
Note 13: Applied to pins Z_0-Z_9.
Note 14: Applied to pins R (0)-R (3).
Note 15: Applied to pins R (0)-R (3), Z_0-Z_9 with pull-down resistors. (pull-down to V_{DSP}.)
Note 16: Applied to pins R (0)-R (3), Z_0-Z_9 with pull-down resistors. (SM5J5: pull-down to V_{DSP} or GND with a mask option. SM5J6: pull-down to GND only.)
Note 17: No load condition. (the oscillation frequency should be 8 times of f_S, and the OSC_{IN}/KT port should be connected to GND.
Note 18: No load condition. (the oscillation frequency should be 8 times of f_S.)
Note 19: No load condition. (the oscillation frequency should be 8 times of f_S), ACL state, the I_{DD} is a current flowing between V_{DD} and V_{DSP}.

*R (0)-R (3): $R0_0$-$R0_3$, $R1_0$-$R1_3$, $R2_0$-$R2_3$, $R3_0$-$R3_3$.

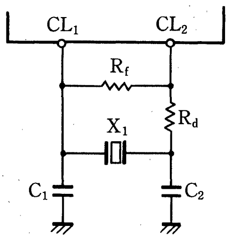
■ A/D Conversion Characteristics

SM5J5 (f_S=312.5kHz, V_{DD}=5.0V, V_{RH}=5.0V)
SM5J6 (f_S=500kHz, V_{DD}=5.0V, V_{RH}=5.0V)

Parameter	V_{RL} pin	MIN.	TYP.	MAX.	Unit
Non-linearity error	GND			±3	LSB
Integration non-linerarity error	GND			±3	LSB
Zero error	GND			±3	LSB
Full-scale error	GND			±3	LSB
V_{RH} pin supply current	—		100	300	μA
Total error	GND			±3	LSB

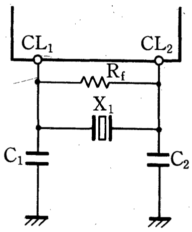
■ Oscillator Circuits

(1) Recommended oscillator circuit for the SM5J5



R_f=1MΩ
R_d=330Ω
C_1=22pF
C_2=22pF
X_1=2.5MHz Ceramic oscillator
Above constants apply to the case where the oscillator is used with the CSA2.5MG (MURATA)

(2) Recommended oscillator circuit for the SM5J6



R_f=1MΩ
C_1=15pF
C_2=15pF
X_1=4.0MHz Ceramic oscillator
Above constants apply to the case where the oscillator is used with the KBR-4.0MS (KYOSERA)

Pin Functions

(1) V_{DD}, V_{DSP}, GND (Power supply)

The V_{DD} pin is the positive power supply (3V to 5V) with respect to GND.

The SM5J5 provides the V_{DSP} which is the negative power supply (−35V) with respect to GND.

The GND pin is the reference power supply for the LSI.

(2) V_{RH}, V_{RL} (A/D conversion)

The V_{RL} pin is a GND pin for the A/D converter.

The V_{RH} pin provides the reference voltage V_{RH} for the A/D converter.

The current consumption and operating accuracy of the A/D converter must be changed according to the case where the V_{RL} pin is used to be left open or provide GND level.

(3) ACL (Reset input)

The ACL pin is used to reset the LSI.

The LSI should be reset with a transition of two instruction cycles after the rising edge of ACL.

Applying a Low level signal to the ACL pin starts execution of the program at field 0, page 0, step 0 after a transition of t_{ACL}.

It is recommended to apply a capacitor between ACL pin and V_{DD} pin in order to prevent from external noise which affects the ACL circuit.

(4) KC_0-KC_3 (Analog inputs)

Executing the KCTA instruction transfers the KC input data to the accumulator A_{CC} through input buffers.

The KC input pin also provides analog input signals given to the A/D conversion block.

(5) KH, KI, OSC_{IN}/KT, KL (Inputs)

The KH and KI input pins are connected to the noise debounce circuit, and the KL and OSC_{IN}/KT input pins to input buffers.

The KL, OSC_{IN}/KT, KH and KI should be loaded into the A_0, A_1, A_2 and A_3 bits of the accumulator A_{CC} upon execution of KLTA instruction.

The noise debounce circuit does not accept the pulse input shorter than two instruction cycle width.

(6) Z_0-Z_9 (Outputs)

The Z_0-Z_9 can be controlled with the output latch F/F to be set or reset by instruction.

- **SM5J5** The Z_0-Z_9 are normally be used as high voltage outputs (−40V). They can be used as the outputs with a pull-down resistor by a mask option. They can also be used with an open-drain transistor structure.
- **SM5J6** The Z_0-Z_9 are normally be used as the outputs with a pull-down resistor. They can be used with an open-drain transistor structure.

(7) Z_{10}-Z_{15} (Input/output)

The Z_{10}-Z_{15} can be controlled with the output latch F/F to be set or reset by instructions.

When used for the inputs, the input mode of Zi specified by lower 4 bits of B register B_L can be tested by instructions.

The Zi pins transfer analog signals into the comparator of A/D converter.

The Zi pins are normally used as I/O pins with pull-down resistors. The Zi pins can be used as CMOS outputs or the open-drain transistor with a protection diode. After an ACL operation, the Zi pins are placed in input mode. When used for the inputs, the Zi pins can be used as the outputs to be pulled down with the output latch F/F to be reset.

(8) P_0-P_3 (Input/output)

The P_0-P_3 are three-state I/O pins.

Executing the ATP instruction transfers the accumulator A_{CC} to the output latch F/F which is loaded into the P_0-P_3.

The P_0-P_3 can be loaded into the A_{CC} upon execution of the PTA instructions. Then the P_0-P_3 remain high impedance.

(9) Q_0-Q_3 (Input/output)

Executing the ATQ instruction transfers the accumulator A_{CC} to the output latch F/F which is loaded into the Q_0-Q_3.

While, the Q_0-Q_3 can be loaded into the A_{CC} upon execution of the QTA instruction. Then, the Q_0-Q_3 should be used with the outputs to be pulled down.

The Qi pins are normally used as I/O pins with pull-down resistors. The Qi pins can be used as CMOS outputs or the open-drain transistor with a protection diode. After an ACL operation, the Qi pins are placed in input mode. When used for the inputs, the Qi pins can be used as the outputs to be pulled down with the output latch F/F to be reset.

(10) $R0_0$-$R0_3$, $R1_0$-$R1_3$, $R2_0$-$R2_3$, $R3_0$-$R3_3$ (input/output)

Upon execution of the ATR instruction, the

R0i-R3i outputs the accumulator A_{CC} specified by the lower 4 bits (BL) of the B register.

8-bit data transfer can be performed in parallel among the R1i, R0i and A_{CC} or X register by the RTAX or AXTR instruction.

- **SM5J5** The Ri pins are normally used as high voltage I/O pins (−40V). They can be used as the I/O pins with pull-down resistors with a mask option.

When used for the inputs, the Ri pins can be used as the I/O pins to be pulled down with the output latch F/F to be reset.

The Ri pins can also be used as open-drain transistor structure.

- **SM5J6** The Ri pins are normally used as the I/O pins with pull-down resistors.

When used for the inputs, the Ri pins can be used as the input pins to be pulled down with the output latch F/F to be reset. They can also be used as an open-drain transistor structure.

(11) F (Sound output)

The F output pin is used for a sound output pin as well as a general-purpose output.

(12) f_{OUT}

The f_{OUT} pin outputs the the signal in synchronizing with the system clock f_S.

The system clock immediately after power on is a frequency of one eighth the reference clock frequency.

■ Hardware Configuration

(1) Program counter and stack

The program counter (PC) is used to address a ROM location.

The PC consists of 13 bits allocated 4 bits (ET, P_U) to the field specification of ROM, 2 bits (P_M) to the page specification, and 7 bits (P_L) to the step specification. The P_M and P_L are binary counter for the page specifications.

The stack register (SR) consists of 6 stages available for up to 6 levels of subroutine nesting.

(2) Program memory (ROM)

The ROM is used to store programs.

The SM5J5/SM5J6 has a 8,192×9-bit ROM which consists of 16 fields×4 pages×128 steps.

When power on with the ACL to be reset, the program starts execution at field 0, page 0, step 0.

Fig. 1 shows the example of a jump to the ROM address by a ROM address instruction.

The TR instruction is used to jump within a page, and the TL instruction is used to jump to any address. A subroutine jump is executed by a TLS or TRS instruction. However, when the ET value may change due to a jump or subroutine jump, execute a TL or TLS instruction following a COMET instruction.

(3) Data memory (RAM) and B register

The RAM is used to store data.

The SM5J5/SM5J6 has a 1,024-bit RAM organized as 16×16×4 which consists of 16 files as shown in Fig. 2. A file consists of 16 words×4 bits.

The RAM address is specified by a B register which consists of a 4-bit B_M for the file specification and a 4-bit B_L for the word specification.

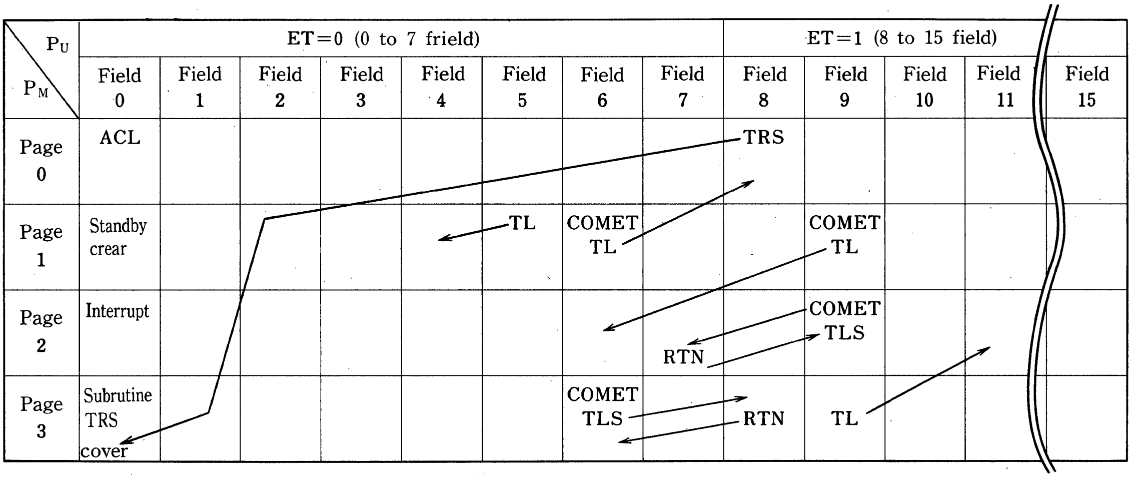


Fig. 1 ROM configuration

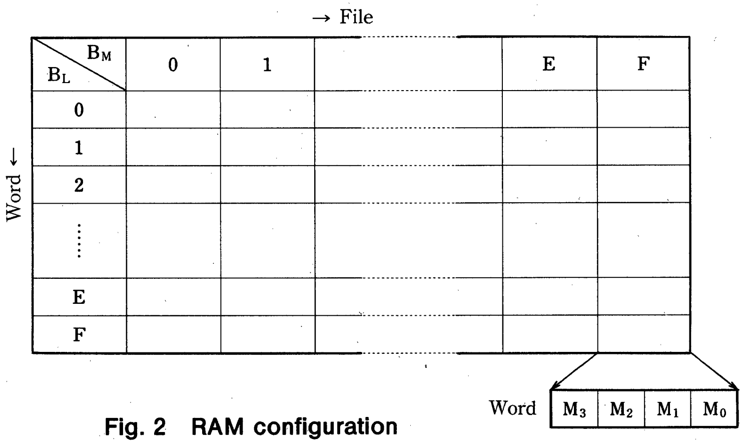


Fig. 2 RAM configuration

(4) Accumulator A_{CC}, X and G registers

The accumulator A_{CC} is a 4-bit general-purpose register which transfers numerics and data. The A_{CC} can be decremented and shifted to the left in combination with the carry flag (C). Furthermore, the A_{CC} together with the arithmetic logic unit (ALU), a carry flag (C) and RAM executes arithmetic operations. It also transfers data to I/O ports.

The X register is a 4-bit register which can be used for a temporary register. It is incremented by instructions. It performs, in conjunction with the A_{CC}, logical sum and logical product.

An 8-bit parallel data of the A_{CC} and X register can be transferred to R [0] and R [1], a G register or a counter/timer.

On the other hand, each data on R0i and R1i, a G register or a counter timer can also be transferred to the A_{CC} and X register with an 8-bit parallel data.

The G register is an 8-bit register which is used for A/D conversion or comparison of analog signals.

(5) Arithmetic and logic unit (ALU), carry flag (C)

The arithmetic and logic unit (ALU) performs, in conjunction with a RAM, a carry flag C and an accumulator A_{CC}, binary addition on a 4-bit basis by instructions.

The carry flag C latches the data incremented by the ADC or ADCS instruction.

(6) SB register

The SB register is an 8-bit register used for a save register.

(7) P, Q, R [3]-R [0], Z (Output latch registers)

Registers P, Q, R0, R2, R3, Z connect with the output latch F/F.

The accumulator A_{CC} can be transferred to registers P, Q, R [3]-R [0], and an 8-bit data of the A_{CC} and X register can be transferred, at the same time, to the output latch registers R [0] and R [1].

(8) System clock generator circuit

The system clock generator circuit generates a system clock of a reference frequency input from the CL_1 pin divided by 4 or 8.

The system clock speed can be controlled with a program. If it is not required for high speed operation, the system clock can be switched to the low speed in order to save the power consumption. This function is also applicable to the case where the power supply is replaced by a battery backup power.

The system clock when reset is the reference frequency divided by 8.

The system clock f_S is used to determine the instruction execution cycle, and the system clock cycle should be identical to the instruction execution cycle. However, the instruction execution cycle of a two-word instruction should be two times as long as a one-word instruction.

(9) Counter/timer

A timer 1 and a timer 2 are 8-bit counter/timers. The data incremented by a count up is latched into the flags TF1 and TF2 to be used for an interrupt request. Executing the TTF1 and TTF2 flags checks the flags TF1 and TF2.

- Timer 1: An 8-bit data of the A_{CC} and X register can be transferred to the timer 1. To the contrary, the timer 1 can be read out from the A_{CC} and X register.
- Timer 2: The timer 2 contains a modulo register (MR register). The contents of the modulo register (TM) are loaded into the timer 2 each time the register is incremented by one.
  An 8-bit data can be loaded into the MR register by instructions, and executing the next instruction cycle transfers the data to the timer 2 which can be read out from the A_{CC} and X register.

The count up pulses of a counter/timer include $(1/2)^9 f_S$, $(1/2)^6 f_S$, $(1/2)^3 f_S$, $(1/2)^6 f_T$ and f_T, under conditions of a system clock f_S and KT input pulse f_T, which can be selected by a program.

A carry output of one counter can be used for a count up pulse of the other counter.

(10) Interrupt

A KI input, the timer 1 and timer 2 carry and an analog input are available for the interrupt request, and the interrupt request flags include the IF, TF1, TF2, and AF flags.

The interrupt block consists of the mask flags (E_3, E_2, E_1 and E_0), E flag and interrupt processing circuits.
(See Fig. 3)

Table 1 shows the jump address caused by an interrupt request.

(11) A/D converter

The A/D conversion block consists of an 8-bit D/A converter, a comparator, an AM flag and AF flag.

The KC and Z pins input the analog signals.

Executing the COMP instruction allows the A/D conversion and the large/small comparison automatically. (See Fig. 4.)

The result of A/D conversion is stored in the G register with the interval of 32 instruction cycles after the COMP instruction is executed.

The result of the large/small comparison is stored in the AF flag with the interval of 6 instruction cycles.

The G register is an 8-bit register which can be transferred to the A_{CC} and X register with the GTAX instruction.

The KC_0 pin can also be used for external interrupt.

The D/A converter generates the voltage V_{REF} according to the contents of the G register.

Assuming that the "n" is placed in the G register as a result of A/D conversion, the analog input voltage may be regarded as a below expression.

$$\frac{256-n}{256}V_{RH}\ (n=0 \text{ to } 255)$$

* V_{RH} is a reference voltage supply from the V_{RH} pin.

When even more strict accuracy is required in the A/D conversion block, an external GND level may be applied to the V_{RL} pin.

The A/D conversion is executed by the comparison among a G register, a D/A converter and a comparator in order.

The large/small comparison is executed by the comparator output V_{REF} according to the G register value and the analog signal of the KC_0. The result of comparison is stored in the AF flag.

(12) Sound output block

The F pin outputs the frequency obtained by a count-up pulse generator circuit.

The pulse frequency can be selected from $(1/2)^7 f_S$, $(1/2)^9 f_S$, $(1/2)^4 f_r$ and $(1/2)^6 f_T$ by programs.

* The f_T is a timer clock frequency input from the OSC_{IN}/KT pin, and the f_S is a system clock frequency.

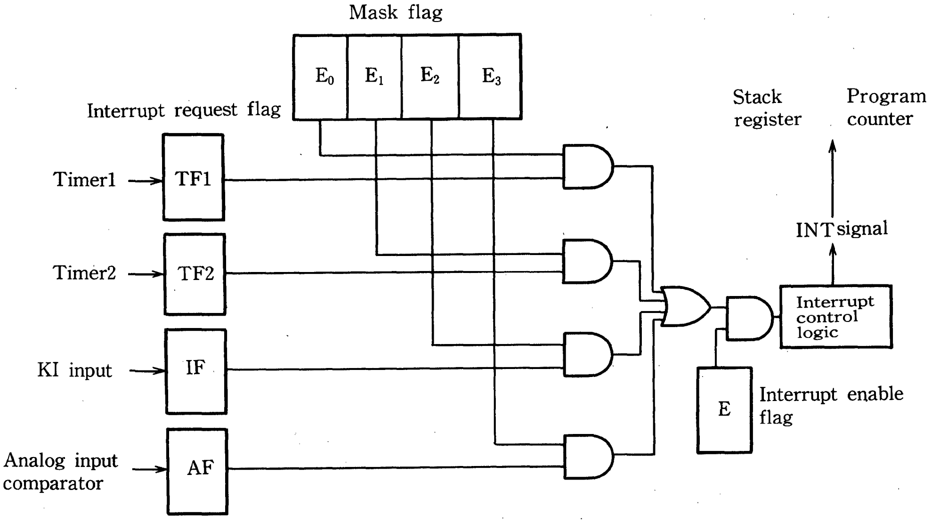


Fig. 3 Interrupt block

Table 1 Interrupt jump address

Interrupt request flag	Jump destination address			Priority
	Field	Page	Step (PL)	
Timer 1 carry (TF1)	0	2	0	1
Timer 2 carry (TF2)	0	2	2	2
KI input (IF)	0	2	4	3
Analog input comparator (AF)	0	2	6	4

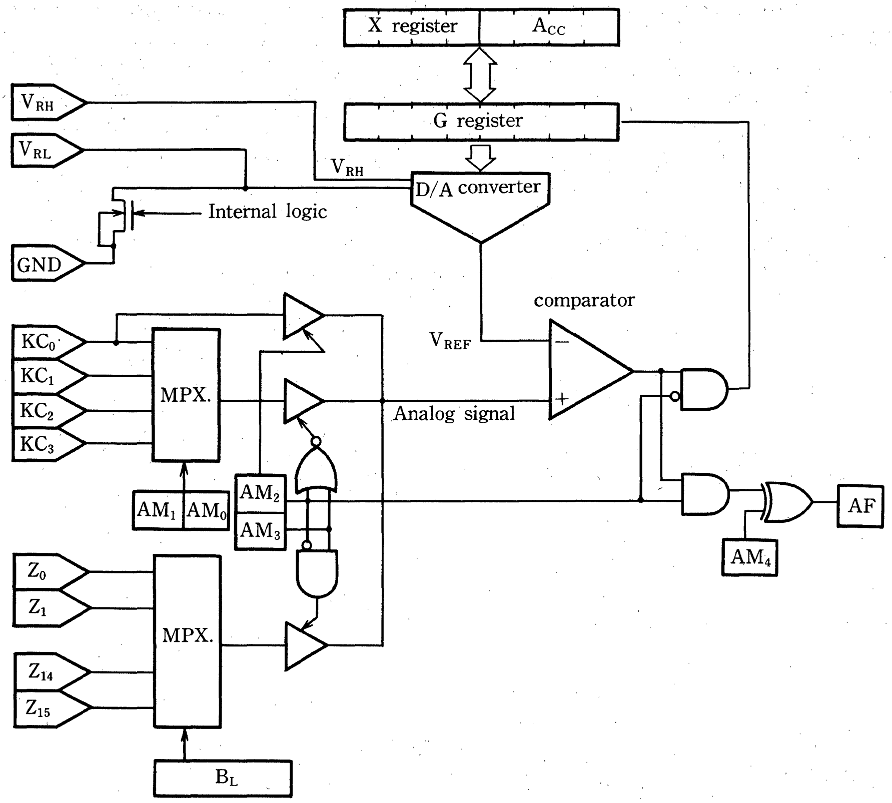


Fig. 4 A/D converter block

(13) Standby mode (for SM5J6)

To reduce power consumption, the device is placed in standby mode, and the program execution is inactivated.

The following two types of standby mode can be selected.

- **Off mode** In the off mode, the system clock generator circuits except for a counter/timer and a count-up pulse generator circuit are inactivated.
- **Hold mode** In the hold mode, the systems except for a system clock generator circuit, a counter/timer and a count-up pulse generator circuit are inactivated.

While in standby mode, if a KH input or an interrupt request from an unmasked KI, timer 1 or timer 2, the device exits standby mode and starts program execution.

(14) Reset function (ACL)

The device is reset with the interval of two instruction cycles from the rising edge of the ACL pin.

Applying a High level signal to the ACL pin resets the internal logic of the device and applying a Low level signal starts execution of the program at address 0, page 0.

In case the noise may harm the ACL operation, apply a capacitor between ACL pin and V_{DD} pin.

(15) Serial I/O

The serial I/O consists of an 8-bit shift register, a 3-bit counter and a 6-bit mode flag, which have the following features.

- Selectable either an 8-bit, a 4-bit, a 2-bit or a 1-bit transfer system.
- Interrupt request available at the end of transfer.
- Selectable transfer clock among a system clock, a timer 2 output or an external clock.
- Connectable to multiple chips.
- Usable in standby mode.
- An 8-bit shift register replaceable by the R/W register when the serial I/O is not used.

Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TR x	100-17F	Jump within a page, P_L←I_6-I_0
TL x	0F0-0F7	Jump
	000-1FF	Pu←I_{11}-I_9, P_M←I_8-I_7; P_L←I_6-I_0
MTPL	08A	Jump within a page, (P_L←A_2-A_0 M_3-M_0)
TRS x	180-1FF	CALL indirect address
JUMP	000-1FF	P_U←1, P_M←I_8, I_7, P_L←I_6-I_0, if DI=1
TLS x	0F8-0FF	CALL to sulbroutine
	000-1FF	P_U←I_{11}-I_9, P_M←I_8-I_7, P_L←I_6-I_0
RTN	0C0	Return
RTNS	0C1	Return and skip
RTNI	0C2	Return from interrupt
COMET	08B	ET←$\overline{ET}$

(2) Data transfer instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
LAX x	040-04F	Acc←I_3-I_0, Skip if last instruction is LAX
WLAX x	087	X←I_7-I_4, Acc←I_3-I_0
	000-0FF	
LBMX x	0E0-0EF	B_M←I_3-I_0
LBLX x	020-02F	B_L←I_3-I_0
STXI x	050-05F	M←I_3-I_0, B_L←B_L+1, Skip if CY=1
EXCI x	070-077	M↔Acc, B_M←B_M⊕I_2-I_0 B_L←B_L+1, Skip if CY=1
EXCD x	078-07F	M↔Acc, B_M←B_M⊕I_2-I_0 B_L←B_L+F_H, Skip if CY=1
EXC x	068-06F	M↔Acc, B_M←B_M⊕I_2-I_0
LDA x	060-067	Acc←M, B_M←B_M⊕I_2-I_0
STR	09E	M←Acc
EXAX	0A6	Acc↔X
ATX	0AE	X←Acc
GTAX	0BD	X←G_7-G_4, Acc←G_3-G_0
AXTG	08D	G_7-G_4←X, G_3-G_0←Acc
	0BD	
XBLA	0B3	B_L↔Acc
BLTA	0B1	Acc←B_L
XBMA	0B2	B_M↔Acc
BMTA	0BA	Acc←B_M
XBSB	084	B↔SB
BTSB	085	SB↔B
SAG	08D	B_M←0 only next step
SGL x	0C8-0CF	B_M=I_2-I_0 B_L=F_H only next step
ATIM	0B4	Ei←Ai (i=3 to 0)

(3) Arithmetic instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
ADX x	000-00F	Acc←Acc+I_3-I_0, Skip if CY=1
ADA	09A	Acc←Acc+A_H
ADD	090	Acc←Acc+M
ADS	091	Acc←Acc+M, Skip if CY=1
ADC	092	Acc←Acc+M+C, C←CY
ADCS	093	Acc←Acc+M+C, C←CY, Skip if CY=1
ADBL	0BB	B_L←Acc+B_L
AND	0A1	Acc←Acc∧x
OR	0B0	Acc←Acc∨x
COMA	086	Acc←$\overline{\text{Acc}}$
ROT	09B	C←A_3←A_2←A_1←A_0←C
DECA	09F	Acc←Acc+F_H, Skip if CY=0
INCX	0A7	X←X+1, Skip if CY=1
INBL	0A3	B_L←B_L+1, Skip if CY=1
DNBL	0AB	B_L←B_L+F_H, Skip if CY=0
INBM	0A2	B_M←B_M+1, Skip if CY=1
DEBM	0AA	B_M←B_M+F_H, Skip if CY=0

(4) Test instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TAX x	010-01F	Skip if Acc=I_3-I_0
TBA x	0D4-0D7	Skip if Ai=1 (i=3 to 0)
TM x	0D0-0D3	Skip if Mi=1 (i=3 to 0)
TAM	096	Skip if Acc=M
TXM	0B6	Skip if X=M
TBLX x	030-03F	Skip if B_L=I_3-I_0
TC	0B8	Skip if C=1
TS	0B9	Skip if S=1
TIF	0C7	Skip and reset if IF=1
THAF	0C6	Skip and reset if HF=1 (AM_5=0) if AF=1 (AM_5=1)
TTF1	0C5	Skip and reset if TF_1=1
TTF2	0C4	Skip and reset if TF_2=1
TQZ	0A0	Skip if Q=0
TZ	080	Skip if Z [B_L]=1

(5) Bit manipulation instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
SM x	0DC-0DF	Mi←1 (i=3 to 0)
RM x	0D8-0DB	Mi←0 (i=3 to 0)
SC	099	C←1
RC	098	C←0
SS	0A9	S←1
RS	0A8	S←0
IE	095	E←1
ID	094	E←0

(6) I/O instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
ATQ	08E	Q←Acc
QTA	0BE	Acc←Q
ATP	08F	P←Acc,P_M←1
PTA	0A4	Acc←P, P_M←0
ATR	08C	R [B_L]←Acc
RTA	0C3	Acc←R [B_L]
AXTR	0AC	R [1]←X, R [0]←Acc
RTAX	0AD	X←R [1], Acc←R [0]
MTR	09C	R [B_L]←M
KCTA	0BC	Acc←KC
KITA	0BF	A_3←K_1, A_2←K_H, A_1←K_T, A_0←K_L
SZ	083	Z [BL]←1
RZ	082	Z [BL]←0
SF	089	F←1, FM_1←A_1, FM_0←A_0
RF	088	F←0

(7) Timer control instructions

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
TCTRL x	0AF	LM←I_7-I_0
	000-0FF	
STM1	097	TIMER1←X, Acc
LTM1	09D	X, Acc←TIMER1
STM2 x	0B7	TIMER2←MR, MR←I_7-I_0
	000-0FF	
LTM2	081	X, Acc←TIMER2

(8) A/D conversion instruction

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
COMP	0A5	$AM_5 \leftarrow A_2 \wedge A_1$, $A_4 \leftarrow A_2 \wedge A_0$, $AM_3 \leftarrow A_3$ $AM_2 \leftarrow A_2$, $AM_1 \leftarrow A_1$, $AM_3 \leftarrow A_0$ A/D Conversion or Comparing

(9) Standby instruction

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
CCTRL	0B5	$CM_2 \leftarrow A_2$, $CM_1 \leftarrow A_1$, $CM_0 \leftarrow A_0$ Standby mode if $A_3 = 0$

(10) Table reference instruction

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	
LAT	08A 000-0FF	PUSH (SP←SP+1, SR←PC+1) PL_6-$PL_4 \leftarrow A_2$-A_0, PL_3-$PL_0 \leftarrow M_3$-M_0 POP (SP←SP−1, PC←SR) $X \leftarrow I_7$-I_4, $A \leftarrow I_3$-I_0

■ Comparison Table Between SM5J5 and SM5J6

The SM5J5 directly drives a fluorescent display tubes. The SM5J6 provides two modes of standby function for power saving.

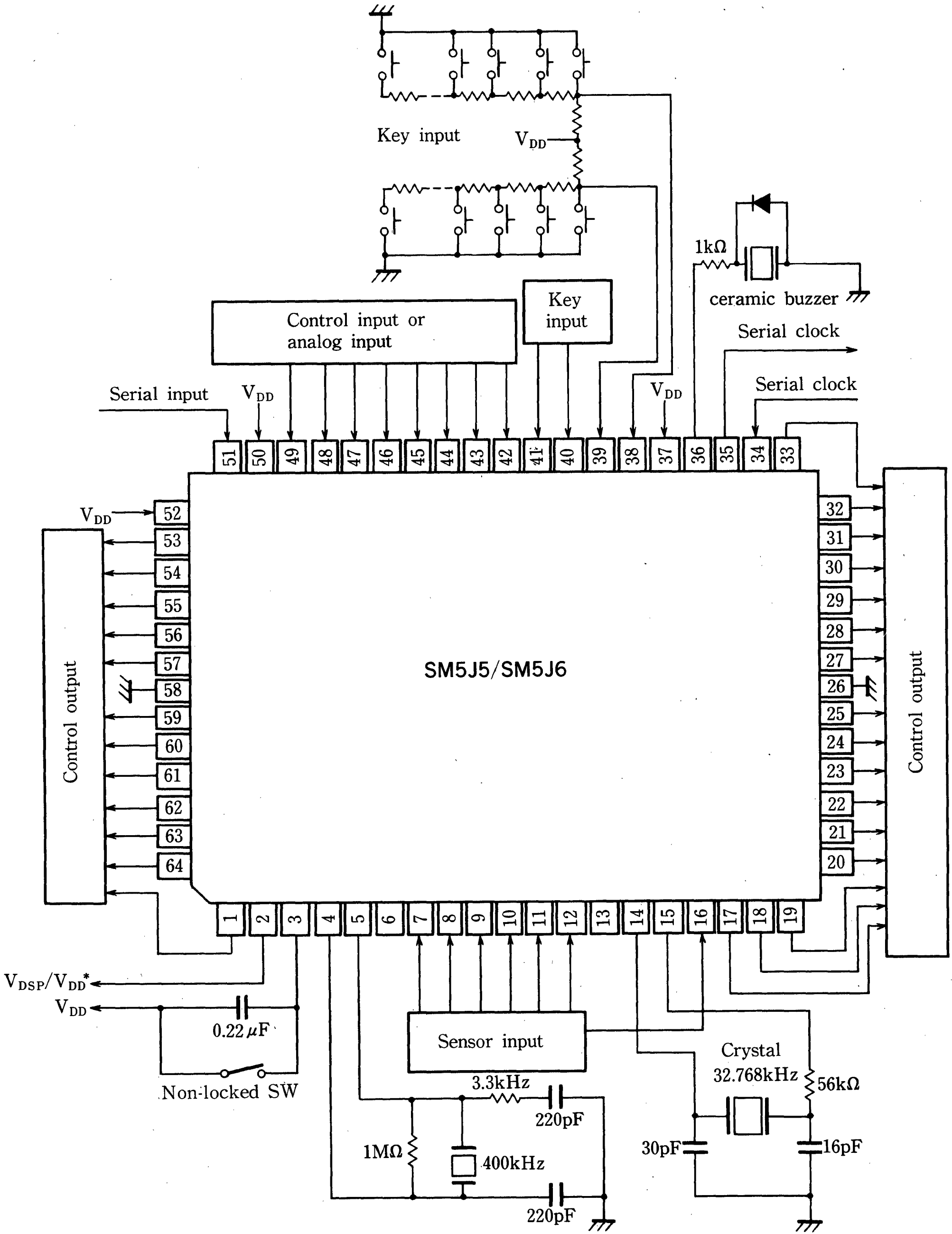
Both models have the same function except for the specifications in the electrical characteristics and I/O ports. See the related sections for details.

Table 2

	SM5J5	SM5J6
ROM	8,192×9 bits	
RAM	256×4 bits	
Instruction set	94 set	
Power supply	+4.5 to +5.5V	+2.7 to +5.5V
Instruction cycle	2.5 μs (MIN.)	2 μs (MIN.)
System clock	50 to 400kHz (312.5kHz TYP.)	50 to 500kHz
Ports R [0]-R[3]*	High voltage (−40V) I/O ports	I/O ports with a pull-down resistors
Ports Z_0-Z_9	High votage (−40V) output ports	Output ports with a pull-down resistor
Operating temperature	−10 to +80℃	
Package	64-pin SDIP 64-pin QFP	

*R (0)-R (3): $R0_0$-$R0_3$, $R1_0$-$R1_3$, $R2_0$-$R2_3$, $R3_0$-$R3_3$.

■ System Configuration Example



*SM5J5: $V_{DSP}=-30V$ (TYP.)
SM5J6: $V_{DD}=+5V$ (TYP.)

SM530 4-Bit Microcomputer (LCD Driver)

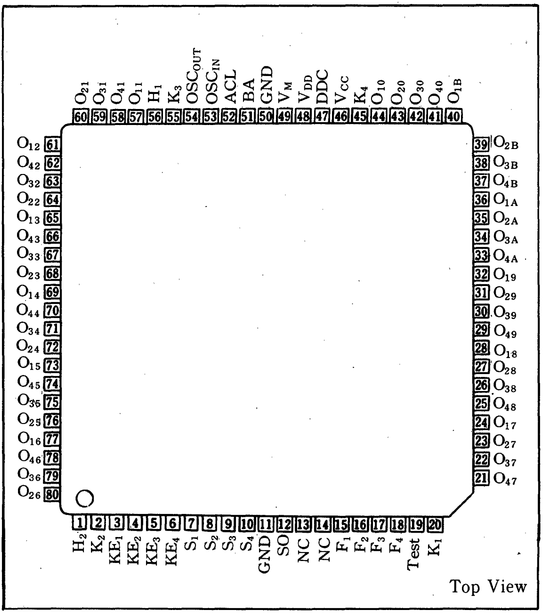
Description

The SM530 is a CMOS 4-bit microcomputer, operated on a single 1.5V power supply with a 1.5 μA power consumption in standby mode. This microcomputer integrates a 4-bit parallel processing function, a 2K byte ROM, an 88 word RAM, a 96-segment LCD driver, a real-time counter circuit, and a melody generator circuit in a single chip. Provided with 1.5V single power supply and a low power consumption design, it is applicable to compact systems required for battery back-up operation.

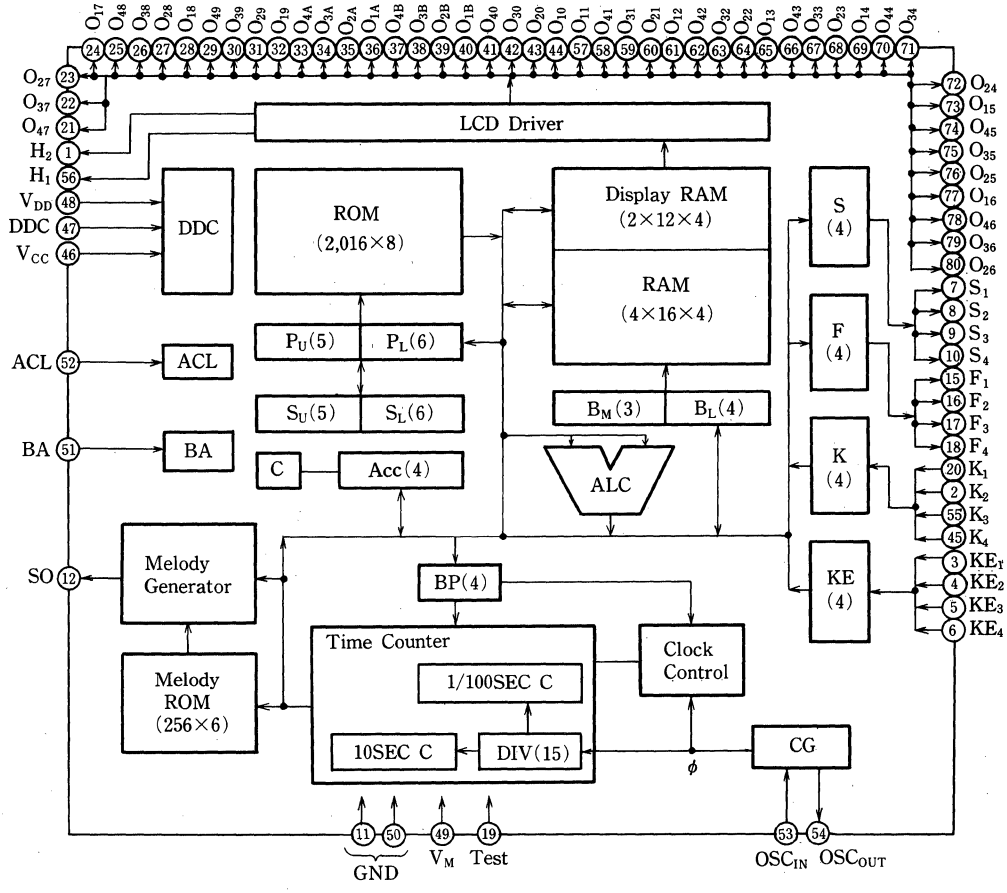
Features

1. CMOS process
2. ROM capacity: 2,016×8 bits
3. RAM capacity: 88×4 bits
4. Instruction set: 49
5. Subroutine nesting: 1 level
6. Instruction cycle: 91.6 μs (TYP.)
7. Input/output ports
   Input ports: 9
   Output ports: 8
   LCD output ports: 48 for segment
   2 for common
   Melody output port: 1
8. On-chip clock divider
9. On-chip crystal oscillator (32.768kHz)
10. Programmable interval timer
    (10 sec, 1 sec, 1/2 sec, 1/10 sec)
11. 1/100 sec counter
12. Melody generator circuit
13. Standby function
14. Single power supply: −1.5V (TYP.)
15. 80-pin QFP (QFP80-P-1818)

Pin Connections



Block Diagram


LCD Driver
ROM (2,016×8)
Display RAM (2×12×4)
RAM (4×16×4)
DDC
ACL
BA
$P_U(5)$ $P_L(6)$
$S_U(5)$ $S_L(6)$
C
Acc(4)
$B_M(3)$ $B_L(4)$
ALC
S (4)
F (4)
K (4)
KE (4)
Melody Generator
Melody ROM (256×6)
BP(4)
Time Counter
1/100SEC C
10SEC C
DIV(15)
Clock Control
CG
ϕ
O_{27} O_{37} O_{47} H_2 H_1 V_{DD} DDC V_{CC} ACL BA SO
O_{24} O_{15} O_{45} O_{35} O_{25} O_{16} O_{46} O_{36} O_{26}
S_1 S_2 S_3 S_4 F_1 F_2 F_3 F_4 K_1 K_2 K_3 K_4 KE_1 KE_2 KE_3 KE_4
GND V_M Test OSC_{IN} OSC_{OUT}

Symbol description

ALU	: Arithmetic logic nuit	DDC	: LCD supply voltage generator
Acc	: Accumulator	DIV	: Divider
ACL	: Auto clear	CG	: Clock generator
C	: Carry F/F	BA	: Battery alarm circuit
P_U, P_L	: Program counter	B_M, B_L	: RAM address register
S_U, S_L	: Stack register of program counter		

Pin Description

Symbol	I/O	Circuit type	Function
K_1-K_4, KE_1-KE_4	I	pull-down	Input ports
S_1-S_4, F_1-F_4	O		Output ports
O_{10}-O_{4B}	O		Segment signal output ports
H_1-H_2	O		Common signal output ports
OSC_{IN}, OSC_{OUT}			Crystal oscillator
SO	O		Melody output port
ACL	I	Pull-down	Auto clear input port
BA	I		Battery alarm input port
V_{CC}, DDC, V_{DD}			Power supply for booster circuit
V_M, GND			Power supply
Test	I	Pull-down	Test input (normally connected to V_M)

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_M	−2.0 to +0.3	V	1
	V_{DD}	−4.0 to +0.3	V	1, 2
	V_{IN1}	V_M−0.3 to +0.3	V	1, 3
	V_{IN2}	V_{DD}−0.3 to +0.3	V	1
Operating temperature	Topr	0 to +50	℃	
Storage temperature	Tstg	−20 to +70	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.
Note 2: Applied to pins K_1−K_4, KE_1−KE_4, S_1−S_4, F_1−F_4, SO, Test, DDC, BA, ACL, OSC_{IN}, OSC_{OUT}.
Note 3: Applied to pins O_{ij} (i=1 to 4, j=0 to B) H_1, H_2, V_{CC}.

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_M	−1.8 to −1.2	V	
	V_{DD}	−3.6 to −2.3	V	
Oscillation start voltage	V_{osc}	−1.4	V	1
Oscillator frequency	f_{osc}	32.768(TYP.)	kHz	

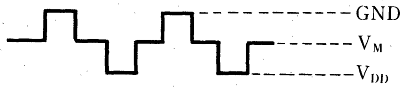
Note 1: Oscillation circuit constants: C_G=15pF, C_D=22pF
The oscillation start time should be within 10 sec.

Electrical Characteristics

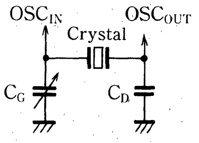
(V_M=−1.45 to 1.55V, V_{DD}=−2.9 to −3.1V, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH}		−0.5			V	1
	V_{IL}				V_M+0.5	V	
Input current	I_{IH1}	V_{IN}=0V	0.155		3	μA	2
	I_{IH2}	V_{IN}=0V	1.55		30	μA	3
Boost output voltage	V_{DD1}	V_M=−1.55V, R_L=5MΩ			−2.80	V	4
	V_{DD2}	V_M=−1.30V, R_L=5MΩ			−2.30	V	
Output current	I_{O1}	V_{DS}=0.5V	10			μA	5
	I_{O2}	V_{DS}=0.5V	60			μA	6
	I_{O3}	V_{DS}=0.5V	60			μA	7
	I_{O4}	V_{DS}=0.5V	120			μA	8
	I_{OH1}	V_{OUT}=−0.5V	160			μA	9
	I_{OL1}	V_{OUT}=V_M+0.5V	10			μA	
	I_{OH2}	V_{OUT}=−0.5V	10			μA	10
	I_{OL2}	V_{OUT}=V_M+0.5V	1.5			μA	
	I_{OH3}	V_{OUT}=−0.5V	100			μA	11
	I_{OL3}	V_{OUT}=V_M+0.5V	3			μA	
	I_{O5}	V_{DS}=0.5V	100			μA	12
Current consumption	I_{DO}	During full-range operation		12		μA	13
	I_{DS}	During system clock stop		1.5		μA	
Oscillation start time	t_{OSC}			10		s	14

Note 1: Applied to pins K_1-K_4, KE_1-KE_4, Test, ACL, OSC_{IN}
Note 2: Applied to pins K_1-K_4, KE_1-KE_4, ACL
Note 3: Applied to pin Test
Note 4: Applied to pin V_{DD}
Note 5: Applied to pins O_{ij} (i=1 to 4, j=0 to B)
Note 6: Applied to pins H_1, H_2
Note 7: Applied to pin DDC
Note 8: Applied to pin V_{CC}
Note 9: Applied to pin SO
Note 10: Applied to pins S_1-S_4
Note 11: Applied to pin F_1
Note 12: Applied to pins F_2-F_4
Note 13: Current consumption at 32.768kHz
Note 14: Oscillation circuit constant, C_G=15pF, C_P=22pF

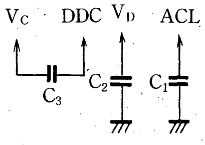


H_1, H_2 waveform



C_G=15pF, C_D=22pF

Oscillation circuit



C_1=C_2=C_3=0.1μF

Boost circuit

Hardware Configuration

(1) Program memory (ROM)

The on-chip ROM has a configuration of 32 pages×63 steps×8 bits (see Fig. 1).

The program counter consists of a 5-bit page address counter (PU) used to specify the pages 0 to 31, and a 6-bit polynomial counter (P_L) used to specify the steps 0 to 62.

The stack register is an 11 bit register which allows 1 level of subroutine nesting.

(2) Data memory (RAM) and data address register (B_M, B_L)

The data memory has a 352 bit organized as $4\times16\times4+2\times12\times4$, and its address is specified by the data address registers (B_M, B_L). The B_M is used to specify the file in the data memory, and the B_L used to specify the word within a file.

2 files (B_M=4 to 7) of data memory are allocated to the display RAM. The data set herein is loaded into the LCD segment pins. Fig. 2 shows the RAM configuration.

(3) K_1-K_4, KE_1-KE_4,BA (Inputs)

Ports K and KE are 4-bit input ports with pull-down resistors. The contents of these pins can be loaded into the accumulator A_{CC} by instructions.

PU_5, PU_4 / PU_3~PU_1	00	01	10	11
000	Page 0 Note 1 and 4	8	16	24
001	1 Note 4	9	17	25
010	2 Note 4	10	18	26
011	3 Note 4	11	19	27
100	4 Note 4	12	20	28
101	5 Note 4	13	21	29
110	6 Note 4	14 Note 3	22	30
111	7 Note 4	15 Note 2	23	31

Note 1: Page 0: The address where the clock restarts from the standby mode.
Note 2: Page 15: The starting address with the ACL.
Note 3: Page 14: Subroutine cover page
Note 4: Pages 0 through 7: The address which allows a jump by the JUMP instruction. P_{L6}=0

Fig. 1 ROM configulation

B_M \ B_L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4, 6*	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨				
5, 7*	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨				

Note: The shadowed area is allocated for a display RAM.
* Executing the SABM instruction at the file 2 or 3 of the B_M register specifies the file 6 or 7 of the B_M in the same location of the file 4 or 5.

Fig. 2 RAM configuration

While in standby mode, if the K or KE pin ac cepts an input signal, the CPU is initialized and sta rts execution of the program at $P_U=0_H$, $P_L=00_H$.

The BA pin can be used as an input pin which allows testing the input fixed at High or Low, by instructions.

(4) F_1-F_4, S_1-S_4 (Outputs)

Ports F and S are 4-bit output ports.The accumulator A_{CC} can be transferred to these ports by instructions.

(5) Divider and clock counter

The device contains a real-time clock divider, 1 sec counter and 1/100 sec counter. These counters generate signals of 10 sec, 1 sec, 1/2 sec, 1/10 sec which can be tested by instructions, and constitute a real-time clock.

Either 1 sec counter or 1/100 counter can be directly indicated on an LCD screen through a decoder. The contents of 1/100 sec counter can be loaded into the accumulator A_{CC} by instructions.

(6) LCD driver

The SM530 contains an on-chip LCD driver which can directly drive a 96-segment LCD with a 1/2 duty and 1/2 bias scheme.

Fig. 3 shows the common and segment waveform. The display frequency is 128Hz, and 3V of display voltage is obtained through an internal booster circuit.

The display data is transferred through an LCD driver circuit, and displayed on an LCD screen.

Fig. 4 shows an example of a seven-segment numeric LCD digit. The RAM data of $B_M=4$, 6 corresponds to the H_1 segment, and $B_M=5$, 7 corresponds to the H_2 segment.

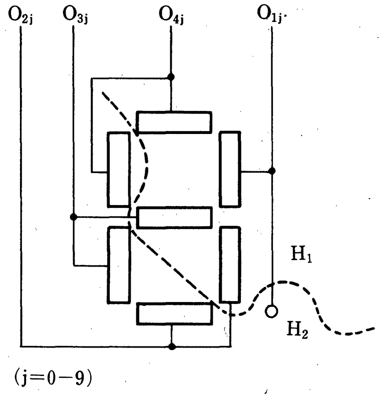


Fig. 4 7-segment numeric LCD digit

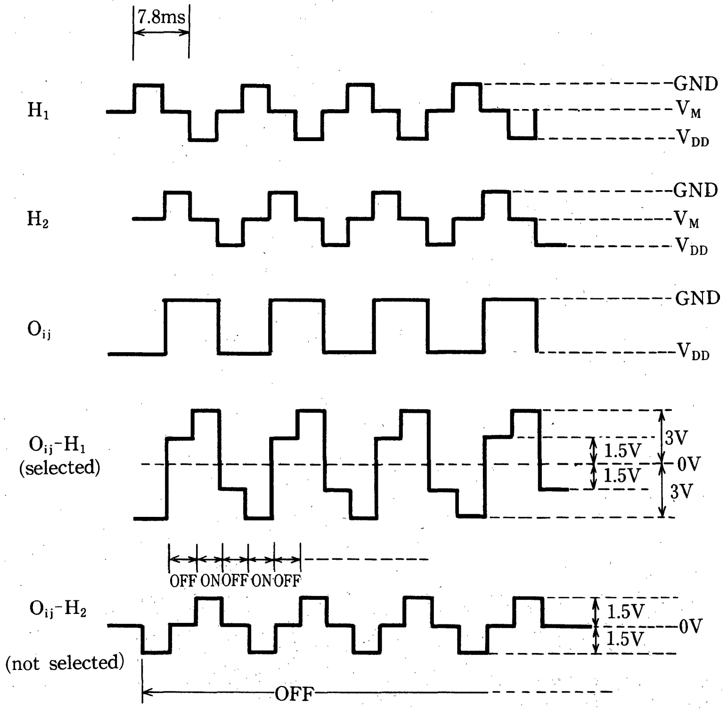


Fig. 3 LCD driving signal waveform

(7) Display decoder

The 1 sec counter or 1/100 sec counter is loaded into the display decoder, and output as segment signals through pins $O_{10}-O_{40}$. The display decoder can not be used when the RAM data is displayed on an LCD.

Fig. 5 shows the relationship between the display RAM and pins Oij. Table 1 shows the truth table of the display decoder.

Table 1 Display decoder truth table

1set or 1/100sec counter	Display character	1sec or 1/100sec counter	Display character
0000	0	0101	5
0001	1	0110	6
0010	2	0111	7
0011	3	1000	8
0100	4	1001	9

Note: The display segment of a floating point is specified by the first bit of a display RAM $(B_M, B_L)=(5, 0)$

(8) Melody generator circuit

The contents of a melody ROM can be output with standard 12 musical scales (555 to 2114Hz) in two octaves from the SO pin.

The tone length can be selected between 250ms and 125ms depending on the melody ROM. The melody ROM provides a pause and a stop instruction.

Controlling the melody F/F (ME F/F) by instructions starts and stops melody. The melody ROM stores up to 256 steps of musical notes.

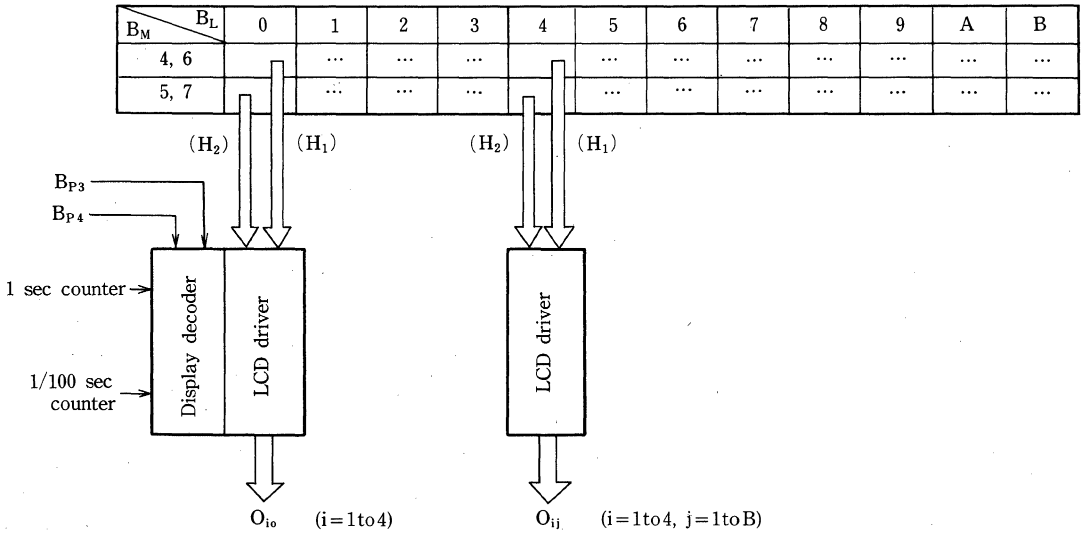
Table 2 shows the musical scales in one octave. Executing an instruction from a melody ROM outputs half frequencies of the standard frequencies shown in table 2, and generates lower 12 musical scales by one octave.

(9) ACL circuit

The ACL circuit contains a resistor and a capacitor, which does not require any external circuits. The ACL may be cleared with the interval of about 0.5 sec after a crystal oscillator circuit starts oscillation when the power is turned on, and starts execution of the program at $P_U=F_H$, $P_L=00_H$.

(10) Standby mode

Executing an instruction places the device in standby mode. In this mode, the system clock is inactivated to reduce power consumption. While in standby mode, if the K or KE pin receives an input signal, or the selecting γ F/F is set, the device exits standby mode and the CPU starts execution of the program at $P_U=0_H$, $P_L=00_H$.



B_M \ B_L	0	1	2	3	4	5	6	7	8	9	A	B
4, 6		···	···	···		···	···	···	···	···	···	···
5, 7		···	···	···		···	···	···	···	···	···	···

Fig. 5 Display RAM and Oij

Table 2 Melody output frequency

Musical scale	do#	re	re#	mi	fa	fa#	so	so#	la	la#	si	do
Frequency (Hz)	1110.8	1170.3	1236.5	1310.7	1394.4	1489.5	1560.4	1680.4	1771.2	1872.4	1985.9	2114.1

(11) Booster circuit

The device contains a booster circuit which generates a voltage two times higher than the 1.5V power supply.

Then, it is necessary to apply external capacitors between DDC pin and V_{CC} pin as well as V_{DD} pin and GND (see Fig. 6).

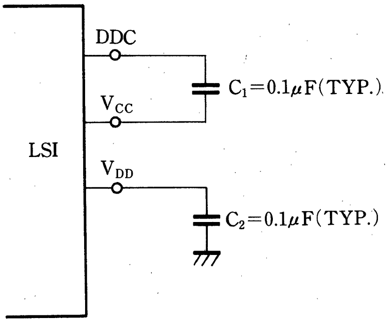


Fig. 6 Booster circuit

(12) System clock

The system clock has a frequency of one third that of a 32.768kHz clock.

The instruction cycle time should be 91.5μs.

■ Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
INCB	4C	$B_L \leftarrow B_L+1$ skip if $B_L=7$ or F
DECB	4D	$B_L \leftarrow B_L-1$ skip if $B_L=0$
LB xy	30-3F	$B_{M3} \leftarrow 0$ $B_{M2}, B_{M1} \leftarrow x(I_4, I_3)$ $B_{L4}, B_{L1} \leftarrow y(I_2, I_1)$ $B_{L3}, B_{L2} \leftarrow (1, 1)$
LBL xy	6B 00-FF	$B_M \leftarrow x(I_7\text{-}I_5)$ $B_L \leftarrow y(I_4\text{-}I_1)$
SABM	72	$B_{M3} \leftarrow 1$ next step only
SABL	73	$B_{L4} \leftarrow 1$ next step only
EXBL	5A	$A_{CC} \leftrightarrow B_L$

(2) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BF	$P_L \leftarrow x(I_6\text{-}I_1)$
TL xy	60-67 00-FF	$P_U \leftarrow x(I_{11}\text{-}I_7)$ $P_L \leftarrow y(I_6\text{-}I_1)$
TRS x	C0-FF	$P_U \leftarrow 01110$, $P_L \leftarrow x(I_6\text{-}I_1)$ $SR \leftarrow PC+1$
JUMP xy	00-FF	$P_{U5}, P_{U4} \leftarrow (0, 0)$ $P_{U3}\text{-}P_{U1} \leftarrow x(I_6, I_8, I_7)$ $P_{L6} \leftarrow 0$ $P_{L5}\text{-}P_{L1} \leftarrow y(I_5\text{-}I_1)$
ATPL	6A	$P_{L4}\text{-}P_{L1} \leftarrow A_{CC}$
RTN	68	$PC \leftarrow SR$
RTNS	69	$PC \leftarrow SR$ skip the next step

(3) Data transfer instructions

Mnemonic	Machine code	Operation
LAX x	10-1F	$A_{CC} \leftarrow x(I_4\text{-}I_1)$
LDA x	20-23	$A_{CC} \leftarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x(I_2, I_1)$
EXC x	24-27	$A_{CC} \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x(I_2, I_1)$
EXCI x	28-2B	$A_{CC} \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x(I_2, I_1)$ $B_L \leftarrow B_L+1$ skip if $B_L=7$ or F
EXCD x	2C-2F	$A_{CC} \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x(I_2, I_1)$ $B_L \leftarrow B_L-1$ skip if $B_L=0$
DTA	52	$A_{CC} \leftarrow 1/100$ SEC. C.

(4) Arithmetic instructions

Mnemonic	Machine code	Operation
ADD	54	$A_{CC} \leftarrow A_{CC}+M$
ADDC	55	$A_{CC} \leftarrow A_{CC}+M+C$ $C \leftarrow C_4$ skip if $C_4=1$
ADX x	00-0F	$A_{CC} \leftarrow A_{CC}+x(I_4\text{-}I_1)$ skip if $C_4=1$
COMA	53	$A_{CC} \leftarrow A_{CC}$

(5) Test instructions

Mnemonic	Machine code	Operation
TM x	48-4B	skip if $Mi=1$ $(i=I_2I_1)$
TC	5B	skip if $C=0$
TAM	59	skip if $A_{CC}=M$
TABL	58	skip if $A_{CC}=B_L$
TG x	6C-6F	skip if $\gamma=1$, $\gamma \leftarrow 0$ $(\gamma_{10S}, \gamma_{1S}, \gamma_{0.5S}, \gamma_{0.1S})$
TBA	79	skip if $BA=1$

(6) Bit manipulation instructions

Mnemonic	Machine code	Operation
RM x	40-43	Mi←0 (i=I_2I_1)
SM x	44-47	Mi←1 (i=I_2I_1)
RC	56	C←0
SC	57	C←1

(7) I/O instructions

Mnemonic	Machine code	Operation
KTA	50	A_{CC}←K
KETA	51	A_{CC}←KE
ATS	5C	S←A_{CC}
ATF	5D	F←A_{CC}
ATBP	5E	BP←A_{CC}
SDS	4F	DS←1
RDS	4E	DS←0

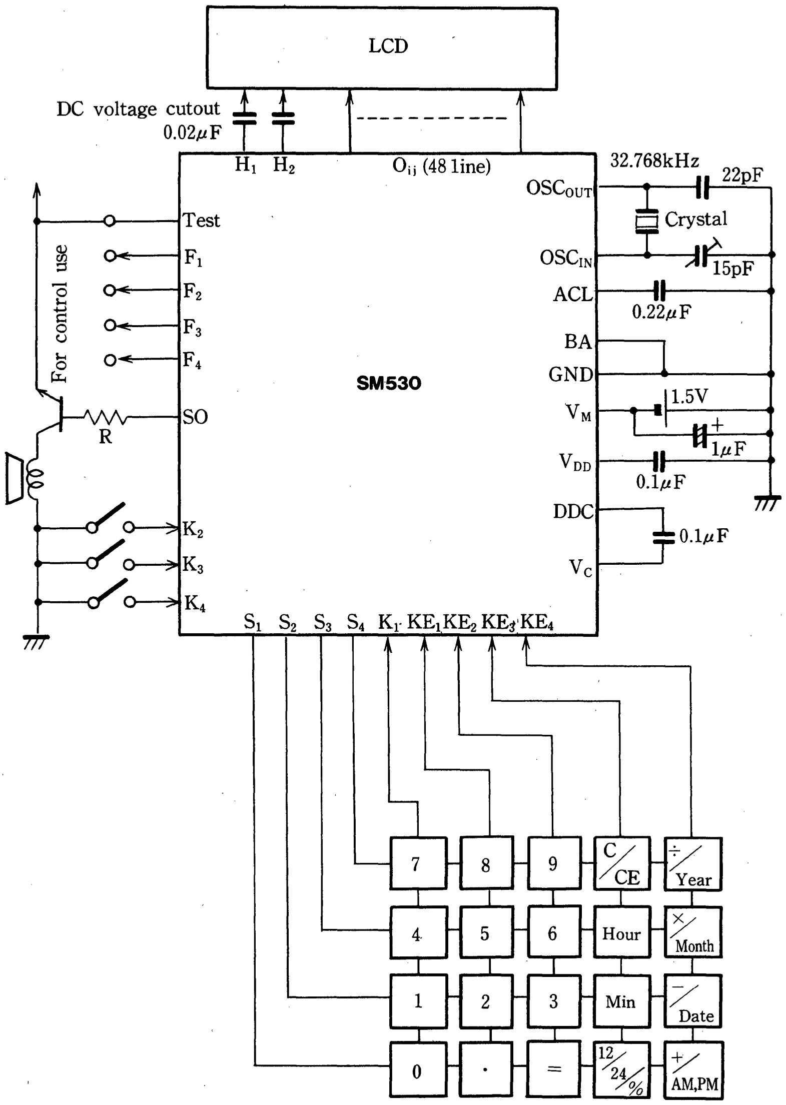
(8) Melody instructions

Mnemonic	Machine code	Operation
PRE x	78 00-FF	Melody ROM pointer preset
SME	77	ME←1
RME	76	ME←0
TMEL	75	skip if MES=1 MES←0

(9) Special instructions

Mnemonic	Machine code	Operation
CEND	74	System clock stop
IDIV	70	DIV←0 1 SEC. C.←0
INIS	71	1/100 SEC. C.←0
SKIP	00	No operation

System Configuration Example (Calculator watch)



Key matrix

SM531 4-Bit Microcomputer (LCD Driver)

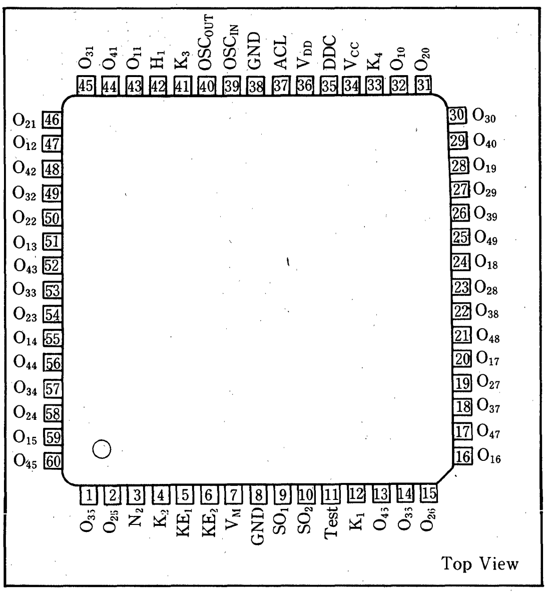
■ Description

The SM531 is a CMOS 4-bit microcomputer, operated on a single 1.5V power supply with a 1.5 μA power cosumption in standby mode. This microcomputer integrates a 4-bit parallel processing function, a 1.2K byte ROM, a 52 word RAM, an 80-segment LCD driver, a real-time counter circuit, and a melody generator circuit in a single chip. Provided with 1.5V single power supply and a low power consumption design, it is applicable to compact systems required for battery back-up operation.

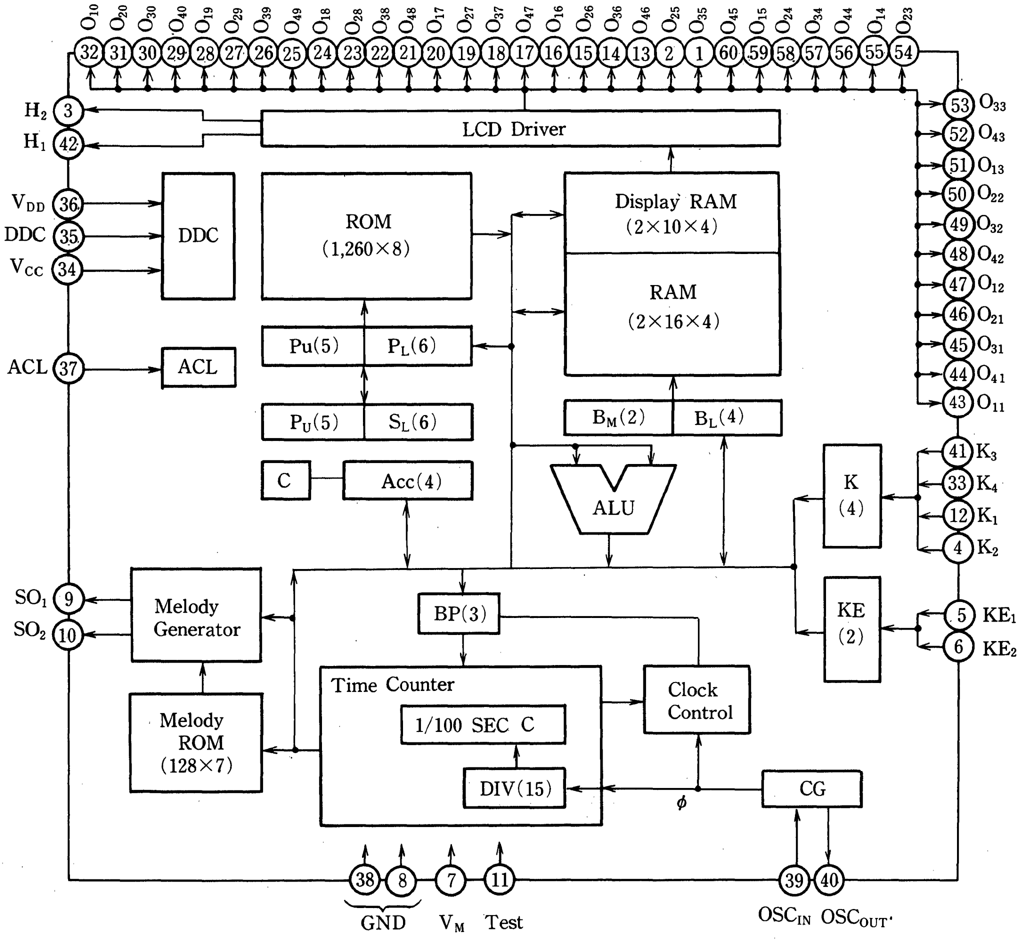
■ Features

1. CMOS process
2. ROM capacity: 1,260×8 bits
3. RAM capacity: 52×4 bits
4. Instruction set: 45
5. Subroutine nesting: 1 level
6. Insturction cycle: 91.6 μs (TYP.)
7. Input/output ports
   Input ports: 6
   LCD output ports: 40 for segment
   2 for common
   Melody output ports: 2
8. On-chip clock divider
9. On-chip crystal oscillator (32.768kHz)
10. Programmable interval timer
    (1 sec, 1/2 sec, 1/10 sec)
11. 1/100 sec counter
12. Melody generator circuit
13. Standby function
14. Single power supply: −1.5V (TYP.)
15. 60-pin QFP (QFP60−P−1414)

■ Pin Connections



Block Diagram



Symbol description

ALU	: Arithmetic logic unit	S_U, S_L	: Stack register of program counter
A_{CC}	: Accumulator	DDC	: LCD supply voltage generator
ACL	: Auto clear	DIV	: Divider
C	: Carry F/F	CG	: Clock generator
P_U, P_L	: Program counter	B_M, B_L	: RAM address register

Pin Description

Symbol	I/O	Circuit type	Function
K_1-K_4, KE_1, KE_2	I	Pull-down	Input ports
O_{10}-O_{49}	O		Segment signal output ports
H_1-H_2	O		Common signal output ports
OSC_{IN}, OSC_{OUT}			Crystal oscillator
SO_1, SO_2	O		Melody output ports
ACL	I		Auto clear input port
BA	I		Battery alarm input port
V_{CC}, DDC, V_{DD}			Power supply for booster circuit
V_M, GND			Power supply
Test	I	Pull-down	Test input (normally connected to V_M)

■ Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_M	-2.0 to $+0.3$	V	1
	V_{DD}	-4.0 to $+0.3$	V	1
	V_{IN1}	$V_M-0.3$ to $+0.3$	V	1, 2
	V_{IN2}	$V_{DD}-0.3$ to $+0.3$	V	1, 3
Operating temperature	Topr	0 to +50	℃	
Storage temperature	Tstg	−20 to +70	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.
Note 2: Applied to pins K_1-K_4, KE_1, KE_2, SO_1, SO_2, Test, DDC, ACL, OSC_{IN}, OSC_{OUT}.
Note 3: Applied to pins O_{ij} (i=1 to 4, j=0 to 9), H_1, H_2, Vc.

■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_M	-1.8 to -1.2	V	
	V_{DD}	-3.6 to -2.3	V	
Oscillation start valtage	V_{OSC}	-1.4	V	1
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz	

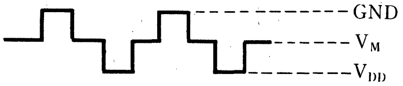
Note 1: Oscillation circuit constant, C_G=15pF, C_D=22pF.
Oscillation start time: within 10 seconds.

■ Electrical Characteristics

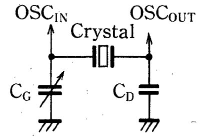
($V_M=-1.45$ to -1.55V, $V_{DD}=-2.9$ to -3.1V, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH}		-0.5			V	1
	V_{IL}				$V_M+0.5$	V	
Input current	I_{IH1}	V_{IN}=0V	0.155		3	μA	2
	I_{IH2}	V_{IN}=0V	1.55		50	μA	3
Boost output voltage	V_{DD1}	$V_M=-1.55$V, R_L=5MΩ			-2.80	V	4
	V_{DD2}	$V_M=-1.30$V, R_L=5MΩ			-2.30	V	
Output current	I_{O1}	V_{DS}=0.5V	10			μA	5
	I_{O2}	V_{DS}=0.5V	60			μA	6
	I_{O3}	V_{DS}=0.5V	60			μA	7
	I_{O4}	V_{DS}=0.5V	120			μA	8
	I_{O5}	V_{DS}=0.5V	900			μA	9
Current consumption	I_{DO}	During full-range operation		10		μA	10
	I_{DS}	During system clock stop		1.5		μA	
Oscillation starting time	T_{OSC}			10		s	11

Note 1: Applied to pins K_1-K_4, KE_1-KE_4, ACL, OSC_{IN}
Note 2: Applied to pins K_1-K_4, KE_1-KE_4, ACL
Note 3: Applied to pin Test
Note 4: Applied to pin V_{DD}
Note 5: Applied to pins O_{ij} (i=1 to 4, j=0 to 9)
Note 6: Applied to pins H_1, H_2
Note 7: Applied to pin DDC
Note 8: Applied to pin V_{CC}
Note 9: Applied to pins SO_1, SO_2
Note 10: Current consumption at 32.768kHz
Note 11: Oscillation circuit constant, C_G=15pF, C_D=22pF

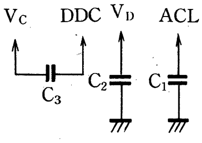


● H_1, H_2 waveforms



C_G=15pF, C_D=22pF

● Oscillation circuit



$C_1=C_2=C_3$=0.1μF

● Boost circuit

■ Hardware Configuration

(1) Program memory (ROM)

The on-chip ROM has a configuration of 20 pages×63 steps×8 bits (see Fig. 1). The program counter consists of a 5-bit page address counter (PU) used to specify the pages 0 to 19, and a 6-bit polynomial counter (P_L) used to specify the steps 0 to 62.

The stack register is an 11-bit register which allows 1 level of subroutine nesting.

(2) Data memory (RAM) and data address register (B_M, B_L)

The data memory has a 208-bit organized as 2 ×16×4+2×12×4, and its address is specified by the data address registers (B_M, B_L). The B_M is used to specify the file in the data memory, and the B_L used to specify the word within a file.

2 files (B_M=2, 3) of data memory are allocated to the display RAM. The data set herein is loaded into the LCD segment pins. Fig. 2 shows the RAM configuration.

PU_5,PU_4 \ PU_3-PU_1	00	01	10
000	Page 0 Note 1 and 4	8	16
001	1 Note 4	9	17
010	2 Note 4	10	18
011	3 Note 4	11	19
100	4 Note 4	12	
101	5 Note 4	13	
110	6 Note 4	14 Note 3	
111	7 Note 4	15 Note 2	

Note 1: Page 0 shows the address where the clock restarts from the standby mode.
Note 2: Page 15 shows the starting address with the ACL.
Note 3: Page 14 shows the subroutine cover page.
Note 4: Pages 0 through 7 show the addresses which allow a jump by the JUMP instruction, P_{L6}=0.

Fig. 1 ROM configuration

B_M \ B_L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨						
3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨						

Note : The shadowed area is allocated for a display RAM.

Fig. 2 RAM configuration

(3) K_1-K_4, KE_1, KE_2 (Inputs)

Ports K and KE are 4-bit input ports with pull-down resistors. The contents of these pins can be loaded into the accumulator A_{CC} by instructions.

While in standby mode, if the K or KE pin accepts an input signal, the CPU is initialized and starts execution of the program at $P_U = 0_H$, $P_L = 00_H$.

(4) Divider and clock counter

The device contains a real-time clock divider and a 1/100 sec counter. These counters generate signals of 1 sec, 1/2 sec, 1/10 sec which can be tested by instructions, and constitute a real-time clock.

The 1/100 counter can be directly indicated on an LCD screen through a decoder. The contents of 1/100 sec counter can be loaded into the accumulator A_{CC} by instructions.

(5) LCD driver

The SM531 contains an on-chip LCD driver which can directly drive an 80-segment LCD with a 1/2 duty and 1/2 bias scheme.

Fig. 3 shows the common and segment waveform. The display frequency is 128Hz, and 3V of display voltage is obtained through an internal booster circuit.

The display data is transferred through an LCD driver circuit, and displayed on an LCD screen.

Fig. 4 shows an example of a seven-segment numeric LCD digit. The RAM data of $B_M=2$ correspond to the H_1 segment, and $B_M=3$ corresponds to the H_2 segment.

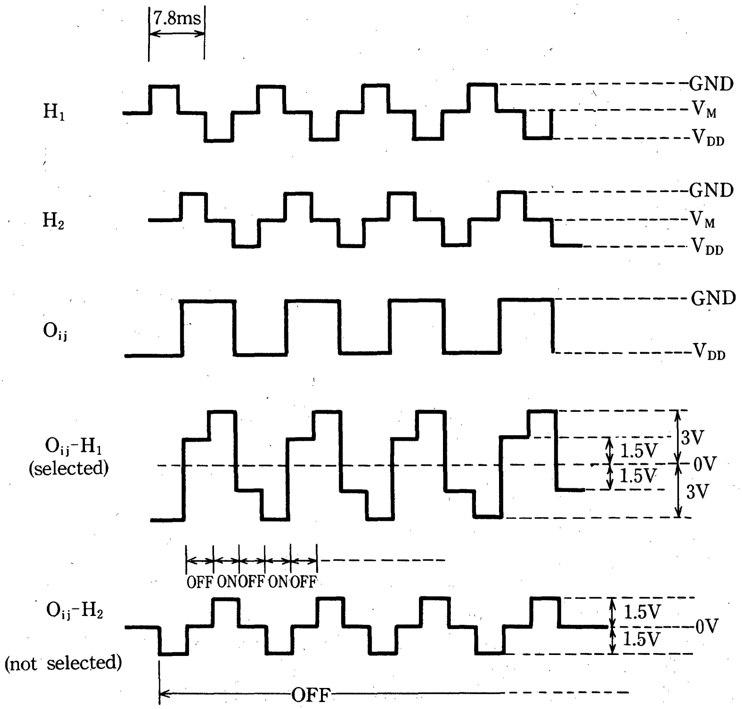


Fig. 3 LCD driving signal waveform

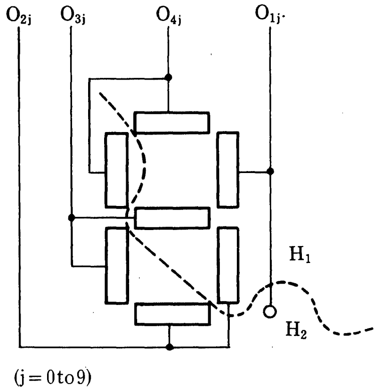


Fig. 4 7-segment numeric LCD digit

Table 1 Display decoder truth table

1/100 sec counter	Display character	1/100 sec counter	Display character
0000	0	0101	5
0001	1	0110	6
0010	2	0111	7
0011	3	1000	8
0100	4	1001	9

Note : The display segment of a floating point is specified by the first bit of display RAM (B_M, B_L)=(3, 0)

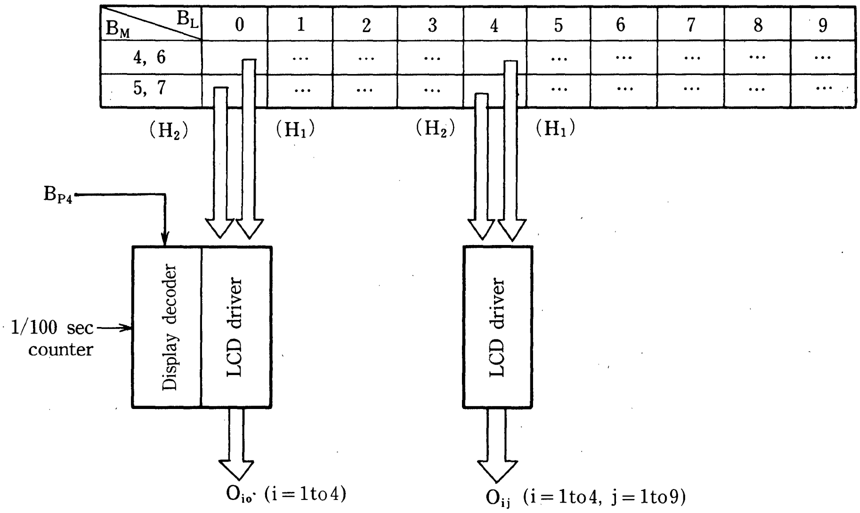


Fig. 5 Display RAM and Oij

(6) Display decoder

The 1/100 sec counter is loaded into the display decoder, and output as segment signals through pins O_{10}-O_{40}. The display decoder can not be used when the RAM data is displayed on an LCD.

Fig. 5 shows the relationship between the display RAM and pins O_{ij}. Table 1 shows the truth table of the display decoder.

(7) Melody generator circuit

The contents of a melody ROM can be output with standard 12 musical scales (555 to 2114Hz) in two octaves from the SO_1 and SO_2 pins. The tone length can be selected between 250ms and 125ms depending on the melody ROM. The melody ROM provides a pause and a stop instruction.

Controlling the melody F/F (ME F/F) by instructions starts and stops the melody. The melody ROM stores up to 128 steps of musical notes.

Table 2 shows the musical scales in one octave. Executing an instruction from a melody ROM outputs half frequencies of the standard frequencies shown in the table 2, and generates lower 12 musical scales by one octave. Executing an instruction allows an envelope control for melodies. The SO_1 output has an opposite phase with the SO_2 output.

Table 2 Melody output frequency

Musical scale	do#	re	re#	mi	fa	fa#	so	so#	la	la#	si	do
Frequency (Hz)	1110.8	1170.3	1236.5	1310.7	1394.4	1489.5	1560.4	1680.4	1771.2	1872.4	1985.9	2114.1

(8) ACL circuit

The ACL circuit contains a resistor and a capacitor, which does not require any external circuits. The ACL may by cleared with the interval of about 0.5 sec after a crystal oscillator circuit starts oscillation when the power is turned on, and starts execution of the program at $P_U = F_H$, $P_L = 00_H$.

(9) Standby mode

Executing an instruction places the device in standby mode. In this mode, the system clock is inactivated to reduce power consumption. While in standby mode, if the K or KE pin receives an input signal, or the selected γF/F is set, the device exits standby mode and the CPU starts execution of the program at $P_U = 0_H$, $P_L = 00_H$.

(10) Booster circuit

The device contains a booster circuit which generates a voltage two times higher than the 1.5V power supply.

Then, it is necessary to apply external capacitors between DDC pin and V_{CC} pin as well as V_{DD} pin and GND (see Fig. 6).

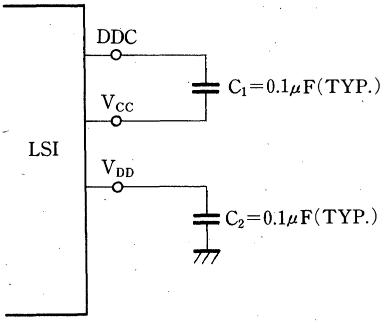


Fig. 6 Booster circuit

(11) System clock

The system clock has a frequency of one third that of a 32.768kHz clock.

The instruction cycle time should be 91.5μs.

■ Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
INCB	4C	$B_L \leftarrow B_L + 1$ skip if $B_L = 7$ or F
DECB	4D	$B_L \leftarrow B_L - 1$ skip if $B_L = 0$
LB_{xy}	30-3F	$B_{M2}, B_{M1} \leftarrow x\ (I_4, I_3)$ $B_{L4}, B_{L1} \leftarrow y\ (I_2, I_1)$ $B_{L3}, B_{L2} \leftarrow (1, 1)$
LBL xy	6B 00-FF	$B_M \leftarrow x\ (I_7\text{-}I_5)$ $B_L \leftarrow y\ (I_4\text{-}I_1)$
SABL	73	$B_{L4} \leftarrow 1$ next step only
EXBL	5A	$Acc \leftrightarrow B_L$

(2) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BE	$P_L \leftarrow x\ (I_6\text{-}I_1)$
TL xy	60-64 00-FE	$P_U \leftarrow x\ (I_{11}\text{-}I_7)$ $P_L \leftarrow y\ (I_6\text{-}I_1)$
TRS x	C0-FE	$P_U \leftarrow 01110$, $P_L \leftarrow x\ (I_6\text{-}I_1)$ $SR \leftarrow PC + 1$
JUMP xy	00-FF	$P_{U5}, P_{U4} \leftarrow (0, 0)$, $P_{U3}\text{-}P_{U1} \leftarrow x\ (I_6, I_8, I_7)$ $P_{L6} \leftarrow 0$ $P_{L5}\text{-}P_{L1} \leftarrow y\ (I_5\text{-}I_1)$
ATPL	6A	$P_{L4}\text{-}P_{L1} \leftarrow Acc$
RTN	68	$PC \leftarrow SR$
RTNS	69	$PC \leftarrow SR$ skip the next step

(3) Data transfer instructions

Mnemonic	Machine code	Operation
LAX x	10-1F	$Acc \leftarrow x\ (I_4\text{-}I_1)$
LDA x	20-23	$Acc \leftarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x\ (I_2, I_1)$
EXC x	24-27	$Acc \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x\ (I_2, I_1)$
EXCI x	28-2B	$Acc \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x\ (I_2, I_1)$ $B_L \leftarrow B_L + 1$ skip if $B_L = 7$ or F
EXCD x	2C-2F	$Acc \leftrightarrow M$ $B_{M2}, B_{M1} \leftarrow B_{M2}, B_{M1} \oplus x\ (I_2, I_1)$ $B_L \leftarrow B_L - 1$ skip if $B_L = 0$
DTA	52	Acc←1/100 SEC. C.

(4) Arithmetic instructions

Mnemonic	Machine code	Operation
ADD	54	$Acc \leftarrow Acc + M$
ADDC	55	$Acc \leftarrow Acc + M + C$ $C \leftarrow C_4$ skip if $C_4 = 1$
ADX x	00-0F	$Acc \leftarrow Acc + x\ (I_4\text{-}I_1)$ skip if $C_4 = 1$
COMA	53	$Acc \leftarrow \overline{Acc}$

(5) Test instructions

Mnemonic	Machine code	Operation
TM x	48-4B	skip if $Mi = 1\ (i = I_2 I_1)$
TC	5B	skip if $C = 0$
TAM	59	skip if $Acc = M$
TABL	58	skip if $Acc = B_L$
TG x	6C-6F	skip if $\gamma = 1$, $\gamma \leftarrow 0$ (γ_{1S}, $\gamma_{0.5S}$, $\gamma_{0.1S}$)

(6) Bit manipulation instructions

Mnemonic	Machine code	Operation
RM x	40-43	$Mi \leftarrow 0\ (i = I_2 I_1)$
SM x	44-47	$Mi \leftarrow 1\ (i = I_2 I_1)$
RC	56	$C \leftarrow 0$
SC	57	$C \leftarrow 1$

(7) I/O instructions

Mnemonic	Machine code	Operation
KTA	50	Acc←K
KETA	51	Acc←KE
ATBP	5E	BP←Acc
SDS	4F	DS←1
RDS	4E	DS←0

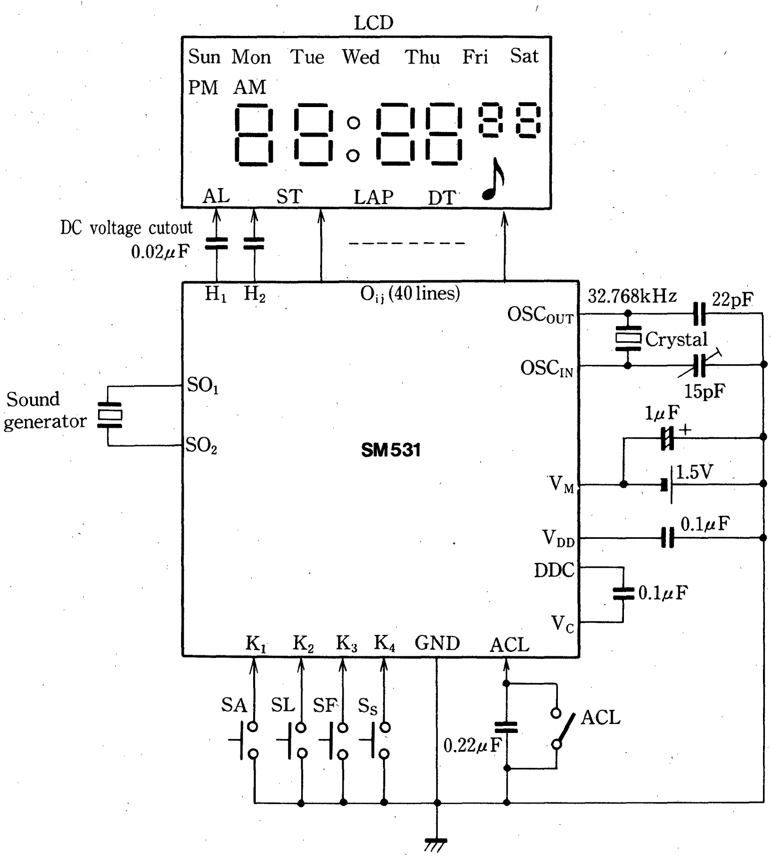
(8) Melody instructions

Mnemonic	Machine code	Operation
PRE x	78 00-FF	Melody ROM pointer preset
SME	77	ME←1
RME	76	ME←0
TMEL	75	skip if MES=1 MES←0

(9) Special instructions

Mnemonic	Machine code	Operation
CEND	74	System clock stop
IDIV	70	DIV←0
INIS	71	1/100 SEC. C.←0
SKIP	00	No operation

■ System Configuration Example (Melody alarm watch)



SM500 4-Bit Microcomputer (LCD Driver)

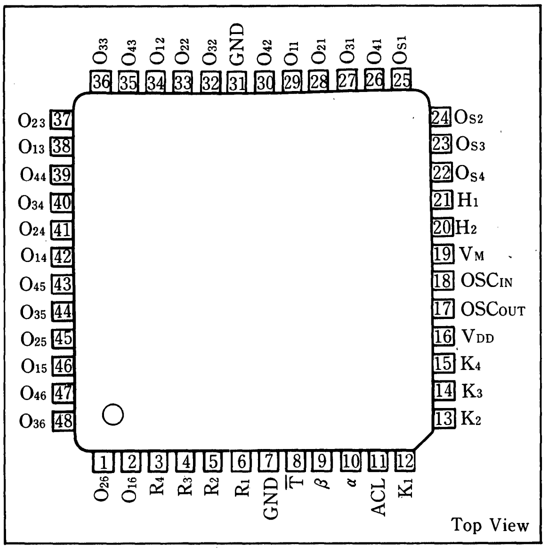
■ Description

The SM500 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a 1,197-byte ROM, a 40-word RAM, a 15-stage divider and a 56-segment LCD driver circuit in a single chip. This microcomputer is applicable to LCD systems with low power consumption and reduced cost.

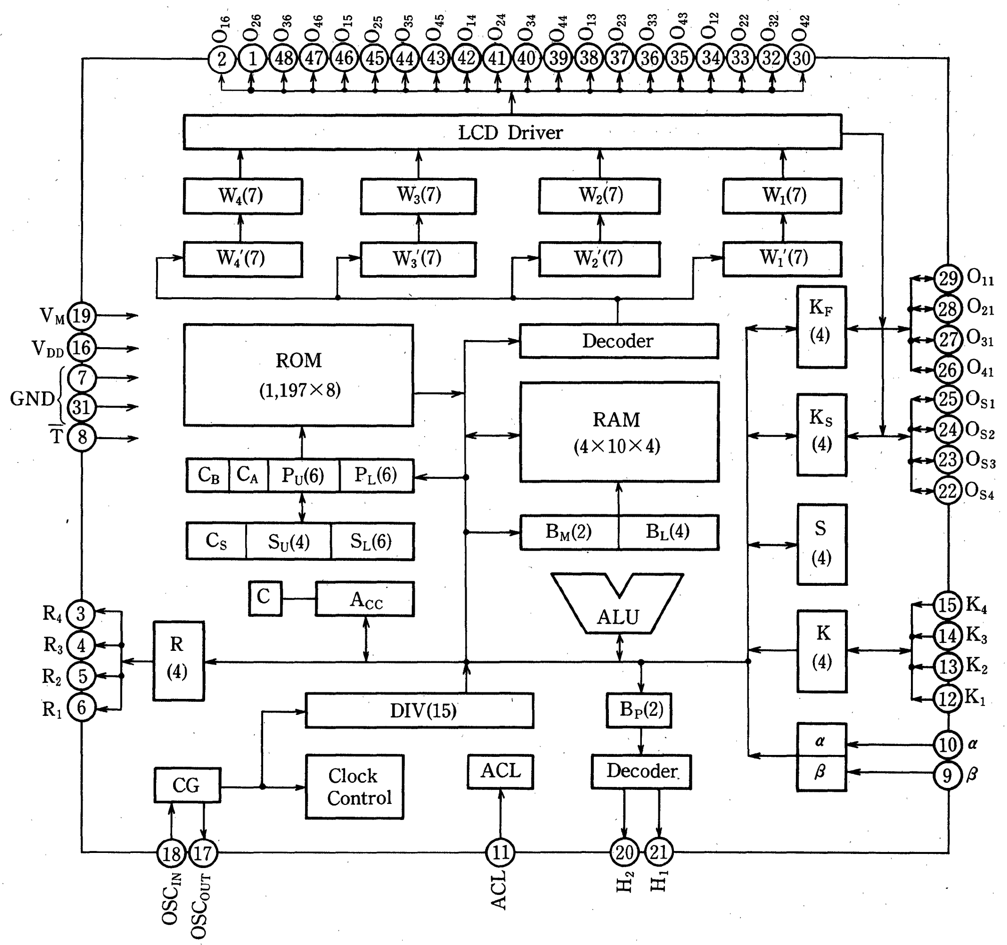
■ Features

1. CMOS process
2. ROM capacity: 1,197×8 bits
3. RAM capacity: 40×4 bits
4. Instruction set: 52
5. Subroutine nesting: 1 level
6. Instruction cycle: 61 µs (TYP.)
7. Input/output ports
   I/O ports: 8
   (for switching with segment pin)
   Input ports: 6
   Output ports: 4
   LCD output ports: 28 for segment
   (including 8 I/O ports)
   :2 for common
8. On-chip divider circuit for clock
9. On-chip crystal oscillator circuit
10. LCD driver circuit
    (56-segment, 1/2 bias, 1/2 duty)
11. Standby function
12. Single power supply: −3V or −5V (TYP.)
13. 48-pin QFP (QFP48-P-1010)

■ Pin Connections



Block Diagram



Symbol description

ALU	: Arithmetic logic unit	W_1-W_4, W_1'-W_4'	: Static shift register
Acc	: Accumulator	B_M, B_L	: RAM address register
ACL	: Auto clear	B_P	: Backplate signal generator circuit
C	: Carry F/F	K_F	: 4-bit F/F
C_A, C_B, P_U, P_L	: Program counter	K_S	: 4-bit F/F
C_S S_U, S_U	: Stack register of program counterr	S	: 4-bit F/F (status register)
CG	: Clock generator	K	: Key input F/F
DIV	: Frequency divider		

Pin Description

Symbol	I/O	Circuit type	Function
K_1-K_4	I	Pull down	Acc←K_1-K_4
α, β	I	Pull up	Independent test possible
O_{11}-O_{41}	I/O		W and W′ registers output or input/output to/from K_F register
O_{S1}-O_{S4}	I/O		W and W′ registers output or input/output to/from K_S register
O_{12}-O_{46}	O		W and W′ registers output ; used for LCD segment output
H_1, H_2	O		3-state level output possible ; used for LCD common output
R_1-R_4	O		R_1-R_4←Acc
$\overline{T}$	I	Pull up	For test (Connected to GND normally)
ACL	I	pull down	Auto clear
OSC_{IN}, OSC_{OUT}			For clock oscillation
V_M			Power supply for LCD driver
V_{DD}, GND			Power supply for logic circuit

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_{DD}	−6.0 to +0.3	V	1
	V_M	V_{DD} to +0.3	V	
	V_{IN}	V_{DD}−0.3 to +0.3	V	
	V_{OUT}	V_{DD}−0.3 to +0.3	V	
Operating temperature	Topr	−20 to +70	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.

Recommended Operating Conditions

(1) 3V power supply specification

(GND=0V)

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−2.7 to −3.3	V	
	V_M	V_{DD}/2 (TYP.)	V	
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz	
Oscillation start voltage	V_{OSC}	−2.7	V	1

(2) 5V power supply specification

(GND=0V)

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−4.5 to −5.5	V	
	V_M	V_{DD}/2 (TYP.)	V	
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz	
Oscillation start voltage	V_{OSC}	−4.5	V	1

Note 1: The oscillation start time should be within 10 sec.

Electrical Characteristics

(1) 3V power supply specification

($V_{DD}=-3.0V\pm10\%$, GND=0V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH}		−0.6			V	1
	V_{IL}				$V_{DD}+0.6$	V	
Input current	I_{H1}	$V_{IN}=0V$			15	μA	2
	I_{H2}	$V_{IN}=0V$			3	μA	3
	I_{L3}	$V_{IN}=V_{DD}$			1	μA	4
Output voltage	V_{OA}	No load $V_M=V_{DD}/2$	−0.3			V	5
	V_{OB}		$V_M-0.3$		$V_M+0.3$	V	
	V_{OC}				$V_{DD}+0.3$	V	
Output current	I_{OH1}	$V_{OUT}=-0.5V$	30			μA	6
	I_{OL1}	$V_{OUT}=V_{DD}+0.5V$	10			μA	
	I_{OH2}	$V_{OUT}=-0.5V$	100			μA	7
	I_{OL2}	$V_{OUT}=V_{DD}+0.5V$	10			μA	
	I_{O3}	$V_{DS}=0.3V$	100			μA	8
	I_{O4}	$V_{DS}=0.5V$	100			μA	9
Supply current	I_{DA}	During full-range operation		20		μA	10
	I_{DS}	When system clock is stationary		3		μA	

(2) 5V power supply specification

($V_{DD}=-5.0V\pm10\%$, GND=0V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH}		−0.6			V	1
	V_{IL}				$V_{DD}+0.6$	V	
Input current	I_{H1}	$V_{IN}=0V$			50	μA	2
	I_{H2}	$V_{IN}=0V$			10	μA	3
	I_{L3}	$V_{IN}=V_{DD}$			3	μA	4
Output voltage	V_{OA}	No load $V_M=V_{DD}/2$	−0.3			V	5
	V_{OB}		$V_M-0.3$		$V_M+0.4$	V	
	V_{OC}				$V_{DD}+0.4$	V	
Output current	I_{OH1}	$V_{OUT}=-0.5V$	35			μA	6
	I_{OL1}	$V_{OUT}=V_{DD}+0.5V$	12			μA	
	I_{OH2}	$V_{OUT}=-0.5V$	120			μA	7
	I_{OL2}	$V_{OUT}=V_{DD}+0.5V$	12			μA	
	I_{O3}	$V_{DS}=0.3V$	120			μA	8
	I_{O4}	$V_{DS}=0.5V$	120			μA	9
Supply current	I_{DA}	During full-range operation		50	100	μA	10
	I_{DS}	When system clock is stationary		10	30	μA	

Note 1: Applied to pins K_1-K_4, α, β, ACL, O_{11}, O_{21}, O_{31}, O_{41}, O_{S1}-O_{S4}
Note 2: Applied to pins K_1-K_4, O_{11}, O_{21}, O_{31}, O_{41}, O_{S1}-O_{S4}
Note 3: Applied to pin ACL
Note 4: Applied to pins α, β
Note 5: Applied to pins H_1, H_2
Note 6: Applied to pins O_{ij} (i=1 to 4, j=2 to 6)
Note 7: Applied to pins $O_{11}-O_{41}$, O_{S1}-O_{S4}
Note 8: Applied to pin R_1
Note 9: Applied to pins R_2, R_3, R_4
Note 10: $f_{OSC}=32.768kHz$, supply current with no load, oscillator circuit parameter: $C_D=C_G=22pF$

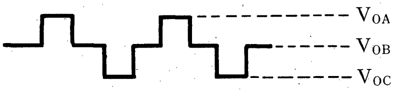
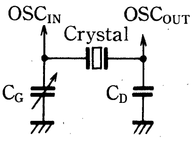


Fig. 1 H1, H2 waveforms



$C_G=15pF$, $C_D=22pF$

Fig. 2 Oscillator circuit

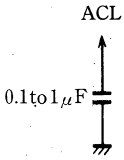


Fig. 3 ACL circuit

Pin Functions

(1) K_1-K_4 (Inputs)

The K_1-K_4 are 4-bit parallel input ports which are connected to the accumulator Acc. The contents of the K_1-K_4 are loaded into the A_{CC} by the KTA instruction.

When a system clock is inactivated, if a High level signal is input to any one bit of ports K_1-K_4, the system clock restarts, and the program counter starts at page 0, step 0.

(2) α, β (Inputs)

The input ports α and β can be independently tested by the TA and TB instructions respectively.

These ports are pulled-up to the High level within a chip.

(3) R_1-R_4 (Outputs)

The R_1-R_4 are 4-bit parallel output ports which generate the data stored in the R register.

The R register is connected to the accumulator A_{CC}. The contents of the Acc are loaded into the R register by the ATR instruction, which can be output at ports R_1-R_4.

The R_1 of the R register performs, in conjunction with the f_1, f_4 or f_{12} of a divider, the logical product. It can also provide an alarm output.

(4) H_2, H_1 (LCD common outputs)

The H_2 and H_1 pins are used to drive the common of an LCD with a 1/2 duty, 1/2 bias scheme, and provide a 3-level output.

The display can be turned on or off by the common outputs with the BP register.

(5) O_{ij} (Segment output ports)

The segment output ports O_{ij} ($i=1$ to 4, $j=2$ to 6) consist of 20 bits, which are used to output the contents of W′ and W registers for the display on or off with the BP register.

(6) O_{11}, O_{21}, O_{31}, O_{41} (Input/output ports)

The I/O ports O_{11}-O_{41} are used as segment output ports to generate the contents of W′ and W registers with the S register. The I/O ports can also be used as output ports as well as input ports for the K_F register. After ACL operation, it should be input ports with pull-down resistors.

(7) O_{S1}, O_{S2}, O_{S3}, O_{S4} (Input/output ports)

The I/O ports O_{S1}-O_{S4} are used as segment output ports to generate the contents of W′ and W registers with the S register. The I/O ports can also be used for output ports as well af input ports for the K_S register. After ACL operation, it should be input ports with pull-down resistor.

■ Hardware Configuration

(1) Program memory (ROM)

The on-chip ROM has 1,197 bytes organized as 19 pages×63 steps×8 bits. Fig. 1 shows the ROM configuration.

The program counter consists of a 1-bit C_A, C_B, a 4-bit page address counter P_U register and a 6-bit polynomial counter P_L (inhibit code: P_L= 111111).

The C_A is used to specify the field, the P_U for the page, P_L for the steps within a page and the C_B for the case where the field boundary is crossed.

(2) Data memory (RAM)

The data memory has 160 bits organized as 4×10×4 bits. Fig. 2 shows the RAM configuration.

The RAM address is specified by a 2-bit B_M register for the file specification, and a 4-bit B_L register for the word (4-bit) specification.

(3) Crystal oscillator and Divider (DIV)

The device contains a crystal oscillator circuit for the system clock and timer oscillator. A 16.384kHz system clock can be provided and 1 sec signal can be obtained from the final stage of a divider by connecting an external 32.768kHz crystal oscillator between the oscillator pins.

The divider consists of 15 stages, and lower 4 stages can be loaded into the accumulator by the DTA instruction. The lowest 9 stages (f_9-f_1) can be reset with the IDIV instruction or an ACL operation.

(4) Segment decoder

The SM500 contains an on-chip LCD driver which can directly drive an LCD with a 3V, 1/2 duty, 1/2 bias scheme. The device also contains a segment decoder which helps the software to be reduced.

The truth table of a segment decoder is shown in Fig. 5, the LCD segments relative to the decoder shown in Fig. 4, and the LCD driving signal waveform shown in Fig. 6. The display characters other than those described in Fig. 5 are available by directly setting data to W′ with the WR or WS instruction.

(5) Standby mode

The SM500 is a low power consumption design due to CMOS process. For further low power requirement, executing the CEND instruction places the device in standby mode. To reduce power consumption, the system clock is inactivated.

Field

Page ↓	← C_A=0 →	← C_A=1 →
	0	10
	1	11
	2	12
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	A	
	B	
	C	
	D	
	E	
	F	

Fig. 1 ROM configuration

→ File

Word ↓ B_L \ B_M	0	1	2	3
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

Fig. 2 RAM configuration

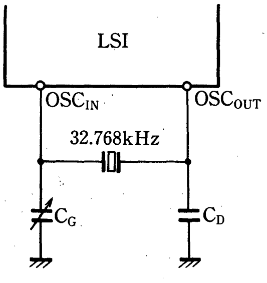


Fig. 3 Crystal oscillator circuit

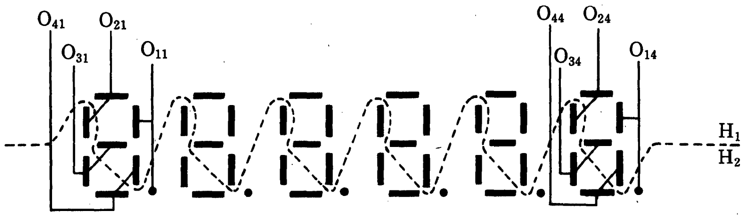


Fig. 4 LCD segment layout for segment decoders

While in standby mode, if more than one input of K_1-K_4 goes High, or γ F/F is reset, the device exits its standby mode and starts execution of the program at address 0000 ($C_A=0$, $P_U=0$, $P_L=0$).

(6) Reset function

Connecting a capacitor between the ACL pin and the GND activates the ACL circuit when it is powered up. The ACL is cleared in about 0.5 sec from a crystal oscillator circuit starts oscillation after power on, and starts execution of the program at $C_A=0$, $P_U=F_H$, $P_L=0$.

While in power on, applying a High level signal to the ACL pin activates the ACL operation. However, it takes about 0.5 sec to start execution of the program after the ACL goes Low. The lowest 9 stages of a divider are reset during the ACL goes High.

Acc	Display character	Acc	Display character
0	0	6	6
1	1	7	7
2	2	8	8
3	3	9	9
4	4	A	–
5	5	B	Blank

Fig. 5 Display decoder truth table

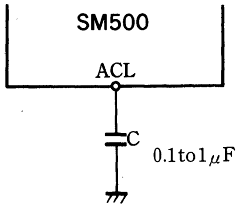


Fig. 7 ACL external circuit

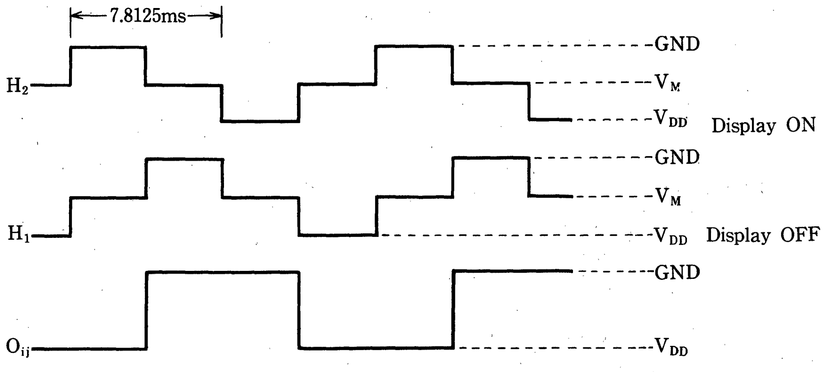


Fig. 6 LCD driving signal waveforms

Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
LB xy	40-4F	$B_L \leftarrow x\ (I_3, I_2)$, $B_M \leftarrow y\ (I_1, I_0)$
LBL xy (2 step)	5F 00-FF	$B_M \leftarrow x\ (I_5, I_4)$, $B_L \leftarrow y\ (I_3, I_0)$
EXBLA	0B	$Acc \leftrightarrow B_L$
INCB	64	$B_L \leftarrow B_L + 1$, Skip if $B_L = 7$
DECB	6C	$B_L \leftarrow B_L - 1$, Skip if $B_L = 0$

(2) ROM address instructions

Mnemonic	Machine code	Operation
COMCB	6D	$C_B \leftarrow \underline{C_B}$
RTN	6E	$C_A \leftarrow C_S$, $P_U \leftarrow S_U$, $P_L \leftarrow S_L$, $R \leftarrow 0$
RTNS	6F	$C_A \leftarrow C_S$, $P_U \leftarrow S_U$, $P_L \leftarrow S_L$ $R \leftarrow 0$, skip the next step
SSRx	70-7F	$S_U \leftarrow x\ (I_3\text{-}I_0)$, $E \leftarrow 1$ next step only

TRx	80-BE	if $R=0$;
		$P_L \leftarrow X\ (I_5\text{-}I_0)$, $P_U \leftarrow S_U$, $C_A \leftarrow C_B$
		if $R=1$; $P_L \leftarrow I_5\text{-}I_0$

TRSx	C0-FF	if $R=0$, $E=0$; $P_L \leftarrow x\ (I_5\text{-}I_0)$ $P_{U3} \leftarrow 1$, $P_{U2}\text{-}P_{U0} \leftarrow 0$, $S_L \leftarrow P_L + 1$, $S_U \leftarrow P_U$, $C_S \leftarrow C_A \leftarrow 0$, $R \leftarrow 1$ if $R=0$, $E=1$; $P_L \leftarrow x\ (I_5, I_0)$, $P_U \leftrightarrow S_U$, $S_L \leftarrow P_L + 1$, $C_S \leftarrow C_A \leftarrow C_B$, $R \leftarrow 1$
TRSAxy	C0-FF	if $R=1$; $P_{U1}, P_{U0} \leftarrow x\ (I_5, I_4)$ $P_{L3}\text{-}P_{L0} \leftarrow y\ (I_3\text{-}I_0)$, $P_{L5}, P_{L4} \leftarrow 0$

(3) Arithmetic instructions

Mnemonic	Machine code	Operation
ADD	08	$Acc \leftarrow Acc + M$
ADDC	09	$Acc \leftarrow Acc + M + C$, $C \leftarrow C_Y$, Skip if $C_Y = 1$
ADX x	31-3F	$Acc \leftarrow Acc + x\ (I_3\text{-}I_0)$ Skip if $C_Y = 1$ No skip if $I_3I_2I_1I_0 = 1010$ (30 defines inhibit)
COMA	0A	$Acc \leftarrow \overline{Acc}$

(4) Data transfer instruction

Mnemonic	Machine code	Operation
EXCx	10-13	$Acc \leftrightarrow M$, $B_M \leftarrow B_M \oplus x\ (I_1, I_0)$
EXCIx	14-17	$Acc \leftrightarrow M$, $B_M \leftarrow B_M \oplus x\ (I_1, I_0)$ $B_L \leftarrow B_L + 1$, Skip if $B_L = 7$
EXCDx	1C-1F	$Acc \leftrightarrow M$, $B_M \leftarrow B_M \oplus x\ (I_1, I_0)$ $B_L \leftarrow B_L - 1$, Skip if $B_L = 0$
LAXx	20-2F	$Acc \leftarrow x\ (I_3\text{-}I_0)$
LDAx	18-1B	$Acc \leftarrow M$, $B_M \oplus x\ (I_1, I_0)$
ATBP	03	$B_P \leftarrow Acc$
PTW	59	$W_{i6} \leftarrow W'_{i6}$, $W_{i5} \leftarrow W'_{i5}$ ($i=1$ to 4)
PDTW	61	$W'_{15} \leftarrow W'_{16} \leftarrow DECi$ ($i=1$ to 4)
TW	5C	$W_{ij} \leftarrow W'_{ij}$ ($i=1$, $j=0$ to 6)
DTW	5D	$W'_{i6} \leftarrow DEC_i$ W'_{ij} write shift ($i=1$ to 4, $j=0$ to 6)
WR	62	$W'_{46} \leftarrow 0$, $W'_{36} \leftarrow Acc_2$, $W'_{26} \leftarrow Acc_1$ $W'_{16} \leftarrow Acc_0$, W'_{ij} write shift ($i=1$ to 4, $j=0$ to 6)
WS	63	$W'_{46} \leftarrow 1$, $W'_{36} \leftarrow Acc_2$ $W'_{26} \leftarrow Acc_1$, $W'_{16} \leftarrow Acc_0$ W'_{ij} write shift ($i=1$ to 4, $j=0$ to 6)

(5) I/O control instructions

Mnemonic	Machine code	Operation
ATR	01	$R \leftarrow Acc$
KTA	6A	$Acc \leftarrow K$
ATS	30	$S \leftarrow Acc$

EXKSA	02	if $S_2 = 1$; $Acc \leftarrow K_S$
		if $S_2 = 0$; $K_S \leftarrow Acc$

EXKFA	6B	if $S_4 = 1$; $Acc \leftarrow K_F$
		if $S_4 = 0$; $K_F \leftarrow Acc$

(6) Divider manipulation instructions

Mnemonic	Machine code	Operation
DTA (2 step)	5E 04	$Acc_3 \leftarrow f_1$, $Acc_2 \leftarrow f_2$ $Acc_1 \leftarrow f_3$, $Acc_0 \leftarrow f_4$
IDIV	65	$f_9\text{-}f_1 \leftarrow 0$

(7) Bit manipulation instructions

Mnemonic	Machine code	Operation
RMx	04-07	$Mx \leftarrow 0$
SMx	0C-0F	$Mx \leftarrow 1$
RMF	68	$m' \leftarrow 0$, $Acc \leftarrow 0$
SMF	69	$m' \leftarrow 1$
COMCN	60	$C_N \leftarrow \overline{C_N}$
RC	66	$C \leftarrow 0$
SC	67	$C \leftarrow 1$

(8) Test instructions

Mnemonic	Machine code	Operation
TA	50	Skip if $\alpha=1$
TB	51	Skip if $\beta=1$
TC	52	Skip if C=0
TAM	53	Skip if Acc=M
TMx	54-57	Skip if Mx=1
TG	58	Skip if $\gamma=0$, $\gamma \leftarrow 0$
TA0	5A	Skip if Acc=0
TABL	5B	Skip if Acc=B_L

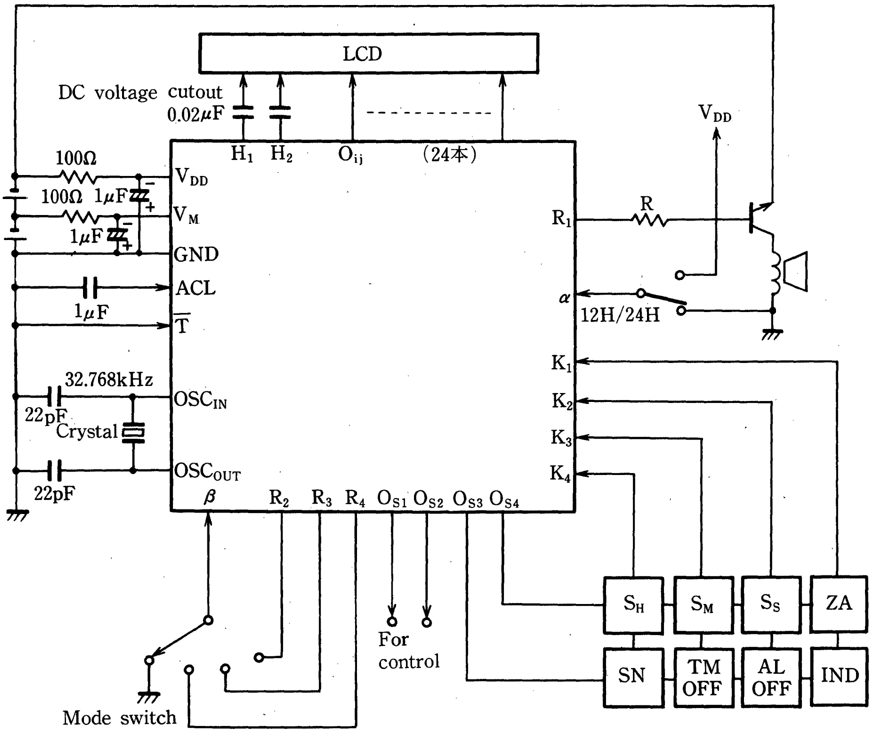
(9) Clock control instruction

Mnemonic	Machine code	Operation
GEND (2 step)	5E 00	clock stop

(10) Special instruction

Mnemonic	Machine code	Operation
SKIP	00	No operation

■ System Configuration Example (Digital watch)



SM5K1 4-Bit Microcomputer (LCD Driver)

■ Description

The SM5K1 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a 1,280×8-bit ROM, a 16×4-bit RAM, a 15-stage divider circuit in a single chip.

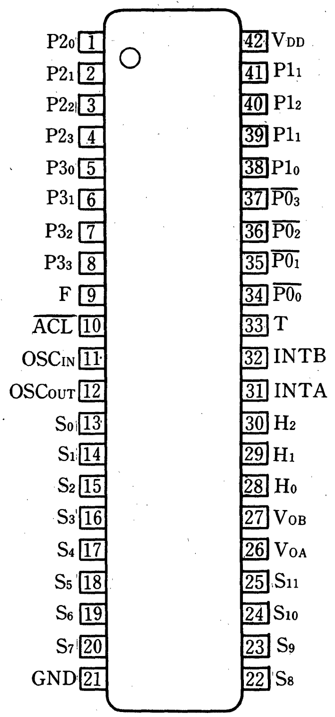
Provided with three kinds of interrupt, four levels of subroutine stack, 64-segment of LCD driver, two modes of standby, 4 bits of large current drive port (LED directly drive port), and 2 kinds of sound output functions, this microcomputer is applicable to battery back-up compact systems to home appliances such as an electronic microwave oven with a minimal external parts count and low power consumption.

■ Features

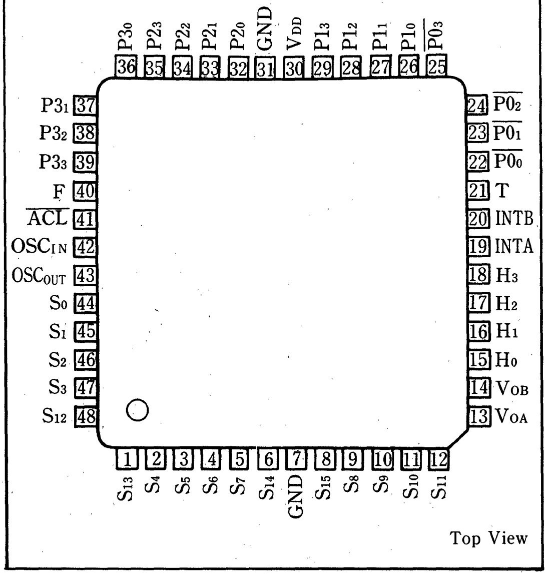
1. CMOS process
2. ROM capacity: 1,280×8 bits
3. RAM capactity
   Data RAM: 64×4 bits
   Display RAM: 16×4 bits
4. Instruction set: 51
5. Subroutine nesting: 4 levels
6. Instruction cycle: 5 to 61 μs
7. Interrupts
   External interrupts: 2
   Internal interrupt: 1
8. Input/output ports
   I/O ports: 8
   Input ports: 6
   Output ports: 5
   LCD output ports: 16 for segment
   4 for common
9. Built-in LCD driver circuit
   1/3 bias
   1/3 or 1/4 duty selectable
10. Sound (pulse) output
    2kHz or 2.5kHz
    (400kHz ceramic oscillator)
    2,048kHz or 4,096kHz
    (32.768kHz crystal oscillator)
11. LED direct drive ($\overline{P0_0}$-$\overline{P0_3}$)
    15mA (V_{DD}=5V±10%)
12. Built-in oscillator circuit
    Ceramic oscillator (400kHz)
    Crystal oscillator (32.768kHz)
13. 15-stage divider circuit
14. Single power supply: 2.4 to 5.5V
15. 42-pin SDIP (SDIP42-P-600)
    48-pin QFP (QFP 48-P-1010)

■ Pin Connections

42SDIP

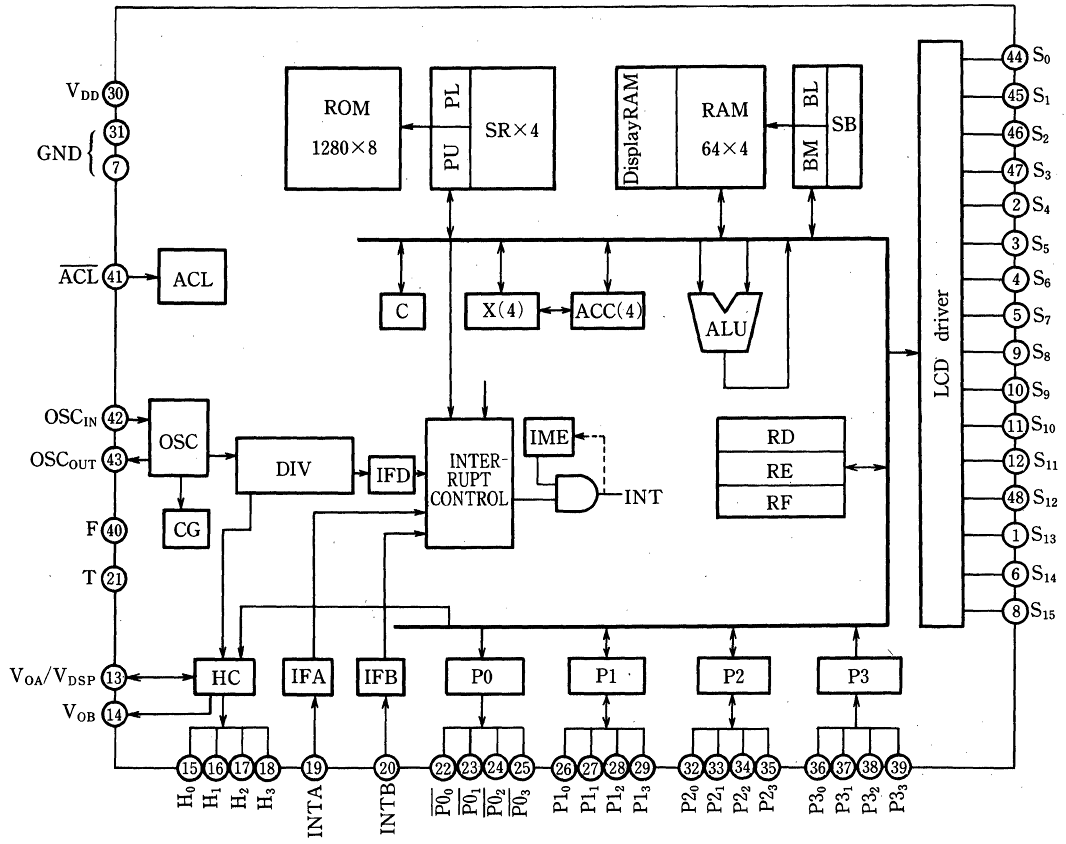


48QFP



Top View

Block Diagram



Acc	: Accumulator
ACL	: Auto clear
ALU	: Arithmetic logic unit
B_L, B_M	: RAM address register
C	: Carry flag
CG	: Clock generator
DIV	: Divider
HC	: Common signal circuit
IFA, IFB, IFD	: Interrupt request
IME	: Interrupt mask enable F/F
OSC	: Clock generator
P_U, P_L	: Program counter
RD, RE, RF	: Mode register
SB	: SB register
SR	: Stack register
X	: X register

Note: Pin numbers apply to a 48-pin QFP.

Pin Description

Symbol	Function	Symbol	Function
$\overline{P0}_0$-$\overline{P0}_3$	Output ports: 15mA Max. sink current at 5V±10%	F	Sound output port
$P1_0$-$P1_3$ $P2_0$-$P2_3$	Input/output ports	T	Test input port (normally connected to GND)
$P3_0$-$P3_3$	Input ports	OSC_{IN}, OSC_{OUT}	Ceramic or crystal oscillator
H_0-H_3	Common signal output ports	$\overline{ACL}$	Reset input port
S_0-S_{15}	Segment signal output ports	V_{DD}, GND	Power supply
INTA, INTB	External interrupt input ports	V_{OA}/V_{DSP}, V_{OB}	LCD drive power supply

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−0.3 to 6.5	V	
Input voltage	V_I	−0.3 to V_{DD}+0.3	V	
Output voltage	V_O	−0.3 to V_{DD}+0.3	V	
Source output current on each pin	I_{O1}	4	mA	1
	I_{O2}	4	mA	2
	I_{O3}	4	mA	3
	I_{O4}	2	mA	4
Sink output current on each pin	I_{O5}	30	mA	1
	I_{O6}	200	μA	2
	I_{O7}	4	mA	3
	I_{O8}	2	mA	4
Sum of source output current	ΣI_{OH}	20	mA	
Sum of sink output current	ΣI_{OL}	80	mA	
Operating temperature	Topr	−20 to 70	℃	
Storage temperature	Tstg	−55 to 150	℃	

Note 1: Applied to pins $\overline{P0}_i$ (i=3 to 0)
Note 2: Applied to pins $P1_i$, $P2_i$ (i=3 to 0)
Note 3: Applied to pin F
Note 4: Applied to pins H_0-H_3, S_0-S_{15}

Recommended Operating Conditions

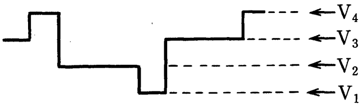
Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	2.4 to 5.5	V	
Instruction cycle time	t_{sys}	61 to 5	μs	
Operating temperature	Topr	−20 to 70	℃	

DC Characteristics

(V_{DD}=2.4V to 5.5V, Ta=−20℃ to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			$0.8V_{DD}$		V_{DD}	V	1
	V_{IL1}			0		$0.2V_{DD}$		
	V_{IH2}			$0.6V_{DD}$		V_{DD}	V	2
	V_{IL2}			0		0.5		
	V_{IH3}			$V_{DD}-0.5$		V_{DD}	V	3
	V_{IL3}			0		0.5		
Input current	I_{IH1}	$V_{IH}=V_{DD}$	V_{DD}=4.5V to 5.5V			1	μA	1
			V_{DD}=2.4V to 3.3V			2		
	I_{IH2}	$V_{IH}=V_{DD}$	V_{DD}=4.5V to 5.5V	8	25	70	μA	2
			V_{DD}=2.4V to 3.3V	1	7	20		
	I_{IH3}	$V_{IH}=V_{DD}$	V_{DD}=4.5V to 5.5V	20	55	150	μA	4
			V_{DD}=2.4V to 3.3V	2	15	50		
	I_{IL1}	$V_{IL}=0$	V_{DD}=4.5V to 5.5V	25	70	150	μA	5
			V_{DD}=2.4V to 3.3V	3	20	55		
Output voltage	V_4		V_{DD}=4.5V	4.2		4.5	V	6
	V_3			2.7	3.0	3.3		
	V_2			1.2	1.5	1.8		
	V_1			0		0.3		
	V_4		V_{DD}=3.0V	2.7		3.0	V	
	V_3			1.7	2.0	2.3		
	V_2			0.7	1.0	1.3		
	V_1			0		0.3		
Output current	$-I_{OH1}$	$V_{OH}=V_{DD}-0.5V$	V_{DD}=4.5V to 5.5V	1.0			mA	7
			V_{DD}=2.4V to 3.3V	0.3				
	I_{OL1}	$V_{OL}=1.0V$	V_{DD}=4.5V to 5.5V	15				
			V_{DD}=2.4V to 3.3V	4.5				
	$-I_{OH2}$	$V_{OH}=V_{DD}-0.5V$	V_{DD}=4.5V to 5.5V	1.0			mA	8
			V_{DD}=2.4V to 3.3V	0.3				
	I_{OL2}	$V_{OH}=0.5V$	V_{DD}=4.5V to 5.5V	1.0				
			V_{DD}=2.4V to 3.3V	0.3				
	$-I_{OH3}$	$V_{OH}=V_{DD}-0.5V$	V_{DD}=4.5V to 5.5V	1.0			mA	1
			V_{DD}=2.4V to 3.3V	0.3				
	I_{OL3}	$V_{OL}=0.5V$	V_{DD}=4.5V to 5.5V			100	μA	
			V_{DD}=2.4V to 3.3V			30		
	$-I_{OH4}$	$V_{OH}=V_{DD}-0.5V$	V_{DD}=4.5V to 5.5V	100			μA	9
			V_{DD}=2.4V to 3.3V	30				
	I_{OL4}	$V_{OL}=0.5V$	V_{DD}=4.5V to 5.5V	100				
			V_{DD}=2.4V to 3.3V	30				
Output impedance	D_{COM}		V_{DD}=4.5V to 5.5V		5	15	kΩ	10
			V_{DD}=2.4V to 3.3V		15	60		
	D_S		V_{DD}=4.5V to 5.5V		10	40		11
			V_{DD}=2.4V to 3.3V		30	120		
Current cosumption (Operating)	I_{DA}	$T_{SYS}=5.0\mu s$	V_{DD}=4.5V to 5.5V		350	850	μA	12
			V_{DD}=2.4V to 3.3V		150	250		
		$T_{SYS}=61.0\mu s$	V_{DD}=4.5V to 5.5V		220	650		
			V_{DD}=2.4V to 3.3V		60	150		
Current consumption (Hold mode)	I_{DH1}	Ceramic oscillation $T_{SYS}=5.0\mu s$	V_{DD}=4.5V to 5.5V		35	70	μA	13
			V_{DD}=2.4V to 3.3V		20	40		
		Crystal oscillation $T_{SYS}=61.0\mu s$	V_{DD}=4.5V to 5.5V		20	45		
			V_{DD}=2.4V to 3.3V		10	20		
	I_{DH2}	Ceramic oscillation $T_{SYS}=5.0\mu s$	V_{DD}=4.5V to 5.5V		25	60	μA	14
			V_{DD}=2.4V to 3.3V		15	35		
		Crystal oscillation $T_{SYS}=61.0\mu s$	V_{DD}=4.5V to 5.5V		8.0	15		
			V_{DD}=2.4V to 3.3V		4.5	8.0		
Current consumption (Stop mode)	I_{DS}	Ceramic oscillation	V_{DD}=2.4V to 3.3V			1.0	μA	15
		Crystal oscillation $T_{SYS}=61.0\mu s$	V_{DD}=4.5V to 5.5V		3.5	7.0		16
			V_{DD}=2.4V to 3.3V		1.5	3.0		

Note 1: Applied to pins $P1_i$, $P2_i$ (i=0 to 3)
Note 2: Applied to pins $P3_i$ (i=0 to 3)
Note 3: Applied to pins OSC_{IN}, T, INTA, INTB, and $\overline{ACL}$
Note 4: Applied to pins T, INTA and INTB
Note 5: Applied to pin $\overline{ACL}$
Note 6: Applied to pins H_0-H_3, S_0-S_{15}



LCD terminal output waveform example

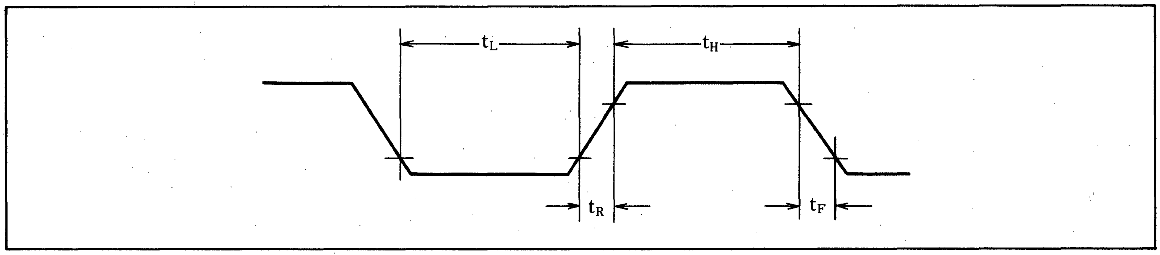
Note 7: Applied to pins $\overline{PO}_i$ (i=0 to 3)
Note 8: Applied to pin OSC_{OUT}
Note 9: Applied to pin F
Note 10: Applied to pins H_0-H_3
Note 11: Applied to pins S_0-S_{15}
Note 12: No load condition. Current consumption under the operation with an external clock input. LCD should be turned on.
Note 13: No load condition. Current consumption when driving an oscillator and turning LCD ON placed the device in hold mode.
Note 14: No load condition. Current consumption when driving an oscillator and turning an LCD bleeder resistor OFF placed the device in hold mode.
Note 15: No load condition. Current consumption when the entire system including ceramic oscillation is inactivated.
Note 16: No load condition. Current consumption when the entire system except for crystal oscillator is inactivated.

External clock Input characteristics

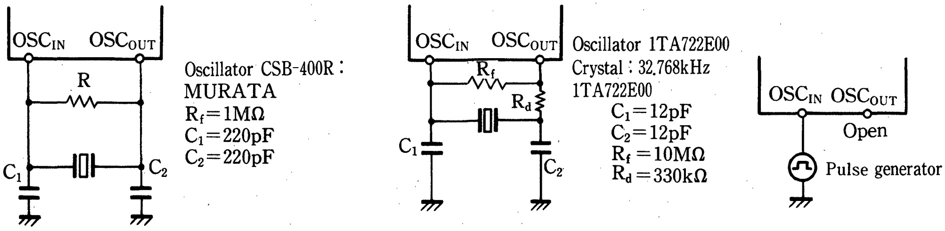
(V_{DD}=2.4 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input rise time	t_M				50	ns
Input fall time	t_F				50	ns
Clock pulse width	t_L		1.20		30.47	μs
	t_H		1.20		30.47	

External Input Clock Timing



Oscillator Circuit



(a) 400kHz clock

(b) 32.768kHz clock

(c) External clock input circuit

Note: The resistors, capacitors and crystal oscillators should be located as close to the LSI chip as possible to minimize influence of stray capacitance.

■ Pin Functions

(1) GND, V_{DD} (Power supply)

The GND pin should be grounded.

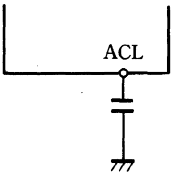
The V_{DD} pin is the power supply input which should be positive with respect to GND.

(2) T (Test input)

The test pin must be grounded and should not be used. It is connected to GND with a pull-down resistor.

(3) ACL (Reset input)

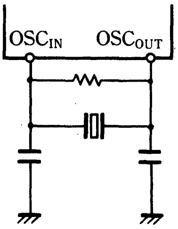
The ACL accepts an active-Low level which initializes the internal logic of the device. Normally a capacitor is connected between this pin and GND to provide a power-on reset function.



(4) OSC_{IN}, OSC_{OUT} (Crystal or ceramic oscillators)

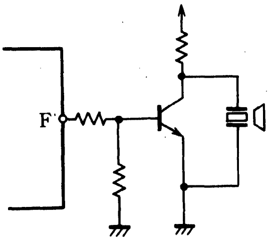
The OSC_{IN} and OSC_{OUT} pins connect with an external crystal or ceramic oscillator, in conjunction with an on-chip oscillator circuit, constitute a real-time clock.

Either a crystal or ceramic oscillator is selectable with a mask option.



(5) F (Sound output)

The pin F serves exclusively as a sound output pin which can be selected between 2kHz and 2.5kHz at the base frequency of a 400kHz ceramic oscillator, 2,048kHz and 4,096kHz at a 32.768 kHz crystal oscillator.



(6) H_0-H_3 (Common drive outputs)

The H_0-H_3 pins are used to drive the common output of an LCD.

(7) S_0-S_{15} (Segment drive outputs)

The S_0-S_{15} pins are used to drive LCD segments.

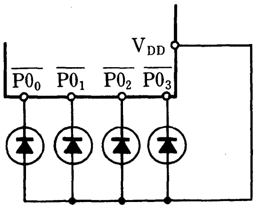
(8) INTA, INTB (External interrupt inputs)

The IFA flag is set at the rising edge of INTA input pin, and the IFB flag is set at the falling edge of INTB input pin.

Note: Both INTA and INTB pins are connected to the noise debounce circuit which does not accept the pulse shorter than two instruction cycles.

(9) $\overline{P0}_0$-$\overline{P0}_3$ (Output ports)

The $\overline{P0}$ port, output pins $\overline{P0}_0$-$\overline{P0}_3$ are used to directly drive an LED with a maximum of 15mA of sink current (V_{DD}=5V±10%).



(10) $P1_0$-$P1_3$, $P2_0$-$P2_3$ (I/O ports)

The P1 and P2 are I/O ports which can be switched between Input and Output modes through an instruction at a 4-bit unit.

These ports can also be used as output ports for a keymatrix.

(11) $P3_0$-$P3_3$ (Input ports)

The P3 port, input pins P_{30}-P_{33} are connected to the positive supply with pull-down resistors, which can be used for a keymatrix.

■ Hardware Construction

(1) Program counter and stack register

The program counter (PC) is used to specify the ROM address.

The PC consists of 12 bits including a 6-bit page address count register (P_U) and a 6-bit binary counter (P_L) which addresses steps within each page.

The stack register (SR) consists of 4 stages which provides up to 4 levels of subroutine nesting.

(2) Program memory (ROM)

The SM5K1 has 1,280 steps of on-chip ROM organized as 20 pages×64 steps.

When the ACL resets the device (power-on), it starts execution of the program at page 0, step 0. Fig. 1 shows the jump address with a ROM address instruction.

A jump within a page is executed by a TR instruction, and a jump out of a page is executed by a TL instruction.

A subroutine jump is executed by a CALL or TRS instruction.

(3) Data memory (RAM) and B register

The RAM consists of a 256-bit data RAM organized as 4×16×4 bits and a 64-bit display RAM organized as 8×2×4 bits.

Fig. 2 shows the RAM configuration with 6 files of architecture.

The B register consists of a 4-bit BM which address files and a 4-bit BL which address words. 2×8×4 bits of RAM space, (4×16+2×8)×4 bits, is used as a display RAM area from which data is output to LCD segment driving pins. An LCD with a 1/4 or 1/3 duty and 1/3 bias format can be directly driven by writing display data into the display RAM area.

Fig. 3 shows the relationship between the display RAM and LCD segments.

B_L / B_M	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
8	▨	▨	▨	▨	▨	▨	▨	▨								
9	▨	▨	▨	▨	▨	▨	▨	▨								

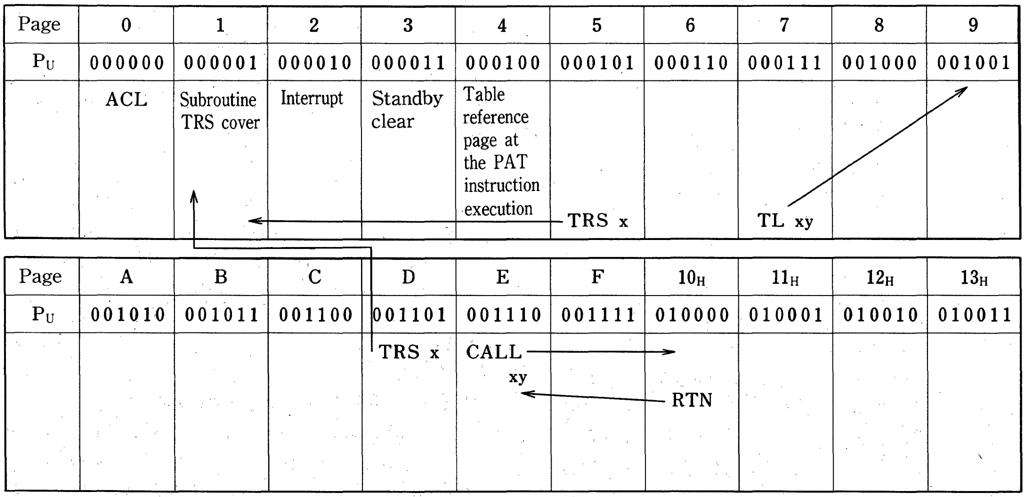
The shadowed area is allocated for a display RAM

Note: The file can be specified as long as the BM should be 0, 1, 2, 3, 8 or 9.

Fig. 2 RAM configuration

(4) Accumulator (Acc), X register, arithmetic and logic unit (ALU)

The accumulator (Acc) is a 4-bit general-purpose register which transfers data and numerics to memory, I/O ports, and registers. The Acc performs arithmetic operations in conjuction with a RAM, a carry flag and an ALU.



Page	0	1	2	3	4	5	6	7	8	9
P_U	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001
	ACL	Subroutine TRS cover	Interrupt	Standby clear	Table reference page at the PAT instruction execution	TRS x		TL xy		

Page	A	B	C	D	E	F	10_H	11_H	12_H	13_H
P_U	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011
				TRS x	CALL xy		RTN			

Fig. 1 ROM configuration

The X register is a 4-bit register used as a temporary register which transfers and compares data with the Acc. The ROM data can be loaded into the X register and Acc using a table reference instruction.

The arithmetic and logic unit (ALU) performs binary addition, in conjunction with a RAM, a carry flag and an Acc.

(5) SB register

The SB register is an 8-bit register which can be used as a save register.

(6) Output latch register and mode register

Ports P0, P1 and P2 connect with output latch registers, and transfer the contents of the Acc to the output latch registers with an instruction.

The SM5K1 has mode registers RD, RE and RF for controlling an LCD and interrupt functions.

(7) System clock generator circuit, divider

The OSC_{IN} and OSC_{OUT} provide a system clock f_S with the base frequency divided by two. One cycle of the system clock is identical to the instruction cycle time.

The divider consists of 15 stages. The lower 8-stage is reset with an instruction, and the lowest 4-stage is transferred to the Acc by a DTA instruction.

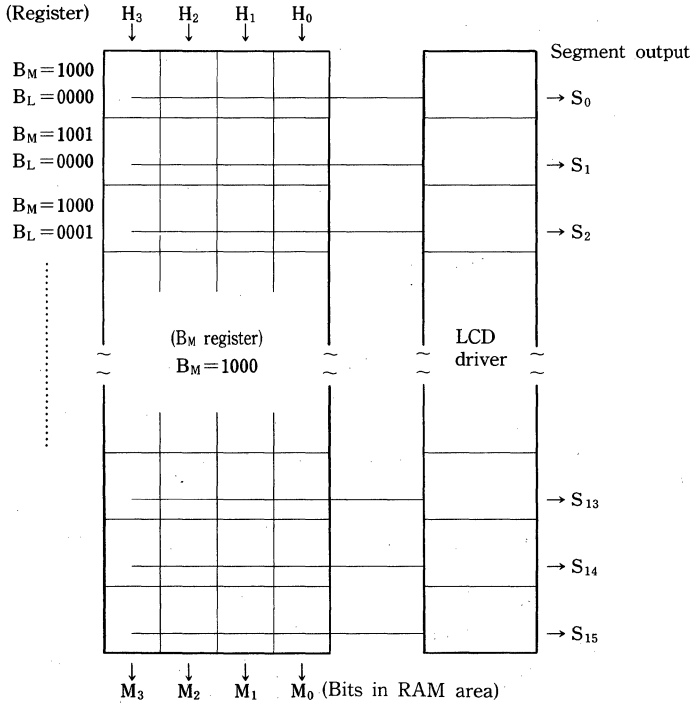
The oscillator can be selected between the ceramic and crystal with a mask option.

The least stage of a divider f_C can be selected between 2Hz and 1Hz under crystal oscillation with a mask option (See Fig. 4).

(8) Sound output

The frequency obtained by a system clock generator circuit can be output from the F pin as a sound pulse.

Setting the RD register outputs and stops the sound pulse, and switches the frequency. The frequency can be selected between 2kHz and 2.5kHz at 400kHz of a ceramic oscillator, while 2,048kHz and 4,096kHz at 32.768kHz of a crystal oscillator.



*The common outputs cannot be used when applied to an LCD with a 1/3 duty, 1/3 bias scheme.

Fig. 3 Display RAM and LCD segment outputs.

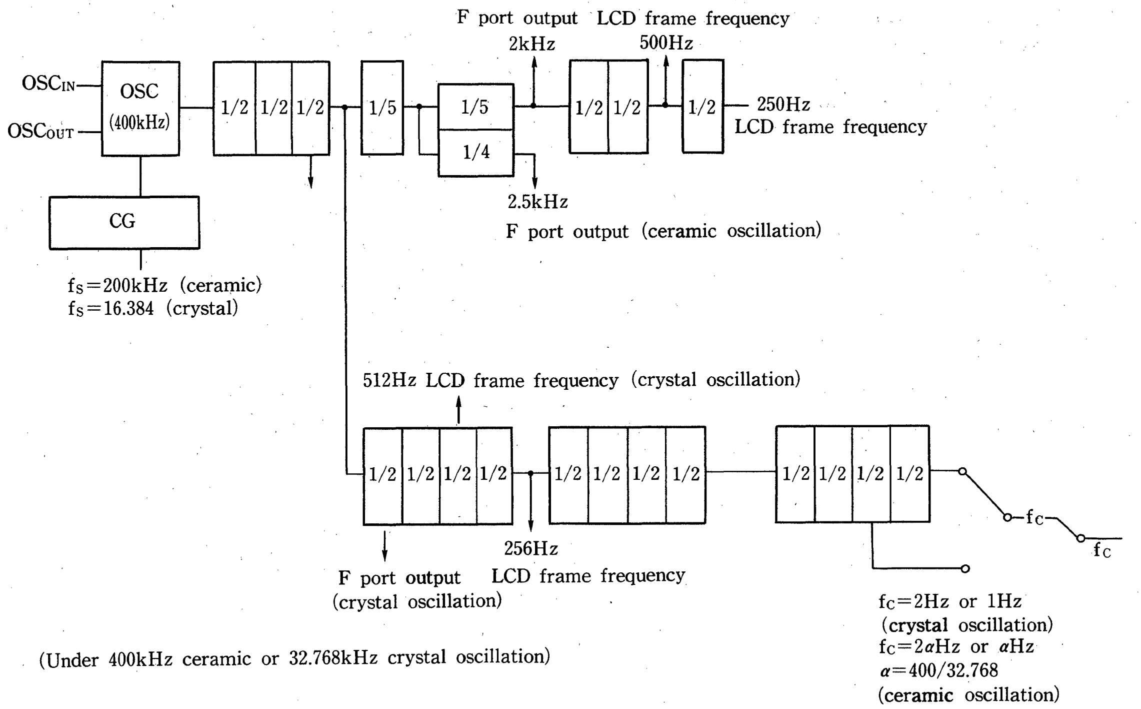


Fig. 4 System clock generator circuit (fc=1Hz or 0.5Hz)

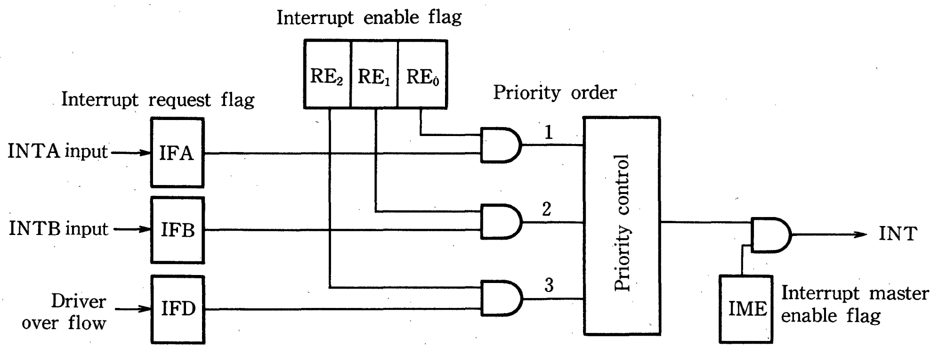


Fig. 5 Interrupt handling

(9) Interrupts

The INTA, INTB inputs and the divider overflow flag can be used for the interrupt request. The IFA, IFB and IFD flags can be used as the interrupt request flag.

The interrupt block consists of mask flags (RE_0, RE_1, RE_2), an IME flag, and an interrupt processing circuit.

(10) Standby mode

To reduce power consumption, the device is placed in standby mode, and the program execution is inactivated.

- Stop mode

In the stop mode, the entire system clock is inactivated under ceramic oscillation, however, only a reference clock is operative under crystal oscillation.

- Hold mode

Only a system clock generator circuit (CG circuit) is inactivated, while the OSC and DIV circuit is in operative (see block diagram).

While in standby mode, if any one bit of P3 input port goes High during ACL, or an interrupt occurs from unmasked INTA, INTB or a divider, the device exits standby mode and starts program execution.

(11) Reset function (ACL)

Applying a Low level signal to the $\overline{\text{ACL}}$ pin resets the internal logic of the device and applying a High level signal starts execution of the program at address 0, page 0.

Once the device is reset, all I/O ports are placed in input mode, and the mode registers RD, RE and RF are cleared. The output port P0 is cleared to output a High level signal.

The interrupt enable flags IFA, IFB and IFD, and the interrupt master enable flag IME are reset to disable all interrupts.

In case the noise may harm the ACL operation, apply a capacitor between $\overline{\text{ACL}}$ pin and V_{DD} pin (see Fig. 6).

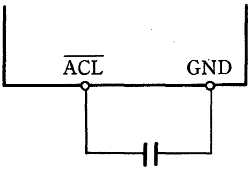


Fig. 6 ACL circuit

(12) LCD driver

- Display segment

The SM5K1 contains an on-chip LCD driver which can directly drive an LCD with a 1/4 duty and 1/3 bias as well as 1/3 duty and 1/3 bias scheme.

Fig. 7 shows an example of LCD segment configuration for 1/4 duty.

Each segment of the LCD can be turned on or off by software control of the setting of the corresponding bit "1" or "0" in the display RAM area (see Fig. 3).

The LCD digit may have any shape, provided that the maximum number of segments does not exceed 64 (see Fig. 7).

Fig. 8 shows an example of a seven-segment numeric LCD digit.

- LCD driving signal waveform

Fig. 9 shows the LCD signal driving waveforms required to display the number "5" on the 7-segment display for 1/4 duty shown in Fig. 8 (a).

Fig. 10 shows the LCD signal driving waveforms required to display the number "2" on the 7-segment display for 1/3 duty shown in Fig. 8 (a).

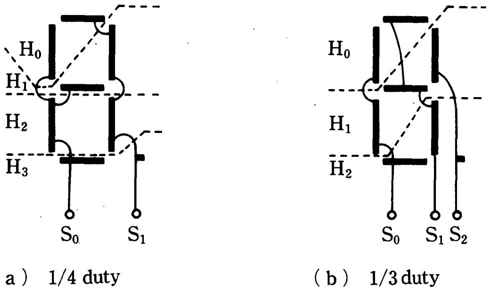


Fig. 8 7-segment numeric LCD digit

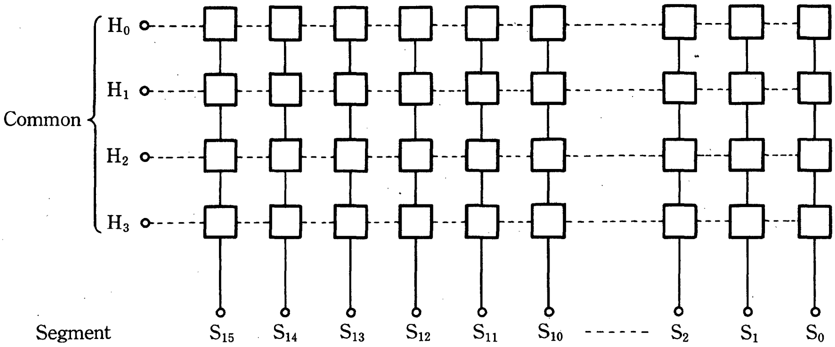
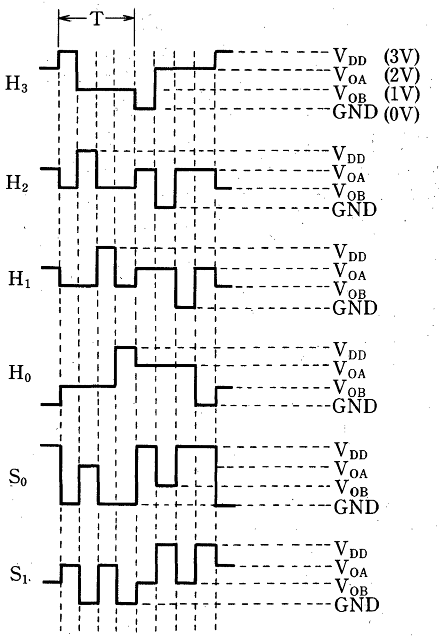
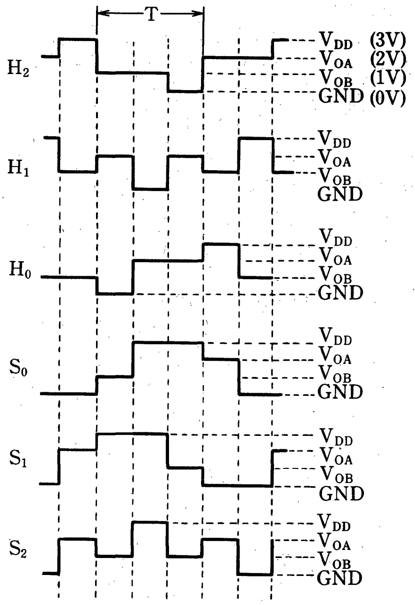


Fig. 7 LCD configuration for 1/4 duty



1/4 duty
Frame frequency=1/T=62.5Hz or 125Hz

Fig. 9 LCD driving signal waveform



1/3 duty
Frame frequency=1/T=83.3Hz or 166.7Hz

Fig. 10 LCD driving signal waveform

- V_{OA} and V_{OB} pins

The device contains bleeder resistors to allow 1/3 bias driving. When V_{DD} is 3V, voltages of 2V and 1V are applied to pins V_{OA} and V_{OB} respectively.

Normally pins V_{OA} and V_{OB} are left open. When an LCD with a large display area is driven, connect capacitors across pins V_{OA} and V_{DD} and across V_{OB} and V_{DD} to improve the rise time of the LCD driving signal.

■ Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BF	$P_L \leftarrow x$ (I_5-I_0)
TL xy	E0-E4 00-FF	$P_U \leftarrow x$ (I_{11}-I_6) $P_L \leftarrow y$ (I_5-I_0)
TRS x	C0-DF	Push, $P_U \leftarrow$01H, $P_L \leftarrow x$ (I_4 I_3 I_2 I_1 I_0 0)
CALL xy	F0-F4 00-FF	Push, $P_U \leftarrow x$ (I_{11}-I_6) $P_L \leftarrow y$ (I_5-I_0)
RTN	7D	Pop
RTNS	7E	Pop, Skip the next step
RTNI	7F	Pop, IME←1

(2) Data transfer instructions

Mnemonic	Machine code	Operation
LAX x	10-1F	Acc←x (I_3-I_0)
LBMX x	30-3F	$B_M \leftarrow x$ (I_3-I_0)
LBLX x	20-2F	$B_L \leftarrow x$ (I_3-I_0)
LDA x	50-53	Acc←M, $B_{Mi} \leftarrow B_{Mi} \oplus \times(I_1, I_0)$, (i=1, 0)
EXC x	54-57	M←Acc, $B_{Mi} \leftarrow B_{Mi} \oplus \times(I_1, I_0)$, (i=1, 0)
EXCI x	58-5B	M↔Acc, $B_L \leftarrow B_L + 1$ $B_{Mi} \leftarrow B_{Mi} \oplus \times(I_1, I_0)$, (i=1, 0) Skip if $B_L = F_H$
EXCD x	5C-5F	M↔Acc, $B_L \leftarrow B_L - 1$ $B_{Mi} \leftarrow B_{Mi} \oplus \times(I_1, I_0)$, (i=1, 0) Skip if $B_L = 0$
EXAX	64	Acc↔X
ATX	65	X←Acc
EXBM	66	$B_M \leftrightarrow$ Acc
EXBL	67	$B_L \leftrightarrow$ Acc
EX	68	B↔SB

(3) Arithmetic instructions

Mnemonic	Machine code	Operation
ADX x	00-0F	Acc←Acc+x (I_3-I_0) Skip if CY=1
ADD	7A	Acc←Acc+M
ADC	7B	Acc←Acc+M+C, C←CY Skip if CY=1
COMA	79	Acc←Acc
INCB	78	$B_L \leftarrow B_L + 1$, Skip if $B_L = F_H$
DECB	7C	$B_L \leftarrow B_L - 1$, Skip if $B_L = 0$

(4) Test instructions

Mnemonic	Machine code	Operation
TAM	6F	Skip if Acc=M
TC x	6E	Skip if C=1
TM	48-4B	Skip if M_i=1, (i=3 to 0)
TABL	6B	Skip if Acc=B_L
TPB x	4C-4F	Skip if P $(R)_i$=1, (i=I_1, I_0)
TA	6C	Skip if IFA=1 IFA←0
TB	6D	Skip if IFB=1 IFB←0
TD	69 02	Skip if IFD=1 IFD←0

(5) Bit manipulation instructions

Mnemonic	Machine code	Operation
SM x	44-47	M_i←1 (i=3 to 0)
RM x	40-43	M_i←0 (i=3 to 0)
SC	61	C←1
RC	60	C←0
IE	63	IME←1
ID	62	IME←0

(6) I/O instructions

Mnemonic	Machine code	Operation
INL	70	Acc←$P1_i$ (i=3 to 0)
OUTL	71	$P0_i$←Acc (i=3 to 0)
ANP	72	Pj←Pj ∧ Acc (j=2 to 0)
ORP	73	Pj←Pj ∨ Acc (j=2 to 0)
IN	74	Acc←Pj (j=3 to 1)
OUT	75	Pj←Acc (j=2 to 0), Pj←Acc (j=F_H-D_H)

(7) Table reference instructions

Mnemonic	Machine code	Operation
PAT	6A	Push P_U←04H, PL←(X_1, X_0, Acc) X←ROM H, Acc←ROM L Pop

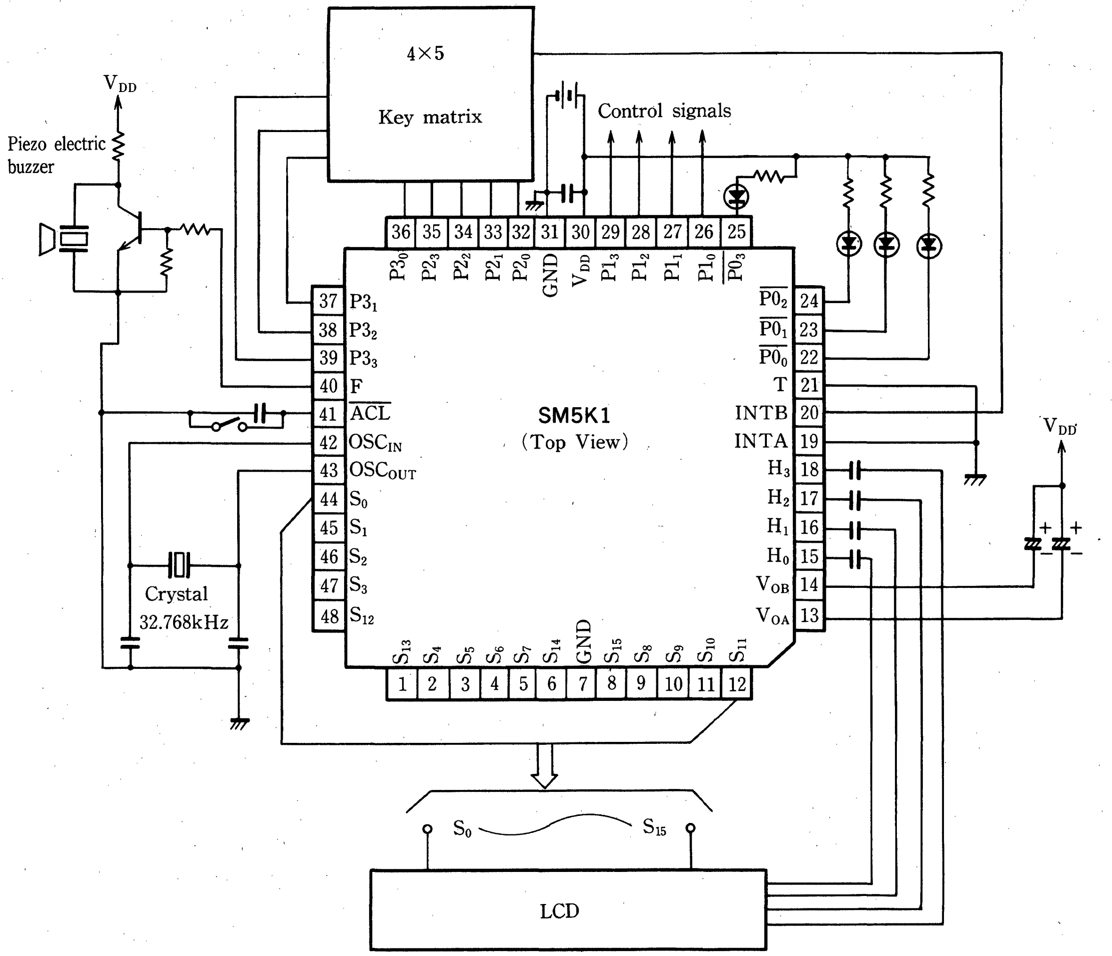
(8) Divider operation instructions

Mnemonic	Machine code	Operation
DR	69 03	DIV (f_7-f_0) Reset
DTA	69 04	Acc←DIV (f_3-f_0)

(9) Special instructions

Mnemonic	Machine code	Operation
STOP	76	Standby mode (STOP)
HALT	77	Standby mode (HALT)
NOP	00	No operation

System Configuration Example (Audio-timer)



SM4A 4-Bit Microcomputer (LCD Driver)

■ Description

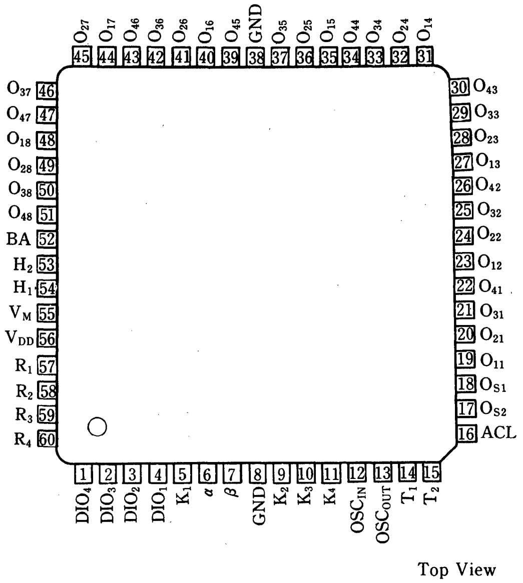
The SM4A is a CMOS 4-bit microcomputer which integrates a 4-bit parallel procesing function, a 2,268-byte ROM, a 96-word RAM, a 15-stage divider, and a 68-segment LCD driver circuit in a single chip.

This microcomputer is applicable to the system having multiple LCD segment, with low power consumption.

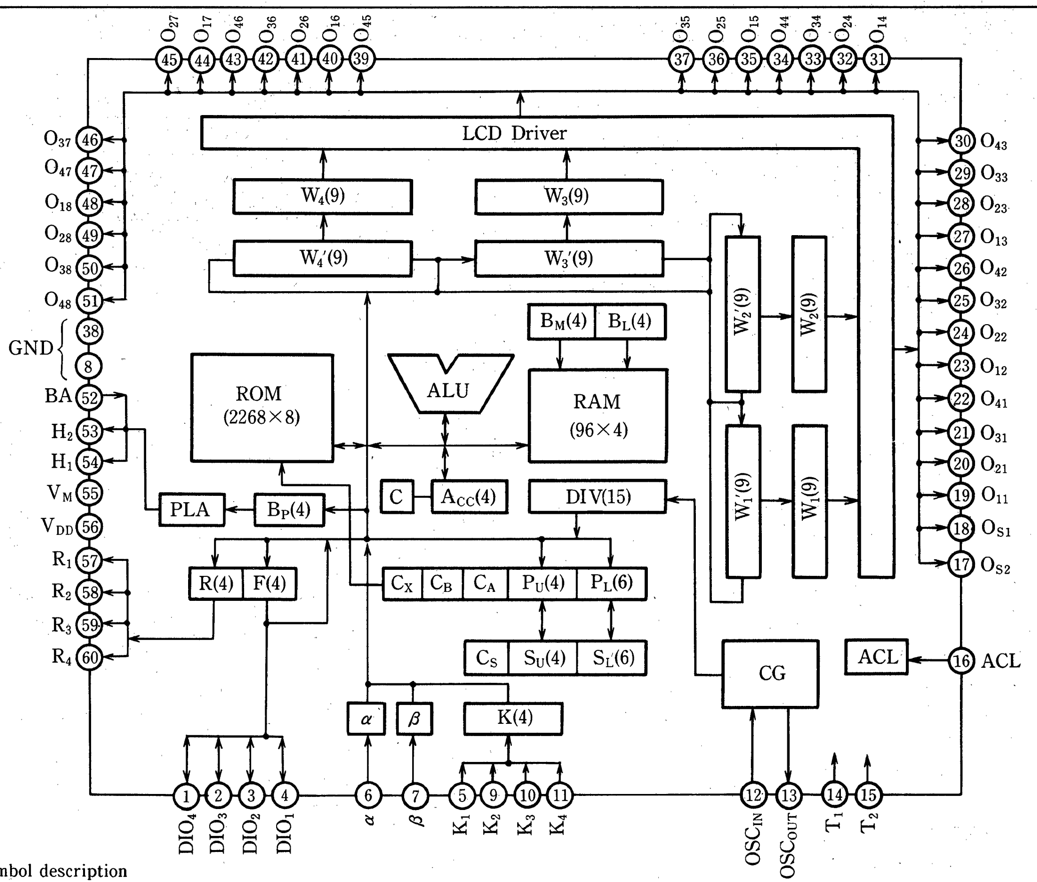
■ Features

1. CMOS process
2. ROM capacity: 2,268×8 bits
3. RAM capacity: 96×4 bits
4. Instruction set: 54
5. Subroutine nesting: 1 level
6. Instruction cycle: 61 μs (TYP.)
7. Input/output ports
   I/O ports: 4
   Input ports: 6
   Output ports: 4
   LCD output ports: 34 for segment
   2 for common
8. On-chip clock divider
9. On-chip crystal oscillator
10. External RAM access
11. LCD driver circuit
    (68-segment, 1/2 bias, 1/2 duty)
12. Standby function
13. Single power supply: −3V (TYP.)
14. 60-pin QFP (QFP60-P-1414)

■ Pin Connections



Block Diagram



Symbol description

ALU	: Arithmetic logic unit	B_M, B_L	: RAM address register
Acc	: Accumulator	DIV	: Frequency divider
ACL	: Auto clear	PLA	: Programmable logic array
C	: Carry F/F	CG	: Clock generator
C_X, C_A, P_U, P_L	: Program counter	W_1-W_4, W'_1-W'_4	: Static shift register
C_S, S_U, S_L	: Stack register of program counter		

Pin Description

Symbol	I/O	Circuit type	Function
K_1-K_4	I	Pull down	Acc←K_1-K_4
α	I	Pull down	Set by ↑, reset after test instruction execution
β	I	Pull down	Input signal is held for 1 instruction cycle, test possible
DIO_1-DIO_4	I/O	3-state output	Acc←→DIO_1-DIO_4
R_1-R_4	O	Complementary	R_1-R_4←Acc
O_{11}-O_{48} OS_1,OS_2	O		W and W′ registers output: used for LCD segment output
H_1, H_2	O		3-state level output possible, used for LCD common output
BA	I	Pull up	For test the input signal of High or Low
T_1, T_2	I		For test (Connected to V_{DD} normally)
ACL	I		Auto clear
OSC_{IN}, OSC_{OUT}			For clock oscillation
V_M			Power supply for LCD driver
GND, V_{DD}			Power supply for logic circuit

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_{DD}	-3.5 to $+0.3$	V	1
	V_M	-3.5 to $+0.3$	V	
	V_{IN}	$V_{DD}-0.3$ to $+0.3$	V	
Operating temperature	Topr	-5 to $+55$	℃	
Storage temperature	Tstg	-55 to $+150$	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	-3.2 to -2.6	V
	V_M	$V_{DD}/2$ (TYP.)	V
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz

Electrical Characteristics

($V_{DD}=-3.2$ to -2.6V, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		-0.6			V	1
	V_{IL1}				$V_{DD}+0.6$	V	
	V_{IH2}		-0.3			V	2
	V_{IL2}				$V_{DD}+0.3$	V	
Output voltage	V_{OH1}	$I_{OUT}=50\mu$A to V_{DD}	-0.5			V	3
	V_{OL1}	$I_{OUT}=5\mu$A to GND			$V_{DD}+0.5$	V	
	V_{OH2}	$I_{OUT}=50\mu$A to V_{DD}	-0.5			V	4
	V_{OL2}	$I_{OUT}=30\mu$A to GND			$V_{DD}+0.5$	V	
	V_{OH3}	$I_{OUT}=50\mu$A to V_{DD}	-0.5			V	5
	V_{OL3}	$I_{OUT}=50\mu$A to GND			$V_{DD}+0.5$	V	
	V_{OA}	No load	-0.3			V	6
	V_{OB}	$V_{DD}=-3.0$V		-1.5		V	
	V_{OC}	$V_M=-1.5$V			-2.7	V	
Output current	I_{SO}	$V_{OUT}=-0.2$V	100			μA	7
	I_{SIN}	$V_{OUT}=V_{DD}+0.2$V	100			μA	
Supply current	I_{DA}	During full-range operation		50	100	μA	
	I_{DS}	When system clock is stationary		10	20	μA	

Note 1: Applied to pins K_1, K_2, K_3, K_4, α, β
Note 2: Applied to pin ACL
Note 3: Applied to pins O_{48}-O_{11}, O_{S1}, O_{S2}
Note 4: Applied to pins DIO_1-DIO_4
Note 5: Applied to pins R_2, R_3, R_4
Note 6: Applied to pins H_1, H_2
Note 7: Applied to pin R_1

Pin Functions

(1) K_1-K_4 (Inputs)

The input ports K_1-K_4 are connected to the accumulator Acc. The contents of the K_1-K_4 are loaded into the Acc.

(2) α, β (Inputs)

The input ports α and β can be independently tested. The α input latches the α F/F at the rising edge of the input, and can be tested by the TA instruction. The α F/F is reset after the test. The β is used to put the input signal into the β F/F for the interval of one instruction, and can be tested by the TB instruction.

(3) DIO_1-DIO_4 (I/O ports)

The DIO_1-DIO_4 pins normally output the contents of the F_1-F_4 F/F. The F_1-F_4 F/F data can be changed on transferring the accumulator Acc by the ATF instruction. Connecting the DIO_1-DIO_4 with the Acc allows the data transfer between the Acc and an external RAM by the READ and WRITE instructions. The output buffer of the F_1-F_4 F/F is designed to be a three-state output, and it is kept high impedance when the DIO input is loaded into the Acc by the READ instruction.

(4) R_1-R_4 (Outputs)

Connecting the DIO_1-DIO_4 with the Acc outputs the contents of the Acc. And selecting the programmable logic array PLA generates a sound output, and allows a segment output on pins O_{S3} and O_{S4}.

(5) O_{ij} (i=1 to 4, j=1 to 8), O_{S1}, O_{S2} (Outputs)

34-bits of output ports O_{ij}, O_{S1} and O_{S2} are used to output the contents of the static shift register W'_{in}, W_{in} (i=1 to 4, n=0 to 8). The output signal can be used as a segment signal for a 1/2 duty scheme, and a strobe signal for the key-scan, according to the display mode. These ports output the address of the external RAM upon execution of the READ or WRITE instruction.

(6) H_1, H_2 (Output)

The H_1 and H_2 are used to output the common signal of an LCD with 1/2 bias, 1/2 duty scheme in a three output level inculding V_{DD}, GND and V_M.

(7) BA (Inputs)

The BA pin is used to test the input level of High or Low by instructions.

Hardware Cnfiguration

(1) Program memory (ROM)

The on-chip ROM has a 2,268 byte organized as 36 pages×63 steps×8 bits. The program counter consists of 1-bit registers C_X and C_A, a 4-bit register P_U, and a 6-bit polynomial counter P_L. The P_L is used to specify the steps, the P_U specify the pages, and the C_A specify the fields. The C_X register is only used to specify the subroutine pages.

→Field

Page↓

$C_X=0$		$C_X=1$
$C_A=0$	$C_A=1$	—
0	16	32
1	17	33
2	18	34
3	19	35
4	20	
5	21	
6	22	
7	23	
8	24	
9	25	
10	26	
11	27	
12	28	
13	29	
14	30	
15	31	

Fig. 1 ROM configuration (fields and pages)

(2) Data memory (RAM)

Data memory has a 6×16 word×4-bit configuration, and is addressed by a 4-bit B_L and a 4-bit B_M.

(3) Oscillator circuit

An on-chip crystal oscillator allows the oscillation with the external circuit shown in Fig. 3.

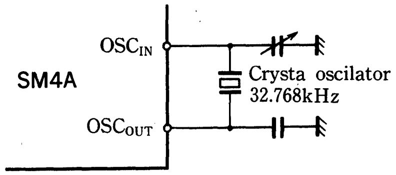


Fig. 3

(4) Divider

A 15-stage resettable divider outputs a 1 Hz signal at the lowest stage when a 32.768kHz crystal oscillator is used. The output on each stage can be loaded into the accumulator Acc on an 4-bit basis.

(5) Reset function (ACL)

An on-chip reset circuit may sometimes require a capacitor between the ACL pin and GND pin. It takes 1 sec on an internal timer from the beginning of oscillation to clear the ACL mode when power on.

BM_3		0				1	
BM_2		0	0	1	1	—	
BM_1		0	1	0	1	0	1
B_L	0						
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	10						
	11						
	12						
	13						
	14						
	15						
		X	Y	Z	M	U	T

Fig. 2 RAM configuration

Instruction Set

Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1	
SBM	02	$1 \rightarrow B_{M3}$ ($B_{M3}=1$ for next step only)
LB	40-4F	$I_4, I_3 \rightarrow B_{L2}, B_{L1}$ I_2, I_1-B_{M2}, B_{M1}
LBL	5F	I_8-$I_5 \rightarrow B_{M4}$-B_{M1} I_4-$I_1 \rightarrow B_{L4}$-B_{L1}
	00-FF	
INCB	64	$B_L+1 \rightarrow B_L$ if $B_L=a$; skip
DECB	6C	$B_L-1 \rightarrow B_L$ if $B_L=b$; skip
RM	04-07	$0 \rightarrow Mi$ ($i=I_2\ I_1$)
SM	0C-0F	$1 \rightarrow Mi$ ($i=I_2\ I_1$)
ATPL	03	$Acc \rightarrow P_{L4}$-P_{L1}
ADD	08	$Acc+M \rightarrow Acc$
ADD11	09	$Acc+M+C \rightarrow Acc$ $C_4 \rightarrow C$ if $C_4=1$; skip
COMA	0A	$\overline{Acc} \rightarrow Acc$
EXBLA	0B	$Acc \leftrightarrow B_L$
EXC	10-13	$Acc \leftrightarrow M$ $B_{M2}, B_{M1} \oplus I_2, I_1 \rightarrow B_{M2}, B_{M1}$
EXCI	14-17	$Acc \rightarrow M$ $B_{M2}, B_{M1} \oplus I_2, I_1 \rightarrow B_{M2}, B_{M1}$ $B_L+1 \rightarrow B_L$ if $B_L=a$; skip
EXCD	1C-1F	$Acc \rightarrow M$ $B_{M2}, B_{M1} \oplus I_2, I_1 \rightarrow B_{M2}, B_{M1}$ $B_L-1 \rightarrow B_L$ if $B_L=b$; skip
LDA	18-1B	$M \rightarrow Acc$ $B_{M2}, B_{M1} \oplus I_2, I_1 \rightarrow B_{M2}, B_{M1}$
LAX	20-2F	I_4-$I_1 \rightarrow Acc$
ADX	30-3F	I_4-$I_1+Acc \rightarrow Acc$ if $C_4=1$; skip
DC	3A	$10+Acc \rightarrow Acc$
DTA	5E	$DIV \rightarrow Acc$
	04-07	
ROT	6B	$C \rightarrow A_4 \rightarrow A_3 \rightarrow A_2 \rightarrow A_1 \rightarrow C$
ATBP	01	$Acc \rightarrow Bp$
ATW	5D	$Acc \rightarrow W'_{i8}$ ($i=1$ to 4) W'in Right Shift ($i=1$ to 4, $n=7$ to 0)
PATW	00	$Acc \rightarrow W'_{i8}$ $W'_{i8} \rightarrow W'_{i7}$ ($i=1$ to 4)
ATF	60	$Acc \rightarrow F$
ATR	61	$Acc \rightarrow R$
READ	68	$DIO \rightarrow Acc$
WRITE	69	$Acc \rightarrow DIO$
KTA	6A	$K_i \rightarrow Acc$
RC	66	$0 \rightarrow C$
SC	67	$1 \rightarrow C$

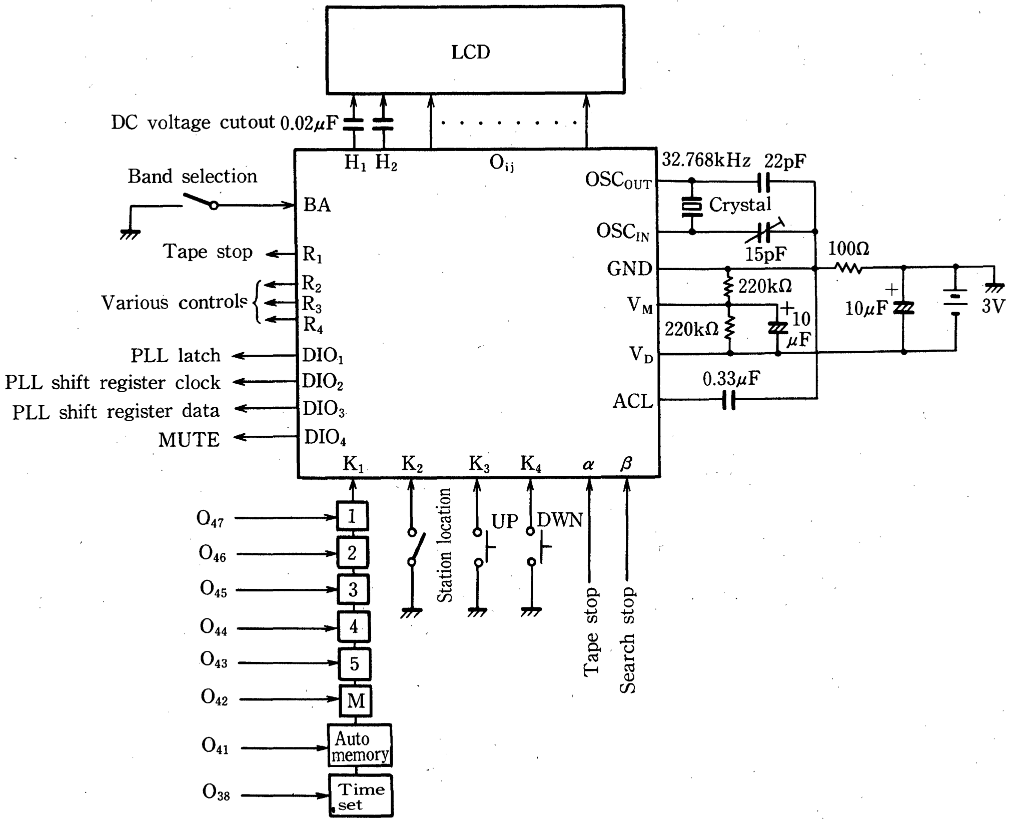
Mnemonic	Machine code	Operation
	I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1	
TW	5C	$W'_{in} \rightarrow W_{in}$ (i=1 to 4, n=8 to 0)
PTW	59	$W'_{in} \rightarrow W_{in}$ (i=1 to 4, n=8,7)
WR	62	$0 \rightarrow W'_{48}$ W_{in} Right Shift
WS	63	$1 \rightarrow W'_{48}$ W_{in} Right Shift
IDIV	65	0→DIV
TA	50	if $\alpha = 1$; skip
TB	51	if $\beta = 1$; skip
TC	52	if C=0; skip
TAM	53	if Acc=M; skip
TM	54-57	if Mi=1 ($i = I_2 I_1$); skip
TA0	5A	if Acc=0; skip
TABL	5B	if Acc=B_L; skip
TIS	58	if 1S=0; skip
② TAL	5E	if BA=1; skip
	02	
② CEND	5E	Clock stop
	00	
② ST	5E	1→T
	03	
COMCB	6D	$\overline{C_B} \rightarrow C_B$
SSR	70-7F	I_4-$I_1 \rightarrow S_{U4}$-S_{U1} 1→E (next step only)

TR0	80-BF	if R=0; I_6-$I_1 \rightarrow P_{L6}$-P_{L1} $S_U \rightarrow P_U$ $C_B \rightarrow C_A$
		if R=1; I_6-$I_1 \rightarrow P_{L6}$-P_{L1}

TR1	C0-FF	if R=0, E=0; $\left(\begin{array}{l} I_6\text{-}I_1 \rightarrow P_{L6}\text{-}P_{L1} \quad 0 \rightarrow P_U \rightarrow S_U \quad P_L+1 \rightarrow S_L \\ 1 \rightarrow R \quad 1 \rightarrow C_A \rightarrow C_S \quad 1 \rightarrow D \end{array}\right.$
		if R=0, E=1; $\left(\begin{array}{l} I_6\text{-}I_1 \rightarrow P_{L6}\text{-}P_{L1} \quad P_U \leftrightarrow S_U \quad P_L+1 \rightarrow S_L \\ C_B \rightarrow C_A \rightarrow C_S \quad 1 \rightarrow R \end{array}\right.$
		if R=1; I_6, $I_5 \rightarrow P_{U2}$, P_{U1} I_4-$I_1 \rightarrow P_{L4}$-P_{L1}

RTN0	6E	$C_S \rightarrow C_A$ $S_U \rightarrow P_U$ $S_L \rightarrow P_L$ 0→R
RTN1	6F	$C_S \rightarrow C_A$ $S_U \rightarrow P_U$ $S_L \rightarrow P_L$ 0→R skip next step
JUMP	00-FF	if D=1 I_8-$I_6 \rightarrow P_{U4}$, P_{U3}, P_{U1} I_5-$I_1 \rightarrow P_{L5}$-P_{L1}

System Configuration Example (Radio PLL controller)



SM510 4-Bit Microcomputer (LCD Driver)

■ Description

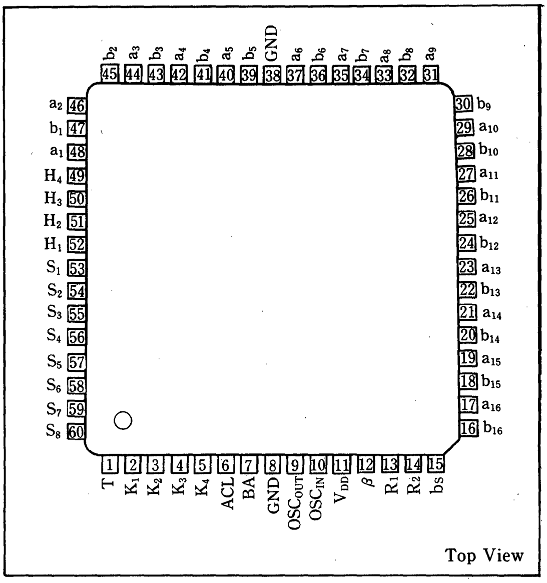
The SM510 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a 2,772×8-bit ROM, a 128×4-bit RAM, a 15 stage divider and a 132-segment LCD driver circuit.

This microcomputer is applicable to many applications having multiple LCD segments with low power consumption.

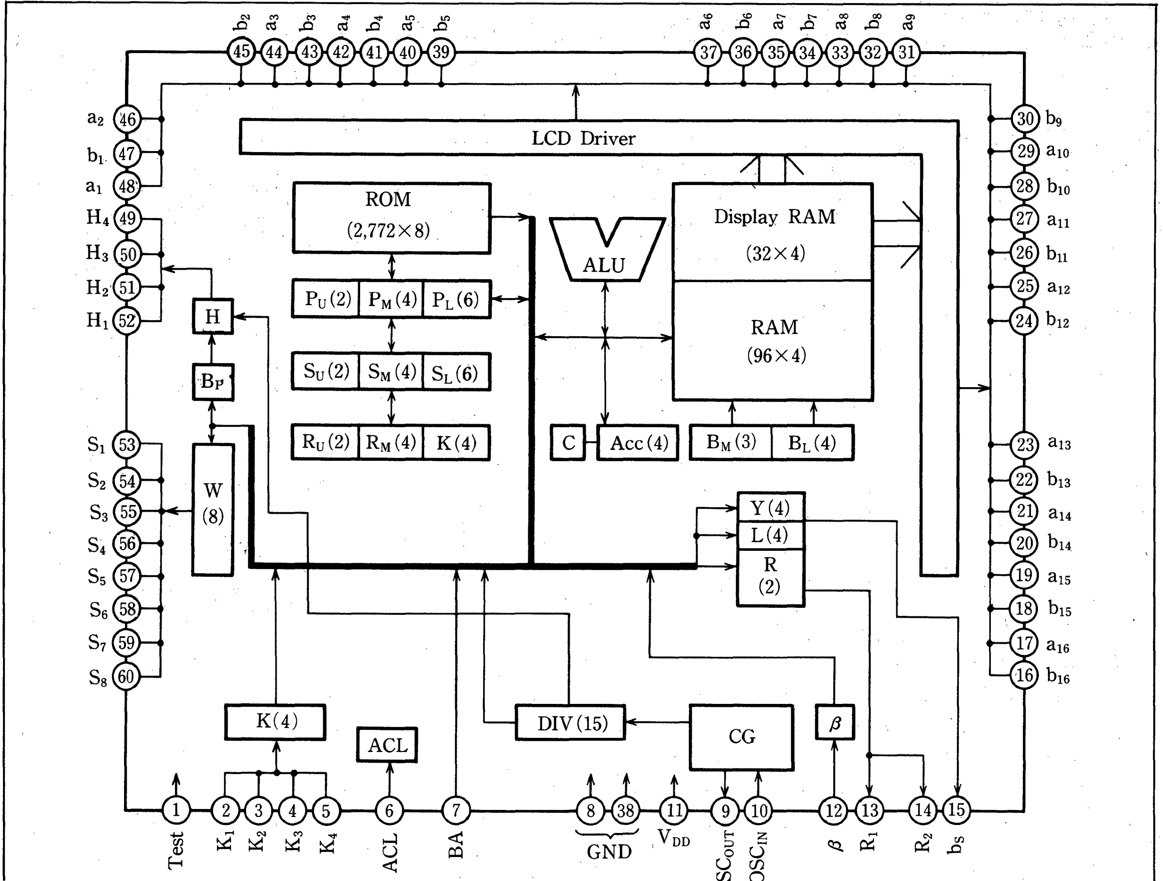
■ Features

1. CMOS process
2. ROM capacity: 2,772×8 bits
3. RAM capacity: 94×4 bits (Data RAM)
   32×4 bits
   (Display RAM)
4. Instruction set: 49
5. Subroutine nesting: 2 levels
6. Instruction cycle: 61 μs (TYP.)
7. Input/output ports
   Input ports: 6 bits
   Output ports: 10 bits
   LCD output ports:
   34 bits for segment
   4 bits for common
8. 15 stage divider with reset
9. LCD drive circuit
   3V, 1/4 duty, 1/3 bias,
   132 segments (MAX.)
10. Crystal oscillator circuit (32.768kHz)
11. Standby mode
12. Single-3V (TYP.) power supply
13. 60-pin QFP (QFP60-P-1414)

■ Pin Connections



Block Diagram



Symbol description

ALU	: Arithmetic logic unit	W	: 8-bit shift register
Acc	: Accumulator	B_M, B_L	: RAM address register
C	: Carry F/F	B_P	: Backplate signal generator circuit
P_U, P_M, P_L	: Program counter	H, L, Y	: 4-bit F/F
S_U, S_M, S_L	: Stack register of program counter	R	: 2-bit F/F
R_U, R_M, R_L	: Stack register of program counter	K	: Key input F/F
DIV	: Divider	CG	: Clock Generator

Pin Description

Symbol	I/O	Circuit type	Function
a_i, b_i	O		Segment output ports (i=1 to 16)
b_S			
H_1-H_4	O		Common output ports
S_1-S_8	O		Strobe output ports
T	I		Test input port (normally connected to GND)
K_1-K_4	I	pull-down	Key input ports
OSC_{IN}			Crystal oscillator
OSC_{OUT}			
BA, β	I	pull-up	Independent input ports
GND, V_{DD}			Power supply
R_1, R_2	O		Melody output ports
ACL	I	pull-down	Auto clear input port

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_{DD}	-3.5 to $+0.3$	V	1
	V_{IN}	V_{DD} to $+0.3$	V	
Operating temperature	Topr	0 to +50	℃	
Storage temperature	Tstg	-55 to $+150$	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	-3.2 to -2.6	V
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz

Electrical Characteristics

($V_{DD}=-3.2$ to -2.6V, Ta=0 to 50℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		-0.6			V	1
	V_{IL1}				$V_{DD}+0.6$	V	
	V_{IH2}		-0.3			V	2
	V_{IL2}				$V_{DD}+0.3$	V	
Input current	I_{IH}	$V_{IN}=0V$	1		15	μA	3
	I_{IL}	$V_{IN}=V_{DD}$	1		15	μA	4
Output voltage	V_{OH}	$I_{OUT}=50\mu A$ to V_{DD}	-0.5			V	5
	V_{OL}	$I_{OUT}=5\mu A$ to GND			$V_{DD}+0.5$	V	
	V_{OA}	$V_{DD}=-3.0V$ No load	-0.3		0	V	6
	V_{OB}		-1.3	-1.0	-0.7	V	
	V_{OC}		-2.3	-2.0	-1.7	V	
	V_{OD}		-3.0		-2.7	V	
Output current	I_{SO}	$V_{OUT}=-0.2V$	100			μA	7
	I_{SIN}	$V_{OUT}=V_{DD}+0.2V$	100			μA	
Supply current	I_{DA}	During full-range operation		40	80	μA	8
	I_{DS}	When system clock is stationary		15	25	μA	

Note 1: Applied to pins K_1-K_4, β_{IN}
Note 2: Applied to pins ACL, BA
Note 3: Applied to pins K_1-K_4
Note 4: Applied to pin β
Note 5: Applied to pins S_1-S_8
Note 6: Applied to pins a_1-a_{16}, b_1-b_{16}, b_S, H_1-H_4
Note 7: Applied to pins R_1, R_2
Note 8: When a bleeder resistor is turned on.

■ Pin Functions

(1) K_1-K_4 (Inputs)

The K_1-K_4 ports normally pulled down are connected to, and loaded into the accumulator (Acc) by instructions.

A matrix composed of K input ports and strobe output ports (S_1-S_8) enables up to 32 kinds of keys to be connected.

In this case, be sure to take the interval at least 1 step between strobe outputs and K inputs.

(2) BA, β (Individual inputs)

The individual input ports BA and β normally pulled up can be tested using the TAL and TB instructions.

Applying a High level to these ports skips the next instruction.

(3) S_1-S_8 (Strobe outputs)

The strobe outputs (S_1-S_8) are used to output an 8-bit W register, and constitute a key input matrix in combination with the input ports K_1-K_4.

The W register is an 8-bit register transferred by the PTW instruction in parallel.

The W'register is an 8-bit shift register of which the least significant bit W_1 is set and reset by WS and WR instructions, and the entire contents of W'register are shifted by one bit.

(4) a_1-a_{16}, b_1-b_{16}, bs

The segment outputs a_1-a_{16}, b_1-b_{16} are connected to the display RAM. By transferring appropriate data to the display RAM, alphanumeric characters are automatically displayed.

The bs is used to output the contents of the L or Y register. Segment output ports are designed to drive an LCD with 1/4 duty cycle. The bs is used to flash the display such as a colon under the control of Y register.

(5) H_1-H_4 (Common outputs)

The H_1-H_4 are used to drive an LCD with 1/4 duty cycle and 1/3 bias, and have the 4 levels of output.

The common outputs control the BP F/F, BC F/F to select the display mode or blanking mode.

Below shows the conditions of a display mode to be selected.

BP=1 and BC=0

(6) R (Buzzer output)

The R_1 and R_2 output ports are used to directly drive a piezo electric buzzer.

The R port can generate the contents of the R register with a mask change, and used as a control signal.

Hardware Configuration

(1) Program counter and stack

The program counter consists of a 2-bit register P_U, a 4-bit register R_M and a 6-bit polynomial counter P_L. The P_U and P_M specify the pages and the P_L specifies the steps within a page.

The stack consists of registers S_U, S_M, S_L and R_U, R_M, R_L, and has 2 levels of nesting.

(2) Program memory (ROM)

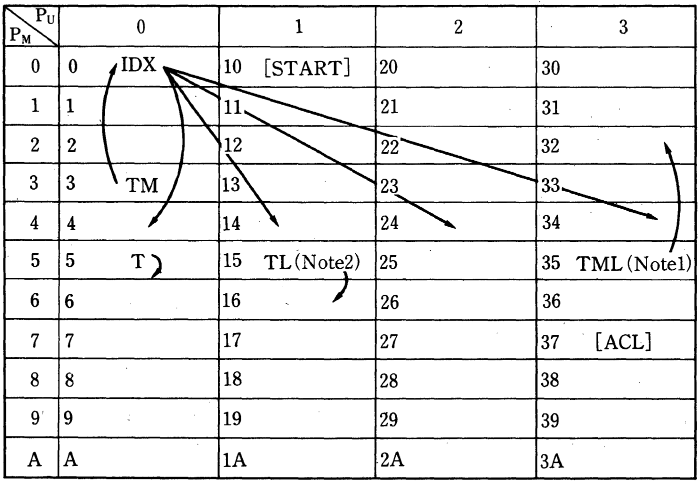
An on-chip 2,772-bit ROM is organized as 44 pages×63 steps. Fig. 1 shows the ROM configuration.

- When power on, the program starts execution from the address $P_U=3$, $P_M=7$, $P_L=0$ specified by an ACL circuit.

P_M \ P_U	0	1	2	3
0	0 Subroutine cover page	10 Start from CEND	20	30
1	1	11	21	31
2	2	12	22	32
3	3	13	23	33
4	4	14	24	34
5	5	15	25	35
6	6	16	26	36
7	7	17	27	37 Power on
8	8	18	28	38
9	9	19	29	39
A	A	1A	2A	3A

Note : 1 page consists of 63 steps.

Fig. 1 ROM configuration



P_M \ P_U	0	1	2	3
0	0 IDX	10 [START]	20	30
1	1	11	21	31
2	2	12	22	32
3	3 TM	13	23	33
4	4	14	24	34
5	5 T	15 TL(Note2)	25	35 TML(Note1)
6	6	16	26	36
7	7	17	27	37 [ACL]
8	8	18	28	38
9	9	19	29	39
A	A	1A	2A	3A

Note 1: Jump address of TML, $P_M=0$ to 3

Note 2: Jump address of TL, all addresses

Fig. 2 Jump instruction and jump addresser

- When the program starts execution from the system clock halt state by a 1S signal or a key input signal, the address starts at $P_U=1$, $P_M=0$, $P_L=0$.
- For the instructions except for a jump instruction, the polynomial counter P_L is shifted by 1 step according to a polynomial code.
- The combination of jump instructions including T, TL, TM, TML, RTN0, RTN1 and ATPL enables to jump to any page or any subroutine. Fig. 2 shows the relationship between jump instructions and jump addresses on a ROM map.

(3) Data memory RAM

A 512-bit data RAM consists of 8×16×4-bits.

The RAM is specified by a 3-bit B_M and a 4-bit B_L. The B_M is used to specify the files and the B_L specify the words. Note that 1 word consists of 4 bits.

The SM510 has 2×16×4 bits of display RAM area out of the entire RAM, and the display RAM is connected to external pins for segment outputs.

Writing data to the display RAM directly drives an LCD with 1/4 duty and 1/3 bias scheme.

Fig. 3 shows the RAM map.

B_M B_L	X 000	Y 001	Z 010	M 011	P 100	Q 101	R 110	S 111
0000								
0001								
0010								
0011								
0100								
0101								
0110								
0111								
1000								
1001								
1010								
1011								
1100								
1101								
1110								
1111								

The area (R, S) enclosed by a thick frame is allocated for a display RAM.

Fig. 3 RAM configuration

(4) Divider circuit for clock function

An internal 15-stage divider circuit is used to make a clock system.

The divider outputs the signal at 1 sec. unit (1S), and γ F/F is set at the rising edge of 1S signal. γ F/F can be tested by an instruction, and reset by the test. A 1 sec. count is notified upon execution of this instruction.

(5) Standby function

The SM511/SM512 is a low power consumption design due to CMOS process. Further low power feature can also be obtained by halting almost all the system clocks through the CEND instruction for low power requirements.

γ F/F must be reset or one or more inputs of K_1-K_4 must go High in order to restart the system clock from the halt state. Then the program starts at the ROM address 1000 ($P_U=1$, $P_M=0$, $P_L=0$).

(6) Clock generator (CG)

The device contains an on-chip crystal oscillator circuit which consists of the external circuit shown in Fig. 4. The system clock has a frequency of one second that of the oscillator frequency.

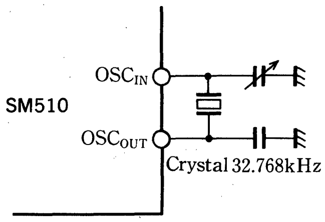


Fig. 4

(7) ACL circuit

Resistors and Capacitors are mounted in an ACL circuit which does not normally require any external circuits.

The ACL will be cleared in about 0.5 sec from a crystal oscillator circuit starts oscillation after power on, and the program starts at $P_U=3$, $P_M=7$, $P_L=0$.

The ACL operations can be obtained by transferring signals into the ACL pin after power on. Note that it takes about 0.5 sec to start execution of the program after the ACL signal is released.

In case noise may harm the ACL operation, apply a 0.01 to 0.1 μF of capacitor between ACL pin and GND pin.

Fig. 5 shows the sample circuit.

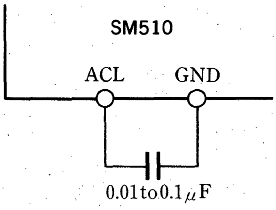
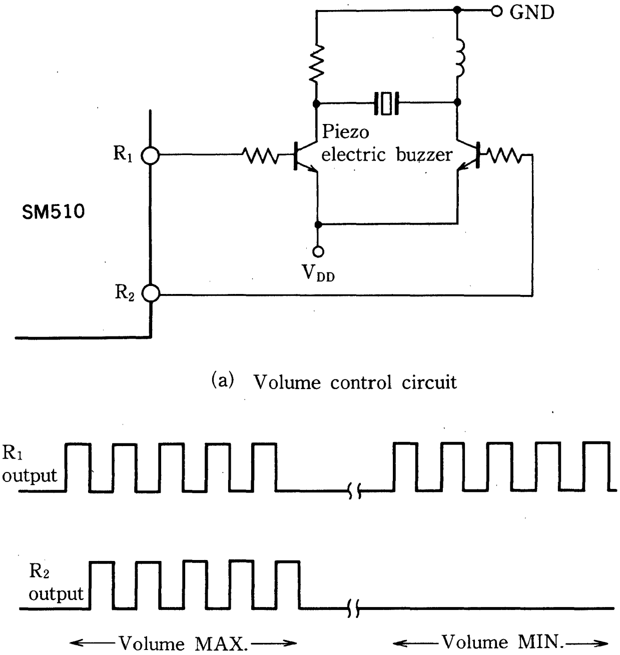


Fig. 5 Compensator for ACL

(8) Buzzer output function

The R_1 and R_2 are buzzer output ports which are used to directly drive a piezo electric buzzer at a frequency of 4,096kHz with a 32.768kHz crystal oscillator.

The R_1 and R_2 ports output different pulse phases which allow the volume control of a buzzer with the circuit shown in fig. 6. These ports can directly drive a piezo electric buzzer. However, it is recommended to use the drive circuits shown in Figs. 6 and 7, to prevent from the malfunction of a system caused by the counter electromotive force from a piezo electric buzzer.



(a) Volume control circuit

(b) Output waveform

Fig. 6 Piezo electric buzzer driver circuit 1

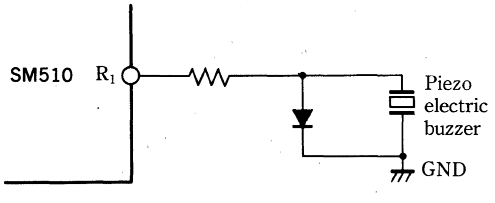


Fig. 7 Piezo electric buzzer driver circuit 2 (Direct driver circuit)

(9) LCD driver

• LCD segment

The SM510 has an on-chip LCD driver circuit which can directly drive an LCD with a 3V, 1/4 duty and 1/3 bias scheme.

The display RAM is connected to segment outputs of ai, bi (i=1 to 16) according to LCD common outputs of H_1-H_4 as shown in fig 8.

The segment outputs provide 1-digit data (M_1-M_4) of the display RAM in synchronizing with H_1-H_4 outputs.

Each segment of the LCD can be turned on or off by controlling the corresponding bit data "1" or "0" in the display RAM area.

The LCD driving waveform relative to the display mode is automatically generated. The device provides the maximum of 132 segments. Fig. 9 shows the segment display example.

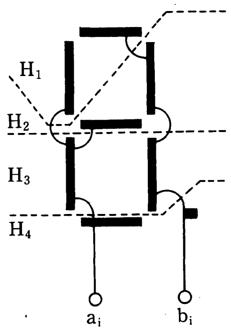


Fig. 9 7-segment numeric LCD digit

• Display waveform

Fig. 10 shows the display waveforms required to display the number "5" on the LCD pattern shown in Fig. 9 (segment outputs a_1, b_1 are used).

In the combination that the potential difference between the common output and segment output is 3V (in the combination of H_4 and a_1), shown in Fig. 10, the segment is turned on, and in the case of 2V or less (in the combinaiton of H_4 and a_1), the segment is turned off.

• LCD flashing output (bs)

The bs output is used to flash symbols displayed on the LCD screen. Otherwise, the bs is used as a segment output in the same way as ai, bi (i=1 to 16).

• Blanking the display

There are two ways for blanking the entire display depending on applications.

1. For blanking the display in a short period of time.

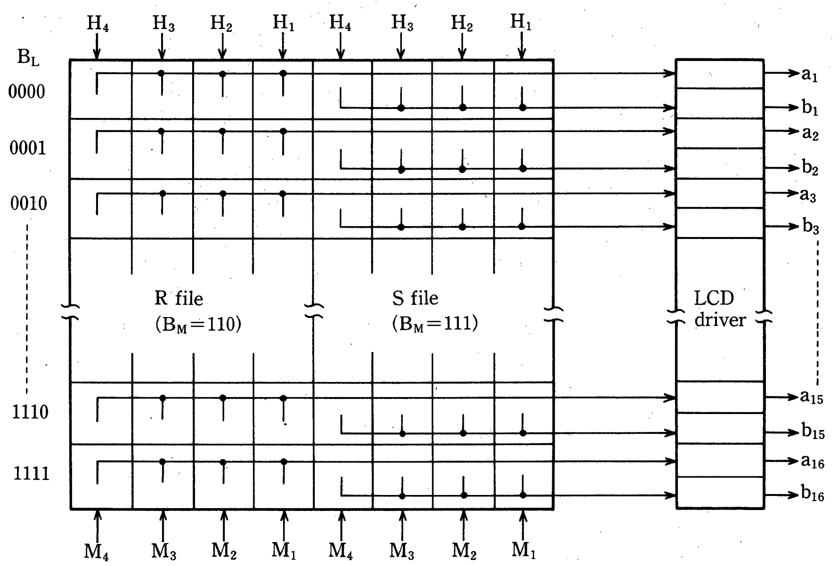


Fig. 8 Display RAM and LCD segment output

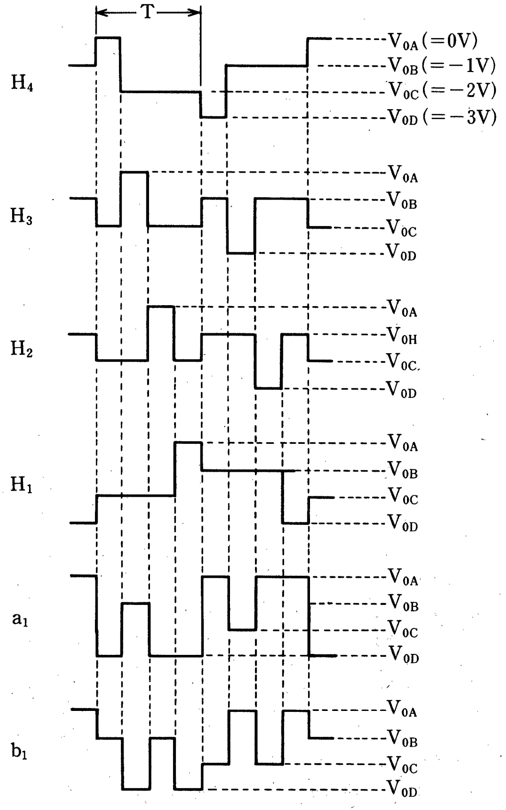


Fig. 10 LCD driving signal waveform

Control the common signal generator circuit by the ATBP instruction.

2. For blanking the display in a long period of time to decrease power consumption.

Use the BDC instruction to turn on and off the liquid crystal bleeder current. In this case, cutting off the bleeder current decreases great amount of power consumption.

Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
LB x	40-4F	$B_{L3}, B_{L2}\leftarrow x\ (I_3)\oplus x\ (I_2)$ $B_{L1}, B_{L0}\leftarrow x\ (I_3, I_2)$ $B_{M1}, B_{M0}\leftarrow x\ (I_1, I_0)$
LBL xy (2-byte)	5F 00-FF	$B_M\leftarrow x\ (I_6\text{-}I_4), B_L\leftarrow y\ (I_3\text{-}I_0)$
SBM	02	$B_{M2}\leftarrow 1$ (only next step)
EXBLA	0B	$A_{CC}\leftrightarrow B_L$
INCB	64	Skip if $B_L=F_H$, $B_L\leftarrow B_L+1$
DECB	6C	Skip if $B_L=0$, $B_L\leftarrow B_L-1$

(2) ROM address instructions

Mnemonic	Machine code	Operation
ATPL	03	$P_{L3}\text{-}P_{L0}\leftarrow \text{Acc}$
RTN0	6E	$P_U\leftarrow S_U\leftarrow R_U$, $P_M\leftarrow S_M\leftarrow R_M$ $P_L\leftarrow S_L\leftarrow R_L$
RTN1	6F	$P_U\leftarrow S_U\leftarrow R_U$, $P_M\leftarrow S_M\leftarrow R_M$ $P_L\leftarrow S_L\leftarrow R_L$, Skip next step
TL xyz (2-byte)	70-7A 00-FE	$P_M\leftarrow x\ (I_3\text{-}I_0)$, $P_U\leftarrow y\ (I_7\text{-}I_6)$ $P_L\leftarrow z\ (I_5\text{-}I_0)$
TML xyz (2-byte)	7C-7F 00-FE	$R\leftarrow S\leftarrow PC+1$, $P_{M3}, P_{M2}\leftarrow(0, 0)$ $P_{M1}, P_{M0}\leftarrow x\ (I_1, I_0)$, $P_U\leftarrow y\ (I_7, I_6)$ $P_L\leftarrow z\ (I_5\text{-}I_0)$
TM x IDX yz (2-byte)	C0-FE 00-FE	$R\leftarrow S\leftarrow PC+1$, $P_U\leftarrow 0$, $P_M\leftarrow 0$ $P_L\leftarrow x\ (I_5\text{-}I_0)$, $P_U\leftarrow y\ (I_7, I_6)$ $P_L\leftarrow z\ (I_5\text{-}I_0)$, $P_M\leftarrow(0100)_2$
T xy	80-BF	$P_L\leftarrow x\ (I_5\text{-}I_0)$

(3) Data transfer instructions

Mnemonic	Machine code	Operation
EXC x	10-13	$\text{Acc}\leftrightarrow M$ $B_{M1}, B_{M0}\leftarrow B_{M1}, B_{M0}\oplus x\ (I_1, I_0)$
BDC	6D	$BC\leftarrow C$ Display on if C=0 Display off if C=1
EXCI x	14-17	$\text{Acc}\leftrightarrow M$ $B_{M1}, B_{M0}\leftarrow B_{M1}, B_{M0}\oplus x\ (I_1, I_0)$ Skip if $B_L=F_H$, $B_L\leftarrow B_L+1$
EXCD x	1C-1F	$\text{Acc}\leftrightarrow M$ $B_{M1}, B_{M0}\leftarrow B_{M1}, B_{M0}\oplus x\ (I_1, I_0)$ Skip if $B_L=0$, $B_L\leftarrow B_L-1$
LDA x	18-1B	$\text{Acc}\leftarrow M$ $B_{M1}, B_{M0}\leftarrow B_{M1}, B_{M0}\oplus x\ (I_1, I_0)$
LAX x	20-2F	$\text{Acc}\leftarrow x\ (I_4\text{-}I_1)$ Skip when in succession
WR	62	$W_7\leftarrow W_6\leftarrow\cdots\leftarrow W_0\leftarrow 0$
WS	63	$W_7\leftarrow W_6\leftarrow\cdots\leftarrow W_0\leftarrow 1$

(4) I/O Instructions

Mnemonic	Machine code	Operation
KTA	6A	$\text{Acc}\leftarrow K$
ATBP	01	$BP\leftarrow \text{Acc}$
ATL	59	$L\leftarrow \text{Acc}$
ATFC	60	$Y\leftarrow \text{Acc}$
ATR	61	$Ri\leftarrow \text{Acc}$, $i=1, 2$

(5) Arithmetic instructions

Mnemonic	Machine code	Operation
ADD	08	$\text{Acc}\leftarrow \text{Acc}+M$
ADD11	09	$\text{Acc}\leftarrow \text{Acc}+M+C$, $C\leftarrow CY$ Skip if CY=1
ADX x	30-3F	$\text{Acc}\leftarrow \text{Acc}+x\ (I_3\text{-}I_0)$ Skip if CY=1
COMA	0A	$\text{Acc}\leftarrow\overline{\text{Acc}}$
DC	3A	$\text{Acc}\leftarrow \text{Acc}+(1010)^2$
ROT	6B	┌$\text{Acc}_0\leftarrow \text{Acc}_1\leftarrow\cdots \text{Acc}_3\leftarrow$┐ └──→ C ──┘
RC	66	$C\leftarrow 0$
SC	67	$C\leftarrow 1$

(6) Test instructions

Mnemonic	Machine code	Operation
TB	51	Skip it $\beta=1$
TC	52	Skip if C=0
TAM	53	Skip if Acc=M
TMI x	54-57	Skip if M=1, $(i=x\ (I_1, I_0))$
TA0	5A	Skip if Acc=0
TABL	5B	Skip if $\text{Acc}=B_L$
TIS	58	Skip if 1S=0, $\gamma\leftarrow 0$
TAL	5E	Skip if BA=1
TF1	68	Skip if $f_1=1$
TF4	69	Skip if $f_4=1$

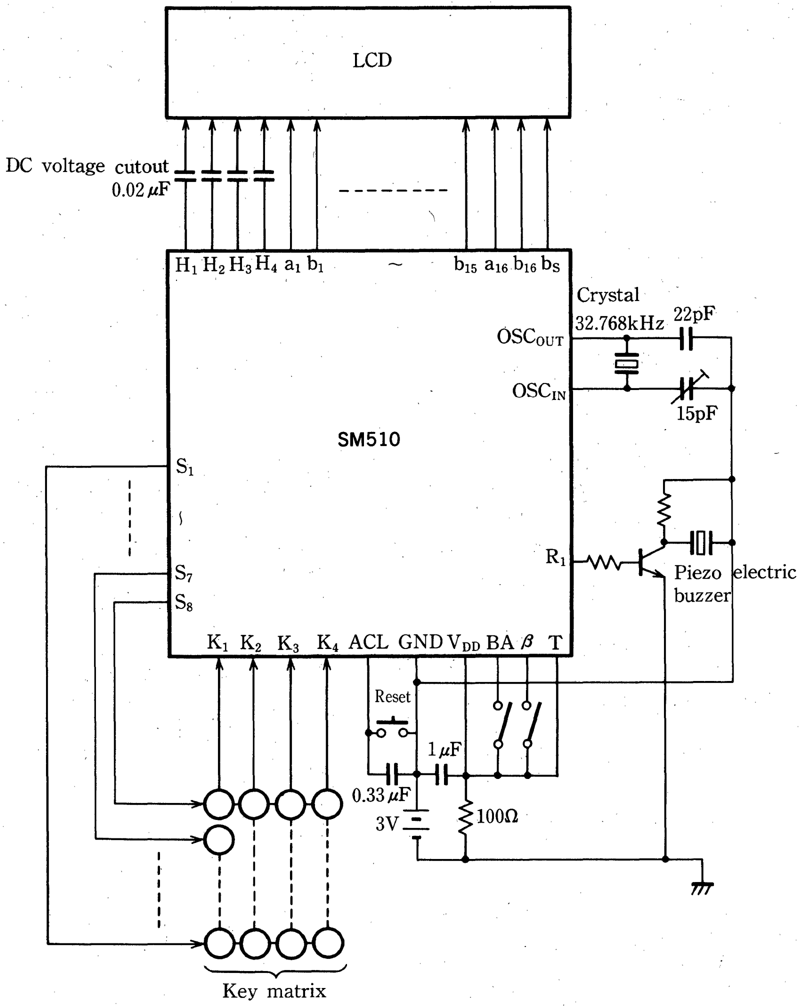
(7) Bit manipulation instructions

Mnemonic	Machine code	Operation
RM x	04-07	$Mi\leftarrow 0$, $i=(I_1, I_0)$
SM x	0C-0F	$Mi\leftarrow 1$, $i=(I_1, I_0)$

(8) Special instructions

Mnemonic	Machine code	Operation
SKIP	00	No operation
CEND	5D	clock stop
IDIV	65	$DIV\leftarrow 0$

System Configuration Example (Electronic calculator with real-time clock function)



SM511/SM512 4-Bit Microcomputer (LCD Driver)

■ Description

The SM511/SM512 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a 4,032×8-bit ROM, a 128/142×4-bit RAM, a 15 stage divider and a 136/200-segment LCD driver circuit.

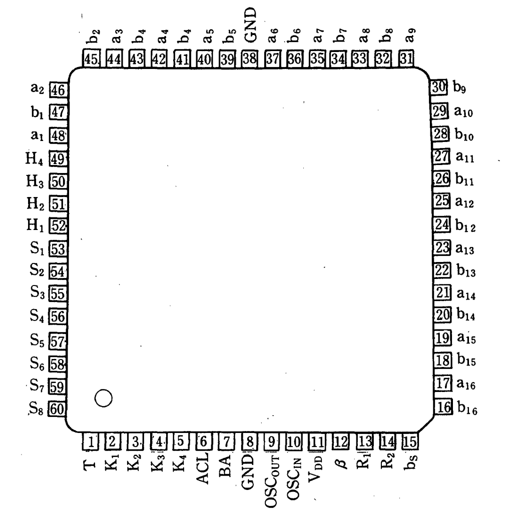
This microcomputer is applicable to many applications having multiple LCD segments with low-power consumption.

■ Features

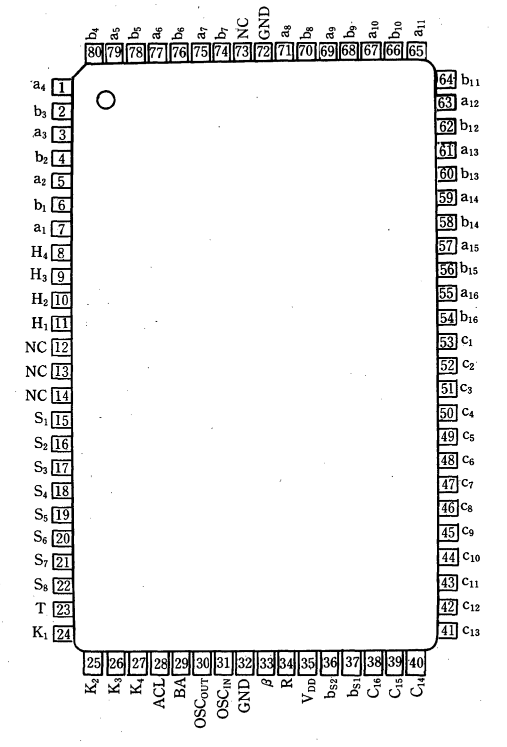
1. CMOS process
2. ROM capacity: 4,032×8 bits
3. RAM capacity:
   Data RAM: 96×4 bits (SM511)
   80×4 bits (SM512)
   Display RAM: 32×4 bits (SM511)
   48×4 bits (SM512)
4. Instruction set: 55
5. Subroutine nesting: 2 levels
6. Instruction cycle: 61 μs (TYP.)
7. I/O ports
   Input ports: 6 bits
   Output ports: 9 bits
   LCD output ports
   Segment: 34 bits for SM511
   : 50 bits for SM512
   Common: 4 bits
8. 15-stage divider with reset
9. Melody generator circuit
   (Output time for up to 32 sec.)
10. LCD drive circuit
    3V, 1/4 duty, 1/3 bias,
    136 segments (MAX.) for SM511
    200 segments (MAX.) for SM512
11. Crystal oscillator circuit (32.768kHz)
12. Standby mode: 20 μA
    (Current consumption at clock halt)
13. Single −3V (TYP.) power supply
14. 60-pin QFP (QFP60-P-1414) for SM511
    80-pin QFP (QFP80-P-1420) for SM512

■ Pin Connections

SM511

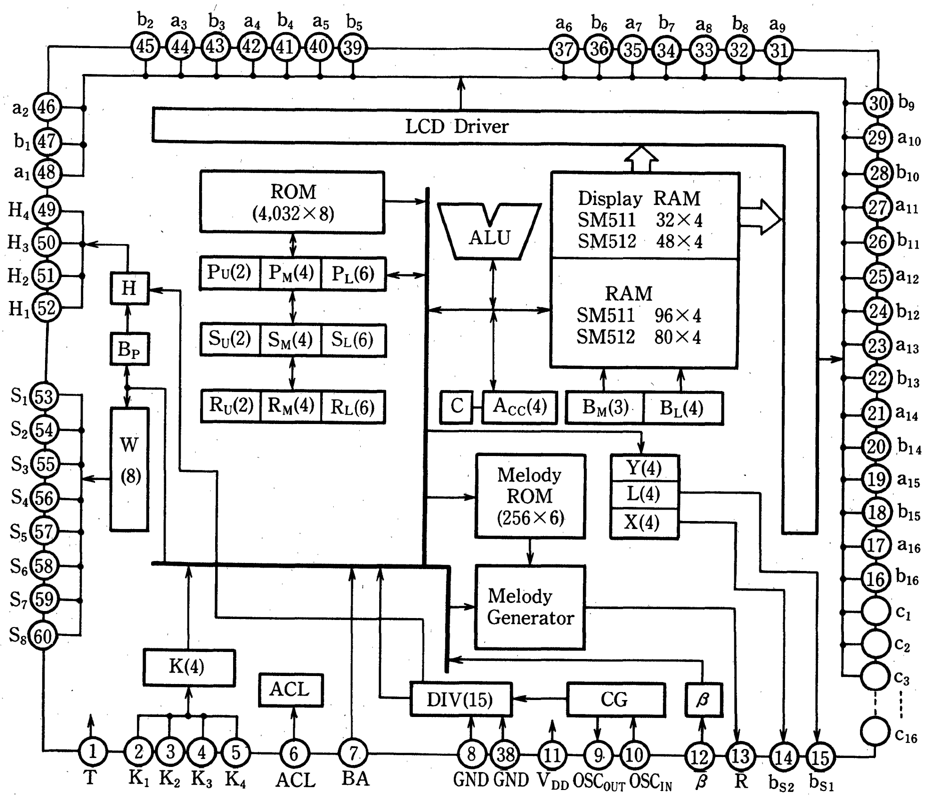


SM512



Top View

Block Diagram



Symbol description

ALU	: Arithmetic logic unit	W	: 8-bit shift register
Acc	: Accumulator	B$_M$, B$_L$	: RAM address register
C	: Carry F/F	B$_P$	: Backplate signal generator circuit
P$_U$, P$_M$, P$_L$	: Program counter	H, L, X, Y	: 4-bit F/F
S$_U$, S$_M$, S$_L$	: Stack register of program counter	K	: Key input F/F
R$_U$, R$_M$, R$_L$	: Stack register of program counter	CG	: Clock generator
DIV	: Divider		

Note: Pin numbers apply to the SM511. Signals C$_1$-C$_{16}$ apply to the SM512 only.

Pin Description

Symbol	I/O	Circuit type	Function
ai, bi, ci*	O		Segment output ports (i=1 to 16)
b_{S1}, b_{S2}			
H_1-H_4	O		Common output ports
S_1-S_8	O		Strobe output ports
T	I		Test input port (normally connected to V_{DD})
K_1-K_4	I	Pull down	Key input ports
OSC_{IN}			Crystal oscillator
OSC_{OUT}			
BA, β	I	Pull up	Independent input ports
GND, V_{DD}			Power supply
R	O		Melody output port
ACL	I	Pull down	Auto clear

*The Ci port applies to the SM512 only.

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Pin voltage	V_{DD}	−3.5 to +0.3	V	1
	V_{IN}	V_{DD} to +0.3	V	
Operating temperature	Topr	0 to 50	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND (GND=0V).

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	−3.2 to −2.6	V
Oscillator frequency	f_{OSC}	32.768 (TYP.)	kHz

Electrical Characteristics

($V_{DD}=-3V\pm10\%$, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		−0.6			V	1
	V_{IL1}				$V_{DD}+0.6$	V	
	V_{IH2}		−0.3			V	2
	V_{IL2}				$V_{DD}+0.3$	V	
Input current	I_{IH1}	$V_{IN}=0V$	1		15	μA	3
	I_{IH2}	$V_{IN}=V_{DD}$	1		15	μA	4
Output voltage	V_{OH}	$I_{OUT}=50\mu A$ to V_{DD}	−0.5			V	5
	V_{OL}	$I_{OUT}=5\mu A$ to GND			$V_{DD}+0.5$	V	
	V_{OA}	$V_{DD}=-3.0V$ No load	−0.3	0		V	6
	V_{OB}		−1.3	−1.0	−0.7	V	
	V_{OC}		−2.3	−2.0	−1.7	V	
	V_{OD}		−3.0	−3.0	−2.7	V	
Output current	I_{SO}	$V_{OUT}=-0.2V$	100			μA	7
	I_{SIN}	$V_{OUT}=V_{DD}+0.2V$	100			μA	
Supply current	I_{DA1}	During full-range operation		50		μA	8
	I_{DS1}	When system clock is stationary		20		μA	
	I_{DA2}	During full-range operation		35		μA	9
	I_{DS2}	When system clock is stationary		17		μA	

Note 1: Applied to pins K_1-K_4, β
Note 2: Applied to pins ACL, BA
Note 3: Applied to pins K_1-K_4
Note 4: Applied to pin β
Note 5: Applied to pins S_1-S_8
Note 6: Applied to pins a_1-a_{16}, b_1-b_{16}, c_1-c_{16}, b_{S1}, b_{S2}, H_1-H_4 (c_1-c_{15} apply to the SM512 only)
Note 7: Applied to pin R
Note 8: When melody circuit is inoperative with V_{DD} at −3.0V and system clock at 16.384kHz.
Note 9: When melody circuit is inoperative with V_{DD} at −3.0V and system clock at 8.192kHz.

Pin Functions

(1) K_1-K_4 (Inputs)

The K_1-K_4 ports normally pulled down are connected to, and loaded into the accumulator (Acc) by instructions.

A matrix composed of K input ports and strobe output ports (S_1-S_8) enables up to 32 kinds of keys to be connected.

In this case, be sure to take the interval at least 1 step between strobe outputs and K inputs.

(2) BA, β (Individual inputs)

The individual input ports BA and β normally pulled up can be tested using the TAL and TB instructions.

Applying a High level signal to these ports skips the next instruction.

(3) S_1-S_8 (Strobe outputs)

The strobe outputs (S_1-S_8) are used to output an 8-bit W register, and compose a key input matrix in combination with the input ports K_1-K_4.

The W register is an 8-bit register transferred by the PTW instruction in parallel.

The W'register is an 8-bit shift register of which the least significant bit W_1 is set and reset by WS and WR instructions, and the entire contents of W'register are shifted by one bit.

(4) a_1-a_{16}, b_1-b_{16}, c_1-c_{16}, bs_1, bs_2

The segment outputs a_1-a_{16}, b_1-b_{16}, including c_1-c_{16} (for SM512 only) are connected to display RAM. By transferring appropriate data to the display RAM, alphanumeric characters are automatically displayed.

The bs_1 and bs_2 are used to output the contents of L F/F and X F/F.

Segment output ports are designed to drive an LCD with 1/4 duty cycle.

The bs_1 is used to flash the display such as a colon under the control of Y F/F.

(5) H_1-H_4 (Common outputs)

The H_1-H_4 are used to drive an LCD with 1/4 duty cycle and 1/3 bias, and provide a 4 level output.

The common outputs control the BP F/F, BC F/F to select a display mode or blanking mode.

Below shows the conditions of a display mode to be selected.

BP=1 and BC=0

(6) R (Melody output)

An internal melody generator circuit provides a variety of sound signals.

Hardware Configuration

(1) Program counter and stack

The program counter consists of a 2-bit register P_U, a 4-bit register R_M and a 6-bit polynomial counter P_L. The P_U and P_M specify the pages and the P_L specifies the steps within a page.

The stack consists of registers S_U, S_M, S_L and R_U, R_M, R_L, and has 2 levels of nesting.

(2) Program memory (ROM)

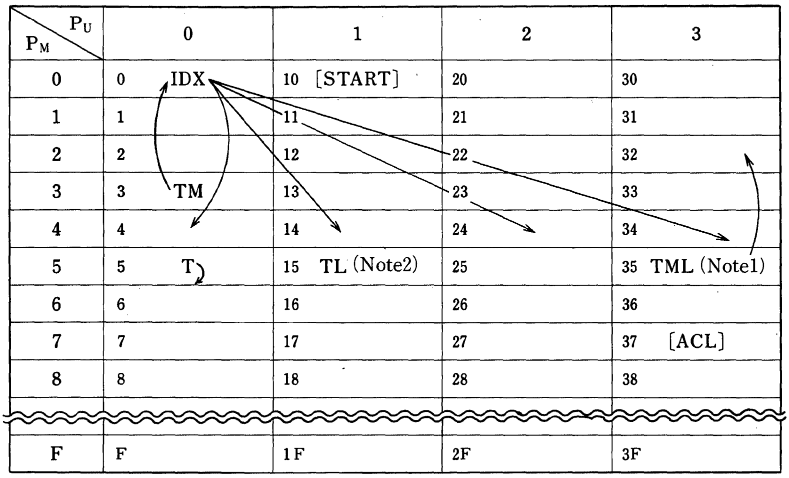
An on-chip 4,032-bit ROM is organized as 64 pages×63 steps. Fig. 1 shows the ROM configuration.

- When power on, the system starts execution from the address $P_U=3$, $P_M=7$, $P_L=0$ specified by an ACL circuit.
- When the system starts execution from the system clock halt state by a 1S signal or a key input signal, the address starts at $P_U=1$, $P_M=0$, $P_L=0$.
- For the instructions except for a jump instruction, the polynomial counter P_L is shifted by 1 step according to a polynomial code.
- The combination of jump instructions including T, TL, TM, TML, RTN0, RTN1 and ATPL enables to jump to any page or any subroutine. Fig. 2 shows the relationship between jump instructions and jump addresses on a ROM map.

P_U P_M	0	1	2	3
0	0 Subroutine cover page	10 Start from CEND	20	30
1	1	11	21	31
2	2	12	22	32
3	3	13	23	33
4	4	14	24	34
5	5	15	25	35
6	6	16	26	36
7	7	17	27	37 Power on
8	8	18	28	38
9	9	19	29	39
A	A	1A	2A	3A
B	B	1B	2B	3B
C	C	1C	2C	3C
D	D	1D	2D	3D
E	E	1E	2E	3E
F	F	1F	2F	3F

Note: 1 page consists of 63 steps.

Fig. 1 ROM configuration



P_M \ P_U	0	1	2	3
0	0 IDX	10 [START]	20	30
1	1	11	21	31
2	2	12	22	32
3	3 TM	13	23	33
4	4	14	24	34
5	5 T	15 TL (Note2)	25	35 TML (Note1)
6	6	16	26	36
7	7	17	27	37 [ACL]
8	8	18	28	38
F	F	1F	2F	3F

Note 1: Jump address of TML should be PM=0 to 3
Note 2: Jump address of TL should be all address.

Fig. 2 Jump instructions and jump address

B_L \ B_M	X 000	Y 001	Z 010	M 011	P 100	Q 101	R 110	S 111
0000						▨	▨	▨
0001						▨	▨	▨
0010						▨	▨	▨
0011						▨	▨	▨
0100						▨	▨	▨
0101						▨	▨	▨
0110						▨	▨	▨
0111						▨	▨	▨
1000						▨	▨	▨
1001						▨	▨	▨
1010						▨	▨	▨
1011						▨	▨	▨
1100						▨	▨	▨
1101						▨	▨	▨
1110						▨	▨	▨
1111						▨	▨	▨

The area (R, S) enclosed by a thick frame is allocated for a display RAM of the SM511.

The area (Q, R, S) with oblique lines is allocated for a display RAM of the SM512.

Fig. 3 RAM configuration

(3) Data memory RAM

A 512-bit data RAM consists of 8×16×4 bits.

The RAM is specified by a 3-bit B_M and a 4-bit B_L. The B_M is used to specify the files and the B_L specify the words. Note that 1 word consists of 4-bits.

The SM511 has 2×16×4-bits and the SM512 has 3×16×4 bits of display RAM area out of the entire RAM, and the display RAM is connected to external pins for segment outputs.

Writing data to the display RAM directly drives an LCD with 1/4 duty and 1/3 bias scheme.

Fig. 3 shows the RAM map.

(4) Divider circuit for clock function

An internal 15-stage divider circuit is used to make a clock system.

The divider outputs the signal at 1 sec. unit (1S), and γF/F is set at the rising edge of 1S signal. γ F/F can be tested by an instruction, and reset by the test. A 1 sec. count is notified upon execution of this instruction.

(5) Standby function

The SM511/SM512 is a low power consumption design due to CMOS process. Further low power feature can also be obtained by halting almost all the system clocks by executing the CEND instruction for low power requirements.

γF/F must be reset or one or more inputs of K_1-K_4 must go High in order to restart the system clock from the halt state. Then the program starts at the ROM address 1000 (P_U=1, P_M=0, P_L=0).

(6) Selection of system clock

Either system clock of 16.384kHz or 8.192kHz can be selected by instructions. Selecting 8.192kHz of clock offers low power consumption with low speed of execution time. The rest of functions are the same with the case when 16.384kHz clock is selected.

The system clock immediately after executing ACL is set to 8.192kHz of clock.

(7) ACL circuit

Resistors and Capacitors are mounted in an ACL circuit which does not normally require any external circuits.

The ACL will be cleared in about 0.5 sec from a crystal oscillator circuit starts oscillation after power on, and the program starts at P_U=3, P_M=7, P_L=0.

The ACL operations can be obtained by transferring signals into the ACL pin after power on. Note that it takes about 0.5 sec to start execution of program after the ACL signal is released.

In case noise may harm the ACL operation, apply a 0.01 to 0.1 μF of capacitor between ACL pin and GND pin.

Fig. 4 shows the sample circuit.

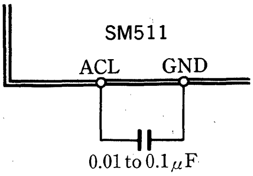


Fig. 4 Compensator for ACL

(8) Melody ROM

A melody ROM is organized as 256 steps×6 bits which executes musical notes, pause and stop instructions.

The melody ROM is used to output 12 musical scales (555 to 2097Hz) in two octaves and select tone length either 125ms or 62.5ms.

It also generates melodies, effect sound, alarms and repeats any part of melodies.

	m_3 m_2 m_1 m_0	Output frequency (Hz)	Clock cycle*	(Note)
do	0 0 1 0	2114.1	15.5	7 8 8 8
si	0 0 1 1	1985.9	16.5	8 8 8 9
la #	0 1 0 0	1872.4	17.5	8 9 9 9
la	0 1 0 1	1771.2	18.5	9 9 9 10
so #	0 1 1 0	1680.4	19.5	9 10 10 10
so	0 1 1 1	1560.4	21.0	10 11 10 11
fa #	1 0 0 0	1489.5	22.0	11 11 11 11
fa	1 0 0 1	1394.4	23.5	11 12 12 12
mi	1 0 1 0	1310.7	25.0	12 13 12 13
re #	1 0 1 1	1236.5	26.5	13 13 13 14
re	1 1 0 0	1170.3	28.0	14 14 14 14
do #	1 1 0 1	1110.8	29.5	14 15 15 15

* : Clock cycle of 32.768kHz

Note: The numerics within waveforms show the clock cycle count of the crystal oscillation frequency 32.768kHz. As shown in table 1, the melody output waveform is generated with the controlled frequency by amplifying the numerics.

Fig. 5 Melody output frequency

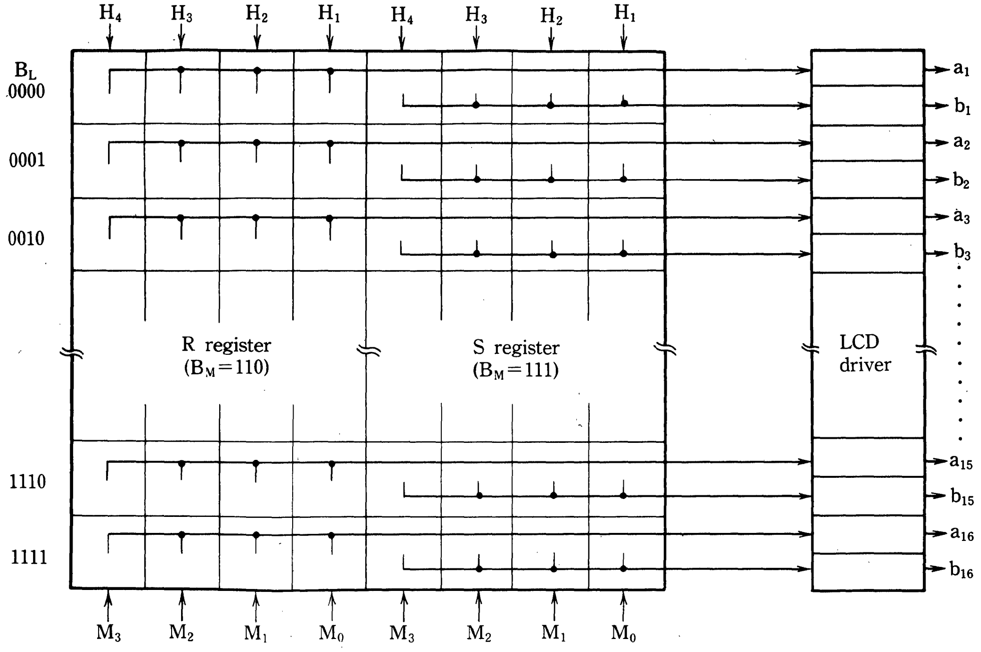


Fig. 6 Display RAM of the SM511 and LCD segment output

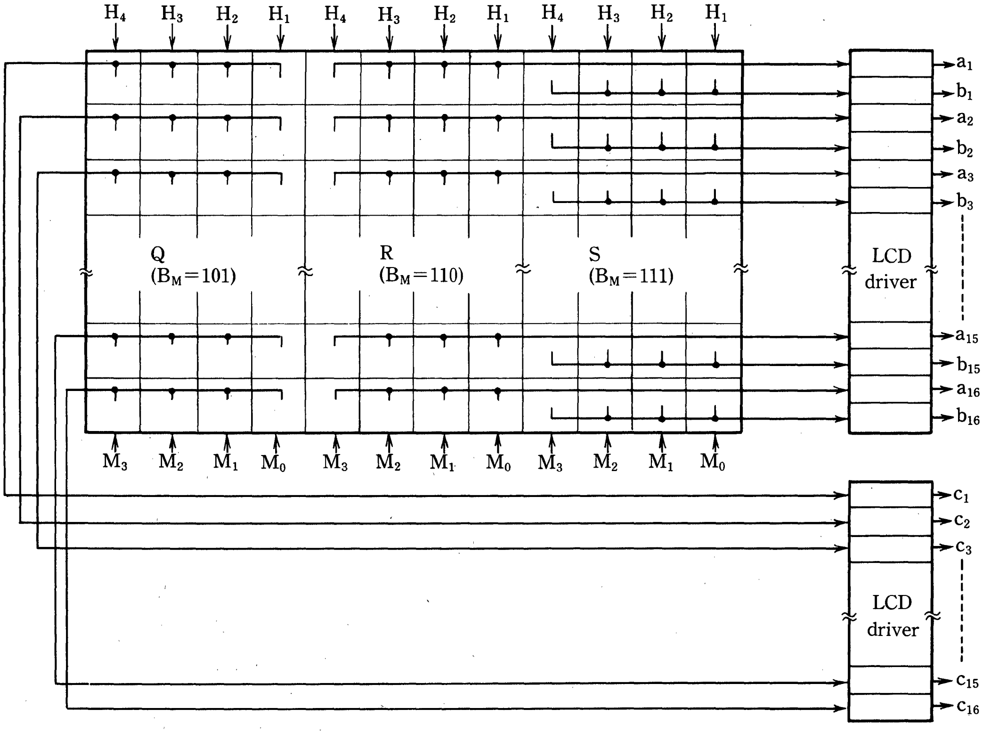


Fig. 7 Display RAM of the SM512 and LCD segment output

(9) LCD driver

• LCD segment

The SM511/SM512 has an on-chip LCD driver circuit which can directly drive an LCD with a 3V, 1/4 duty and 1/3 bias scheme.

The display RAM is connected to segment outputs of a_1-a_{16}, b_1-b_{16} for the SM511 and a_1-a_{16}, b_1-b_{16}, c_1-c_{16} for the SM512 according to LCD common outputs of H_1-H_4 as shown in Fig. 6 and 7.

The segment outputs provide 1-digit data (M_0-M_3) of the display RAM in synchronizing with H_1-H_4 outputs.

Each segment of the LCD can be turned on or off by controlling the corresponding bit data "1" or "0" in the display RAM area.

The LCD driving waveform relative to the display mode is automatically generated. The SM511 provides the maximum of 136 segments, and the SM512 provides 200 segments. Fig. 8 shows the segment display example.

• Display waveform

Fig. 9 shows the display waveforms required to display the number "5" on the LCD pattern of the SM511 and "PM 5" on that of the SM512 shown in Fig. 8 (segment outputs a_1, b_1 for SM511 and a_1, b_1, c_1 for SM512 are used).

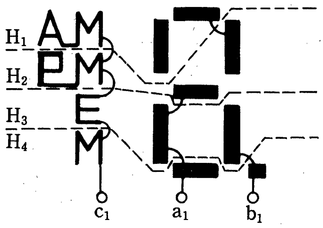


Fig. 8 7-segment numeric LCD digit

• LCD flashing output (bs_1)

The bs_1 output is used to flash symbols displayed on the LCD screen.

For the 4 segments determined by the combination of the segment output bs_1 and common outputs H_1-H_4, each segment can be turned on and off or flashed.

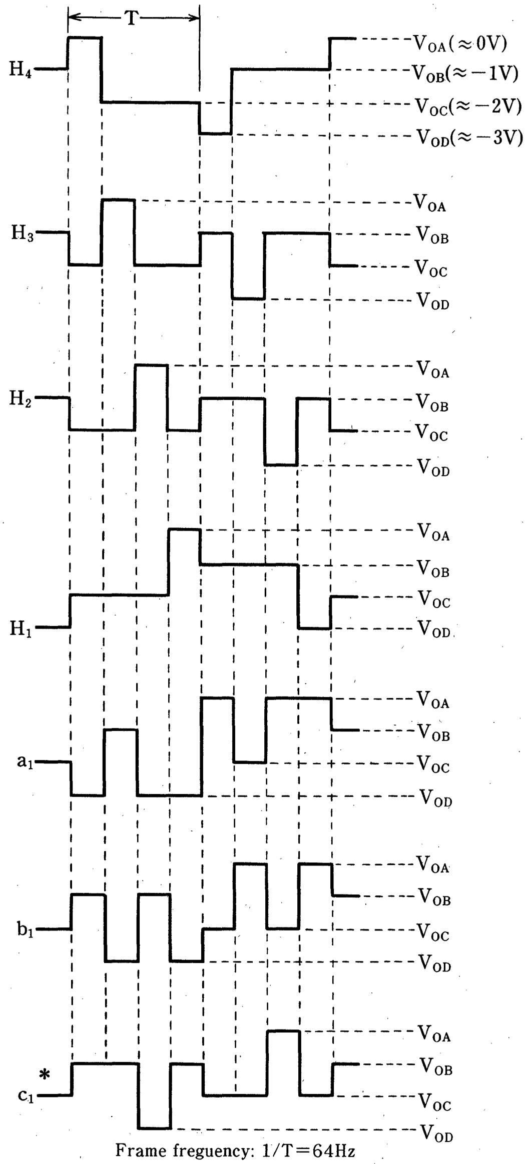
The flashing time should be on for 0.5 sec and off for 0.5 sec.

• Blanking the display

There are two ways for blanking the entire display depending on applications.

1. When blanking the display for a short period of time, control the common signal generator circuit by the ATBP instruction.

2. When blanking the display for a long period of time to decrease power consumption, use the BDC instruction to turn on and off the liquid crystal bleeder current. In this case, to cut off the bleeder current decreases great amount of power consumption.



Frame freguency: 1/T=64Hz

* C_1 is applicable to the SM512 only

Fig. 9 LCD driving signal waveform

■ Comparison Table of The SM510 Series

		SM510	SM511	SM512	
Process		CMOS	CMOS	CMOS	
ROM capacity		2,772×8	4,032×8	4,032×8	bit
RAM capacity		96×4+32×4	96×4+32×4	80×4+48×4	bit
Supply voltage		−3.2 to −2.6	−3.2 to −2.6	−3.2 to −2.6	V
Instruction set		49	55	55	
Subroutine nesting		2	2	2	level
Input port		6	6	6	
Output port		47	47	63	
Instruction cycle (MIN.)		61	61	61	μs
System clock generator circuit		Yes	Yes	Yes	
Current consumption	CEND	15 μA (TYP.) (Note 1)	20 μA (TYP.) (Note 1) 17 μA (TYP.) (Note 2)	20 μA (TYP.) (Note 1) 17 μA (TYP.) (Note 2)	μA
	Operation	40 μA (TYP.) (Note 1)	50 μA (TYP.) (Note 1) 35 μA (TYP.) (Note 2)	50 μA (TYP.) (Note 1) 35 μA (TYP.) (Note 2)	μA
Operating temperature		0 to 50	0 to 50	0 to 50	℃
Package		60QFP	60QFP	80QFP	
Number of LCD segment		132 (MAX.)	136 (MAX.)	200 (MAX.)	
Melody output		No (4,096kHz sound output)	Yes	Yes	

Note 1: System clock: 16.384kHz
Note 2: System clock: 8.192kHz

Instruction Set

(1) RAM address instructions

Mnemonic	Machine code	Operation
LB xy	40-4F	$B_{L1}, B_{L0} \leftarrow x(I_3, I_2)$ $B_{M1}, B_{M0} \leftarrow y(I_2, I_0)$
LBL xy (2-byte)	5F 00-FF	$B_M \leftarrow x(I_6 - I_4)$, $B_L \leftarrow y(I_3 - I_0)$
SBM	02	$B_{M2} \leftarrow 1$ (only next step)
EXBLA	0B	$Acc \leftrightarrow B_L$
INCB	64	Skip if $B_L = F_H$, $B_L \leftarrow B_L + 1$
DECB	6C	Skip if $B_L = 0$, $B_L \leftarrow B_L - 1$

(2) ROM address instructions

Mnemonic	Machine code	Operation
ATPL	03	$P_{L3}\text{-}P_{L0} \leftarrow Acc$
RTN0	6E	$P_U \leftarrow S_U \leftarrow R_U$, $P_M \leftarrow S_M \leftarrow R_M$ $P_L \leftarrow S_L \leftarrow R_L$
RTN1	6F	$P_U \leftarrow S_U \leftarrow R_U$, $P_M \leftarrow S_M \leftarrow R_M$ $P_L \leftarrow S_L \leftarrow R_L$, Skip next step
TL xyz (2-byte)	70-7F 00-FF	$P_M \leftarrow x(I_3\text{-}I_0)$, $P_U \leftarrow y(I_7\text{-}I_6)$ $P_L \leftarrow z(I_5\text{-}I_0)$
TML xyz (2-byte)	68-6B 00-FF	$R \leftarrow S \leftarrow PC+1$, $P_{M3}, P_{M2} \leftarrow (0, 0)$ $P_{M1}, P_{M0} \leftarrow x(I_1, I_0)$, $P_U \leftarrow y(I_7, I_6)$ $P_L \leftarrow z(I_5\text{-}I_0)$
TM x IDX yz (2-byte)	C0-FF 00-FF	$R \leftarrow S \leftarrow PC+1$, $P_U \leftarrow 0$, $P_M \leftarrow 0$ $P_L \leftarrow x(I_5\text{-}I_0)$, $P_U \leftarrow y(I_7, I_6)$ $P_L \leftarrow z(I_5\text{-}I_0)$, $P_M \leftarrow (0100)_2$
T xy	80-BF	$P_L \leftarrow x(I_5\text{-}I_0)$

(3) Data transfer instructions

Mnemonic	Machine code	Operation
EXC x	10-13	$Acc \leftrightarrow M$ $B_{M1}, B_{M0} \leftarrow B_{M1}, B_{M0} \oplus x(I_1, I_0)$
BDC	60-34	$BC \leftarrow C$ Display on if $C=0$ Display off if $C=1$
EXCI x	14-17	$Acc \leftrightarrow M$ $B_{M1}, B_{M0} \leftarrow B_{M1}, B_{M0} \oplus x(I_1, I_0)$ Skip if $B_L = F_H$, $B_L \leftarrow B_L + 1$
EXCD x	1C-1F	$Acc \leftrightarrow M$ $B_{M1}, B_{M0} \leftarrow B_{M1}, B_{M0} \oplus x(I_1, I_0)$ Skip if $B_L = 0$, $B_L \leftarrow B_L - 1$
LDA x	18-1B	$Acc \leftrightarrow M$ $B_{M1}, B_{M0} \leftarrow B_{M1}, B_{M0} \oplus x(I_1, I_0)$
LAX x	20-2F	$Acc \leftarrow x(I_4\text{-}I_1)$ Skip when in succession
PTW	6D	$W_1 \leftarrow W'_i$ $(i=7$ to $0)$
WR	62	$W'_0 \leftarrow 0$, $W'_{i+1} \leftarrow W_i$ $(i=6$ to $0)$
WS	63	$W'_0 \leftarrow 1$, $W'_{i+1} \leftarrow W_i$ $(i=6$ to $0)$

(4) I/O instructions

Mnemonic	Machine code	Operation
KTA	50	$Acc \leftarrow K$
ATBP (2-byte)	60 35	$BP \leftarrow Acc$
ATX	5C	$X \leftarrow Acc$
ATL	59	$L \leftarrow Acc$
ATFC (2-byte)	60 33	$Y \leftarrow Acc$

(5) Arithmetic instructions

Mnemonic	Machine code	Operation
ADD	08	$Acc \leftarrow Acc + M$
ADD11	09	$Acc \leftarrow Acc + M + C$, $C \leftarrow CY$ Skip if $CY=1$
ADX x	30-3F	$Acc \leftarrow Acc + x(I_3\text{-}I_0)$ Skip if $CY=1$
COMA	0A	$Acc \leftarrow \overline{Acc}$
DC	3A	$Acc \leftarrow Acc + (1010)_2$
ROT	00	$C \leftarrow Acc_o$, $Acc_i \leftarrow Acc_{i+1}$ $(i=2$ to $0)$, $Acc_3 \leftarrow C$
RC	66	$C \leftarrow 0$
SC	67	$C \leftarrow 1$

(6) Test instructions

Mnemonic	Machine code	Operation
TB	51	Skip if $\beta = 1$
TC	52	Skip if $C=0$
TAM	53	Skip if $Acc = M$
TMI x	54-57	Skip if $M_i = 1$, $(i = x(I_1, I_0))$
TA0	5A	Skip if $Acc = 0$
TABL	5B	Skip if $Acc = B_L$
TIS	58	Skip if $IS=0$, $r \leftarrow 0$
TAL	5E	Skip if $BA=1$

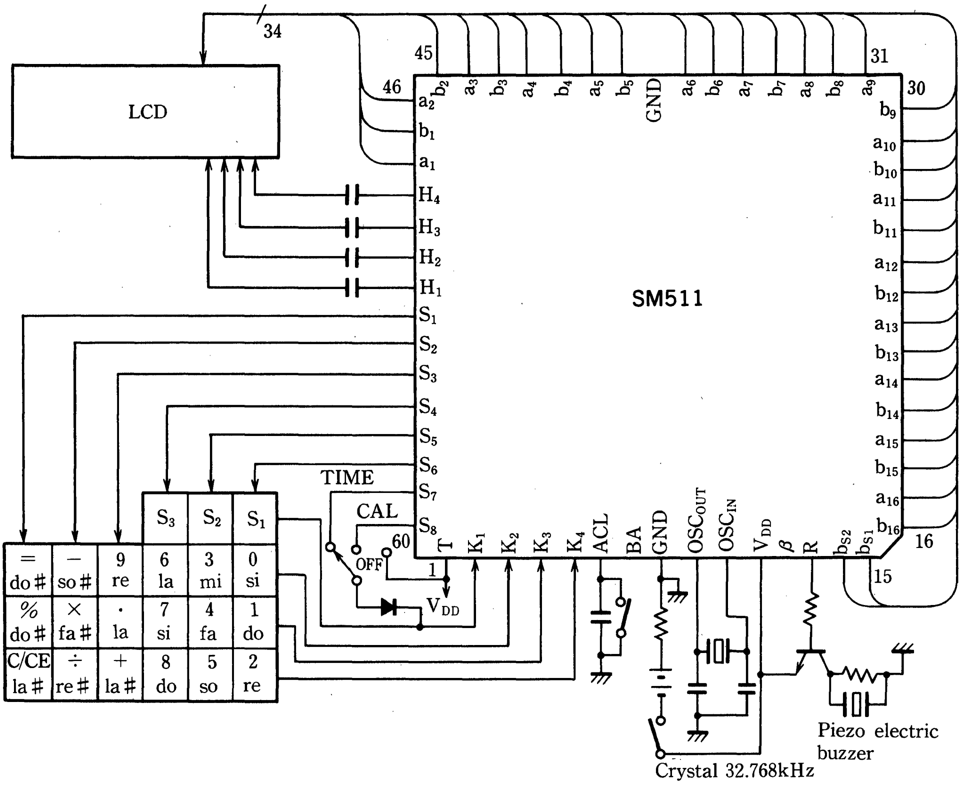
(7) Bit manipulation instructions

Mnemonic	Machine code	Operation
RM x	04-07	$M_i \leftarrow 0$ $(i = x(I_1, I_0))$
SM x	0C-0F	$M_i \leftarrow I$ $(i = x(I_1, I_0))$

(8) Melody control instructions

Mnemonic	Machine code	Operation
PRE x (2-byte)	61 00-FF	Melody ROM pointer preset
SME (2-byte)	60 31	$ME \leftarrow 1$
RME (2-byte)	60 30	$ME \leftarrow 0$
TMEL (2-byte)	60 32	Skip if $MES=1$, $MES \leftarrow 0$

System Configulation Example



SM563 4-Bit Microcomputer (LCD Driver)

Description

The SM563 is a CMOS 4-bit microcomputer which integrates a 4-bit parallel processing function, a ROM, a RAM, I/O ports, a serial interface, a timer/event counter in a single chip.

It provides five kinds of interrupt and a subroutine stack function using the RAM area.

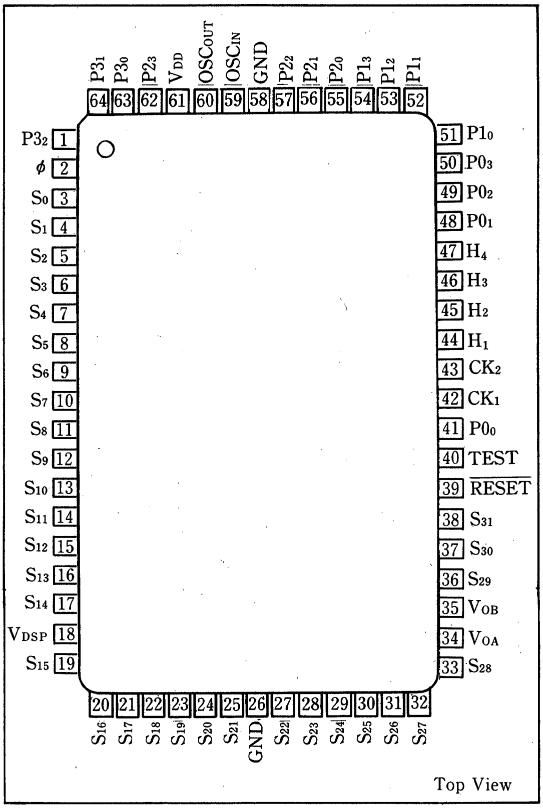
Provided with a 128-segment LCD driver circuit, this microcomputer is applicable to low power systems with multiple LCD segments.

Features

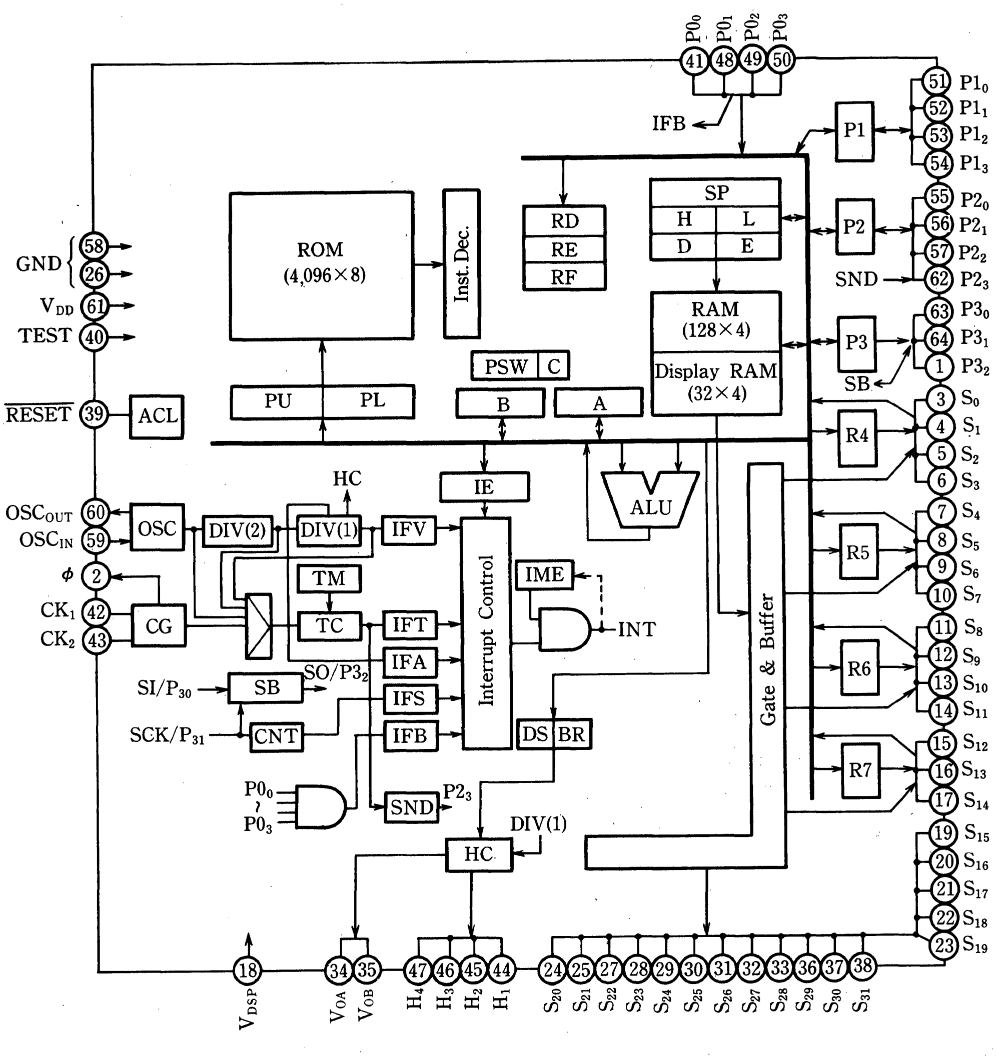
1. CMOS process
2. ROM capacity: 4,096×8 bits
3. RAM capacity
   Data RAM: 128×4 bits
   Display RAM: 32×4 bits
4. Instruction set: 98
5. Subroutine stack: using RAM area
6. Instruction cycle:
   6.67 μs (V_{DD}=3V)
   2.0 μs (V_{DD}=5V)
7. Input/output ports
   I/O ports: 26*
   Input ports: 4
   LCD output ports: 32 (segment)
   4 (common)

   *Including 15 ports available for segment outputs
8. Interrupts
   External interrupt: 1
   Internal interrupts: 4
9. Timer/counter: 1 set
10. On-chip 32.768kHz crystal oscillator and clock divider circuit.
11. On-chip system clock oscillator
12. LCD driver circuit
    (128-segment, 1/3 bias, 1/4 duty)
13. Standby function
14. Supply voltage: 2.7 to 5.5V
15. 64-pin QFP (QFP64-P-1420)

Pin Connections



Block Diagram



Symbol description

A, B	: Accumulators	IME	: Interrupt mask enable F/F
ACL	: Auto clear	P1-P3	: Registers
ALU	: Arithmetic logic unit	PL, PU	: Program counters
BR, DS	: Common signal control F/F	PSW	: Program status word register
CG	: clock generator	R4-R7	: General-purpose registers
DIV	: Divider	RD, RE, RF	: Mode registers
D, E, H, L	: General-purpose registers	SB	: Shift register
HC	: Common signal circuit	SP	: Stack pointer
IE	: Interrupt enable F/F	TC	: Count register
IFA, IFB		TM	: Module register
IFS, IFT, IFV	: Interrupt requests		

Pin Description

Symbol	I/O	Circuit type	Function
$P0_0$-$P0_3$	I	Pull up	Acc←$P0_0$-$P0_3$
$P1_0$-$P1_3$	I/O	Pull up	I/O selectable by instructions
$P2_0$-$P2_3$	I/O	Pull up	I/O selectable independently Sound output only when $P2_3$ pin is used as an output
$P3_0$-$P3_3$	I/O	Pull up	Serial interface I/O capacity using RE register
S_0-S_{14}	O or I/O		Selectable between segment ports and I/O ports through an RC register
S_{15}-S_{31}	O		Display RAM contents output as LCD segment signals
H_1-H_4	O		4-value output capability; used for LCD common output
TEST	I	Pull down	For test (Connected to GND normally)
$\overline{\text{RESET}}$	I	Pull up	Auto clear
ϕ	O		System clock output
CK_1, CK_2			For system clock oscillation
OSC_{IN}, OSC_{OUT}			For clock oscillation
V_{DSP} V_{OA}, V_{OB}			Power supply for LCD driver
V_{DC}, GND			Power supply for logic circuit

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply voltage	V_{DD}	−0.3 to +7	V	1
	V_{DSP}	−0.3 to +7	V	
Input voltage	V_{IN}	−0.3 to V_{DD}+0.3	V	1
Output voltage	V_{OUT}	−0.3 to V_{DD}+0.3	V	1
Output current	I_{OUT}	20	mA	2
Operating temperature	Topr	−20 to +70	℃	
Storage temperature	Tstg	−55 to +150	℃	

Note 1: The maximum applicable voltage on any pin with respect to GND.
Note 2: Sum of current from (or flowing into) output pins.

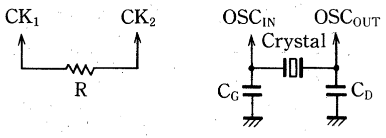
Recommended Operating Conditions

(V_{DD}=2.7 to 5.5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Supply voltage	V_{DD}		2.7		5.5	V	
	V_{DSP}		2.7		V_{DD}	V	
Basic clock oscillator frequency	f		250		600	kHz	1
		V_{DD}=4.5 to 5.5V	250		2,000		
Instruction cycle	t		6.67		16	μs	
		V_{DD}=4.5 to 5.5V	2		16		
Crystal oscillator frequency	f_{OSC}			32.768		kHz	

Note 1: Degree of fluctuation frequency: ±30%

● Oscillation Circuit



C_G=15pF, C_D=22pF

Electrical Characteristics

(V_{DD}=2.7 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}			$0.7V_{DD}$		V_{DD}	V	1
	V_{IL1}			0		$0.3V_{DD}$	V	
	V_{IH2}			$V_{DD}-0.5$		V_{DD}	V	2
	V_{IL2}			0		0.5	V	
Input current	I_{IH}	$V_{IN}=0V$		2		200	μA	1
			V_{DD}=4.5 to 5.5V	20		200		
Output current	I_{OH1}	$V_{OH}=V_{DD}-0.5V$		50			μA	3
	I_{OL1}	$V_{OL}=0.5V$		250			μA	
	I_{OH2}	$V_{OH}=V_{DD}-0.5V$		5		250	μA	4
	I_{OL2}	$V_{OL}=0.5V$		500			μA	
	I_{OH3}	$V_{OH}=V_{DD}-0.5V$		100			μA	5
			V_{DD}=4.5 to 5.5V	400				
	I_{OL3}	$V_{OL}=0.5V$		0.5			mA	
			V_{DD}=4.5 to 5.5V	1.6				
Output impedance	R_C				5	20	kΩ	6
	R_S				10	40	kΩ	7
Output voltage	V_1			2.7		3	V	8
	V_2	$V_{DSP}=3.0V$ No load		1.7	2	2.3	V	
	V_3			0.7	1	1.3	V	
	V_4			0		0.3	V	
Current consumption	I_{OP}	f=600kHz, V_{DD}=3.0V			0.4	1.5	mA	9
	I_{SB}	Standby current	V_{DSP}=3.0V		15	40	μA	10
			V_{DD}=3.0V		8	20		11

Note 1: Applied to pins $P0_0$-$P0_1$, $\overline{RESET}$,
$P1_0$-$P1_3$, $P2_0$-$P2_3$, $P3_0$-$P3_2$ (during input mode).

Note 2: Applied to pins CK_1, Test, OSC_{IN}.

Note 3: Applied to pin CK_2

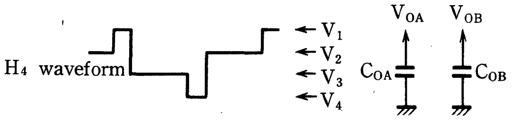
Note 4: Applied to pins $P1_0$-$P1_3$ (during output mode).

Note 5: Applied to pins $P2_0$-$P2_3$, $P3_0$-$P3_2$ (during output mode), ϕ

Note 6: Applied to pins H_1-H_4

Note 7: Applied to pins S_0-S_{31}

Note 8: Applied to pins H_1-H_4, S_0-S_{31}



$C_{OA}=1\mu F$, $C_{OB}=1\mu F$

Note 9: No load condition.

Note 10: No load condition when bleeder resistance is ON.

Note 11: No load condition when bleeder resistance is OFF.

Pin Functions

(1) GND, V_{DD} (Power supply inputs)

Both GND pins 26 and 58 should be grounded.

The V_{DD} pin is the positive power supply with respect to GND.

The V_{DSP} pin is the positive power supply for an LCD driver with respect to GND.

(2) TEST (Test input)

The test pin should be left open or connected to GND with a pull-down resistor.

(3) $\overline{\text{RESET}}$ (Input)

The RESET accepts an active Low system reset which initializes the internal logic of the device. Normally a capacitor of about 1.0 μF is connected between this pin and GND to provide a power on reset function.

(4) OSC_{IN}, OSC_{OUT} (Crystal oscillators)

The OSC_{IN} and OSC_{OUT} pins connect with an external crystal oscillator and these pins and the GND connect with a capacitor, which constitute an oscillator circuit.

The output of the oscillator is coupled to a clock divider for real-time clock operation.

(5) CK_1, CK_2 (System clock CR oscillators)

The CK_1 and CK_2 pins, in conjunction with a resistor between them, provide a system clock oscillator.

(6) H_1-H_4 (Common signal outputs)

The H_1-H_4 pins are used to drive the common of an LCD.

(7) S_0-S_{31} (Segment outputs)

The S_0-S_{31} pins drive LCD segments. Pins S_0 through S_{14} may also be used as I/O ports when specified with the mode register RC.

(8) $P0_0$-$P0_3$ (Inputs)

The P0 pins are normally used to accept key input data. A falling edge at these pins resets the IFB flag.

(9) $P1_0$-$P1_3$ (Input/output)

The P1 are I/O pins connected to the positive supply with pull-up resistors.

They may be switched between Input and Output modes through an instruction.

(10) $P2_0$-$P2_3$ (Input/output)

The $P2_0$-$P2_3$ pins are bit-independent I/O ports which can be independently set to Input or Output mode with the mode register RF.

When the $P2_3$ is used for an output pin, it serves exclusively as a sound output pin, which can output a sound signal with any frequency set up by the timer counter.

Pins $P2_0$ and $P2_1$ output the OD and R/W signals with the mode register RC.

(11) $P3_0$-$P3_2$ (Input/output)

The $P3_0$-$P3_2$ pins are I/O pins which are connected to the positive supply with pull-up resistors.

These pins can be set to I/O mode for use in a serial interface with the mode register RE.

■ Hardware Configuration

(1) ROM and program counter

The on-chip ROM has a configuration of 64 pages×64 steps×8 bits, and stores programs and table data.

The program counter consists of a 6-bit page address counter P_U and a 6-bit binary counter P_L used to specify the steps within a page.

The locations shown in Fig. 1 are allocated in the on-chip ROM.

(2) Stack pointer (SP)

The stack pointer (SP) is an 8-bit shift register which holds the starting address of the stack area of RAM space.

Immediately after the reset, the contents of the stack pointer are uninitialized and must be set to an appropriate value. If, for instance, the initial value of the stack pointer is set to 80_H, the data memory are begining with the highest address (excluding the display RAM area) $7F_H$, is usable as a stack area.

(3) RAM

Data memory has a 160 word×4-bit configuration, and is used to store processing data and other information.

Data memory is also used as a stack area to save register values, the program counter value, and program status word (PSW) at the time a subroutine jump or an interrupt occurs.

Fig. 3 shows the RAM configuration.

2×16×4 bits of entire RAM space is used as a display RAM area from which data is output to LCD segment driving pins. An LCD with a 1/4 duty and 1/3 bias format can be directly driven by writing display data into the display RAM area.

The display RAM outputs are, as shown in Fig. 4, connected to segment output pins S_0-S_{31} for individual set of common outputs H_1-H_4. The segment output pins provide a single digit of display RAM data M_0-M_3, as an LCD driving waveform signal according to H_1-H_4 outputs.

The operations of the display RAM are identical to those of other RAM areas.

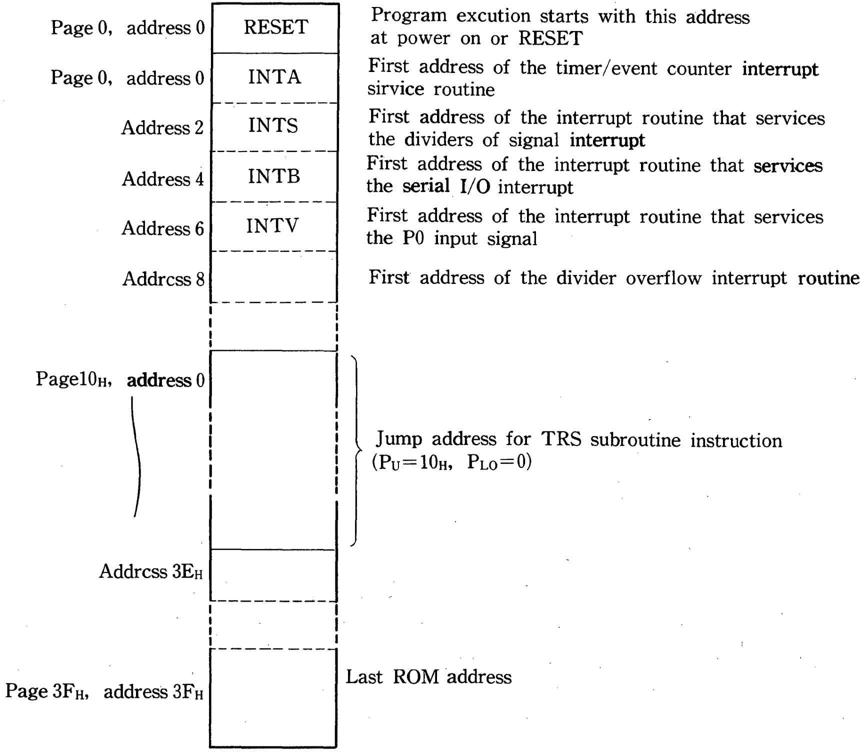

Page 0, address 0
RESET
Program excution starts with this address at power on or RESET
Page 0, address 0
INTA
First address of the timer/event counter interrupt sirvice routine
Address 2
INTS
First address of the interrupt routine that services the dividers of signal interrupt
Address 4
INTB
First address of the interrupt routine that services the serial I/O interrupt
Address 6
INTV
First address of the interrupt routine that services the P0 input signal
Addrcss 8
First address of the divider overflow interrupt routine
Pagel0H, address 0
Jump address for TRS subroutine instruction (PU=10H, PLO=0)
Addrcss 3EH
Page 3FH, address 3FH
Last ROM address

Fig. 1 Program ROM map

L \ H	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
0000									S_0	S_{16}
0001									S_1	S_{17}
0010									S_2	S_{18}
0011									S_3	S_{19}
0100									S_4	S_{20}
0101									S_5	S_{21}
0110									S_6	S_{22}
0111									S_7	S_{23}
1000									S_8	S_{24}
1001									S_9	S_{25}
1010									S_{10}	S_{26}
1011									S_{11}	S_{27}
1100									S_{12}	S_{28}
1101									S_{13}	S_{29}
1110									S_{14}	S_{30}
1111									S_{15}	S_{31}

Note: The area with oblique lines is allocated for a display RAM and the Sn shows the related segment outputs.

Fig. 3 RAM configuration

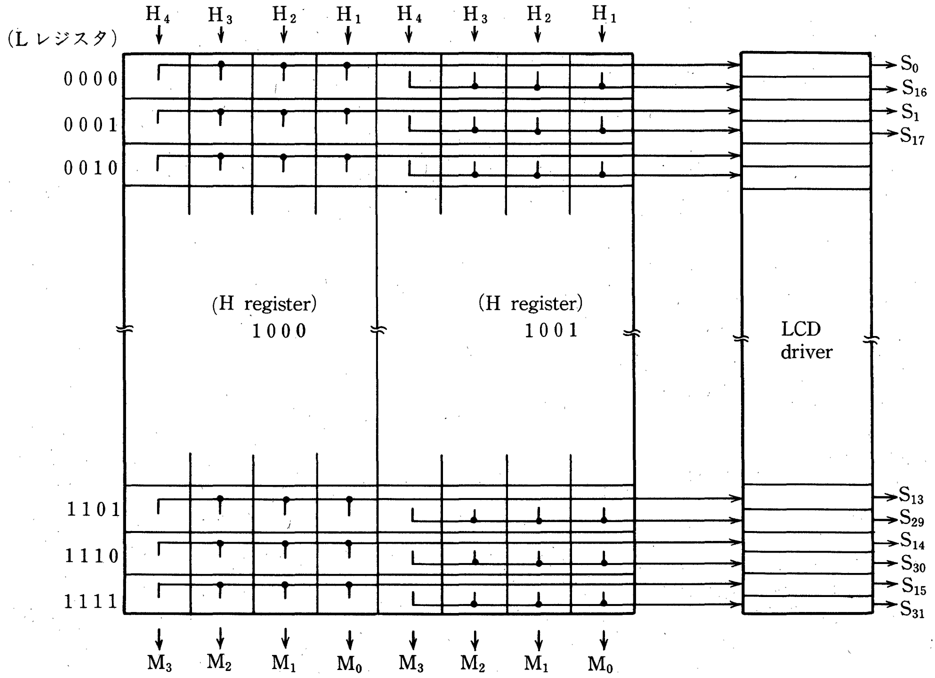


Fig. 4 Display RAM and its LCD segment outputs

(4) Accumulator (A), Subaccumulator (B) and Arithmetic and logic unit (ALU)

The accumulator (A) is a 4-bit working register which is the nucleus of the single chip system. It holds the results of operations and transfers data to memory, I/O ports, and registers.

A sub-accumulator (B) is another 4-bit register. It is used as one of general-purpose registers, and when combined with the Acc to form a B-A register pair, allows data transfer on a byte-by-byte (8-bit) basis.

The arithmetic and logic unit (ALU) performs, in conjunction with a carry flag (C), binary addition, shift operations, and logical operations such as AND, OR, EX-OR, and complement.

(5) General-purpose registers (H, L, D, E)

Registers H and L are 4-bit general-purpose registers. They can transfer and compare data with the Acc on 4-bit basis. Registers D and E are 4-bit registers and can transfer data with the H and L registers on an 8-bit basis.

The H and L as well as the D and E registers can be combined into 8-bit register pairs, and can be used as pointers to data memory locations.

The L register can be incremented or decremented and is used to access I/O ports and mode registers.

(6) Clock divider, IFV flag, IFA flag

The device contians a crystal oscillator and a 15-stage divider, as shown in Fig. 5. A real-time

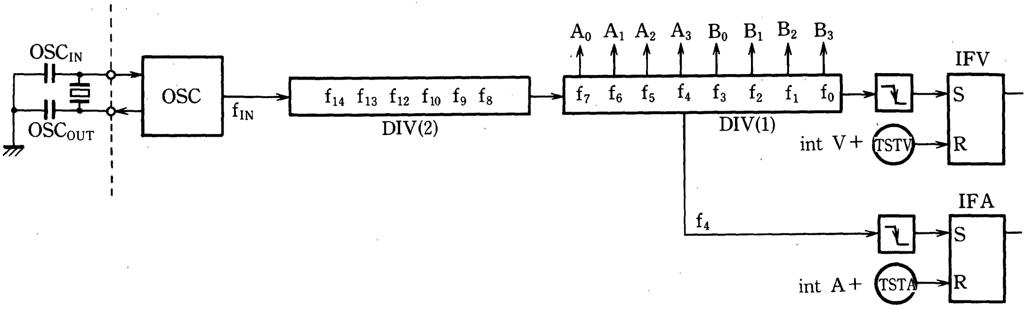


Fig. 5 Real-time clock and divider

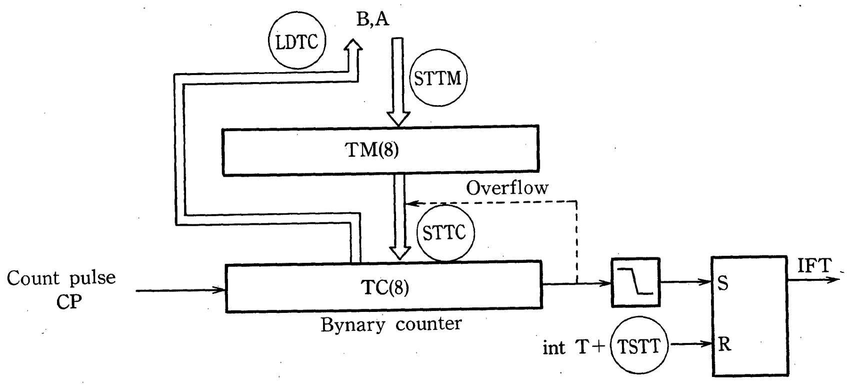


Fig. 6 Timer/event counter

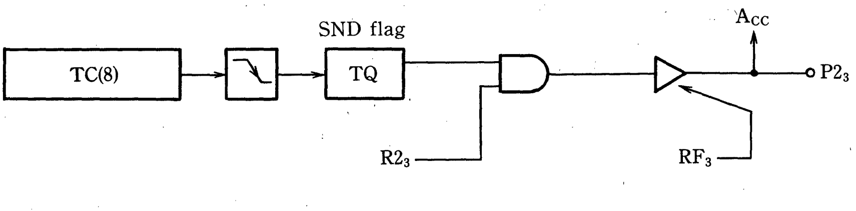


Fig. 7 SND signal

clock can be provided by connecting an external crystal oscillator between the oscillator pins.

When an external 32.768kHz crystal oscillator is used, the f_0 signal is set at a frequency of 1Hz.

(7) Timer/event counter and the SND signal

The timer/event counter consists of an 8-bit count register (TC) and an 8-bit mudulo register (TM).

The count register is an 8-bit incremental binary counter. It is incremented by one at the falling edge of its count pulse (CP) input. If the count register overflows, the timer interrupt request flag IFT is set, and the contents of the modulo register (TM) are loaded into the count register (See Fig. 6).

The count pulse CP can be selected from divider signals f_{IN}, f_8 and f_0, and the system clock, by using the mode register RD.

If the count register (TC) overflows, the SND flag reverses its status at the falling edge of the TC output. A sound signal can be obtained at the TC output by putting P2 in output mode and sending a "1" to pin $P2_3$ (See Fig. 7).

(8) Serial interface and IFS

The serial interface consists of an 8-bit shift register (SB) and a 3-bit counter, which is used to input and output the serial data.

The serial clock can be selected with either an internal clock (system clock) or an external clock.

In Serial shift operations, the highest bit data of the shift register (SB) is output from the SO pin, and the data input from the SI pin at the rising edge of a serial clock is loaded into the lowest bit of the shift register.

When the internal clock is used, immediately after the SIO instruction is executed, the serial op-

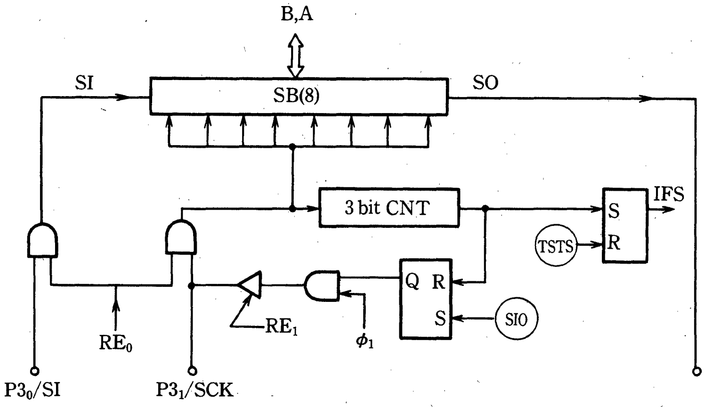
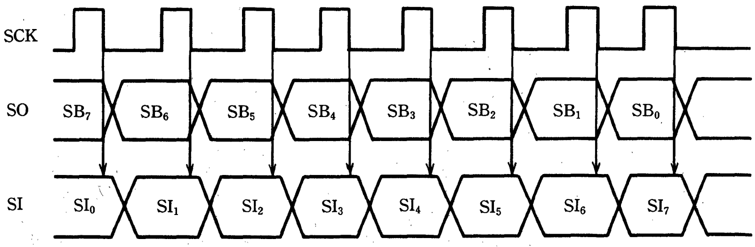


Fig. 8 Serial interface



(注) SO : Serial data output
SI : Serial data input

Note: SO: Serial data output
SI: Serial data input

Fig. 9 Serial I/O timing

eration begins and stops with 8 clocks which are output from the SCK pin.

Upon completion of an 8-bit shift operation, the serial I/O ending flag IFS is set each time a 3-bit counter overflows, and an interrupt request occurs.

(9) Input port P0 and IFB flag

The IFB flag is set at the falling edge of the signal applied to the input port P0 by which the interrupt is enabled.

When port P0 is used as a key input, it can cause an interrupt each time a key is operated.

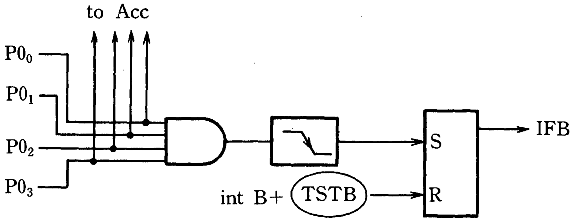


Fig. 10 P0 port

(10) Interrupts

When an interrupt occurs, the corresponding interrupt request flag is set. The CPU acknowledges the interrupt if it is enabled (Master interrupt enable flag and the corresponding interrupt enable flag are set). If more than one interrupt occurs simultaneously, all of the corresponding interrupt request flags will be set, but the CPU will only ac knowledge that interrupt with the highest priority and other interrupts will be queued.

(11) I/O ports

Port P0 is a 4 bit parallel input port. The IFB flag is set at the falling edge of this port.

Port P1 can be switched between input and output modes, 4 bits at a time.

Each bit of port P2 can be independently placed in input or output mode by setting the corresponding bit of mode register RF.

Ports $P2_0$ and $P2_1$ can output the OD and R/W signals, respectively. In those cases, these pins should be kept High in an output mode. Port $P2_3$ outputs the SND signal in the output mode.

Port P3 is a 4-bit I/O port which can be placed in input or output mode, 3 bits at a time. Each bit of port P3 can be set to the I/O modes (SI, SO, SCK) of a serial interface.

Ports P1 and P3 are placed in an output mode when a port output instruction is executed, and in an input mode when a port input instruction is executed. After an ACL operation, ports P1, P2 and P3 are all placed in an input mode.

Every input port has pull-up resistors. (Pull-up resistors for I/O ports are effective only when the ports are placed in an input mode.)

Ports P1 through P3 in an output mode can be independently set or reset by instructions.

When a key-matrix is configured by using I/O ports, if the short on output pins may occur caused by a multiple key depression, port P1 should be used as an output.

(12) Standby mode

Executing the CEND instruction places the de-

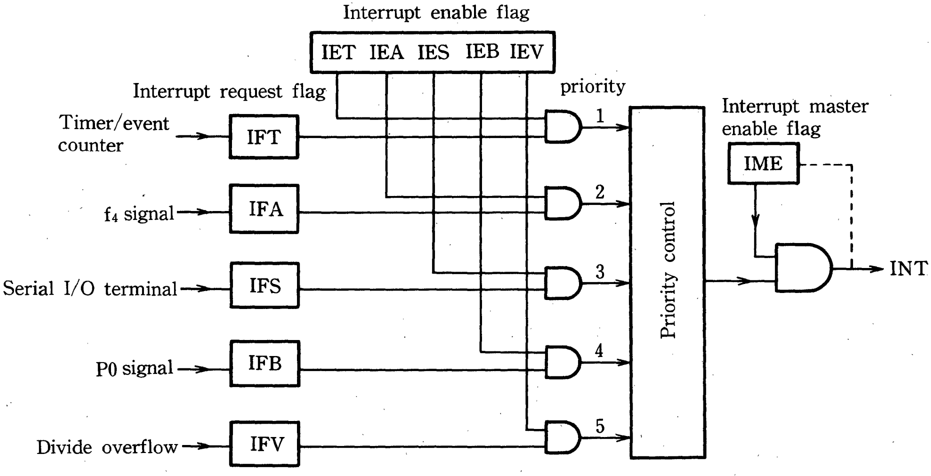


Fig. 11 Interrupt handling

Table 1 Characteristics of I/O ports and registers

Port	Function	Direct 4-bit parallel I/O		IN, OUT instruction		Bit independent output SPn
		Input (INA)	Output (OUTA)	Input (IN)	Output (OUT)	Direct pin-independent output RPn
P0	Input-only port	○	×	○	×	×
P1	I/O port	○	○	×	○	○
P2	I/O port, P_{23}-sound output	○	○	×	×	○
P3	$P3_0$-SI, $P3_1$-SCK, $P3_2$-SO_1, multi-control port	○	○	×	×	○

○: Yes, ×: No

vice in standby mode. To reduce power consumption, the system clock is inactivated.

Standby mode may be cleared with the Interrupt request or the RESET signal.

(13) Reset function (ACL)

Applying a Low level signal to the RESET pin resets the internal logic of the device and applying a High level signal starts execution of the program at address 0, page 0.

Once the device is reset, all I/O ports are placed in input mode to disable all interrupts, and the LCD screen is cleared.

The device is also reset when it is powered up.

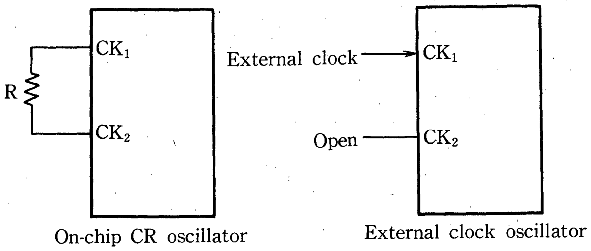


Fig. 12 Master clock sources

(14) Master clock oscillator circuit

The master clock oscillator requires an external resistor across pins CK_1 and CK_2. Instead of using on-chip oscillator, an external clock may be applied to pin CK_1. In this case, pin CK_2 should be left open.

The system clock ϕ is a divided clock equivalent to 1/4 of the clock applied to pin CK_1.

(15) LCD driver

- **Display segment**

The SM563 contains an on-chip LCD driver which can directly drive an LCD with a 1/4 duty and 1/3 bias scheme. Fig. 13 shows an example of LCD segment configuration for 1/4 duty.

Each segment of the LCD can be turned on or off by software control of the setting of the corresponding bit "1" or "0" in the display RAM area (see Fig. 3).

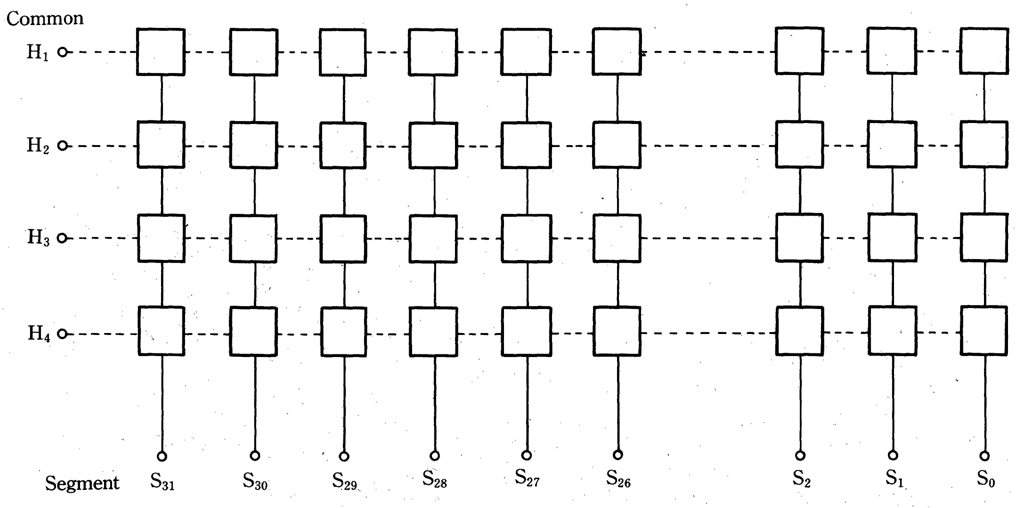


Fig. 13 LCD segment configuration for 1/4 duty

The LCD digit may have any shape, provided that the maximum number of segments does not exceed 128 (see Fig. 13). Fig. 14 shows an example of a seven-segment numeric LCD digit.

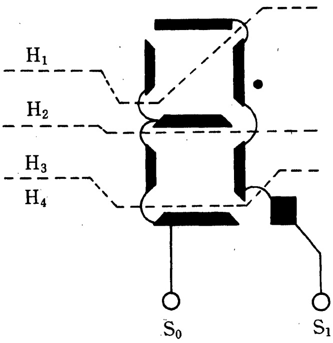


Fig. 14 7-segment numeric LCD digit

- **LCD driving signal waveform**

Fig. 15 shows the LCD signal driving waveforms required to display the number "5" on the 7-segment display shown in Fig. 14 (segment outputs S_0 and S_1 are used). A voltage of 3V is applied to pin V_{DSP} in the Fig. 15.

The frame frequency (I/T) can be selected between 64Hz and 128Hz by mask options.

- **V_{OA} and V_{OB} pins**

The device contains bleeder resistors to allow 1/3 bias driving. When V_{DSP} is 3V, voltages of 2V and 1V are applied to pins V_{OA} and V_{OB} respectively.

Normally pins V_{OA} and V_{OB} are left open. When an LCD with a large display area is driven, connect capacitors across pins V_{OA} and V_{DSP} and across V_{OB} and V_{DSP} to improve the rise time of the LCD driving signal.

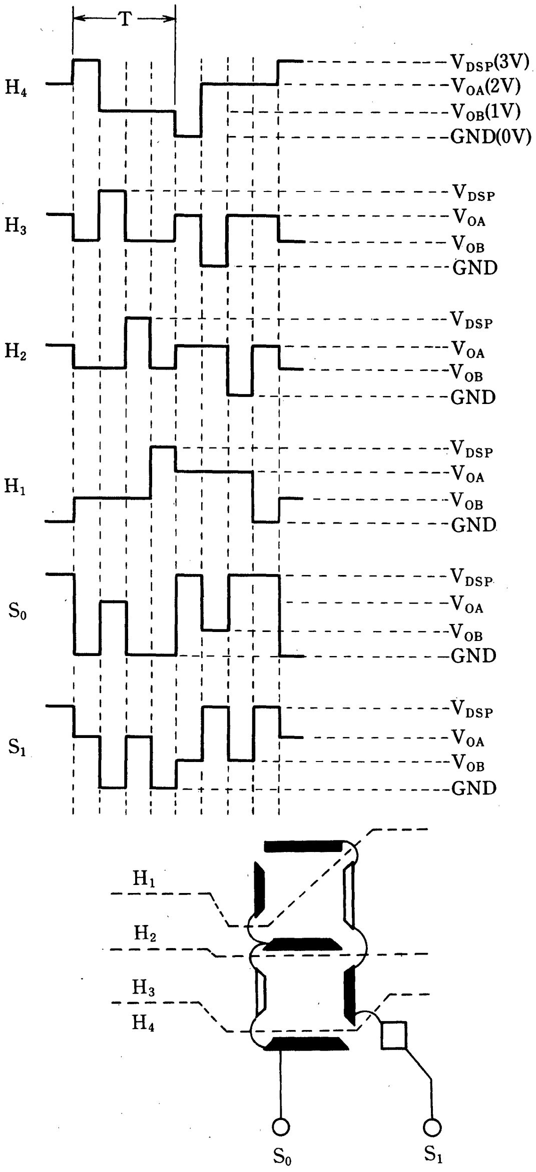


Fig. 15 LCD driving signal waveform (reguired to display the number 5)

■ Instruction Set

(1) ROM address instructions

Mnemonic	Machine code	Operation
TR x	80-BF	PL←x (I_5-I_0)

TL xy
(2-byte)
	E0-EF
	00-FF
		PU←x (I_{11}-I_4)
		PL←y (I_5-I_0)

TRS x	C0-DF	(SP−2), (SP−3), (SP−4)←PC
		SP←SP−4
		PU←$(10000)_2$
		PL←x ($I_4I_3I_2I_1I_0O$)

CALL xy
(2-byte)
	F0-FF
	00-FF
		(SP−2), (SP−3), (SP−4)←PC
		SP←SP−4, PU←x (I_{11}-I_6),
		PL←y (I_5-I_0)

JBA x (2-byte)	7F 30-3F	PU_5-PU_2←x (I_3-I_0), PU_1, PU_0, PL_5, PL_4←B, PL_3-PL_0←A
RTN	61	PU, PL←(SP), (SP+1), (SP+2)
RTNS	62	PU, PL←(SP), (SP+1), (SP+2), SP←SP+4
RTNI	63	PU, PL←(SP), (SP+1), (SP+2), PSW←(SP+3), SP←SP+4, IME←1

(2) RAM address instructions

Mnemonic	Machine code	Operation
STL	69	L←A
STH	68	H←A
EXHD	3F	H↔D L↔E
LIHL xy (2-byte)	3D 00-FF	H←x (I_7-I_4), L←y (I_3-I_0)

(3) Data transfer instructions

Mnemonic	Machine code	Operation
EX pr	5C-5F	A↔(pr)
LDX adr (2-byte)	7D 00-FF	A←(adr)
STX adr (2-byte)	7E 00-FF	(adr)←A
EXX adr (2-byte)	7C 00-FF	A↔(adr)
LAX x	10-1F	A←x (I_3-I_0)

LIBA xy
(2-byte)
	3C
	00-FF
		B←x (I_7-I_4)
		A←y (I_3-I_0)

LBAT	60	B←ROM (Pu_5-Pu_2, B, $A)_H$
		A←ROM (Pu_5-Pu_2, B, $A)_L$

LDL	65	A←L
LD pr	54-57	A←(pr)
ST pr	58-5B	(pr)←A
EXH	6C	A↔H
EXL	6D	A↔L
EXB	6E	A↔B
STB	6A	B←A
LDB	66	A←B
LDH	64	A←H
PSHBA	28	(SP−1)←B, (SP−2)←A, SP←SP−2
PSHHL	29	(SP−1)←H, (SP−2)←L, SP←SP−2

POPBA	38	B←(SP+1), A←(SP),
		SP←SP+2

POPHL	39	H←(SP+1), L←(SP)
		SP←SP+2

STSB	70	SB_H←B, SB_L←A
STSP	71	SP_H←B, SP_L←A
STTC	72	TC←TM
STTM	73	TM_H←B, TM_L←A
LDSB	74	B←SB_H, A←SB_L
LDSP	75	B←SP_H, A←SP_L
LDTC	76	B←TC_H, A←TC_L
LDDIV	77	B←DIV_H, A←DIV_L

(4) Arithmetic instructions

Mnemonic	Machine code	Operation
ADX x	00-0F	A←A+x (I_3-I_0), Skip if C_Y=1
ADD	36	A←A+(HL)
ADDC	37	A←A+(HL)+C, C←C_Y Skip iF C_Y=1
OR	31	A←A ∨ (HL)
AND	32	A←A ∧ (HL)
EOR	33	A←A⊕(HL)
ANDB	22	A←A ∧ B
ORB	21	A←A ∨ B
EORB	23	A←A⊕B
COMA	6F	A←$\bar{A}$
ROTR	25	C→A_3→A_2→A_1→A_0→C
ROTL	35	C←A_3←A_2←A_1←A_0←C
INCB	52	B←B+1, Skip if B=F_H
DECB	53	B←B−1, Skip if B=0
INCL	50	L←L+1, Skip if L=F_H
DECL	51	L←L−1, Skip if L=0
DECM adr	79 00-FF	(adr)←(adr)−1, Skip if (adr)=0
INCM adr	78 00-FF	(adr)←(adr)+1, Skip if (adr)=F_H

(5) Test instructions

Mnemonic	Machine code	Operation
TAM	30	Skip if A=(HL)
TAH	24	Skip if A=H
TAL	34	Skip if A=L
TAB	20	Skip if A=B
TC	2A	Skip if C=0
TM x	48-4B	Skip if (HL) x=1
TA x	4C-4F	Skip if Ax=1
TSTT	2B	Skip if IFT=1, IFT←0
TSTA	2C	Skip if IFA=1, IFA←0
TSTS	2D	Skip if IFS=1, IFS←0
TSTB	2E	Skip if IFB=1, IFB←0
TSTV	2F	Skip if IFV=1, IFB←0

(6) Bit manipulation instructions

Mnemonic	Machine code	Operation
SM x	40-43	(HL) x←1
RM x	44-47	(HL) x←0
RC	26	C←0
SC	27	C←1
RIME	3A	IME←0
SIME	3B	IME←1
DI x (2-byte)	7F C0-DF	IEF←IEF ∧ x
EI x (2-byte)	7F E0-FF	IEF ∨ x

(7) I/O instructions

Mnemonic	Machine code	Operation
IN	67	A←P0
OUT	6B	P1←A
INA x (2-byte)	7F A0-A9	A←P (x), R (x)
OUTA x (2-byte)	7F B0-BF	P (x), R (x)←A

INBA x	7F
	80-81
		B←R (x+1)
		A←R (x)

OUTBA x (2-byte)	7F 90-91	R (x+1)←B R (x)←A
SP xy (2-byte)	7A 00-F3	P (y)←P (y) x
BP xy (2-byte)	7B 00-F3	P (y)←P (y) x
RDS (2-byte)	7F 60	DS←0
RBR (2-byte)	7F 70	BR←0
SDS (2-byte)	7F 61	DS←1
SBR (2-byte)	7F 71	BR←0
READ (2-byte)	7F 62	A←P4 with $\boxed{\text{OD}}$
WRIT (2-byte)	7F 72	P4←A with $\boxed{\text{R/W}}$

READB
(2-byte)
	7F
	63
		A←P4, with $\boxed{\text{OD}}$
		B←P5

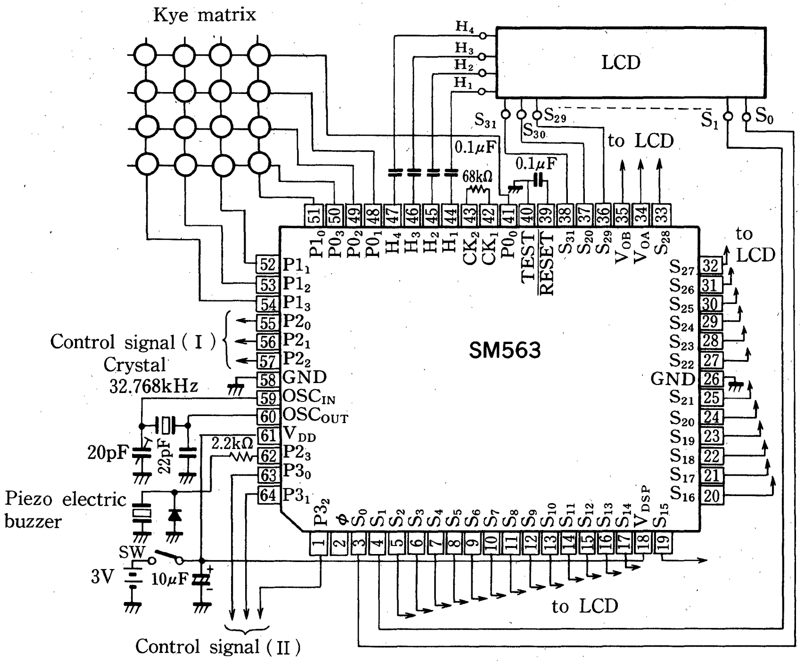
WRITB
(2-byte)
	7F
	73
		P4←A, with $\boxed{\text{R/W}}$
		P5←B

(8) Special instructions

Mnemonic	Machine code	Operation
SIO	3E	Serial I/O start
IDIV (2-byte)	7F 10	DIV←0
SKIP	00	No operation
CEND (2-byte)	7F 00	System clock stop

*The machine code consists of 8-bits of I_7, I_6, I_5, I_4, I_3, I_2, I_1 and I_0.

■ System Configuration Example



8-bit Single-chip Microcomputers

LH0801/LH0801A LH0811/LH0811A

Z8 Microcomputer Unit

■ Description

The LH0801/A, LH0811/A are 8-bit single chip microcomputers (Z8) which have 2K bytes and 4K bytes of ROM respectively.

The Z8 offers faster execution; more efficient use of memory, more sophisticated interrupt, input/output and bit-manipulation capabilities, and easier system expansion.

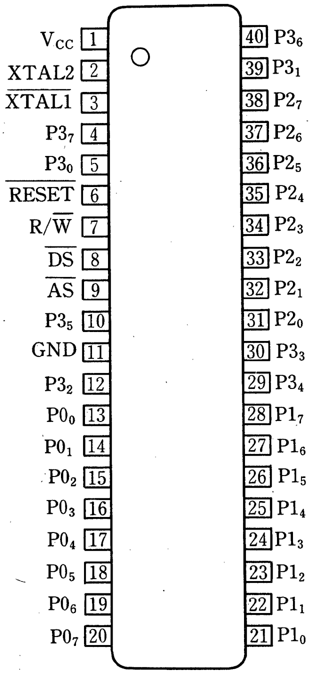
Under program control, the Z8 can be tailored to the needs of its user. It can be configured as a stand-alone microcomputer with 2K bytes for the LH0801/A or 4K bytes for the LH0811/A of internal ROM, a traditional microprocessor that manages up to 124 bytes for the LH0801/A or 120 bytes for the LH0811/A of external memory, or a parallel processing device in a system with other processors and peripheral controllers linked by the Z-BUS. In all configurations, a large number of pins remain available for I/O.

■ Features

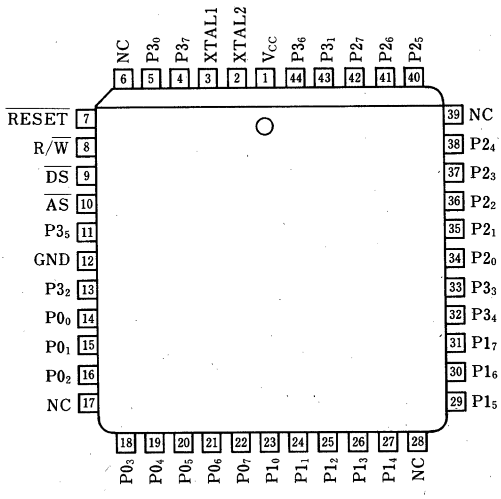
1. Complete single-chip microcomputer with internal ROM, RAM and I/O
   RAM capacity: 124 bytes
   ROM capacity: 2K bytes (LH0801/A)
   4K bytes (LH0811/A)
   I/O ports: 32
2. On-chip two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler
3. Full-duplex UART
4. 144 byte register file
5. Register pointer so that short, fast instructions can access any working register groups
6. Vectored, priority interrupts for I/O, counter/timers, and UART
7. Up to 62K bytes for the LH0801/A or 60K bytes for the LH0811/A addressable external space each for program and data memory
8. On-chip oscillator
9. Maximum clock frequency
   8MHz (internal 4MHz): LH0801/LH0811
   12MHz (internal 6MHz) :LH0801A/LH0811A

■ Pin Conections

LH0801/LH0801A, LH0811/LH0811A



LH0801U/UH0801AU, LH0811U/LH0811AU



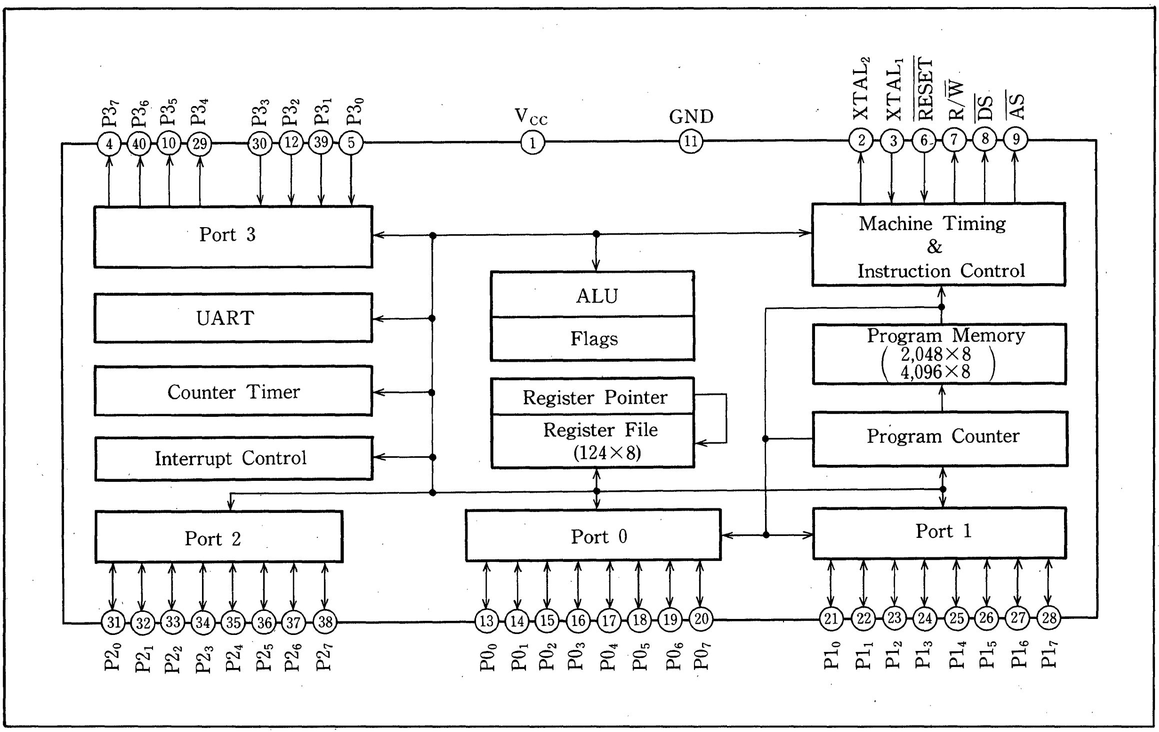
10. High speed instruction execution (8MHz/12MHz)
    Working register execution time: 1.5 μs/1.0 μs
    Average instruction execution time: 2.2 μs/1.5 μs
    Maximum instruction execution time: 5.0 μs/3.3 μs
11. Low power standby option which retains contents of general-purpose registers
12. Single +5V power supply
13. All pins are TTL compatible
14. 40-pin DIP (DIP40-P-600):
    LH0801/A, LH0811/A
    44-pin QFJ (QFJ44-P-S650):
    LH0801U/AU, LH0811U/AU

Ordering Information

LH08XX X X

- Package
  Blank: 40-pin DIP (DIP40-P-600)
  U: 44-pin QFJ (QFJ44-P-S650)
- Clock frequency
  Blank: 8MHz
  A: 12MHz
- Model No.
  LH0801 (On-chip 2K byte ROM)
  LH0811 (On-chip 4K byte ROM)

Block Diagram



(Note) Pin numbers apply to 40-pin DIP.

■ Pin Description

Pin	Meaning	I/O	Function
$P0_0$-$P0_7$	Port 0	I/O	8-bit I/O port, programmable for I/O.
$P1_0$-$P1_7$	Port 1	I/O	Programmable for I/O in bytes.
$P2_0$-$P2_7$	Port 2	I/O	Programmable for I/O in bits.
$P3_0$-$P3_7$	Port 3	I/O	$P3_0$-$P3_3$ for input, $P3_4$-$P3_7$ for output.
$\overline{AS}$	Address Strobe	O	Active "Low", activated for external address memory transfer.
$\overline{DS}$	Data Strobe	O	Active "Low", activated for external data memory transfer.
$R/\overline{W}$	Read/Write	O	Read at "High", Write at "Low".
$\overline{RESET}$	Reset	I	Active "Low", Initializes.
XTALI	Clock 1	I	Clock terminal pin.
XTAL2	Clock 2	O	Clock terminal pin.

■ Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	−0.3 to +7	V	1
Output voltage	V_{OUT}	−0.3 to +7	V	
Operating temperature	Topr	0 to +70	℃	
Storage temperature	Tstg	−65 to +150	℃	

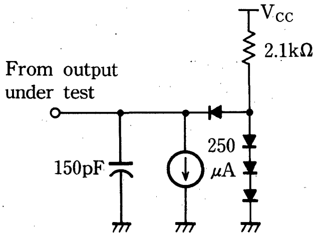
Note 1: The maximum applicable voltage on any pin with respect to GND.

■ DC Characteristics

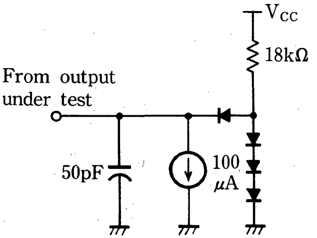
(V_{CC}=5V±5%, Ta=0 to +70℃)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit	Note
Clock input high voltage	V_{CH}	Driven by external clock oscillator	3.8		V_{CC}	V	
Clock input low voltage	V_{CL}	Driven by external clock oscillator	−0.3		0.8	V	
Input high voltage	V_{IH}		2.0		V_{CC}	V	
Input low volltage	V_{IL}		−0.3		0.8	V	
Reset input high voltage	V_{RH}		3.8		V_{CC}	V	
Reset input low voltage	V_{RL}		−0.3		0.8	V	
Output high voltage	V_{OH}	$I_{OH}=-250\mu A$	2.4			V	1
Output low voltage	V_{OL}	$I_{OL}=+2.0mA$			0.4	V	1
Input leakage current	I_{IL}	$0V \leqq V_{IN} \leqq +5.25V$	−10		10	μA	
Output leakage current	I_{OL}	$0V \leqq V_{IN} \leqq +5.25V$	−10		10	μA	
Reset input current	I_{IR}	$V_{CC}=5.25V$, $V_{RL}=0V$			−50	μA	
Supply current	I_{CC}				180	mA	
Back-up power supply	V_{MM}	Power down	3		V_{CC}	V	

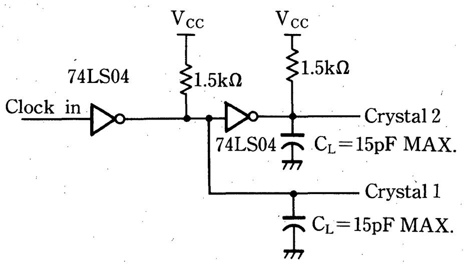
Note 1: $I_{OH}=-100\mu A$ and $I_{OL}=1.0mA$ as to A_0-A_{11}, $\overline{MDS}$, $\overline{SYNC}$, SCLK and IACK in LH0802 / A



Test load 1



Test load 2

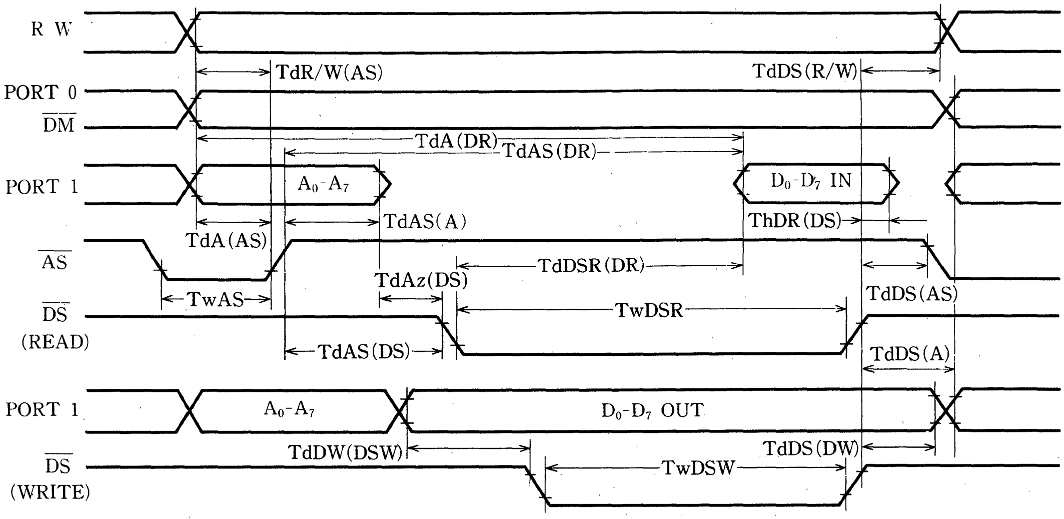


External clock generator circuit

■ External I/O or Mermory Read/Write [Note1, 2] (V_{CC}=5V±5%, Ta=0 to +70℃)

Parameter		Symbol	8MHz		12MHz		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
Address valid to AS ↑ delay		TdA (AS)	50		35		ns	3, 4
$\overline{AS}$ ↑ to address float dalay		TdAS (A)	70		45		ns	3, 4
$\overline{AS}$ ↑ to input data required valid delay		TdAS (DR)		360		220	ns	3, 4, 5
$\overline{AS}$ ↑ low width		TwAS	80		55		ns	3, 4
Address float to $\overline{DS}$ ↓ delay		TdAz (DS)	0		0		ns	
$\overline{DS}$ low width	Read	TwDSR	250		185		ns	3, 4, 5
	Write	TwDSW	160		110			3, 4, 5
$\overline{DS}$ ↓ to input data required valid		TdDSR (DR)		200		130	ns	3, 4, 5
Input data hold time		ThDSR (DS)	0		0		ns	
$\overline{DS}$ ↑ to address active delay		TdDS (A)	70		45		ns	3, 4
$\overline{DS}$ ↑ to $\overline{AS}$ ↓ delay		TdDS (AS)	70		55		ns	3, 4
Read valid to $\overline{AS}$ ↑ delay		TdR/W (AS)	50		30		ns	3, 4
DS ↑ to read not valid		TdDS (R/W)	60		35		ns	3, 4
Output data valid to $\overline{DS}$ ↓ dalay		TdDW (DSW)	50		35		ns	3, 4
$\overline{DS}$ ↑ to output data not valid delay		TdDS (DW)	70		45		ns	3, 4
Write valid to $\overline{AS}$ ↑ dalay		TdA (DR)		410		255	ns	3, 4, 5
$\overline{DS}$ to write not valid delay		TdAS (DS)	80		55		ns	3, 4

Note 1: All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
Note 2: Test load 1
Note 3: The timing is defined at the minimum cycle of TpC.
Note 4: See "Clock Cycle Time Dependency" described later.
Note 5: Apply double cycle of input clock TpC for the expansion memory timing.

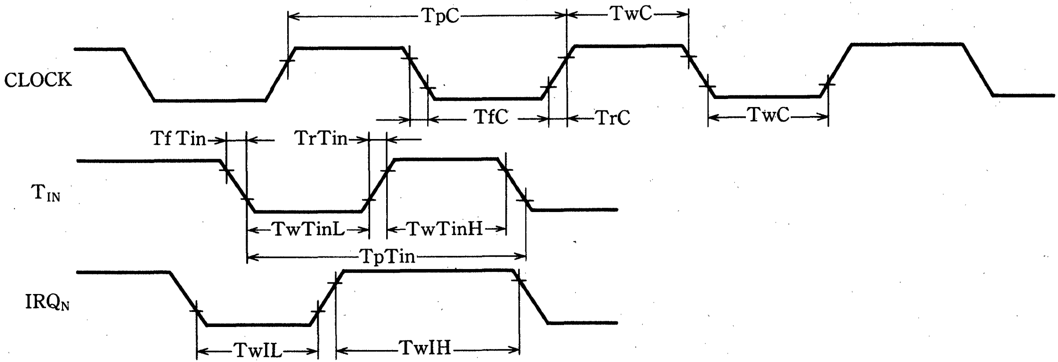


Input Clock, Timer Input, Interrupt Request Input

($V_{CC}=+5V\pm5\%$, Ta=0 to +70℃)

Parameter	Symbol	8MHz		12MHz		Unit	Note
		MIN.	MAX.	MIN.	MAX.		
Input clock cycle	TpC	125	1000	83	1000	ns	1
Input clock rise, fall time	TrC.TfC		25		15	ns	1
Input clock width	TwC	37		26		ns	1
Timer input low width	TwTinL	100		70		ns	2
Timer input high width	TwTinH	3TpC		3TpC		ns	2
Timer input cycle	TpTin	8TpC		8TpC		ns	2
Timer input rise, fall time	TrTin. TfTin		100		100	ns	2
Interrupt request input low time	TwIL	100		70		ns	2, 3
		3TpC		3TpC		ns	2, 4
Interrupt request input high time	TwIH	3TpC		3TpC		ns	2, 3

Note 1: The clock timing references use 3.8V for a logic "1" and 0.8V for a logic "0".
Note 2: The timing references use 0.2V for a logic "1" and 0.8V for a logic "0".
Note 3: Interrupt request from port 3 ($P3_1$-$P3_3$).
Note 4: Interrupt request from port 3 ($P3_0$).

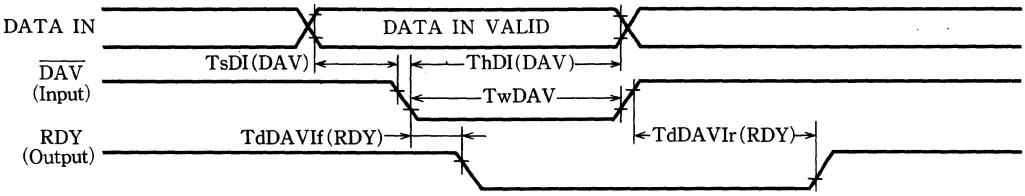


Handshake Timing (Note 4)

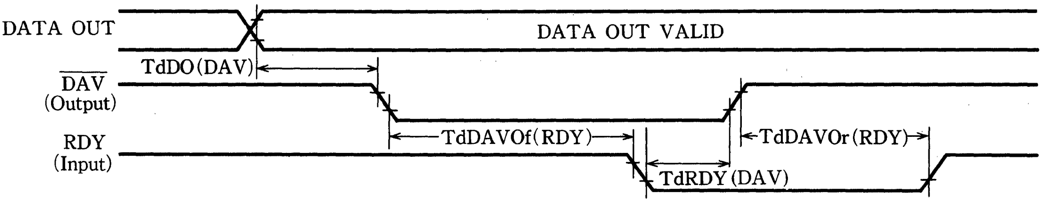
($V_{CC}=5V\pm5\%$, Ta=0 to +70℃)

Parameter	Symbol	8MHz		12MHz		Unit	Note
		MIN.	MAX.	MIN.	MAX.		
Data input setup time	TsDI(DAV)	0		0		ns	
Data input hold time	TsDI (DAV)	230		160		ns	
Data valid signal input width	TwDAV	175		120		ns	
$\overline{\text{DAV}}$ ↓ input to RDY ↓ delay time	TdDAVIf (RDY)		175		120	ns	1, 2
$\overline{\text{DAV}}$ ↓ output to RDY ↓ delay time	TdDAVOf (RDY)	0		0		ns	1, 3
$\overline{\text{DAV}}$ ↑ input to RDY ↑ delay time	TdDAVIr (RDY)		175		120	ns	1, 2
$\overline{\text{DAV}}$ ↑ output to RDY ↑ delay time	TdDAVOr (RDY)	0		0		ns	1, 3
Data output to $\overline{\text{DAV}}$ ↓ delay time	TdDO (DAV)	50		30		ns	1
RDY ↓ input to $\overline{\text{DAV}}$ ↑ delay time	TdRDY (DAV)	0	200	0	140	ns	1

Note 1: Test load 1.
Note 2: Input handshake
Note 3: Output handshake
Note 4: All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".



Input handshake



Output handshake

■ Clock Cycle Time Dependency

(V_{CC}=5V±5%, Ta=0 to +70℃)

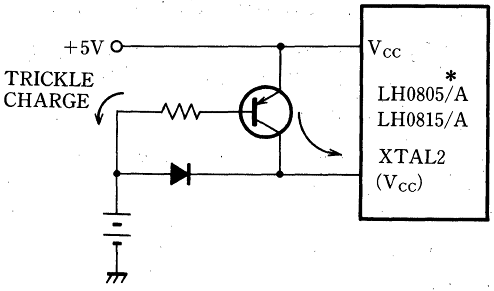
Symbol	8MHz	12MHz	Unit	Note
TdA (AS)	TpC-75	TpC-50	ns	
TdAS (A)	TpC-55	TpC-40	ns	
TdAS (DR)	4TpC-140	4TpC-110	ns	1
TwAS	TpC-45	TpC-30	ns	
TwDSR	3TpC-125	3TpC-65	ns	1
TwDSW	2TpC-90	2TpC-55	ns	1
TdDSR (DR)	3TpC-175	3TpC-120	ns	1
Td (DS) A	TpC-55	TpC-40	ns	
TdDS (AS)	TpC-55	TpC-30	ns	
TdR/W (AS)	TpC-75	TpC-55	ns	
TdDS(R/W)	TpC-65	TpC-50	ns	
TdDW (DSW)	TpC-75	TpC-50	ns	
TdDS (DW)	TpC-55	TpC-40	ns	
TdA (DR)	5TpC-215	5TpC-160	ns	1
TdAS (DS)	TpC-45	TpC-30	ns	

Note 1: Apply double cycle of input clock TpC for the expansion memory timing.

Power Down Standby Option

In low power standby mode, power dissipation can be reduced with retaining the contents of a 124 byte general-purpose register. Use the XTAL2 pin as the V_{MM} power supply input with a bonding option to enter this mode.

Then, an external clock must be input in place of a crystal oscillator through the XTAL1 pin. An appropriate status must be saved in the register file with a software control prior to power reduction caused by a power down function or a lack of power. Fig. 1 shows the example of a power supply circuit with a battery back-up.



* The model numbers with a bonding option are changed from the LH0801/A and LH0811/A to the LH0805/A and LH0815/A respectively.

Fig. 1 Recommended driver circuit for power down option

Architecture

(1) Address Spaces

(i) Program Memory The 16-bit program counter addresses 64K bytes of program memory space. Program memory can be located in two areas : one internal and the other external (Fig. 2). The first 2048 bytes consist of on-chip mask-programmed ROM. At addresses 2048 and greater, the Z8 executes external program memory fetches.

The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts.

(ii) Data Memory The Z8 can address 62K bytes of external data memory beginning at location 2048 (Fig. 3). External data memory may be include with or separated from the external program memory space. DM, an optical I/O function that can be programmed to appear on pin $P3_4$, is used to distinguish between data and program memory space.

(iii) Register File The 144-byte register file includes four I/O port registers (R0-R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255). These registers are assigned the address locations shown in Fig. 4.

Z8 instructions can access registers directly or indirectly with an 8-bit address field. The Z8 also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In the 4-bit mode, the register file is divided into nine working-register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

(iv) Stacks Either the internal register file or the external data memory can be used for the stack. A16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory between locations 2048 and 65535. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4-R127).

(2) I/O ports

The Z8 has 32 lines dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable as input, out-

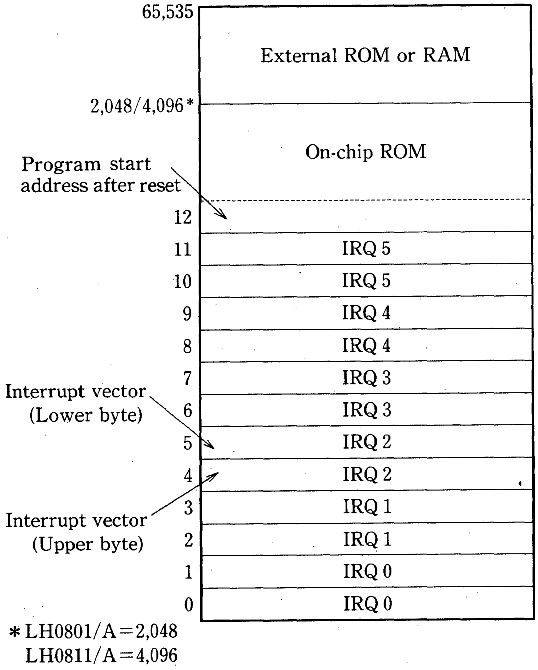


Fig. 2 Program memory map

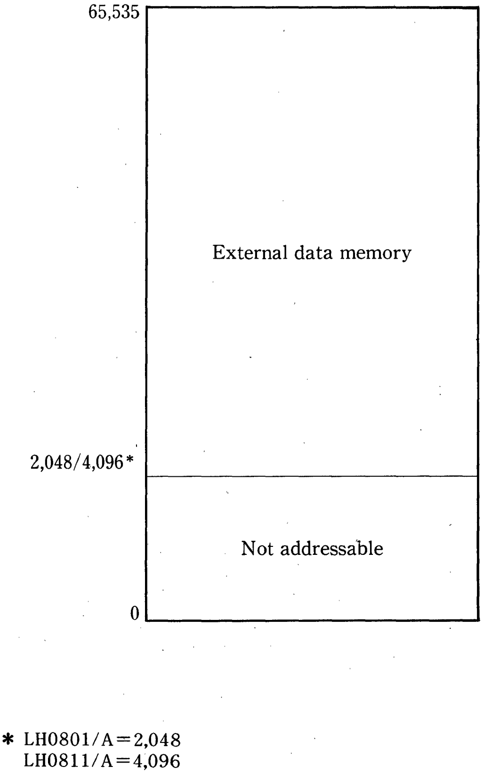


Fig. 3 Data memory map

LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7-0)	SPL
254	STACK POINTER (BITS 15-8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT REQUEST REGISTER	IRQ
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P01M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	TO PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T 1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
	NOT IMPLEMENTED	
127 4	GENERAL-PURPOSE REGISTERS	
3	PORT 3	P3
2	PORT 2	P2
1	PORT 1	P1
0	PORT 0	P0

Fig. 4 The register file

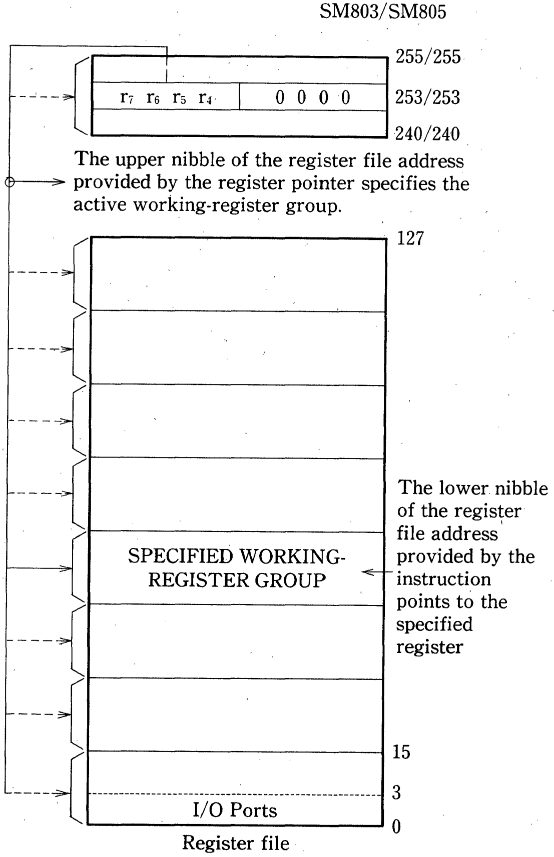

SM803/SM805
255/255
r_7 r_6 r_5 r_4
0 0 0 0
253/253
240/240
The upper nibble of the register file address provided by the register pointer specifies the active working-register group.
127
The lower nibble of the register file address provided by the instruction points to the specified register
SPECIFIED WORKING-REGISTER GROUP
15
3
I/O Ports
0
Register file

Fig. 5 The register pointer

put or address/data. Under software control, the ports can be programmed to provide address outputs, timing, status signals, serial I/O, and parallel I/O with or without handshake. All ports have active pull-ups and pull-downs compatible with TTL loads.

(i) Port 1 can be programmed as a byte I/O port or an address/data port for interfacing external memory.

Memory locations greater than 2048 are referenced through Port 1. To interface external memory. Port 1 must be programmed for the multiplexed Adress/Data mode. If more than 256 external locations are required, Port 0 must output the additional lines.

(ii) Port 0 can be programmed as a nibble I/O port, or as an address port for interfacing external memory.

For external memory references, Port 0 can provide address bits A_8 A_{11} (lower nibble) or A_8-A_{15} (lower and upper nibble) depending on the required address space.

(iii) Port 2 bits can be programmed independently as input or output. The port is always available for I/O operations. In addition, Port 2 can be configured to provide open-drain outputs.

(iv) Port 3 lines can be configured as I/O or control lines. In either case, the direction of the eight lines is fixed as four input ($P3_0$-$P3_3$) and four output ($P3_4$-$P3_7$). For serial I/O, lines $P3_0$ and $P3_7$ are programmed as serial in and serial out respectively.

- handshake for Ports 0, 1 and 2 ($\overline{DAV}$ and RDY)
- four external interrupt request signals (IRQ_0-IRQ_3)
- timer input and output signals (T_{IN} and T_{OUT})
- Data Memory Select (DM).

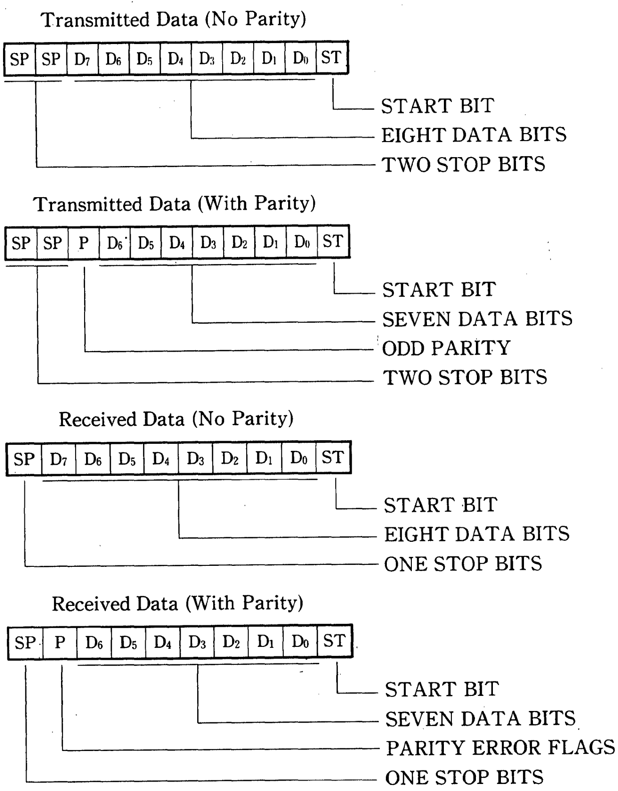

Transmitted Data (No Parity)
SP SP D7 D6 D5 D4 D3 D2 D1 D0 ST
START BIT
EIGHT DATA BITS
TWO STOP BITS
Transmitted Data (With Parity)
SP SP P D6 D5 D4 D3 D2 D1 D0 ST
START BIT
SEVEN DATA BITS
ODD PARITY
TWO STOP BITS
Received Data (No Parity)
SP D7 D6 D5 D4 D3 D2 D1 D0 ST
START BIT
EIGHT DATA BITS
ONE STOP BITS
Received Data (With Parity)
SP P D6 D5 D4 D3 D2 D1 D0 ST
START BIT
SEVEN DATA BITS
PARITY ERROR FLAGS
ONE STOP BITS

Fig. 6 Serial data formats

(3) Serial Input/Output

Port 3 lines $P3_0$ and $P3_7$ can be programmed as serial I/O lines for full-duplex serial asynchronous receiver/transmitter operation. The bit rate is controlled by Counter/Timer 0, with a maximum rate of 62.5K bits/second.

The Z8 automatically adds a start bit and two stop bits to transmitted data (Fig. 6). Odd parity is also available as an option.

(4) Counter/Timer

The Z8 contains two 8-bit programmable counter/timers (T_0 and T_1), each driven by its own 6-bit programmable prescaler. The T_1 prescaler can be driven by internal or external clock sources; however, the T_0 prescaler is driven by the internal clock only.

The counters can be started, stopped, restarted to continue, or restarted from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode). The counters, but not the prescalers, can be read any time without disturbing their value or count mode.

(5) Interrupts

The Z8 allows six different interrupts from eight sources: the four Port 3 lines $P3_0$-$P3_3$, Serial In, Serial Out, and the two counter/timers. These interrupts are both maskable and prioritized.

All Z8 interrupts are vectored. Polled interrupt systems are also supported.

Instruction Set Notation

(1) Addressing modes

The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate
R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working register pair address

(2) Symbols

The following symbols are used in describing the instruction set.

dst	Destination location or contents
src	Source location or contents
cc	Condition code (see list)
@	Indirect address prefix
SP	Stack pointer (control registers 254-255)
PC	Program counter
FLAGS	Flag register (control register 252)
RP	Register pointer (control register 253)
IMR	Interrupt mask register (control register 251)

Assignment of a value is indicated by the symbol "←". For example.

dst ← dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit "n" of a given location. For example, dst (7) refers to bit 7 of the destination operand.

(3) Flags

Control Register R252 contains the following six flags :

C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag

Affected flags are indicated by :

0	Cleared to zero
1	Set to one
*	Set or cleared according to operation
—	Unaffected
×	Undefined

(4) Condition codes

See Table 1.

Table 1 Condition codes

Value	Mnemonic	Meaning	Flags set
1000		Always true	......
0111	C	Carry	C=1
1111	NC	No carry	C=0
0110	Z	Zero	Z=1
1110	NZ	Not Zero	Z=0
1101	PL	Plus	S=0
0101	MI	Minus	S=1
0100	OV	Overflow	V=1
1100	NOV	No overflow	V=0
0110	EQ	Equal	Z=1
1110	NE	Not equal	Z=0
1001	GE	Greater than or equal	(S XOR V) =0
0001	LT	Less than	(S XOR V) =1
1010	GT	Greater than	[Z OR (S XOR V)] =0
0010	LE	Less than or equal	[Z OR (S XOR V)] =1
1111	UGE	Unsigned greater than or equal	C=0
0111	ULT	Unsigned less than	C=1
1011	UGT	Unsigned greater than	(C=0 AND Z=0) =1
0011	ULE	Unsigned less than or equal	(C OR Z) =1
0000		Never true	......

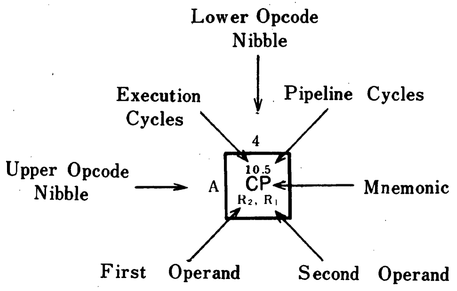
(5) Opcode map

Lower Nibble (Hex)

Upper Nibble (Hex)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6.5 DEC R_1	6.5 DEC IR_1	6.5 ADD r_1, r_2	6.5 ADD r_1, Ir_2	10.5 ADD R_2, R_1	10.5 ADD IR_2, R_1	10.5 ADD R_1, IM	10.5 ADD IR_1, IM	6.5 LD r_1, R_2	6.5 LD r_2, R_1	12/10.5 DJNZ r_1, RA	12/10.0 JR cc, RA	6.5 LD r_1, IM	12/10.0 JP cc, DA	6.5 INC r_1	
1	6.5 RLC R_1	6.5 RLC IR_1	6.5 ADC r_1, r_2	6.5 ADC r_1, Ir_2	10.5 ADC R_2, R_1	10.5 ADC IR_2, R_1	10.5 ADC R_1, IM	10.5 ADC IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
2	6.5 INC R_1	6.5 INC IR_1	6.5 SUB r_1, r_2	6.5 SUB r_1, Ir_2	10.5 SUB R_2, R_1	10.5 SUB IR_2, R_1	10.5 SUB R_1, IM	10.5 SUB IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
3	8.0 JP IRR_1	6.1 SRP IM	6.5 SBC r_1, r_2	6.5 SBC r_1, Ir_2	10.5 SBC R_2, R_1	10.5 SBC IR_2, R_1	10.5 SBC R_1, IM	10.5 SBC IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
4	8.5 DA R_1	8.5 DA IR_1	6.5 OR r_1, r_2	6.5 OR r_1, Ir_2	10.5 OR R_2, R_1	10.5 OR IR_2, R_1	10.5 OR R_1, IM	10.5 OR IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
5	10.5 POP R_1	10.5 POP IR_1	6.5 AND r_1, r_2	6.5 AND r_1, Ir_2	10.5 AND R_2, R_1	10.5 AND IR_2, R_1	10.5 AND R_1, IM	10.5 AND IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
6	6.5 COM R_1	6.5 COM IR_1	6.5 TCM r_1, r_2	6.5 TCM r_1, Ir_2	10.5 TCM R_2, R_1	10.5 TCM IR_2, R_1	10.5 TCM R_1, IM	10.5 TCM IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
7	10/12.1 PUSH R_2	12/14.1 PUSH IR_2	6.5 TM r_1, r_2	6.5 TM r_1, Ir_2	10.5 TM R_2, R_1	10.5 TM IR_2, R_1	10.5 TM R_1, IM	10.5 TM IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
8	10.5 DECW RR_1	10.5 DECW IR_1	12.0 LDE r_1, Irr_2	18.0 LDEI Ir_1, Irr_2					↓	↓	↓	↓	↓	↓	↓	6.1 DI
9	6.5 RL R_1	6.5 RL IR_1	12.0 LDE r_2, Irr_1	18.0 LDEI Ir_2, Irr_1					↓	↓	↓	↓	↓	↓	↓	6.1 EI
A	10.5 INCW RR_1	10.5 INCW IR_1	6.5 CP r_1, r_2	6.5 CP r_1, Ir_2	10.5 CP R_2, R_1	10.5 CP IR_2, R_1	10.5 CP R_1, IM	10.5 CP IR_1, IM	↓	↓	↓	↓	↓	↓	↓	14.0 RET
B	6.5 CLR R_1	6.5 CLR IR_1	6.5 XOR r_1, r_2	6.5 XOR r_1, Ir_2	10.5 XOR R_2, R_1	10.5 XOR IR_2, R_1	10.5 XOR R_1, IM	10.5 XOR IR_1, IM	↓	↓	↓	↓	↓	↓	↓	16.0 IRET
C	6.5 RRC R_1	6.5 RRC IR_1	12.0 LDC r_1, Irr_2	18.0 LDCI Ir_1, Irr_2				10.5 LD r_1, x, R_2	↓	↓	↓	↓	↓	↓	↓	6.5 RCF
D	6.5 SRA R_1	6.5 SRA IR_1	12.0 LDC r_2, Irr_1	18.0 LDCI Ir_2, Irr_1	20.0 CALL* IRR_1		20.0 CALL DA	10.5 LD r_2, x, R_1	↓	↓	↓	↓	↓	↓	↓	6.5 SCF
E	6.5 RR R_1	6.5 RR IR_1		6.5 LD r_1, Ir_2	10.5 LD R_2, R_1	10.5 LD IR_2, R_1	10.5 LD R_1, IM	10.5 LD IR_1, IM	↓	↓	↓	↓	↓	↓	↓	6.5 CCF
F	8.5 SWAP R_1	8.5 SWAP IR_1		6.5 LD Ir_1, r_2		10.5 LD R_2, IR_1			↓	↓	↓	↓	↓	↓	↓	6.0 NOP

Bytes per Instruction 3 2 3 1



Legend:
R = 8-Bit Address
r = 4-Bit Address
R_1 or r_1 = Dst Address
R_2 or r_2 = Src Address

Sequence:
Opcode, First Operand, Second Operand

Note: The blank areas are not defined.

(6) Instruction Summary

Instruction	Addr Mode		Opcode Byte	Flags Affected
and Operation	dst	src	(Hex)	C Z S V D H
ADC dst,src dst←dst+src+C	(Note 1)		1□	* * * * 0 *
ADD dst,src dst←dst+src	(Note 1)		0□	* * * * 0 *
AND dst,src dst←dst AND src	(Note 1)		5□	- * * 0 - -
CALL dst SP←SP−2 @SP←PC;PC←dst	DA IRR		D6 D4	- - - - - -
CCF C←NOT C			E F	* - - - - -
CLR dst dst← 0	R IR		B0 B1	- - - - - -
COM dst dst←NOT dst	R IR		60 61	- * * 0 - -
CP dst,src dst←src	(Note 1)		A□	* * * * - -
DA dst dst←DA dst	R IR		40 41	* * * X - -
DEC dst dst←dst−1	R IR		00 01	- * * * - -
DECW dst dst←dst−1	RR IR		80 81	- * * * - -
DI IMR(7)←0			8F	- - - - - -

DJNZ r,dst
r←r−1
if r 0 PC←PC+dst
Range: +127,−128
	RA		rA
			r=0−F
				- - - - - -

EI IMR(7) ←1			9F	- - - - - -
INC dst dst←dst+1	r R IR		rE r=0−F 20 21	- * * * - -
INCW dst dst←dst+1	RR IR		A0 A1	- * * * - -

IRET
FLAGS←@SP; SP←SP+1
PC←@SP;SP←SP+2;IMR(7)←1
			B F	* * * * * *

JP cc,dst if cc is true PC←dst	DA IRR		cD c=0−F 30	- - - - - -
JR cc,dst if cc is true, PC←PC+dst Range: +127,−128	RA		cB c=0−F	- - - - - -
LD dst,src dst←src	r r R r X r Ir R R R IR IR	IM R r X r Ir r R IR IM IM R	rC r8 r9 r=0−F C7 D7 E3 F3 E4 E5 E6 E7 F5	- - - - - -
LDC dst,src dst←src	r Irr	Irr r	C2 D2	- - - - - -

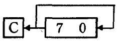
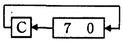
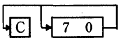
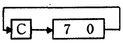
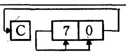
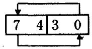
LDCI dst,src
dst←src
r←r+1;rr←rr+1
	Ir
	Irr
		Irr
		Ir
			C3
			D3
				- - - - - -

LDE dst,src dst←src	r Irr	Irr r	82 92	- - - - - -

LDEI dst,src
dst←src
r←r+1;rr←rr+1
	Ir
	Irr
		Irr
		Ir
			83
			93
				- - - - - -

NOP			FF	- - - - - -
OR dst,src dst←dst OR src	(Note 1)		4□	- * * 0 - -

POP dst
dst←@SP
SP←SP+ 1
	R
	IR
			50
			51
				- - - - - -

PUSH src SP←SP−1;@SP←src		R IR	70 71	- - - - - -
RCF C← 0			C F	0 - - - - -
RET PC @SP;SP←SP+ 2			A F	- - - - -
RL dst 	R IR		90 91	* * * * - -
RLC dst 	R IR		10 11	* * * * - -
RR dst 	R IR		E0 E1	* * * * - -
RRC dst 	R IR		C0 C1	* * * * - -
SBC dst,src dst←dst−src−C	(Note 1)		3□	* * * * 1 *
SCF C←1			D F	1 - - - - -
SRA dst 	R IR		D0 D1	* * * 0 - -
SRP src RP←src		IM	31	- - - - - -
SUB dst,src dst←dst−src	(Note 1)		2□	* * * * 1 *
SWAP dst 	R IR		F0 F1	X * * X - -
TCM dst,src (NOT dst) AND src	(Note 1)		6□	- * * 0 - -
TM dst,src dst AND src	(Note 1)		7□	- * * 0 - -
XOR dst,src dst←dst XOR src	(Note 1)		B□	- * * 0 - -

Note 1 These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a □ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

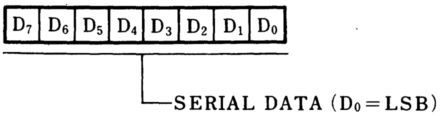
For example, to determine the opcode of an ADC instruction use the addressing modes r (destination) and Ir (source). The result is 13

Addr Mode		Lower Opcode Nibble
dst	src	
r	r	$\boxed{2}$
r	Ir	$\boxed{3}$
R	R	$\boxed{4}$
R	IR	$\boxed{5}$
R	IM	$\boxed{6}$
IR	IM	$\boxed{7}$

Register

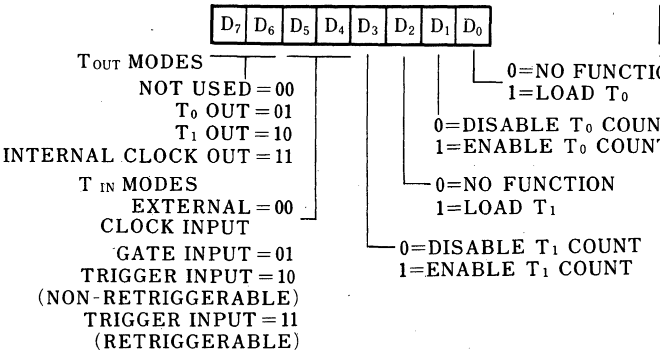
R240 (SIO) Serial I/O Register

($F0_H$: Read/Write)



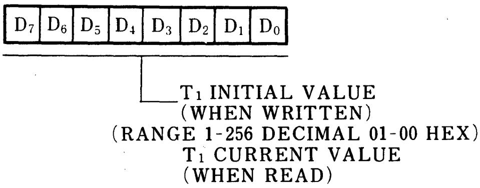
R241 (TMR) Timer Mode Register

($F1_H$: Read/Write)



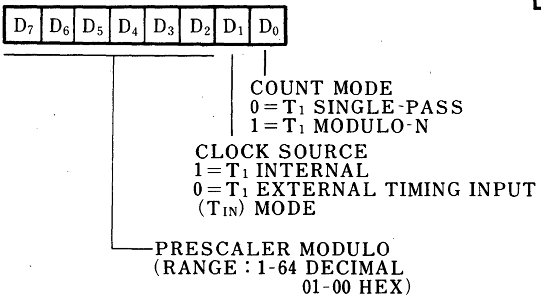
R242 (T1) Counter Timer 1 Register

($F2_H$: Read/Write)


D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
T_1 INITIAL VALUE
(WHEN WRITTEN)
(RANGE 1-256 DECIMAL 01-00 HEX)
T_1 CURRENT VALUE
(WHEN READ)

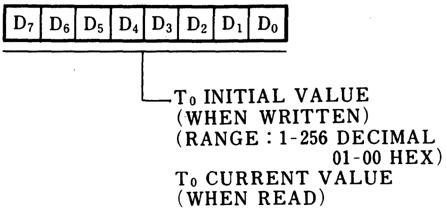
R243 (PRE1) Prescaler 1 Register

($F3_H$: Write Only)



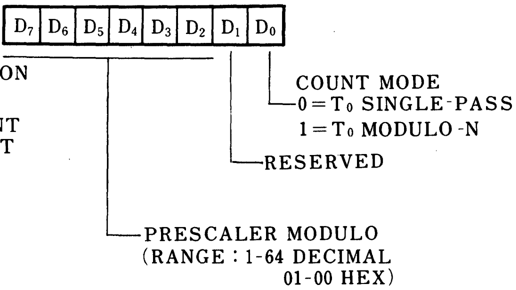
R244 (T0) Counter/Timer 0 Register

($F4_H$: Read/Write)



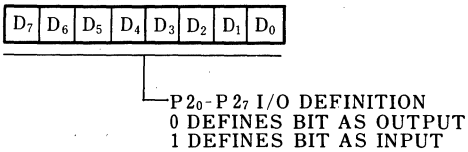
R245 (PRE0) Prescaler 0 Register

($F5_H$: Write Only)



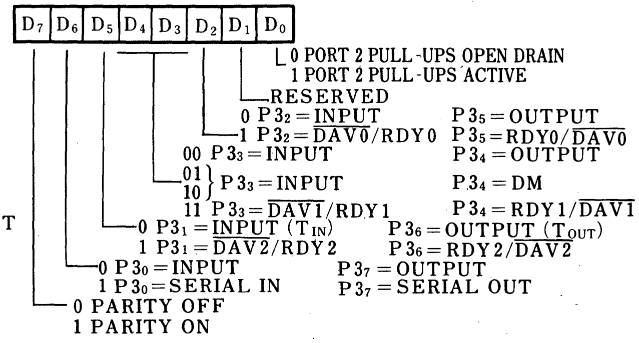
R246 (P2M) Port 2 Mode Register

($F6_H$: Write Only)

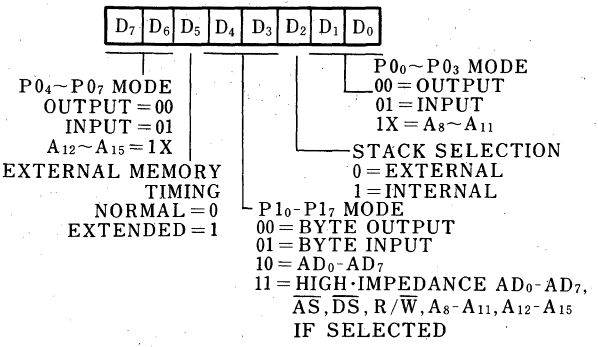

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
$P2_0$-$P2_7$ I/O DEFINITION
0 DEFINES BIT AS OUTPUT
1 DEFINES BIT AS INPUT

R247 (P3M) Port 3 Mode Register

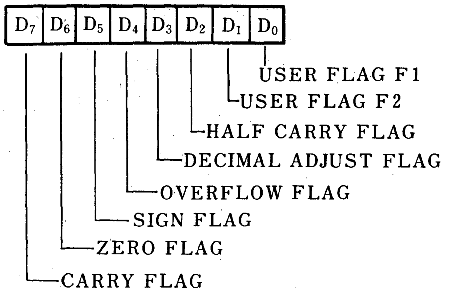
($F7_H$: Write Only)


D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
0 PORT 2 PULL-UPS OPEN DRAIN
1 PORT 2 PULL-UPS ACTIVE
RESERVED
0 $P3_2$=INPUT $P3_5$=OUTPUT
1 $P3_2$=$\overline{DAV0}$/RDY0 $P3_5$=RDY0/$\overline{DAV0}$
00 $P3_3$=INPUT $P3_4$=OUTPUT
01, 10 $P3_3$=INPUT $P3_4$=DM
11 $P3_3$=$\overline{DAV1}$/RDY1 $P3_4$=RDY1/$\overline{DAV1}$
0 $P3_1$=INPUT (T_{IN}) $P3_6$=OUTPUT (T_{OUT})
1 $P3_1$=$\overline{DAV2}$/RDY2 $P3_6$=RDY2/$\overline{DAV2}$
0 $P3_0$=INPUT $P3_7$=OUTPUT
1 $P3_0$=SERIAL IN $P3_7$=SERIAL OUT
0 PARITY OFF
1 PARITY ON

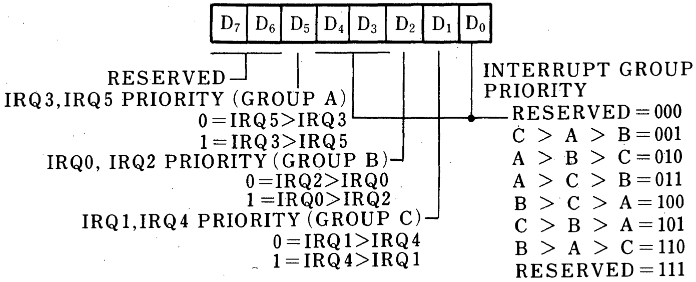
R248 (P01M)
Port 0 and 1 Mode Register
(F8H : Write Only)



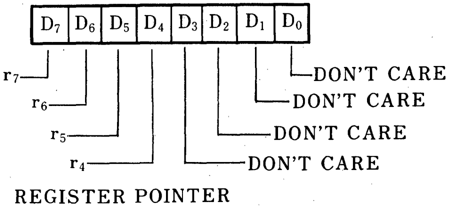
R252 (FLAGS)
Flag Register
(FCH : Read/Write)



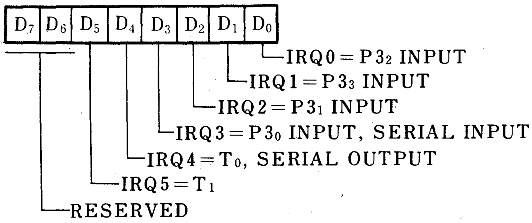
R249 (IPR)
Interrupt Priority Register
(F9H : Write Only)



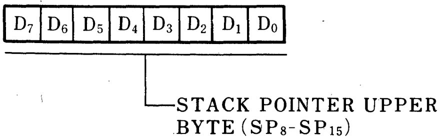
R253 (RP)
Register Pointer
(FDH : Read/Write)



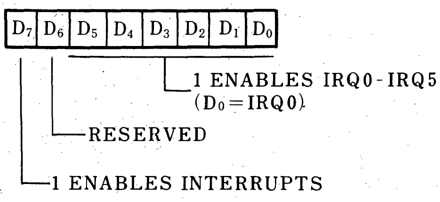
R250 (IRQ)
Interrupt Request Register
(FAH : Read/Write)



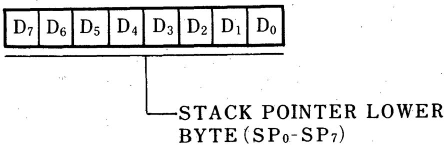
R254 (SPH)
Stack Pointer
(FEH : Read/Write)



R251 (IMR)
Interrupt Mask Register
(FBH : Read/Write)



R255 (SPL)
Stack Pointer
(FFH : Read/Write)



LH0881/LH0881A Z8 Microcomputer Unit (ROMless)

■ Description

The LH0881/A is a ROMless version of the LH0801/A and LH0811/A Z8 single-chip microcomputers and offers the outstanding feature of the Z8 family architecture.

Because some I/O ports are used for address/data bus, this device accesses up to 128K bytes of the external memory space. Using the external memory in place of an on-chip ROM allows designing more powerful microcomputer system.

■ Features

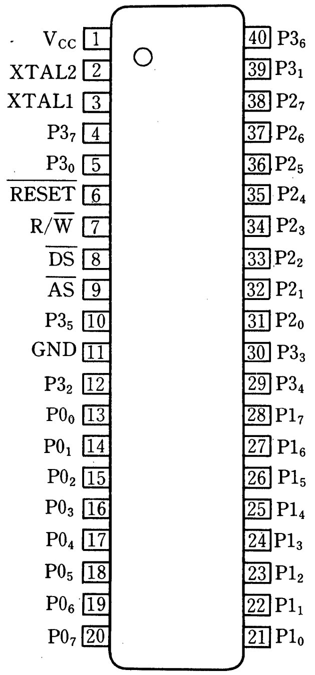
1. Complete microcomputer, 24 I/O lines, and up to 64K bytes addressable external space each for program and data memory.
2. 143 bytes register file
   124 general-purpose registers
   3 I/O port registers
   16 status and control registers
3. Register pointer so that short, fast instructions can access any one of the nine working-register groups.
4. Full-duplex UART and two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler.
5. Vectored priority interrupts for I/O, counter/timers, and UART.
6. On-chip oscillation circuit
7. External clock
   8MHz MAX. (internal 4MHz): LH0881/U
   12MHz MAX. (internal 6MHz): LH0881A/AU
8. Single +5V power supply
9. 40-pin DIP (DIP40-P-600)
   LH0881/LH0881A
   44-pin QFJ (QFJ44-P-S650)
   LH0881U/LH0881AU

■ Ordering Information

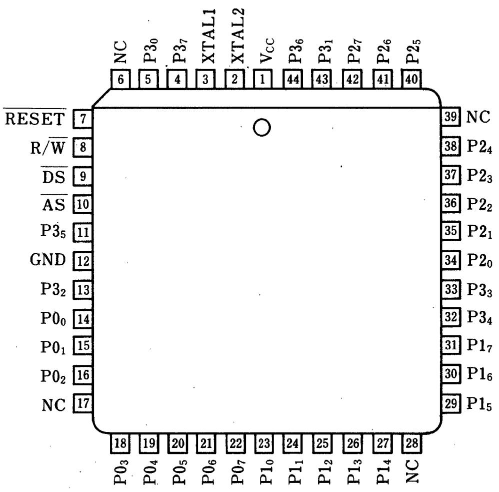
LH0881 X X
- Package
  Blank:40-Pin DIP (DIP40-P-600)
  U: 44-pin QFJ (QFJ44-P-S650)
- Clock frequency
  Blank: 8MHz
  A: 12MHz
- Model No.

■ Pin Connections

LH0881/LH0881A

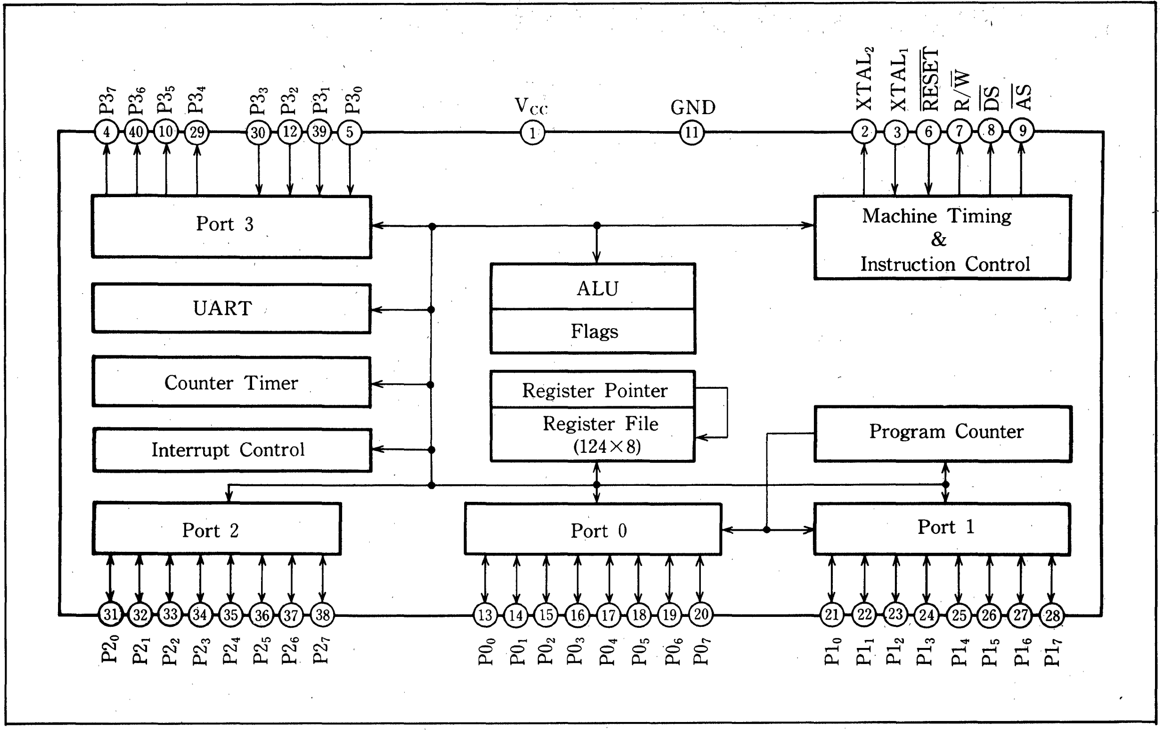


LH0881U/LH0881AU



Top View

Block Diagram



(Note) Pin numbers apply to 40-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
$P0_0$-$P0_7$	Port 0	I/O	4 bits×2, programmable for I/O.
$P1_0$-$P1_7$	Port1・ Address/data bus	I/O	Multiplexed address/data bus
$P2_0$-$P2_7$	Port 2	I/O	Programmable for I/O in bits.
$P3_0$-$P3_7$	Port 3	I/O	$P3_0$-$P3_3$ for input, $P3_4$-$P3_7$ for output.
$\overline{AS}$	Address Strobe	O	Active "Low", activated for external address memory transfer.
$\overline{DS}$	Data Strobe	O	Active "Low", activated for external data memory transfer.
$R/\overline{W}$	Read/Write	O	Read at "High", Write at "Low".
$\overline{RESET}$	Reset	I	Active "Low". Initializes.
XTAL1	Clock 1	I	Clock terminal pin.
XTAL2	Clock 2	O	Clock terminal pin.

Pin functions of the LH0881/A are identical to those of the LH0801/A, LH0811/A, except for pins $P1_0$-$P1_7$.

Address space

(1) Program Memory

The LH0881/A, having a 16-bit program counter, addresses 64K-bytes of external program memory. All the command codes are fatched from these external program memories.

For the LH0881/A, the first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Program execution begins at location 000C$_H$ after a reset.

(2) Data Memory

The LH0881/A can address 64K bytes of external data memory. External data memory may be included with or separated from the external program memory space. $\overline{\text{DM}}$, an optional I/O function that can be programmed to appear on pin P3$_1$, is used to distinguish between data and program memory space.

(3) Register File

The 143-byte register file includes three I/O port registers (R0, R2, R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255).

These registers are assigned the address locations shown in Fig 2.

LH0881/A instructions can access registers directly or indirectly with an 8-bit address field. This also allows short 4-bit register addressing using the Register Pointer (one of control registers). In the 4-bit mode, the register file is divided into nine working register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

(4) Satcks

Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4-R127).

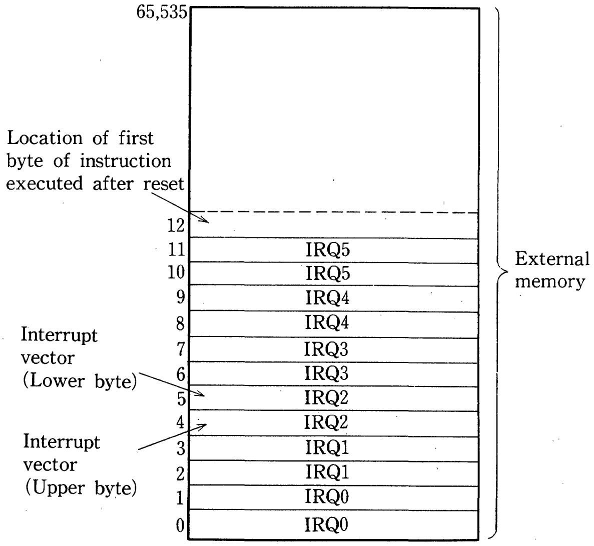

65,535
Location of first byte of instruction executed after reset
12
11
10
9
8
7
6
5
4
3
2
1
0
IRQ5
IRQ5
IRQ4
IRQ4
IRQ3
IRQ3
IRQ2
IRQ2
IRQ1
IRQ1
IRQ0
IRQ0
Interrupt vector (Lower byte)
Interrupt vector (Upper byte)
External memory

Fig. 1 Program memory map

LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7 0)	SPL
254	STACK POINTER (BITS 15 8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT PRIORITY REGISTER	IRO
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P0$_1$M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	T0 PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
	NOT IMPLEMENTED	
127 4	GENERAL-PURPOSE REGISTERS	
3	PORT 3	P3
2	PORT 2	P2
1	NOT IMPLEMENTED	
0	PORT 0	P0

Fig. 2 The register file

Port Functions

The LH0881/A has a dedicated memory interface port (Port 1) and input/output ports (Port 0, 2, 3). These ports are given eight lines each. The funtions of Port 0, 2 and 3 are the same as those of the LH0801/A, LH0811/A.

Port 1 is a dedicated Z-bus compatible memory interface. The operations of Port 1 are supported by the Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) lines, and by the Read/Write (R/$\overline{W}$) and Data Memory ($\overline{DM}$) control lines.

The low-order program and data memory address (A_0-A_7) are output through Port 1 and are multiplexed with data in/out (D_0-D_7). Instruction fetch and data memory read/write operations are done through this port.

Port 1 cannot be used as a register nor can a handshake mode be used with this port.

If more than eight address lines are reqired with the LH0881/A, additional lines can be obtained by programming Port 0 bits as address bits. The least-significant four bits of port 0 can be configured to supply address bits A_8-A_{11} for 4K byte addressing or both nibbles of Port 0 can be configured to supply address bits A_8-A_{15} for 64K byte addressing.

Registers

The LH0881/A control registers are the same as on the LH801/A, LH0811/A, except two bits D_3 and D_4 in the Port 0, 1 Mode Register (R248).

Serial Input/Output

The LH0881/A serial input/output functions are the same as those of the LH0801/A, LH0811/A. (Refer back to the LH0801/A description.)

Counter/Timers

The LH0881/A counter/timer functions are the same as those of the LH0801/A, LH0811/A, (Refer back to the LH0801/A description.)

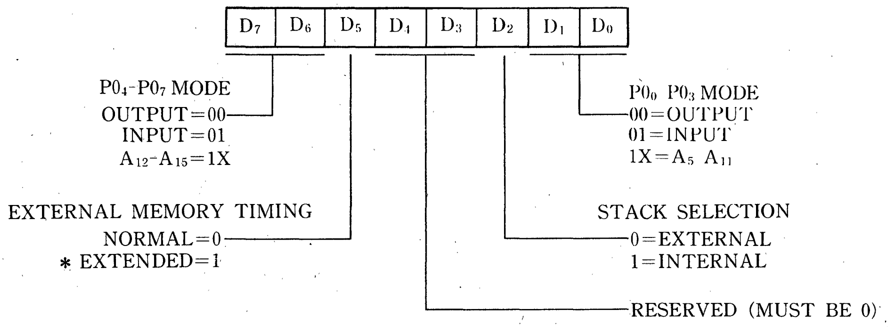
Interrupts

The LH0881/A interrupt functions are the same as those of the LH0801/A, LH0811/A, (Refer back to the LH0801/A description.)

Instructions and AC/DC Characteristics

These data of the LH0881/A are the same as for the LH0881/A, LH0811/A. (Refer back to the LH0801/A description.)

Fig 3 R248 (P01M) Port 0, 1 Mode Register (F8$_H$ Write only)



SM803/SM803A SM805/SM805A

CMOS 8-Bit Single Chip Microcomuputers

Description

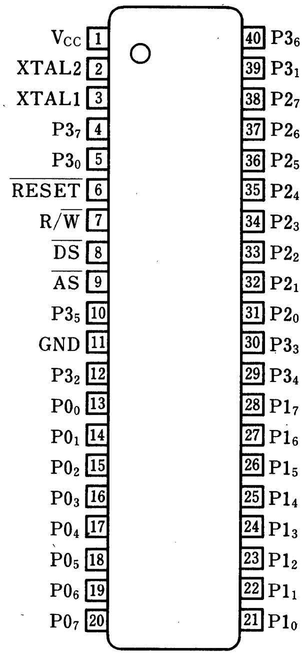
The SM803/A, SM805/A are CMOS 8-bit single chip microcomputers which have 4K bytes and 8K bytes of ROM respectively.

The devices offer faster execution; more efficient use of memory, more sophisticated interrupt, input/output and bit-manipulation capabilities, and easier system expansion.

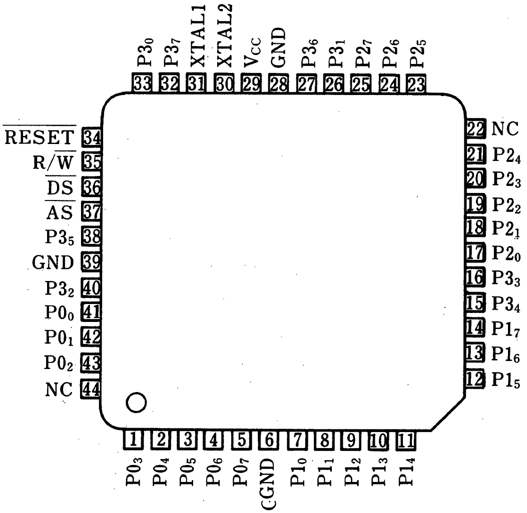
Under program control, the devices can be tailored to the user's needs. It can be configured as a stand-alone microcomputer with 4K bytes for the SM803/A or 8K bytes for the SM805/A of internal ROM, a traditional microprocessor that manages up to 120 bytes for the SM803/A or 112 bytes for the SM805/A of external memory, or a parallel processing device in a system with other processors and peripheral controllers linked by the BUS. In all configurations, a large number of pins remain available for I/O.

Pin Connections

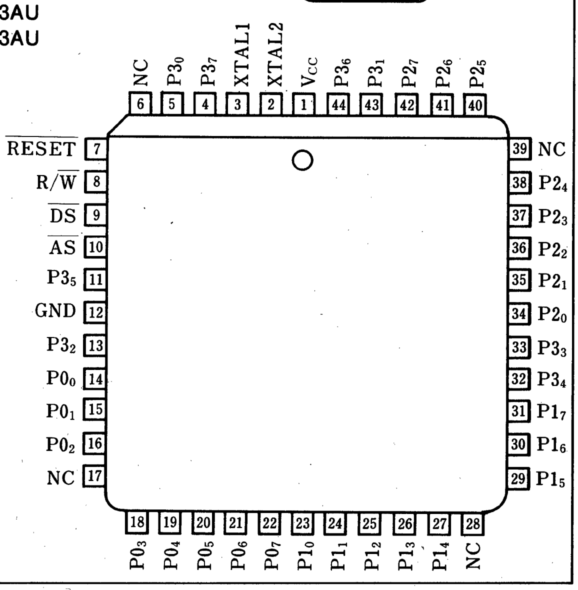
SM803/SM803A
SM805/SM805A



SM803M/SM803AM
SM805M/SM805AM



SM803U/SM803AU
SM805U/SM803AU



Features

1. Complete single-chip microcomputer with internal ROM, RAM and I/O
   RAM capacity: 124 bytes (SM803/A)
   : 236 bytes (SM805/A)
   ROM capacity: 4K bytes (SM803/A)
   : 8K bytes (SM805/A)
   I/O ports: 32
2. On-chip two programmable 8-bit couter/timers, each with a 6-bit programmable prescaler
3. Full-duplex UART
4. 144 byte register file (SM803/A)
   256 byte register file (SM805/A)
5. Register pointer so that short, fast instructions can access any working register groups
6. Vectored, priority interrupts for I/O, counter/timers, and UART
7. Up to 60K bytes for the SM803/A or 56K bytes for the SM805/A addressable external space each for program and data memory
8. On-chip oscillator
9. Maximum clock frequency
   8MHz (internal 4MHz): SM803/SM805
   8MHz (internal 6MHz): SM803A/SM805A
10. High speed instruction execution (8MHz/12MHz)
    Working register execution time:
    1.5 μs/1.0 μs
    Average instruction execution time:
    2.2 μs/1.5 μs
    Maximum instruction execution time:
    5.0 μs/3.3 μs
11. Single +5V power supply
12. 40-pin DIP (DIP40-P-600):
    SM803/A, SM805/A
    44-pin QFP (QFP44-P-1414)
    SM803M/AM, SM805M/AM
    44-pin QFJ (QFJ44-P-S650):
    SM803U/AU, SM805U/AU

Ordering Information

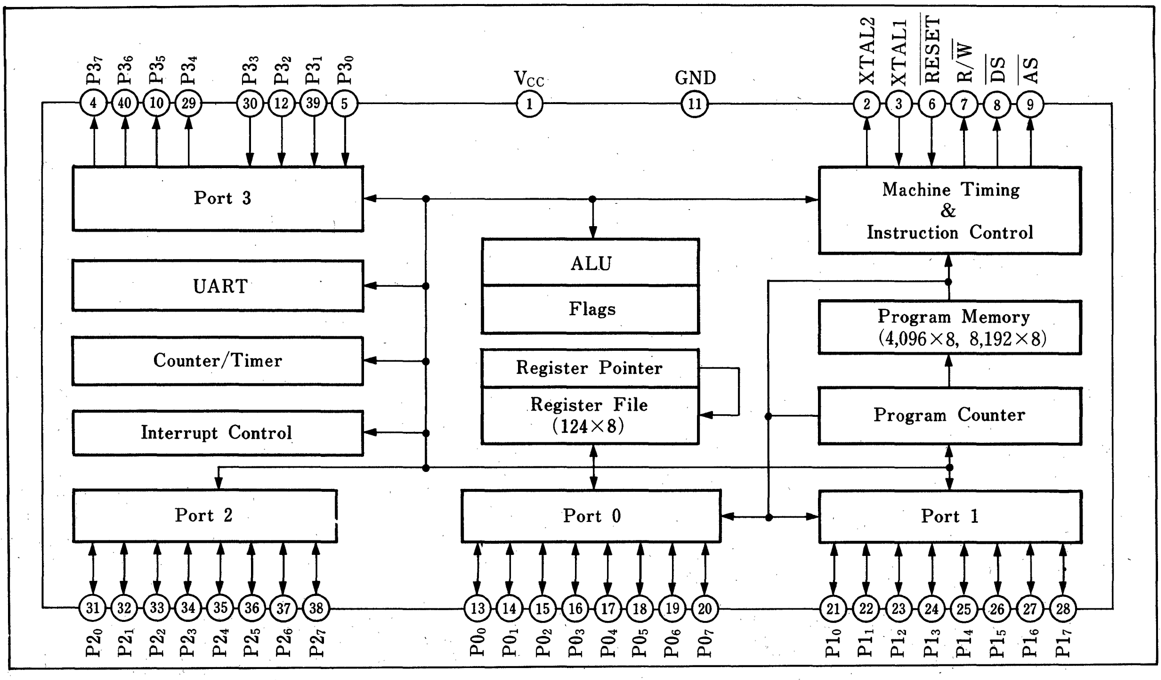
SM80X X X

Package
Blank: 40-pin DIP (DIP40-P-600)
M: 44-pin QFP (QFP44-P-1414)
U: 44-pin QFJ (QFJ44-P-S650)

Clock frequency
Blank: 8MHz
A: 12MHz

Model No.
SM803 (On-chip 4K byte ROM)
SM805 (On-chip 8K byte ROM)

Block Diagram



Note: Pin numbers apply to 40-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
$P0_0-P0_7$	Port 0	I/O	8-bit I/O port, programmable for I/O.
$P1_0-P1_7$	Port 1	I/O	Programmable for I/O in btyes.
$P2_0-P2_7$	Port 2	I/O	Programmable for I/O in bits.
$P3_0-P3_7$	Port 3	I/O	$P3_0$-$P3_3$ for input, $P3_4$-$P3_7$ for output.
$\overline{AS}$	Address Strobe	O	Active "Low", activated for external address memory transfer.
$\overline{DS}$	Data Strobe	O	Active "Low", activated for external data memory transfer.
$R/\overline{W}$	Read/Write	O	Read at "High", Write at "Low".
$\overline{RESET}$	Reset	I	Active "Low". Initializes.
XTAL1	Clock 1	I	Clock terminal pin.
XTAL2	Clock 2	O	Clock terminal pin.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	-0.3 to V_{CC}	V	1
Output voltage	V_{OUT}	-0.3 to V_{CC}	V	
Operating temperature	Topr	0 to +70	℃	
Storage temperatuer	Tstg	-65 to $+150$	℃	

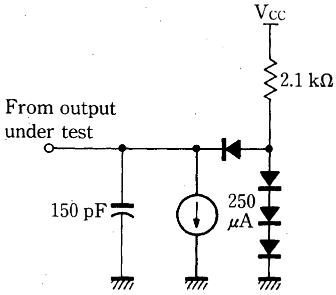
Note: The maximum applicable voltage on any pin with respect to GND.

DC Characteristics

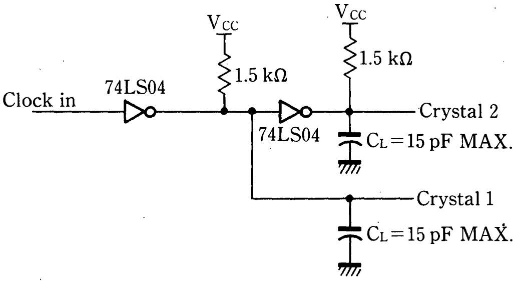
(V_{CC}=5V±10%, Ta=0 to +70℃)

Parameter	Symbol	Condition	8MHz		12MHz		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
Clock input high voltage	V_{CH}	Driven by external clock oscillator	3.8	V_{CC}	3.8	V_{CC}	V	
Clock input low voltage	V_{CL}	Driven by external clock oscillator	-0.3	0.8	-0.3	0.8	V	
Input high voltage (handshaking)	V_{IH}		2.0 (2.2)	V_{CC}	2.0 (2.2)	V_{CC}	V	1
Input low voltage (handshaking)	V_{IL}		-0.3	0.8 (0.5)	-0.3	0.8 (0.5)	V	
Reset input high voltage	V_{RH}		3.8	V_{CC}	3.8	V_{CC}	V	
Reset input low voltage	V_{RL}		-0.3	0.8	-0.3	0.8	V	
Output high voltage	V_{OH}	$I_{OH}=-250\mu A$	2.4		2.4		V	
Output low voltage	V_{OL}	$I_{OL}=+2.0mA$		0.4		0.4	V	
Input leakage current	I_{IL}	$0V \leqq V_{IN} \leqq +5.5V$	-10	10	-10	10	μA	
Output leakage current	I_{OL}	$0V \leqq V_{IN} \leqq +5.5V$	-10	10	-10	10	μA	
Reset input current	I_{IR}	$V_{CC}=5.5V$, $V_{RL}=0V$					μA	
Supply current	I_{CC}						mA	
Standby current	I_{CC1}	HALT instruction		7		10	mA	
	I_{CC2}	STOP intstruction		200		200	μA	

Note1: For the SM805/A, the minimum value should be 2.2V as well as when handshaking.



Test load 1



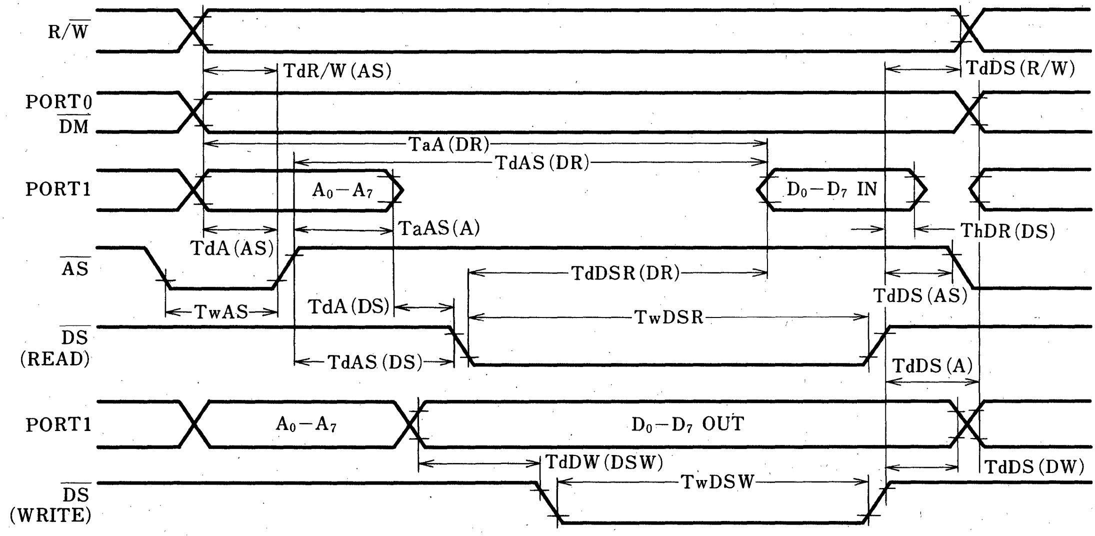
External clock generator circuit

External I/O or Memory Read/Write [Note 1]

(V_{CC}=5V±10%, Ta=0 to +70℃)

Parameter		Symbol	8MHz		12MHz		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
Address valid to $\overline{\text{AS}}$ ↑ delay		TdA (AS)	50		35		ns	2, 3
$\overline{\text{AS}}$ ↑ to input data required valid delay		TdAS (DR)		360		220	ns	2, 3, 4
$\overline{\text{AS}}$ ↑ low width		TwAS	80		55		ns	2, 3
$\overline{\text{DS}}$ low width	Read	TwDSR	250		185		ns	2, 3, 4
	Write	TwDSW	160		110		ns	2, 3, 4
$\overline{\text{DS}}$ ↓ to input data required valid		TdDSR (DR)		200		130	ns	2, 3, 4
Input data hold time		ThDSR (DS)	0		0		ns	2
$\overline{\text{DS}}$ ↑ to address active delay		TdDS (A)	70		45		ns	2, 3
$\overline{\text{DS}}$ ↑ to $\overline{\text{AS}}$ ↓ delay		TdDS (AS)	70		55		ns	2, 3
Read valid to $\overline{\text{AS}}$ ↑ delay		TdR/W (AS)	50		30		ns	2, 3
$\overline{\text{DS}}$ ↑ to read not valid		TdDS (R/W)	60		35		ns	2, 3
Output data valid to $\overline{\text{DS}}$ ↓ delay		TdDW (DSW)	50		35		ns	2, 3
$\overline{\text{DS}}$ ↑ to output data not valid delay		TdDS (DW)	70		45		ns	2, 3
Write valid to $\overline{\text{AS}}$ ↑ delay		TdA (DR)		410		255	ns	2, 3, 4
$\overline{\text{DS}}$ to write not valid delay		TdAS (DS)	80		55		ns	2, 3

Note 1: All timing references use 2.0 V for a logic "1" and 0.8 V for a logic "0".
Note 2: Test load 1
Note 3: The timing is defined at the minimum cycle of TpC.
Note 4: Apply double cycle of input clock TpC for the expansion memory timing.

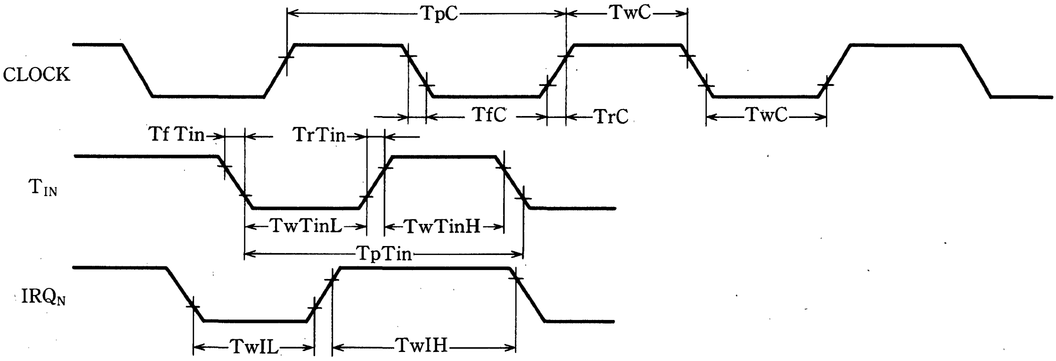


Input Clock, Timer Input, Interrupt Request Input

(V$_{CC}$=5V±10%, to=0 to +70℃)

Parameter	Symbol	8MHz		12MHz		Unit	Note
		MIN.	MAX.	MIN.	MAX.		
Input clock cycle	TpC	125	1000	83	1000	ns	1
Input clock rise, fall time	TrC, TfC		25		15	ns	1
Input clock width	TwC	37		26		ns	1
Timer input low width	TwTinL	100		70		ns	2
Timer input high width	TwTinH	3TpC		3TpC		ns	2
Timer input cycle	TpTin	8TpC		8TpC		ns	2
Timer input rise, fall time	TrTin. TfTin		100		100	ns	2
Interrupt request input low time	TwIL	100		70		ns	2, 3
		3TpC		3TpC		ns	2, 4
Interrupt request input high time	TwIH	3TpC		3TpC		ns	2, 3

Note 1: The clock timing references use 3.8V for a logic "1" and 0.8V for logic "0".
Note 2: The timing references use 2.0V (2.2V for SM805/A) for a logic "1" and 0.8V for a logic "0".
Note 3: Interrupt request from port 3 (P3$_1$-P3$_3$).
Note 4: Interrupt request from port 3 (P3$_0$).

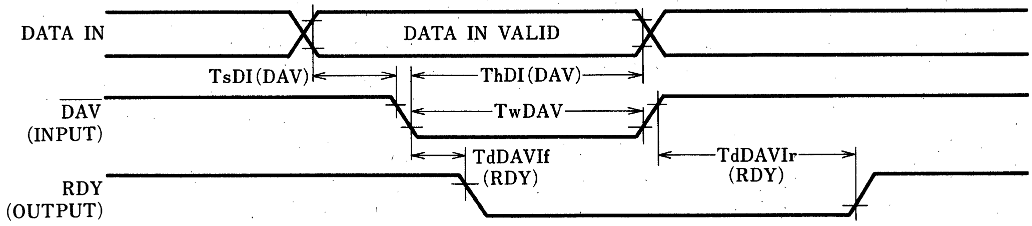


Handshake Timing (Note 1)

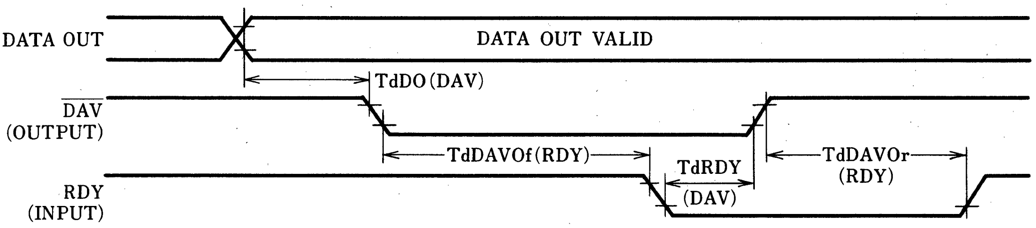
(V$_{CC}$=5V±10%, Ta=0 to +70℃)

Parameter	Symbol	8MHz		12MHz		Unit	Note
		MIN.	MAX.	MIN.	MAX.		
Data input setup time	TsDI (DAV)	0		0		ns	
Data input hold time	ThDI (DAV)	230		160		ns	
Data valid signal input width	TwDAV	175		120		ns	
$\overline{\text{DAV}}$ ↓ input to RDY ↓ delay time	TdDAVIf (RDY)		175		120	ns	2, 3
$\overline{\text{DAV}}$ ↓ output to RDY ↓ delay time	TdDAVOf (RDY)	0		0		ns	2, 4
$\overline{\text{DAV}}$ ↑ input to RDY ↑ delay time	TdDAVIr (RDY)		175		120	ns	2, 3, 5
$\overline{\text{DAV}}$ ↑ output to RDY ↑ delay time	TdDAVOr (RDY)	0		0		ns	2, 4
Data output to $\overline{\text{DAV}}$ ↓ delay time	TdDO (DAV)	50		30		ns	2
RDY ↓ input to $\overline{\text{DAV}}$ ↑ delay time	TdRDY (DAV)	0	200	0	140	ns	2

Note 1: All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
Note 2: Test load 1.
Note 3: Input handshake
Note 4: Output handshake
Note 5: When read out from the port before $\overline{\text{DAV}}$ ↑ input.



Input handshake



Output handshake

Architecture

(1) Address Spaces

(i) Program Memory The 16-bit program counter addresses 64K bytes of program memory space. Program memory can be located in two areas : one internal and the other external (Fig. 2). The first 2048 bytes consist of on-chip mask-programmed ROM. At addresses 2048 and greater, the Z8 executes external program memory fetches.

The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts.

(ii) Data Memory The Z8 can address 62K bytes of external data memory beginning at location 2048 (Fig. 3). External data memory may be include with or separated from the external program memory space. $\overline{\text{DM}}$, an optical I/O function that can be programmed to appear on pin $P3_4$, is used to distinguish between data and program memory space.

(iii) Register File The 144-byte register file includes four I/O port registers (R0-R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255). These registers are assigned the address locations shown in Fig. 4.

Z8 instructions can access registers directly or indirectly with an 8-bit address field. The Z8 also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In the 4-bit mode, the register file is divided into nine working-register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-regsister group.

The 4-bit address specifies the nth (0 to 15) address from the starting location (see Fig. 5).

*The addresses OEO_H-OEF_H of SM805 register file can not be directly accessed due to the essential function of the register pointer. Either of the following two methods is available for accessing those 16 registers.

1) Working register addressing
SRP #OEO_H (set the RP to $0EO_H$)

2) Register indirect addressing
ex.) LD 70H, #$0EO_H$
LD 40H, @70H (read)
LD @70H, 40H (write)

(iv) Stacks Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory between locations 4096 (8192 for SM805/A) and 65535. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 (236 for SM805/A) general-purpose registers.

Either an internal stack or an external stack may be selected with ports 0, 1 and the bit D_2 of mode register (248). The internal stack is specified with the device to be reset.

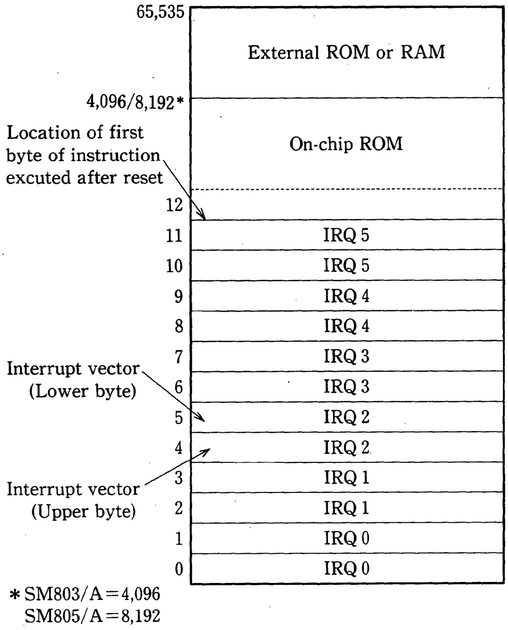


Fig. 2 Program memory map

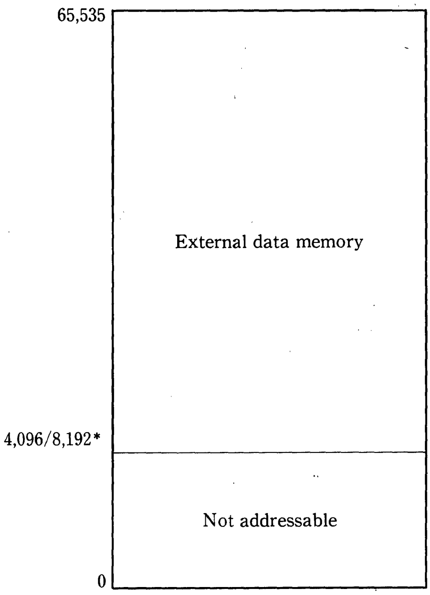


Fig. 3 Data memory map

LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7-0)	SPL
254	STACK POINTER (BITS 15-8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT REQUEST REGISTER	IRQ
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P01M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	TO PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T 1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
239	NOT IMPLEMENTED FOR SM803/A	
127 4	GENERAL-PURPOSE REGISTER	
3	PORT 3	P3
2	PORT 2	P2
1	PORT 1	P1
0	PORT 0	P0

Fig. 4 The register file

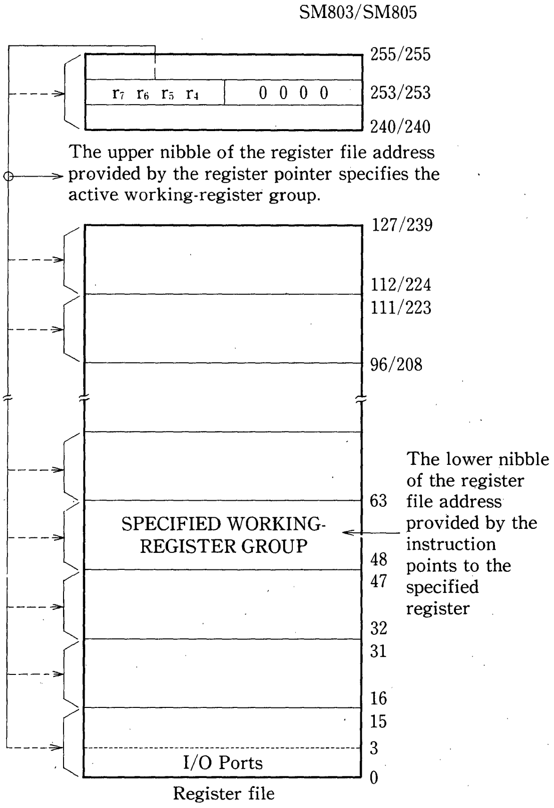

SM803/SM805
255/255
r7 r6 r5 r4
0 0 0 0
253/253
240/240
The upper nibble of the register file address provided by the register pointer specifies the active working-register group.
127/239
112/224
111/223
96/208
SPECIFIED WORKING-REGISTER GROUP
63
48
47
32
31
16
15
3
0
I/O Ports
The lower nibble of the register file address provided by the instruction points to the specified register
Register file

Fig. 5 The register pointer

(2) I/O ports

The Z8 has 32 lines dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable as input, output or address/data. Under software control, the ports can be programmed to provide address outputs, timing, status signals, serial I/O, and parallel I/O with or without handshake. All ports have active pull-ups and pull-downs compatible with TTL loads.

(i) Port 1 can be programmed as a byte I/O port or an address/data port for interfacing external memory.

Memory locations greater than 4095 (8191 for SM805/A) are referenced through Port 1. To interface external memory, Port 1 must be programmed for the multiplexed Adress/Data mode. If more than 257 external locations are required, Port 0 must output the additional lines.

(ii) Port 0 can be programmed as a nibble I/O port, or as an address port for interfacing external memory.

For external memory references, Port 0 can provide address bits A_8A_{11} (lower nibble) or A_8-A_{15} (lower and upper nibble) depending on the required address space.

(iii) Port 2 bits can be programmed independently as input or output. The port is always available for I/O operations. In addition, Port 2 can be configured to provide open-drain outputs.

(iv) Port 3 lines can be configured as I/O or control lines. In either cases, the direction of the eight lines is fixed as four input ($P3_0$-$P3_3$) and four output ($P3_4$-$P3_7$). For serial I/O, lines $P3_0$ and $P3_7$ are programmed as serial in and serial out respectively

- handshake for Ports 0, 1 ind 2 ($\overline{DAV}$ and RDY)
- four external interrupt request signals (IRQ_0-IRQ_3)
- timer input and output signals (T_{IN} and T_{OUT})
- Data Memory Select ($\overline{DM}$).

(3) Serial Input/Output

Port 3 lines $P3_0$ and $P3_7$ can be programmed as serial I/O lines for full-duplex serial asynchronous receiver/transmitter operation. The bit rate is controlled by Counter/Timer 0, with a maximum rate of 62.5K-bits/second.

The device automatically adds a start bit and two stop bits to transmitted data (Fig. 6). Odd parity is also aviliable as an option.

(4) Counter/Timer

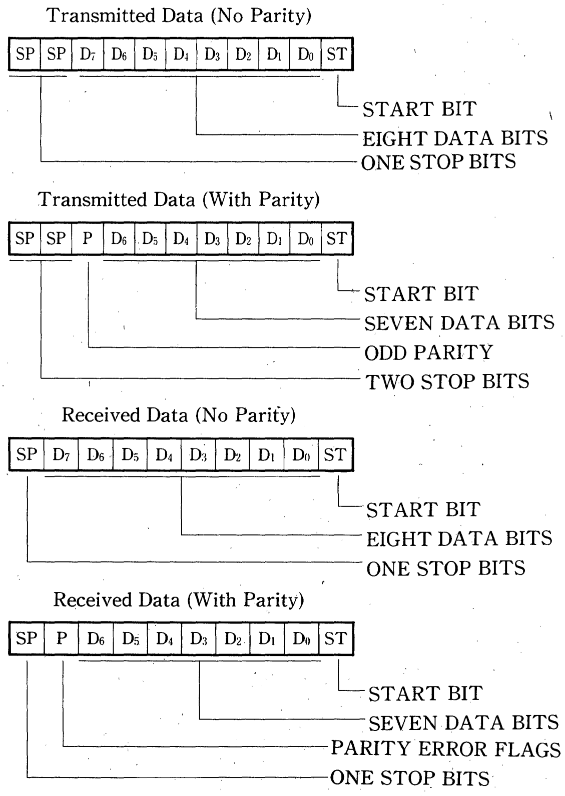
The device contains two 8-bit programmable counter/timers (T_0 and T_1), each driven by its own 6-bit programmable prescaler. The T_1 prescaler can be driven by internal or external clock sources; however, the T_0 prescaler is driven by the internal clock only.

The counters can be started, stopped, restarted to continue, or restarted from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuos mode). The conters, but not the prescalers, can be read any time without disturbing their value or count mode.

(5) Interrupts

The device allows six different interrupts from eight sources: the four Port 3 lines $P3_0$-$P3_3$, Serial In, Serial Out, and the two conter/timers. These interrupts are both maskable and prioritized.

All device interrupts are vectored. Polled interrupt systems are also supported.



Transmitted Data (No Parity)

SP | SP | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | ST
START BIT
EIGHT DATA BITS
ONE STOP BITS

Transmitted Data (With Parity)

SP | SP | P | D6 | D5 | D4 | D3 | D2 | D1 | D0 | ST
START BIT
SEVEN DATA BITS
ODD PARITY
TWO STOP BITS

Received Data (No Parity)

SP | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | ST
START BIT
EIGHT DATA BITS
ONE STOP BITS

Received Data (With Parity)

SP | P | D6 | D5 | D4 | D3 | D2 | D1 | D0 | ST
START BIT
SEVEN DATA BITS
PARITY ERROR FLAGS
ONE STOP BITS

Fig. 6 Serial data formats

■ Instruction Set Notation

(1) Addressing modes

The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate
R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working register pair address

(2) Symbols

The following symbols are used in describing the instruction set.

dst	Destination location or contents
src	Source location or contents
cc	Condition code (see list)
@	Indirect address prefix
SP	Stack pointer (control registers 254-255)
PC	Program counter
FLAGS	Flag register (control register 252)
RP	Register pointer (control register 253)
IMR	Interrupt mask register (control register 251)

Assignment of a value is indicated by the symbol "←". For example.

dst ← dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit "n" of a given location. For example, dst (7) refers to bit 7 of the destination operand.

(3) Flags

Control Register R252 contains the following six flags :

C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag

Affected flags are indicated by :

0	Cleared to zero
1	Set to one
*	Set or cleared according to operation
—	Unaffected
×	Undefined

(4) Condition codes

See Table 1.

Table 1 Condition codes

Value	Mnemonic	Meaning	Flags set
1000		Always true	......
0111	C	Carry	C=1
1111	NC	No carry	C=0
0110	Z	Zero	Z=1
1110	NZ	Not Zero	Z=0
1101	PL	Plus	S=0
0101	MI	Minus	S=1
0100	OV	Overflow	V=1
1100	NOV	No overflow	V=0
0110	EQ	Equal	Z=1
1110	NE	Not equal	Z=0
1001	GE	Greater than or equal	(S XOR V) =0
0001	LT	Less than	(S XOR V) =1
1010	GT	Greater than	[Z OR (S XOR V)] =0
0010	LE	Less than or equal	[Z OR (S XOR V)] =1
1111	UGE	Unsigned greater than or equal	C=0
0111	ULT	Unsigned less than	C=1
1011	UGT	Unsigned greater than	(C=0 AND Z=0) =1
0011	ULE	Unsigned less than or equal	(C OR Z) =1
0000.		Never true	......

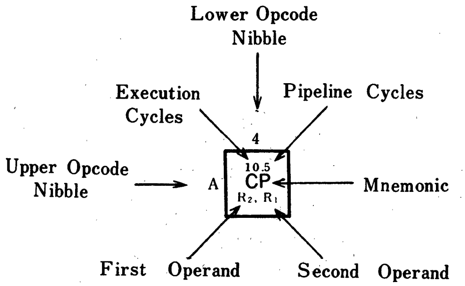
(5) Opcode map

Lower Nibble (Hex)

Upper Nibble (Hex)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6.5 DEC R_1	6.5 DEC IR_1	6.5 ADD r_1, r_2	6.5 ADD r_1, Ir_2	10.5 ADD R_2, R_1	10.5 ADD IR_2, R_1	10.5 ADD R_1, IM	10.5 ADD IR_1, IM	6.5 LD r_1, R_2	6.5 LD r_2, R_1	12/10.5 DJNZ r_1, RA	12/10.0 JR cc, RA	6.5 LD r_1, IM	12/10.0 JP cc, DA	6.5 INC r_1	
1	6.5 RLC R_1	6.5 RLC IR_1	6.5 ADC r_1, r_2	6.5 ADC r_1, Ir_2	10.5 ADC R_2, R_1	10.5 ADC IR_2, R_1	10.5 ADC R_1, IM	10.5 ADC IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
2	6.5 INC R_1	6.5 INC IR_1	6.5 SUB r_1, r_2	6.5 SUB r_1, Ir_2	10.5 SUB R_2, R_1	10.5 SUB IR_2, R_1	10.5 SUB R_1, IM	10.5 SUB IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
3	8.0 JP IRR_1	6.1 SRP IM	6.5 SBC r_1, r_2	6.5 SBC r_1, Ir_2	10.5 SBC R_2, R_1	10.5 SBC IR_2, R_1	10.5 SBC R_1, IM	10.5 SBC IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
4	8.5 DA R_1	8.5 DA IR_1	6.5 OR r_1, r_2	6.5 OR r_1, Ir_2	10.5 OR R_2, R_1	10.5 OR IR_2, R_1	10.5 OR R_1, IM	10.5 OR IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
5	10.5 POP R_1	10.5 POP IR_1	6.5 AND r_1, r_2	6.5 AND r_1, Ir_2	10.5 AND R_2, R_1	10.5 AND IR_2, R_1	10.5 AND R_1, IM	10.5 AND IR_1, IM	↓	↓	↓	↓	↓	↓	↓	
6	6.5 COM R_1	6.5 COM IR_1	6.5 TCM r_1, r_2	6.5 TCM r_1, Ir_2	10.5 TCM R_2, R_1	10.5 TCM IR_2, R_1	10.5 TCM R_1, IM	10.5 TCM IR_1, IM	↓	↓	↓	↓	↓	↓	↓	6.0 STOP
7	10/12.1 PUSH R_2	12/14.1 PUSH IR_2	6.5 TM r_1, r_2	6.5 TM r_1, Ir_2	10.5 TM R_2, R_1	10.5 TM IR_2, R_1	10.5 TM R_1, IM	10.5 TM IR_1, IM	↓	↓	↓	↓	↓	↓	↓	6.0 HALT
8	10.5 DECW RR_1	10.5 DECW IR_1	12.0 LDE r_1, Irr_2	18.0 LDEI Ir_1, Irr_2					↓	↓	↓	↓	↓	↓	↓	6.1 DI
9	6.5 RL R_1	6.5 RL IR_1	12.0 LDE r_2, Irr_1	18.0 LDEI Ir_2, Irr_1					↓	↓	↓	↓	↓	↓	↓	6.1 EI
A	10.5 INCW RR_1	10.5 INCW IR_1	6.5 CP r_1, r_2	6.5 CP r_1, Ir_2	10.5 CP R_2, R_1	10.5 CP IR_2, R_1	10.5 CP R_1, IM	10.5 CP IR_1, IM	↓	↓	↓	↓	↓	↓	↓	14.0 RET
B	6.5 CLR R_1	6.5 CLR IR_1	6.5 XOR r_1, r_2	6.5 XOR r_1, Ir_2	10.5 XOR R_2, R_1	10.5 XOR IR_2, R_1	10.5 XOR R_1, IM	10.5 XOR IR_1, IM	↓	↓	↓	↓	↓	↓	↓	16.0 IRET
C	6.5 RRC R_1	6.5 RRC IR_1	12.0 LDC r_1, Irr_2	18.0 LDCI Ir_1, Irr_2				10.5 LD r_1, x, R_2	↓	↓	↓	↓	↓	↓	↓	6.5 RCF
D	6.5 SRA R_1	6.5 SRA IR_1	12.0 LDC r_2, Irr_1	18.0 LDCI Ir_2, Irr_1	20.0 CALL* IRR_1		20.0 CALL DA	10.5 LD r_2, x, R_1	↓	↓	↓	↓	↓	↓	↓	6.5 SCF
E	6.5 RR R_1	6.5 RR IR_1		6.5 LD r_1, Ir_2	10.5 LD R_2, R_1	10.5 LD IR_2, R_1	10.5 LD R_1, IM	10.5 LD IR_1, IM	↓	↓	↓	↓	↓	↓	↓	6.5 CCF
F	8.5 SWAP R_1	8.5 SWAP IR_1		6.5 LD Ir_1, r_2		10.5 LD R_2, IR_1			↓	↓	↓	↓	↓	↓	↓	6.0 NOP

Bytes per Instruction 3 2 3 1



Legend:
R = 8-Bit Address
r = 4-Bit Address
R_1 or r_1 = Dst Address
R_2 or r_2 = Src Address

Sequence:
Opcode, First Operand, Second Operand

Note: The blank areas are not defined.

(6) Instruction Summary

Instruction	Addr Mode		Opcode Byte	Flags Affected
and Operation	dst	src	(Hex)	C Z S V D H
ADC dst,src dst←dst+src+C	(Note 1)		1□	* * * * 0 *
ADD dst,src dst←dst+src	(Note 1)		0□	* * * * 0 *
AND dst,src dst←dst AND src	(Note 1)		5□	- * * 0 - -
CALL dst SP←SP−2 @SP←PC;PC←dst	DA IRR		D6 D4	- - - - - - -
CCF C←NOT C			E F	* - - - - -
CLR dst dst← 0	R IR		B0 B1	- - - - - - -
COM dst dst←NOT dst	R IR		60 61	- * * 0 - -
CP dst,src dst←src	(Note 1)		A□	* * * * - -
DA dst dst←DA dst	R IR		40 41	* * * X - -
DEC dst dst←dst−1	R IR		00 01	- * * * - -
DECW dst dst←dst−1	RR IR		80 81	- * * * - -
DI IMR(7)←0			8F	- - - - - - -

DJNZ r,dst
r←r−1
if r 0 PC←PC+dst
Range: +127, −128
	RA		rA
			r=0−F
				- - - - - - -

EI IMR(7) ←1			9F	- - - - - - -
HALF			FF 7F	- - - - - - -
INC dst dst←dst+1	r R IR		rE r=0−F 20 21	- * * * - -
INCW dst dst←dst+1	RR IR		A0 A1	- * * * - -

IRET
FLAGS←@SP; SP←SP+1
PC←@SP;SP←SP+2;IMR(7)←1
			B F	* * * * * *

JP cc,dst if cc is true PC←dst	DA IRR		cD c=0−F 30	- - - - - - -
JR cc,dst if cc is true, PC←PC+dst Range: +127, −128	RA		cB c=0−F	- - - - - - -
LD dst,src dst←src	r r R r X r Ir R R R IR IR	IM R r X r Ir r R IR IM IM R	rC r8 r9 r=0−F C7 D7 E3 F3 E4 E5 E6 E7 F5	- - - - - - -
LDC dst,src dst←src	r Irr	Irr r	C2 D2	- - - - - - -

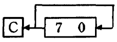
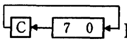
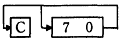
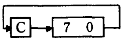
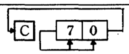
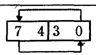
LDCI dst,src
dst←src
r←r+1;rr←rr+1
	Ir
	Irr
		Irr
		Ir
			C3
			D3
				- - - - - - -

LDE dst,src dst←src	r Irr	Irr r	82 92	- - - - - - -

LDEI dst,src
dst←src
r←r+1; rr←rr+1
	Ir
	Irr
		Irr
		Ir
			83
			93
				- - - - - - -

NOP			FF	- - - - - - -
OR dst,src dst←dst OR src	(Note 1)		4□	- * * 0 - -

POP dst
dst←@SP
SP←SP+ 1
	R
	IR
			50
			51
				- - - - - - -

PUSH src SP←SP−1; @SP←src		R IR	70 71	- - - - - - -
RCF C← 0			C F	0 - - - - -
RET PC @SP;SP←SP+ 2			A F	- - - - -
RL dst 	R IR		90 91	* * * * - -
RLC dst 	R IR		10 11	* * * * - -
RR dst 	R IR		E0 E1	* * * * - -
RRC dst 	R IR		C0 C1	* * * * - -
SBC dst,src dst←dst−src−C	(Note 1)		3□	* * * * 1 *
SCF C←1			D F	1 - - - - -
SRA dst 		R IR	D0 D1	* * * 0 - -
SRP src RP←src		IM	31	- - - - - - -
STOP			FF 6F	- - - - - - -
SUB dst,src dst←dst−src	(Note 1)		2□	* * * * 1 *
SWAP dst 	R IR		F0 F1	X * * X - -
TCM dst,src (NOT dst) AND src	(Note 1)		6□	- * * 0 - -
TM dst,src dst AND src	(Note 1)		7□	- * * 0 - -
XOR dst,src dst←dst XOR src	(Note 1)		B□	- * * 0 - -

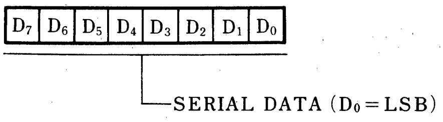
Note 1 These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a □ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, to determine the opcode of an ADC instruction use the addressing modes r (destination) and Ir (source). The result is 13

Addr Mode		Lower Opcode Nibble
dst	src	
r	r	2
r	Ir	3
R	R	4
R	IR	5
R	IM	6
IR	IM	7

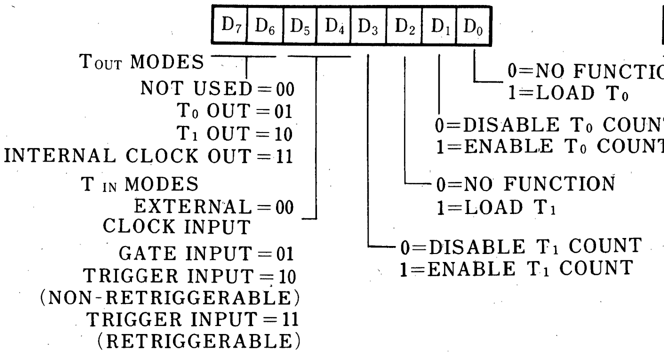
■ Register

R240 (SIO)
Serial I/O Register
(F0H : Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0

SERIAL DATA (D_0=LSB)

R241 (TMR)
Timer Mode Register
(F1H : Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0

T_{OUT} MODES
NOT USED=00
T_0 OUT=01
T_1 OUT=10
INTERNAL CLOCK OUT=11

T_{IN} MODES
EXTERNAL=00
CLOCK INPUT
GATE INPUT=01
TRIGGER INPUT=10
(NON-RETRIGGERABLE)
TRIGGER INPUT=11
(RETRIGGERABLE)

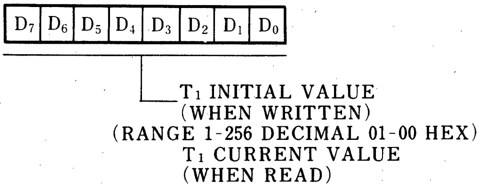
0=NO FUNCTION
1=LOAD T_0

0=DISABLE T_0 COUNT
1=ENABLE T_0 COUNT

0=NO FUNCTION
1=LOAD T_1

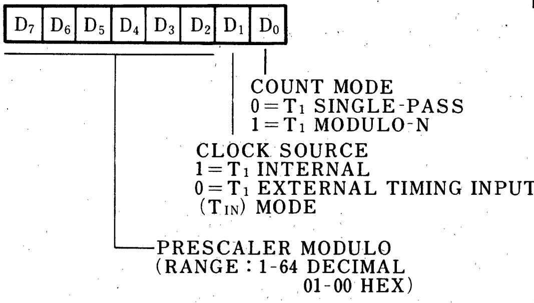
0=DISABLE T_1 COUNT
1=ENABLE T_1 COUNT

R242 (T1)
Counter Timer 1 Register
(F2H : Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0

T_1 INITIAL VALUE
(WHEN WRITTEN)
(RANGE 1-256 DECIMAL 01-00 HEX)
T_1 CURRENT VALUE
(WHEN READ)

R243 (PRE1)
Prescaler 1 Register
(F3H : Write Only)

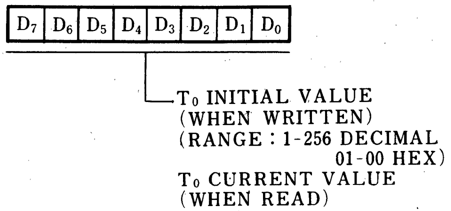

D7 D6 D5 D4 D3 D2 D1 D0

COUNT MODE
0=T_1 SINGLE-PASS
1=T_1 MODULO-N

CLOCK SOURCE
1=T_1 INTERNAL
0=T_1 EXTERNAL TIMING INPUT
(T_{IN}) MODE

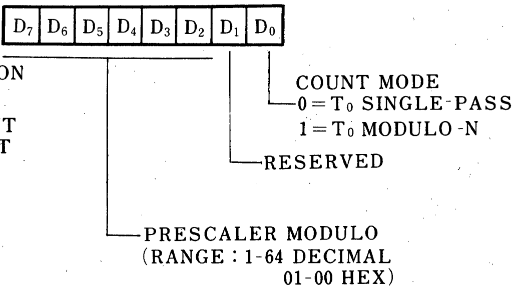
PRESCALER MODULO
(RANGE : 1-64 DECIMAL
01-00 HEX)

R244 (T0)
Counter/Timer 0 Register
(F4H : Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0

T_0 INITIAL VALUE
(WHEN WRITTEN)
(RANGE : 1-256 DECIMAL
01-00 HEX)
T_0 CURRENT VALUE
(WHEN READ)

R245 (PRE0)
Prescaler 0 Register
(F5H : Write Only)

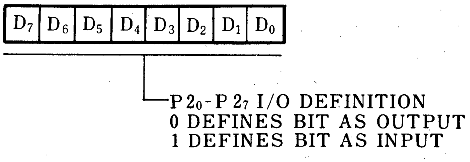

D7 D6 D5 D4 D3 D2 D1 D0

COUNT MODE
0=T_0 SINGLE-PASS
1=T_0 MODULO-N

RESERVED

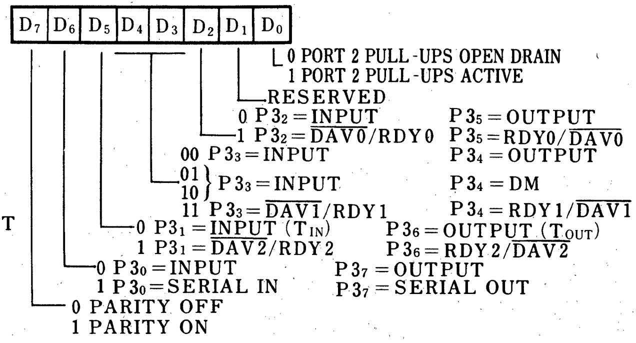
PRESCALER MODULO
(RANGE : 1-64 DECIMAL
01-00 HEX)

R246 (P2M)
Port 2 Mode Register
(F6H : Write Only)


D7 D6 D5 D4 D3 D2 D1 D0

$P2_0$-$P2_7$ I/O DEFINITION
0 DEFINES BIT AS OUTPUT
1 DEFINES BIT AS INPUT

R247 (P3M)
Port 3 Mode Register
(F7H : Write Only)


D7 D6 D5 D4 D3 D2 D1 D0

0 PORT 2 PULL-UPS OPEN DRAIN
1 PORT 2 PULL-UPS ACTIVE

RESERVED

0 $P3_2$=INPUT $P3_5$=OUTPUT
1 $P3_2$=$\overline{DAV0}$/RDY0 $P3_5$=RDY0/$\overline{DAV0}$

00 $P3_3$=INPUT $P3_4$=OUTPUT
01
10 } $P3_3$=INPUT $P3_4$=DM
11 $P3_3$=$\overline{DAV1}$/RDY1 $P3_4$=RDY1/$\overline{DAV1}$

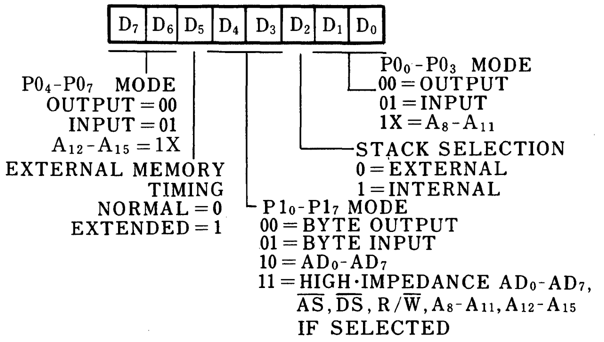
0 $P3_1$=INPUT (T_{IN}) $P3_6$=OUTPUT (T_{OUT})
1 $P3_1$=$\overline{DAV2}$/RDY2 $P3_6$=RDY2/$\overline{DAV2}$

0 $P3_0$=INPUT $P3_7$=OUTPUT
1 $P3_0$=SERIAL IN $P3_7$=SERIAL OUT

0 PARITY OFF
1 PARITY ON

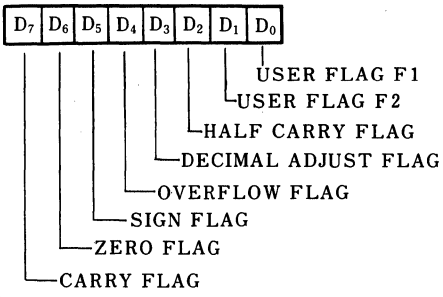
R248 (P01M) Port 0 and 1 Mode Register

(F8H : Write Only)



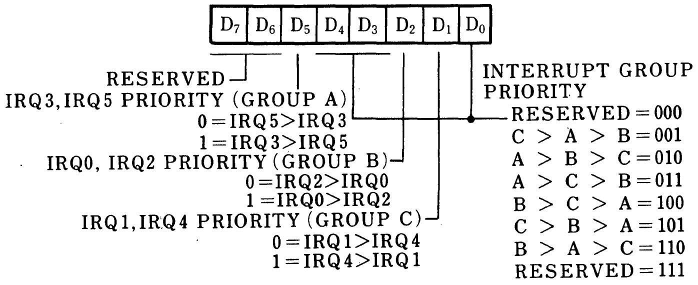
R252 (FLAGS) Flag Register

(FCH : Read/Write)



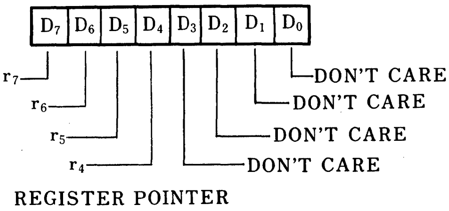
R249 (IPR) Interrupt Priority Register

(F9H : Write Only)



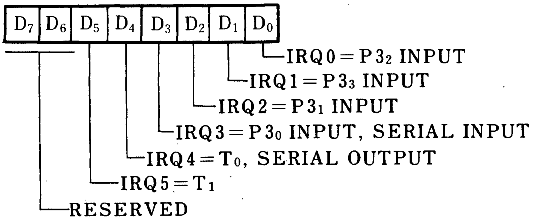
R253 (RP) Register Pointer

(FDH : Read/Write)



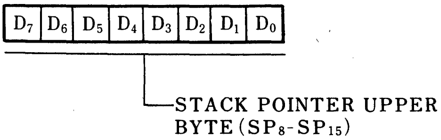
R250 (IRQ) Interrupt Request Register

(FAH : Read/Write)



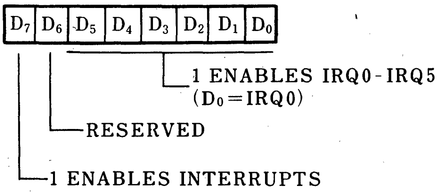
R254 (SPH) Stack Pointer

(FEH : Read/Write)



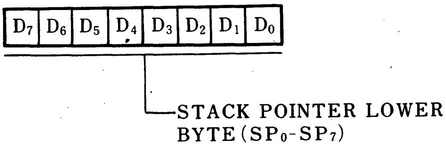
R251 (IMR) Interrupt Mask Register

(FBH : Read/Write)



R255 (SPL) Stack Pointer

(FFH : Read/Write)



LU800V1/LU800AV1/LU805BV2

CMOS 8Bit Single Chip Microcomputers (ROM less)

■ Description

The LU800V1/LU800AV1/LU805BV2 is a ROMless version of the SM803/A and SM805/A CMOS 8-bit single-chip microcomputers and offers the outstanding feature of the Z8 family architecture.

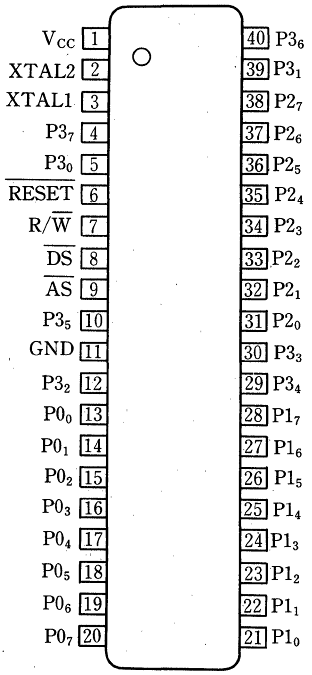
Because some I/O ports are used for address/data bus, this device accesses up to 128K bytes of the external memory space. Using the external memory in place of an on-chip ROM allows designing more powerful microcomputer system.

■ Features

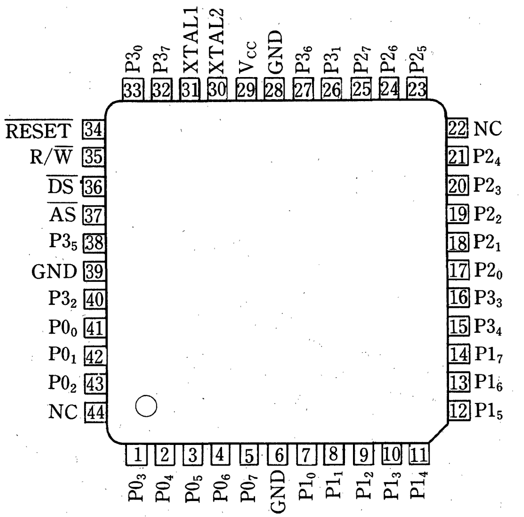
1. Complete microcomputer, 24 I/O lines, and up to 64K bytes addressable external space each for program and data memory.
2. 143 bytes register file
   (255 bytes register file for the LU805BV2)
   124 general-purpose registers
   (236 registers for the LU805BV2)
   3 I/O port registers
   16 status and control registers
3. Register pointer so that short, fast instructions can access any one of the 9 working-register groups.
   (16 groups for the LU805BV2)

■ Pin Connections

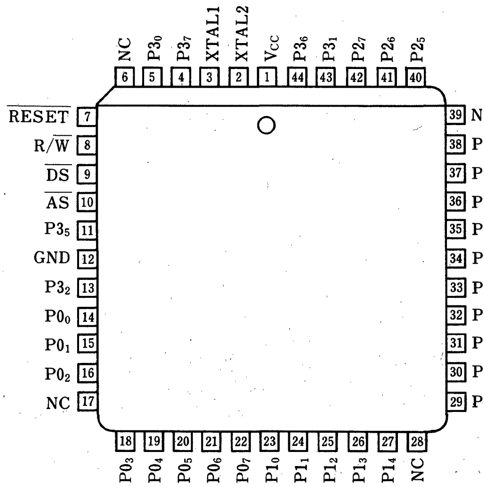
LU800V1/LU800AV1/LU805BV2



LU800V1M/LU800AVM/LU805BVM



LU800V1U/LU800AVU/LU805BVU

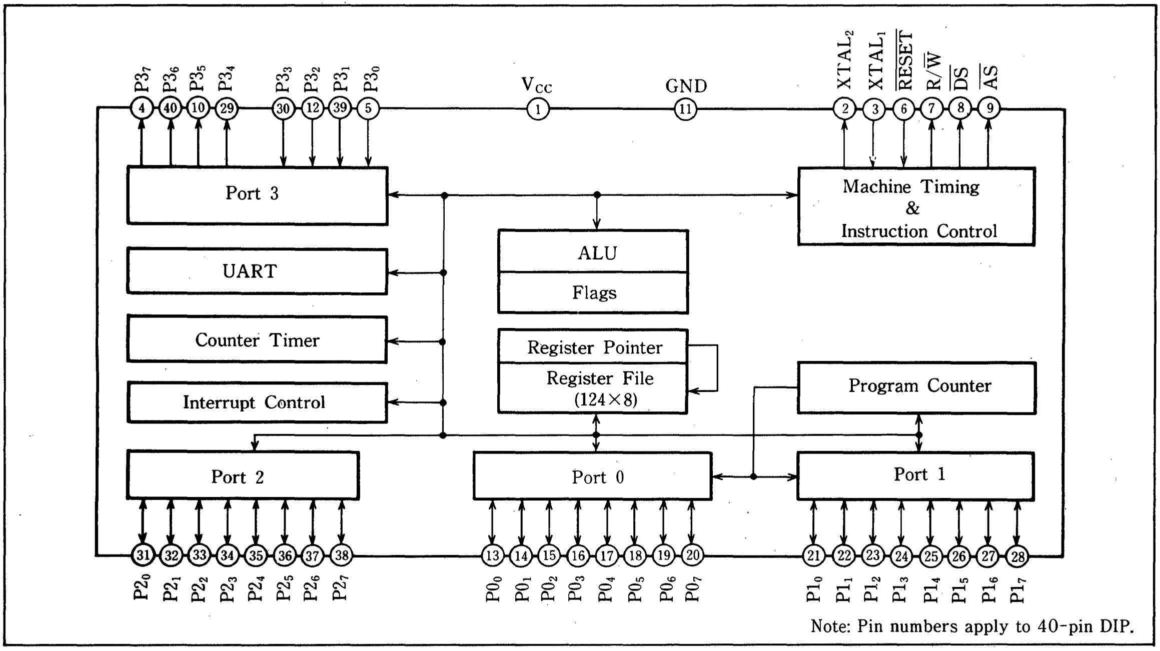


4. Full-duplex UART and two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler.
5. Vectored priority interrupts for I/O, counter/timers, and UART.
6. On-chip oscillation circuit
7. External clock
   8MHz MAX. (internal 4MHz): LU800V1/M
   12MHz MAX. (internal 6MHz): LU800AV1/M
   16MHz MAX. (internal 8MHz): LU805BV2/M
8. Single +5V power supply
9. 40-pin DIP (DIP40-P-600)
   LU800V1/LU800AV1/LU805BV2
   44-pin QFP (QFP44-P-1414)
   LU800V1M/LU800AVM/LU805BVM
   44-pin QFJ (QFP44-P-S650)
   LU800V1U/LU800AVU/LU805BVM

Ordering Information

Model No.	Clock	Package
LU800V1	8MHz	40DIP
LU800V1M		44QFP
LU800V1U		44QFJ
LU800AV1	12MHz	40DIP
LU800AVM		44QFP
LU800AVU		44QFJ
LU805BV2	16MHz	40DIP
LU805BVM		44QFP
LU805BVU		44QFJ

Block Diagram


P3$_7$ P3$_6$ P3$_5$ P3$_4$ P3$_3$ P3$_2$ P3$_1$ P3$_0$
4 40 10 29 30 12 39 5
V$_{CC}$
1
GND
11
XTAL$_2$ XTAL$_1$ $\overline{RESET}$ R/$\overline{W}$ $\overline{DS}$ $\overline{AS}$
2 3 6 7 8 9
Port 3
UART
Counter Timer
Interrupt Control
ALU
Flags
Register Pointer
Register File
(124×8)
Machine Timing
&
Instruction Control
Program Counter
Port 2
Port 0
Port 1
31 32 33 34 35 36 37 38
P2$_0$ P2$_1$ P2$_2$ P2$_3$ P2$_4$ P2$_5$ P2$_6$ P2$_7$
13 14 15 16 17 18 19 20
P0$_0$ P0$_1$ P0$_2$ P0$_3$ P0$_4$ P0$_5$ P0$_6$ P0$_7$
21 22 23 24 25 26 27 28
P1$_0$ P1$_1$ P1$_2$ P1$_3$ P1$_4$ P1$_5$ P1$_6$ P1$_7$

Note: Pin numbers apply to 40-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
P0$_0$-P0$_7$	Port 0	I/O	8-bit I/O port, programmable for I/O.
P1$_1$-P1$_7$	Address/data bus	I/O	Multipiexed Address/data bus
P2$_0$-P2$_7$	Port 2	I/O	Programmable for I/O in bits.
P3$_0$-P3$_7$	Port 3	I/O	P3$_0$-P3$_3$ for input, P3$_4$-P3$_7$ for output.
$\overline{AS}$	Address Strobe	O	Active "Low", activated for external address memory transfer.
$\overline{DS}$	Data Strobe	O	Active "Low", activated for external data memory transfer.
R/$\overline{W}$	Read/Write	O	Read at "High", Write at "Low".
$\overline{RESET}$	Reset	I	Active "Low", Initializes.
XTAL1	Clock 1	I	Clock terminal pin.
XTAL2	Clock 2	O	Clock terminal pin.

Pin functions of the LU800V1/LU800AV1/LU805BV2 are identical to those of the SM803/A, SM805/A, except for pins P1$_0$-P1$_7$.

■ Address space

(1) Program Memory

The ROMless device, having a 16-bit program counter, addresses 64K-bytes of external program memory. All the command codes are fetched from these external program memories.

For the ROMless device, the first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Program execution begins at location 000C$_H$ after a resed.

(2) Data Memory*

The ROMless device can address 64K bytes of external data memory. External data memory may be included with or separated from the external program memory space. $\overline{DM}$, an optional I/O function that can be programmed to appear on pin P3$_1$, is used to distinguish between data and program memory space.

(3) Register File

The 143-byte register file inculdes three I/O port registers (R0, R2, R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255).

These registers are assigned the address locations shown in Fig 2.

The instructions can access registers directly or indirectly with an 8-bit address field. This also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In the 4-bit mode. the regiter file is divided into nine working register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

(4) Stacks

Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124/236 for the LU805BV2 general-purpose registers (R4-R127/R4-R239).

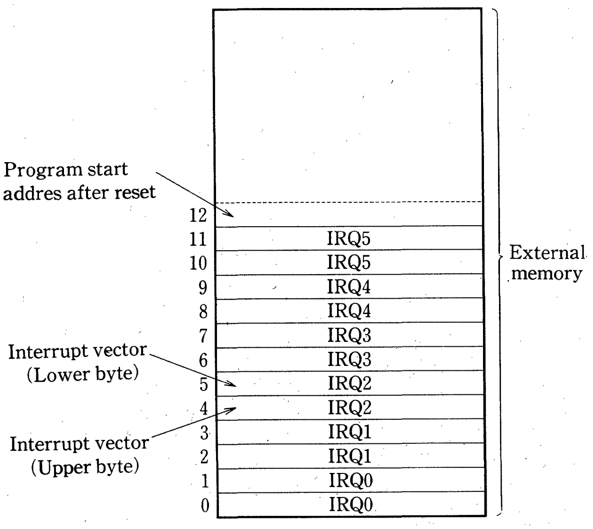
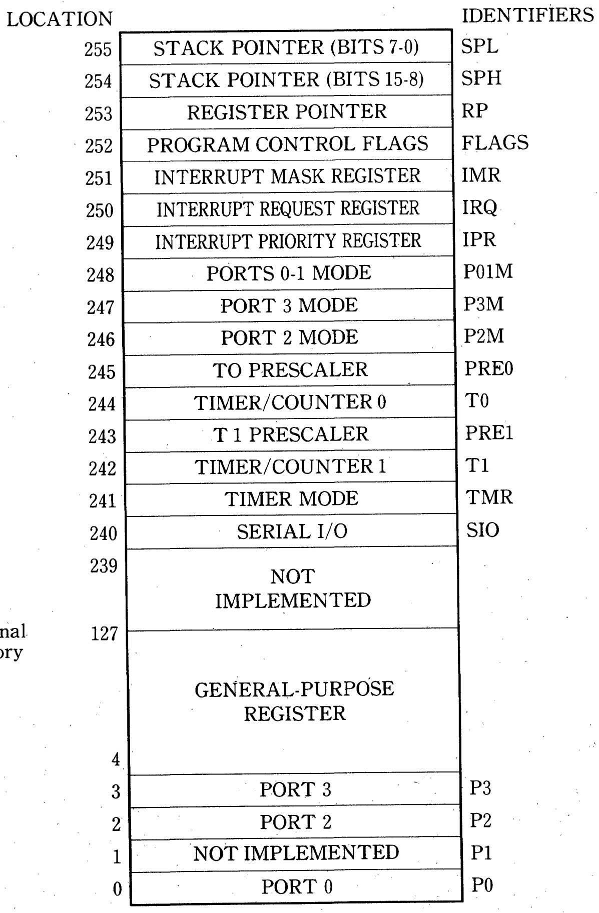


Fig. 1 Program memory map



LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7-0)	SPL
254	STACK POINTER (BITS 15-8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT REQUEST REGISTER	IRQ
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P01M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	TO PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T 1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
239	NOT IMPLEMENTED	
127 4	GENERAL-PURPOSE REGISTER	
3	PORT 3	P3
2	PORT 2	P2
1	NOT IMPLEMENTED	P1
0	PORT 0	P0

Fig. 2 The register file

Port Functions

The LU800V1/LU800AV1/LU805BV2 has a dedicated memory interface port (Port 1) and input/output ports (Port 0, 2, 3). These ports are given eight lines each. The functions of port 0, 2 and 3 are the same as those of the SM803/A, SM805/A.

Port 1 is a dedicated Z-bus compatible memory interface. The operations of Port 1 are supported by the Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) lines, and by the Read/Write ($R/\overline{W}$) and Data Memory ($\overline{DM}$) control lines.

The low-order program and data memory address (A_0-A_7) are output through Port 1 and are multiplexed with data in/out (D_0-D_7). Instruction fetch and data memory read/write operations are done through this port.

Port 1 cannot be used as a register nor can a handshake mode be used with this port.

If more than address lines are required with the ROMless device, additional lines can be obtained by programming Port 0 bits as address bits. The least significant four bits of Port 0 can be configured to supply address bits A_8-A_{11} for 4K byte addressing or both nibbles of Port 0 can be configured to supply address bits A_8-A_{15} for 64K byte addressing.

Registers

The LU800V1/LU800AV1/LU805BV2 control registers are the same as on the SM803/A, SM805/A, except two bits D_3 and D_4 in the port 0, 1 mode register (R248).

Serial Input/Output

The LU800V1/LU800AV1/LU805BV2 serial input/output functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

Counter/Timers

The LU800V1/LU800AV1/LU805BV2 counter/timer functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

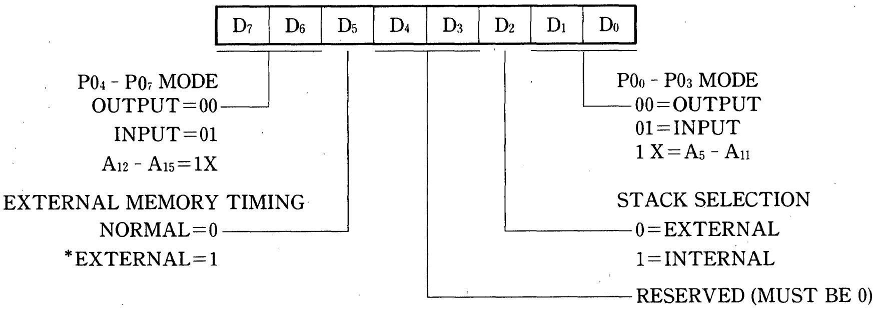
Interrupts

The LU800V1/LU800AV1/LU805BV2 interrupt functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

Instructions and AC/DC Characteristics

These data of the LU800V1/LU800AV1/LU805BV2 are the same as for the SM803/A, SM805/A. (Refer back to the SM803/A, SM805/A description.)

R248 (P01M) Port 0, 1 Mode Register ($F8_H$ Write only)



Reset

When the NU800V1 is reset, the device must be kept Low for at least 50msec from the device is stabled with the reset switch is turned on, or for 18 clock cycles from the power supply and clock oscillator are stabled.

The intervals reset the LU800Vl is obtained by connecting external capacitor of 1μF and resistor of 100kΩ as shown in Fig. 3.

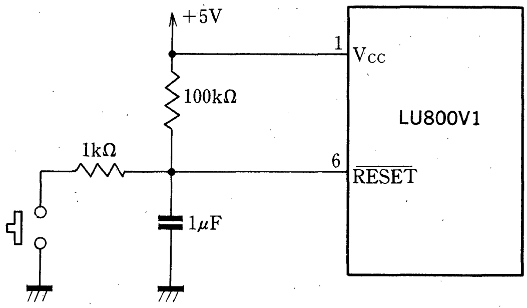


Fig. 3

After reset, ports 0 and 2 are used as input ports, the program counter is reset at 000_H and the interputs are disabled.

When the reset input inactivated, the program memory starts execution at $000C_H$.

Initialization

When the program, after reset, starts execution at $000C_H$, the device must be initialized. Ports 0 and 2 after reset are used as inputs, and an expanded memory timing and an internal stack are selected with the $\overline{\text{DM}}$ signal not to be output. The valid address lines include 8 lines of port 1 only. Usable memory sholud be limited to the first 256 bytes and the port 0 must be programmed as address lines within 256 bytes of memory for use of more than 257 bytes of memory.

Port 0, $P0_0$-$P0_7$, is used as input port after reset, if is used as address lines, a constant address value must be held with an external circuit until it is initialized.

The port initialization sequence is:

(1) Write upper byte of address of an initialization routine to port 0 register.

(2) Configure port 0 and 1 mode register P01M. ($D_1=1$; lower 4 bits of port 0 should be A_8-A_{11} address lines, $D_7=1$ every bit of port 0 should be A_8-A_{15} address lines.)

Note: While the next byte of instruction indicated in the above item (2) are being fetched, be sure not to make a difference between an address output from port 0 and an address held in an external circuit. (This is because the instruction is executed in pipeline system.)

Initiflization with Pull-up Resistors

When connecting the lower bits $P0_0$-$P0_3$ of port 0 to a 4K byte memory with a pull-up resistor, Fig.4 shows that the addresses A_8-A_{11} are kept during the port 0 is an unknown state, and must be phisically located in the latter address of $F0C_H$ of a 4K byte memory. The A_8-A_{11} will change according to the address output from port $P0_0$-$P0_3$, if the port 0 is used as address lines.

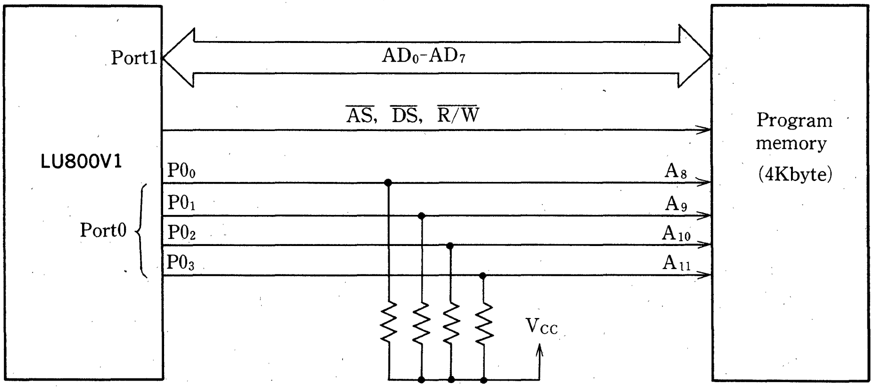


Fig. 4 Memory interface with pull-up resistors

Initialization of port 0

(1) Jump to the address FXX_H in order to match the program counter to the address being accessed.

(2) Write $0F_H$ (upper byte of the address)into port 0 register at FXX_H.

(3) Set the proper bits in the port 0 and 1 mode register to output the port 0.

(4) Set the proper bits in the port 0 and 1 mode register to use the port 0 as address lines.

Initialization with The LS157

Fig. 5 shows the memory interface between upper 4 bits ($P0_0$-$P0_3$) of port 0 and a 4K byte memory with the LS157 and a flip-flop.

After reset, in this case, the "b" inputs are selected and address bits A_8-A_{11} go Low because the SELECT input to the LS157 is kept High until the $R/\overline{W}$ goes Low. If the $R/\overline{W}$ goes Low, the SELECT goes Low and $P0_0$-$P0_3$ will be valid.

Initialization of port 0

(1) Write 00_H (upper byte address of initialization routine) to port 0 register.

(2) Set the proper bits in the port 0 and 1 mode registers.

(3) Write into the external memory upon execution of an LDC or LDE intruction. (This allows the $R/\overline{W}$ to go Low and the LS157 to switch to the "a" inputs.

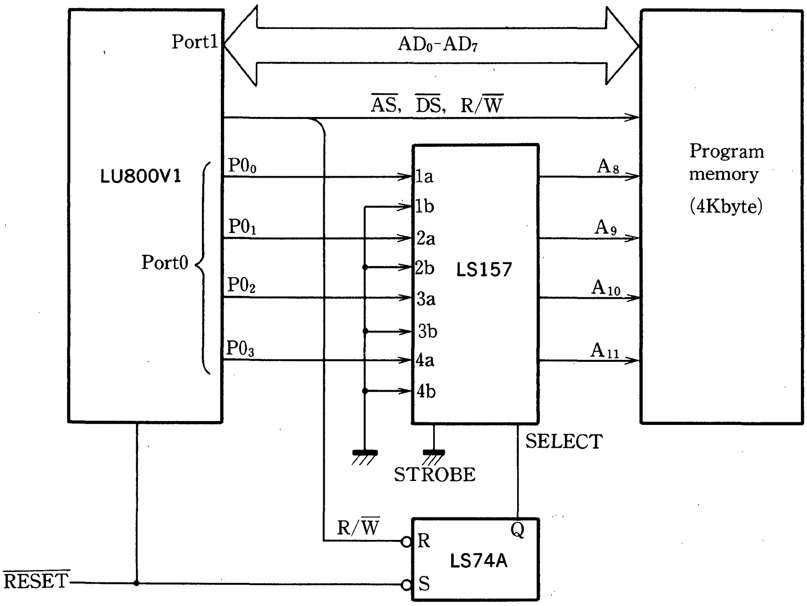


Fig. 5 Memory interface with the LS157

SM8202/SM8203 8-Bit Microcomputer (VCR System Controller)

Description

The SM8202/SM8203 is an 8-bit microcomputer which integrates an 8-bit CPU core, a ROM, a RAM, serial I/O ports, a timer, an A/D converter and a digital servo controller in a single chip.

An on-chip CPU core is organized as a new architecture with a full lineup of instruction sets, which allows the software to be easily developed.

An on-chip servo controller contains the hardware for controlling the speed and the phase of capstan and drum motors as well as for special playback with a software control. This microcomputer is applicable to a variety of VCR systems for an NTSC, a PAL, a movie, a 4-head mechanism, etc.

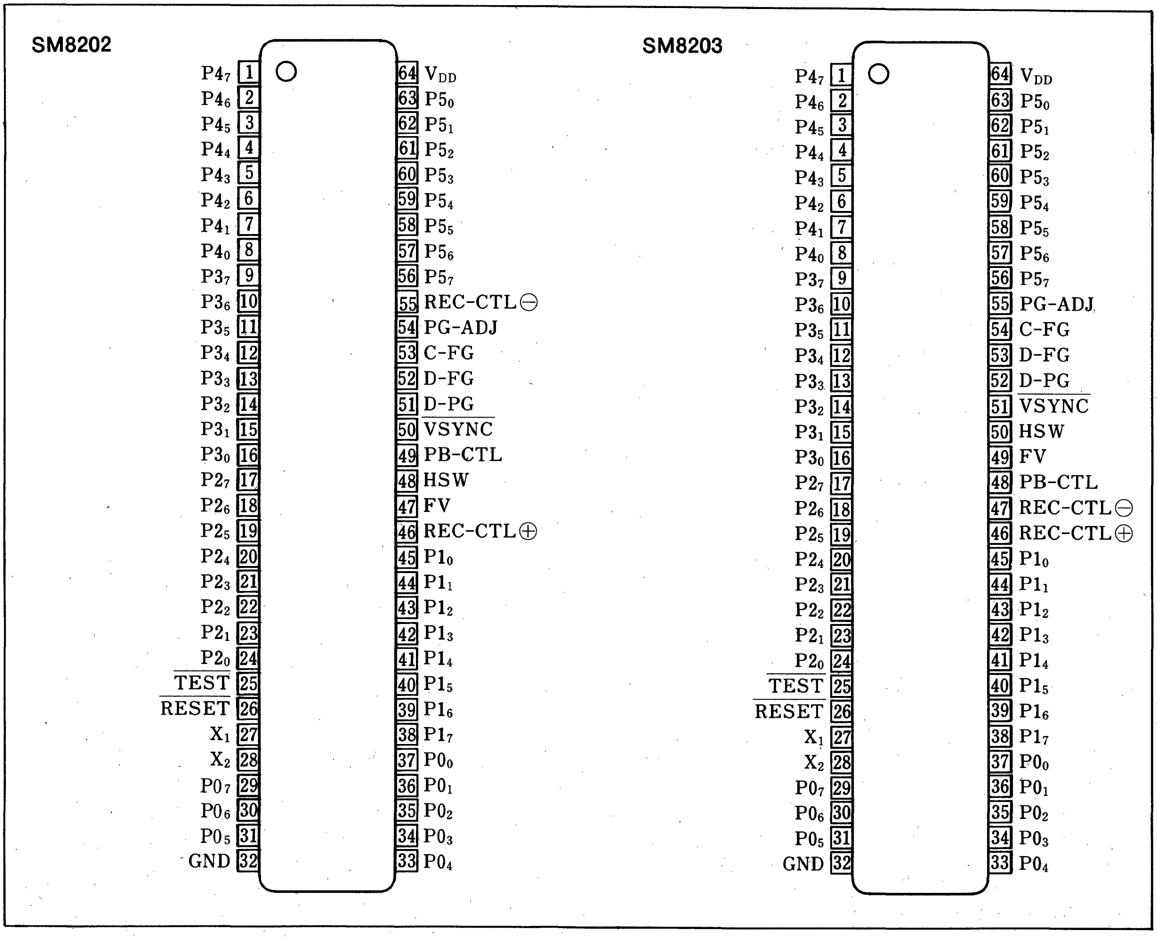
The SM8203 omits the CTL duty discriminating output circuit (VISS) and adds the following two functions from the SM8202.

- An internal 4-head switching circuit for special playback of slow and search modes decreases external parts count.
- Switching the CTL signal with the timer input simplifies programming of a real time count.

Features

1. CMOS process
2. ROM capacity: 10,240 × 8 bits
3. RAM capacity: 256 × 8 bits
4. SM82 core

Pin Connections

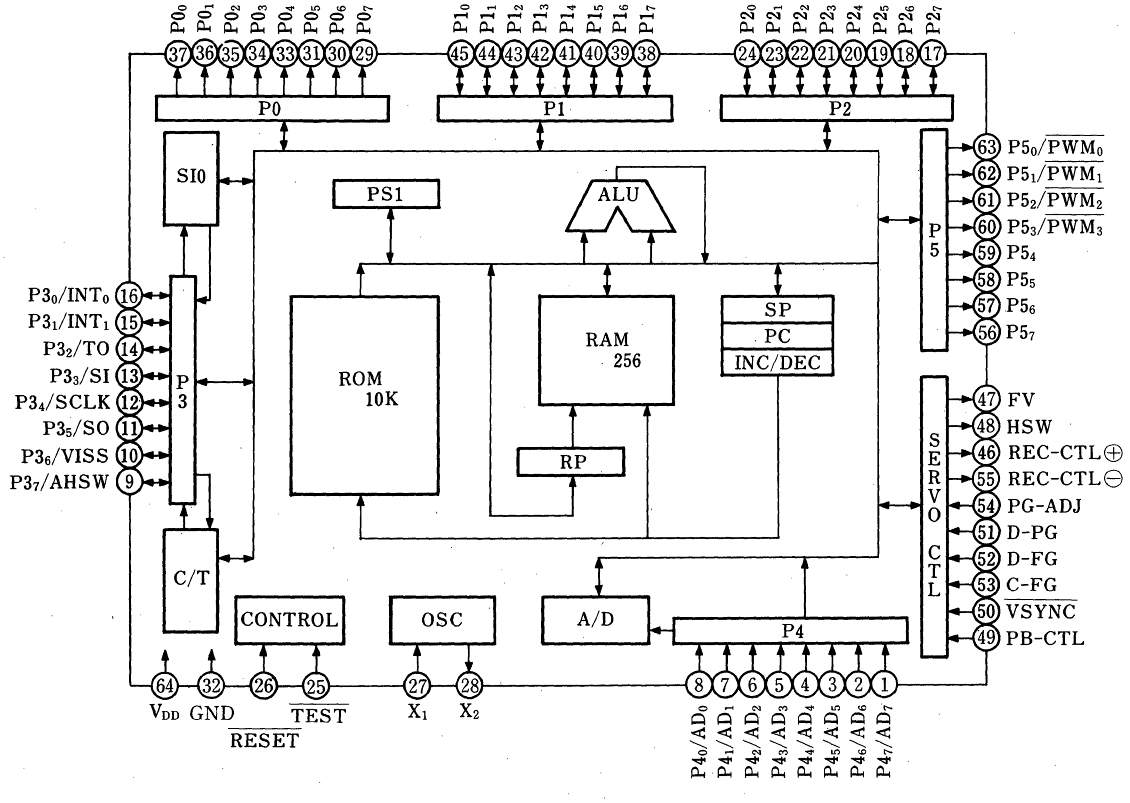


- Instruction set: 64 (including for multiple and division function, a bit manipulation)
- Addressing mode: 22
- General-purpose register: 8-bit × 8
  16-bit × 4

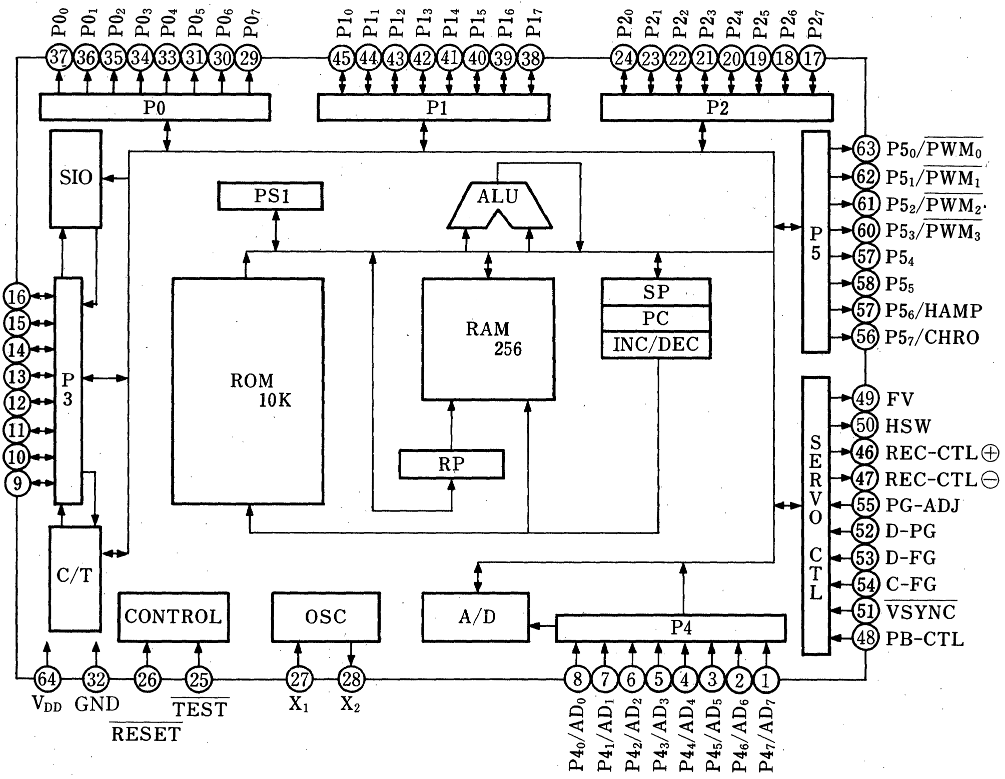
5. An on-chip servo controller
   - Applicable to NTSC, PAL systems
   - PWM: 10 bits/4 channels
6. Applicable to a 4-head VCR
7. Instruction cycle: 0.8 μs (MIN.)
8. Interrupt
   - External interrupts: 2
   - Internal interrupts: 8
9. Input/output ports
   - I/O ports: 24
   - Input ports: 8 (for switching with A/D converter)
   - Output ports: 16 (for switching with a servo controller)
10. Serial I/O
11. Timer: 8 bits × 3
12. A/D covnerter: 8 bits/8 channels
13. Built-in watchdog timer
14. Built-in crystal oscillation circuit
15. Supply voltage: 5V±10%
16. 64-pin SDIP (SDIP64-P-750)

Block Diagram

SM8202



SM8203


P0₀ P0₁ P0₂ P0₃ P0₄ P0₅ P0₆ P0₇
37 36 35 34 33 31 30 29
P1₀ P1₁ P1₂ P1₃ P1₄ P1₅ P1₆ P1₇
45 44 43 42 41 40 39 38
P2₀ P2₁ P2₂ P2₃ P2₄ P2₅ P2₆ P2₇
24 23 22 21 20 19 18 17
P0
P1
P2
SIO
PS1
ALU
P5
63 P5₀/PWM₀
62 P5₁/PWM₁
61 P5₂/PWM₂
60 P5₃/PWM₃
57 P5₄
58 P5₅
57 P5₆/HAMP
56 P5₇/CHRO
P3₀/INT₀ 16
P3₁/INT₁ 15
P3₂/TO 14
P3₃/SI 13
P3₄/SCLK 12
P3₅/SO 11
P3₆/AHSW 10
P3₇/ENV 9
P3
ROM 10K
RAM 256
SP
PC
INC/DEC
RP
SERVO CTL
49 FV
50 HSW
46 REC-CTL⊕
47 REC-CTL⊖
55 PG-ADJ
52 D-PG
53 D-FG
54 C-FG
51 VSYNC
48 PB-CTL
C/T
CONTROL
OSC
A/D
P4
64 V_DD
32 GND
26 RESET
25 TEST
27 X₁
28 X₂
8 7 6 5 4 3 2 1
P4₀/AD₀ P4₁/AD₁ P4₂/AD₂ P4₃/AD₃ P4₄/AD₄ P4₅/AD₅ P4₆/AD₆ P4₇/AD₇

Pin Description

Signal	I/O	Function	
$P0_0$-$P0_7$	O	Output ports (medium voltage)	
$P1_0$-$P1_7$	I/O	I/O ports (resettable I/O on each bit)	
$P2_0$-$P2_7$	I/O	I/O ports (resettable I/O on each bit)	
$P3_0/INT_0$, $P3_1/INT_1$	I/O / I	I/O ports/External interrupt input port	
$P3_2$/TO	I/O / O	I/O port/Timer output port	
$P3_3$/SI	I/O / I	I/O portt/Serial input port	
$P3_4$/SCLK	I/O	I/O port/Serial clock	
$P3_5$/SO	I/O / O	I/O port/Serial output port	
$P3_6$/VISS	I/O / O	I/O port/CTL duty output port for VISS	SM8202
$P3_6$/A-HSW	I/O / O	I/O port/Audio HSW	SM8203
$P3_7$/A-HSW	I/O / O	I/O port/Audio HSW	SM8202
$P3_7$/ENV	I/O / I	I/O port/ENV comparator input port	SM8203
$P4_0/AD_0$-$P4_7/AD_7$	I	Input port/Analog input port	
$P5_0/\overline{PWM_0}$	O	Output port/PWM output port (negative)	
$P5_1/\overline{PWM_1}$	O	Output port/PWM output port (negative)	
$P5_2/\overline{PWM_2}$	O	Output port/PWM output port (negative)	
$P5_3/\overline{PWM_3}$	O	Output port/PWM output port (negative)	
$P5_4$, $P5_5$	O	Output ports	
$P5_6$, $P5_7$	O	Output ports	SM8202
$P5_6$/HAMP	O	Output port/Head amp. control signal output port	SM8203
$P5_7$/CHRO	O	Output port/Chroma rotation signal output port	
PG-ADJ	I	HSW adjustment	
D-PG	I	Drum PG input port	
D-FG	I	Drum FG input port	
$\overline{\text{VSYNC}}$	I	Vertical synchronous signal (negative)	
FV	O	False synchronous signal output port	
HSW	I/O	Head switching pulse	
C-FG	I	Capstan FG input	
PB-CTL	I	PB-CTL input port	
REC-CTL⊕	O	REC-CTL output port	
REC-CTL⊖	O	REC-CTL inverting output port	
X_1, X_2		System clock oscillator	
$\overline{\text{RESET}}$	I	Reset input port	
$\overline{\text{TEST}}$	I	Test input port	
V_{DD}, GND		Power supply, Ground	

Absolute Maximum Ratings

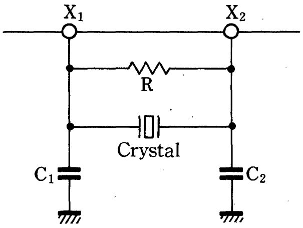
Parameter	Symbol	Condition	Rating	Unit	Note
Supply voltage	V_{DD}		−0.3 to +7.0	V	
Input voltage	V_{IN}		−0.3 to V_{DD}+0.3	V	
Output voltage	V_{OUT}		−0.3 to V_{DD}+0.3	V	
	V_{OUT0}	Applied to port 0	−0.3 to +12	V	
Output current	I_O		20	mA	1
Operating temperature	Topr		−20 to +70	℃	
Storage temperature	Tstg		−55 to +150	℃	

Note 1: The sum of source current or sink current from output ports.

Operating Conditions

Parameter	Symbol	Condition	Rating	Unit
Oscillation frequency	f	V_{DD}=5V±10%	5	MHz

Oscillation circuit



DC Characteristics

(V_{DD}=5V±10%, Ta=−20 to +70℃)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		0.8V_{DD}		V_{DD}	V	1
	V_{IL1}		0		0.2V_{DD}	V	
	V_{IH2}		0.75V_{DD}		V_{DD}	V	2
	V_{IL2}		0		0.15V_{DD}	V	
Input current	I_{IH1}	V_{IN}=V_{DD}			10	μA	3
	I_{IL1}	V_{IN}=0			10	μA	
Output current	V_{OH1}	I_{OH}=−1.5mA	V_{DD}−0.5			V	4
		I_{OH}=−6mA	V_{DD}−2.0			V	
	V_{OL1}	I_{OL}=1.5mA			0.5	V	
		I_{OL}=6mA			2.0	V	
	V_{OH2}	I_{OH}=−4mA	V_{DD}−0.5			V	5
		I_{OH}=−16mA	V_{DD}−2.0			V	
	V_{OL2}	I_{OL}=4mA			0.5	V	
		I_{OL}=16mA			2.0	V	
	V_{OL3}	I_{OL}=2mA	V_{DD}−0.5			V	6
		I_{OL}=8mA	V_{DD}−2.0			V	
	V_{OL4}	I_{OL}=2mA			0.5	V	7
		I_{OL}=8mA			2.0	V	
Current consumption	I_{OP}			10		mA	8

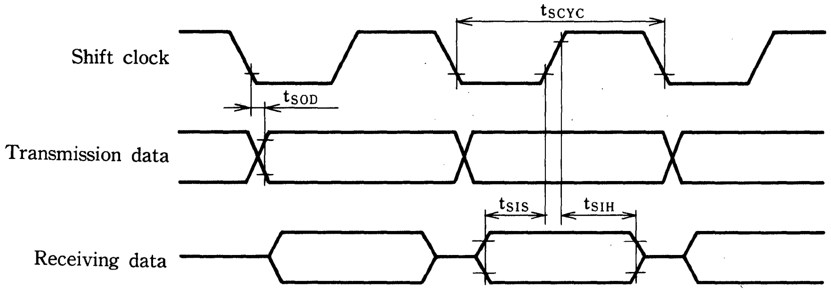
Note 1: Applied to all pins except for D-PG, D-FG, C-FG, PB-CTL
Note 2: Applied to pins D-FG, D-PG, C-FG, PB-CTL
Note 3: Applied to all input pins.
Note 4: Applied to all output pins except for REC-CTL⊕, REC-CTL⊖, PG-ADJ, $P0_0$
Note 5: Applied to REC-CTL⊕
Note 6: Applied to pins REC-CTL⊖, PG-ADJ
Note 7: Applied to pins $P0_0$-$P0_7$
Note 8: No load condition

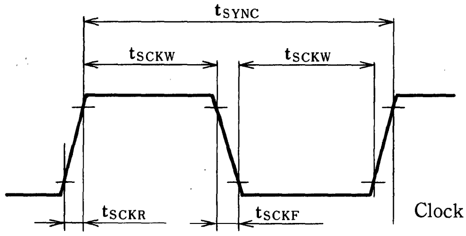
■ AC Characteristics

● SIO Characteristics (V_{DD}=5V±10%, Ta=−20 to +70℃)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit	Note
SIO clock cycle	t_{8CYC}		16			t_{CYC}	1
SIO transmission data delay	t_{SOD}				300	ns	
SIO receiving data setup time	t_{SIS}		300			ns	
SIO receiving data hold time	t_{SIH}				1	t_{CYC}	1
SIO input clock pulse width	t_{SCKW}		0.4		0.6	t_{SCYC}	
SIO input clock pulse width	t_{SCKR}				200	ns	
SIO input clock fall time	t_{SCKF}				200	ns	

Note 1: t_{CYC}: 2×(oscillation frequency)$^{-1}$





Clock timing

■ A/D Converter Accuracy (V_{DD}=5V, Ta=25℃)

Parameter	MIN.	TYP.	MAX.	Unit
Non linearity error			6	LSB
Differential non-linearity			6	LSB
Zero-scale error			6	LSB
Full-scale error			6	LSB
Total tolerance			6	LSB

Pin Function

The device is provided with 58 I/O ports which include 8 × 6 ports and 10 ports only for servo controller.

Figures 1 and 2 show the pin function diagram. The function on each ports can be set with a software.

- I/O ports
- External interrupt input ports
- Serial I/O ports
- Timer output ports
- A/D converter input ports
- PWM output ports
- A-HSW output ports
- VISS output ports (for the SM8202 only)
- HAMP output port (for the SM8203 only)
- CHROMA output port (for the SM8203 only)

(1) Ports P0, P1, P2

Ports P0, P1, P2 are 8-bit I/O ports which can be switched between input and output modes with a directional register.

Port P0 is a medium voltage output port, and serves exclusively as an 8-bit output port.

(2) Port P3

Port P3 is an 8-bit I/O port the same as ports P1 and P2. Port P3 is allocated $P3_0$-$P3_1$ for interrupt input ports, P3 for a timer output port, $P3_3$-$P3_5$ for serial I/O ports. For the SM8202, the $P3_6$ is allocated for a VISS output port and $P3_7$ for an A-HSW output port. For the SM8203, the $P3_6$ is allocated for the A-HSW output, and the $P3_7$ is allocated for the ENV comparator input port for $P5_6$/HAMP, $P5_7$/CHRO outputs, when used as the input port.

(3) Port P4

Port P4 is an only input port. This port can be

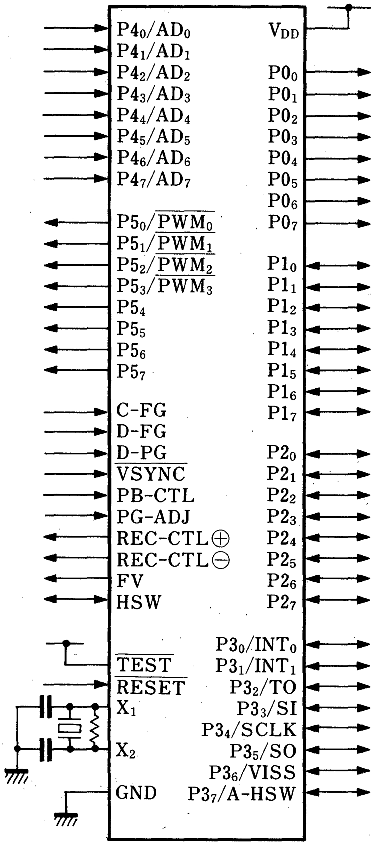


Fig. 1 SM8202 pin functions

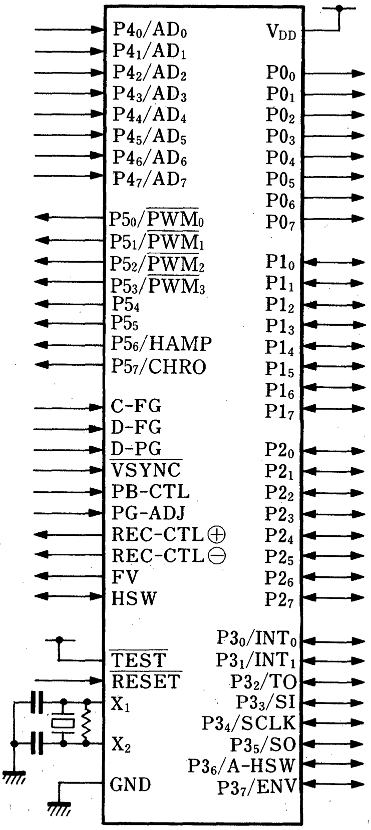


Fig. 2 SM8203 pin functions

set to an analog data input port of A/D converter with a program control.

(4) Port P5

Port P5 is only output port. Each bit of $P5_0$-$P5_3$ may be set to PWM output mode.

For the SM8203, the $P5_6$/HAMP may be set to the HAMP switching signal output port, and the $P5_7$/CHROMA to the CHROMA switching signal output port.

(5) C-FG, D-FG, DPG

The C-FG pin is used to input the signals from the FG (frequency generator) of a capstan motor, and control the speed and the phase, compared to the reference signals.

The D-FG pin is used to input the signals from FG of a drum motor, and control the speed, compared to the reference signals.

The D-PG pin is used to input the signals from the PG (pulse generator) of a drum motor, and control the drum phase, compared to the reference signals.

(6) PB-CTL, REC-CTL⊕, REC-CTL⊖

The PB-CTL pin is used to input the signals for the speed control of a capstan motor when playback.

The REC-CTL+ and REC-CTL− control pins are used to record the control signals when recording.

(7) HSW, PG-ADJ

The HSW pin outputs the head switching signals or inputs the head switching pulses.

The PG-ADJ input pin is used to contorl the phase of a head positionning with an external capacitor and a variable resistor.

(8) V_{SYNC}, FV

The V_{SYNC} pin inputs the vertical synchronous signals, and the FV pin outputs the synchronous signals when a trick motion.

Hardware Configuration

(1) Address space

The device contains an internal RAM, an I/O register and a status register which are so-called register file. The registers are located in the 64K-byte address at the same memory space with the program memory. Fig. 3 shows the address space.

(2) Program memory (ROM)

The program memory space is allocated in the addresses from 1000 to FFFF within a program/ register memory. The first 10K bytes of the program memory is a mask ROM within a chip. 16 interrupt vectors should be inserted into the addresses from 1000 to 101F. After the device is reset, user's program starts execution at the address 1020 (see Fig. 4).

(3) Register file

The addresses from 0000 to 00FF are allocated for a register file, and 0100 to 0FFF for an internal RAM expansion.

The register file consists of a 16 bit general--purpose register, a 7 bit I/O register, a 256 byte internal RAM and some control registers (see Fig. 5). The file may be accessed from an 8 bit address field.

(4) Register configuration

General-purpose registers R_0-R_7 can be used as an 8 bit register as well as a 16 bit register with a couple of registers. Registers R_8-R_{15} can be used as a 16 bit register with a couple of registers.

(5) Interrupt

The device has 10 different interrupt functions (see Table 1), and the priority order should be 7→6→...2→1 shown in table 1. For the interrupt inhibit, the device accepts the interrupt with higher priority than that specified by an interrupt mask IM of a processor status 0 (PS0).

Reseting the bit "I" of the processor status 1 (PS1) through DI instruction inhibits all of maskable interrupts (see Fig. 6).

(6) Reset function

Applying a High level signal to the RESET pin resets the internal logic of the device and starts execution of the program at address 1020.

After reset, the following blocks are initialized.

- The output port is set to "0", and I/O ports may be placed in input mode.
- The interrupt enable flag is reset.
- The peripheral I/O registers are initialized.

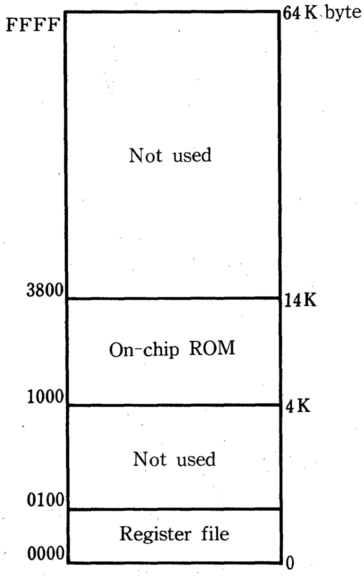


Fig. 3 Address architecture

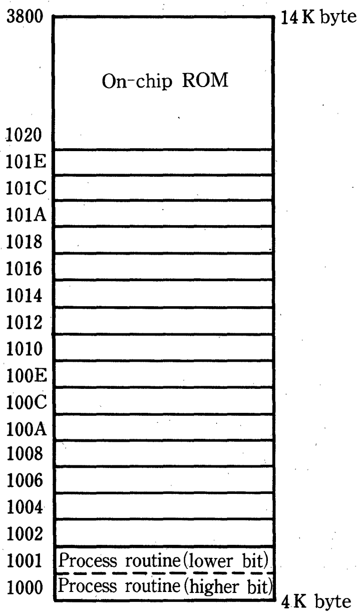
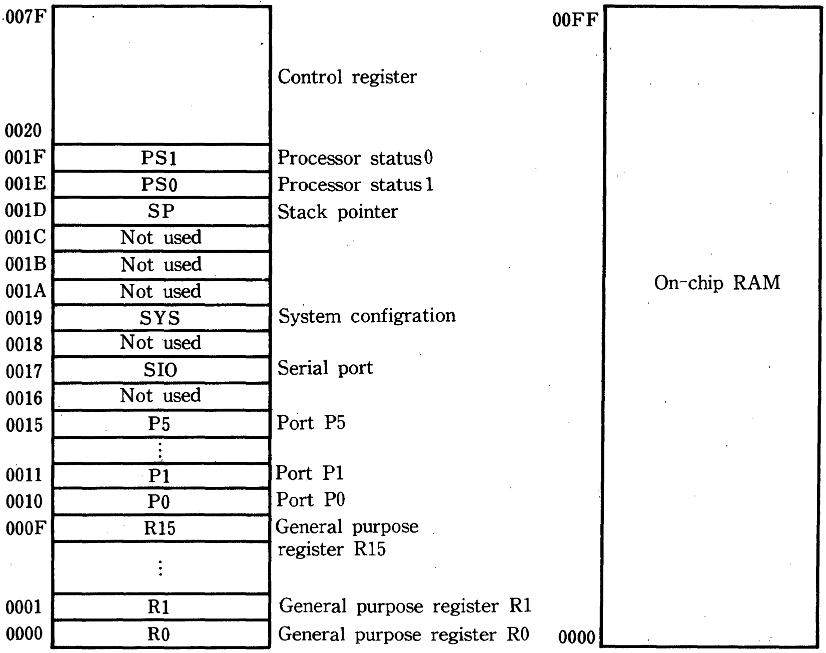


Fig. 4 Program memory map



Address	Register	Description
007F 0020		Control register
001F	PS1	Processor status0
001E	PS0	Processor status1
001D	SP	Stack pointer
001C	Not used	
001B	Not used	
001A	Not used	
0019	SYS	System configration
0018	Not used	
0017	SIO	Serial port
0016	Not used	
0015	P5	Port P5
	⋮	
0011	P1	Port P1
0010	P0	Port P0
000F	R15	General purpose register R15
	⋮	
0001	R1	General purpose register R1
0000	R0	General purpose register R0
00FF 0000	On-chip RAM	

Fig. 5 The register file

Table 1 Interrupt

Vector location	Source	Priority
1000	External interrupt INT0	7 (Most priority)
1002	External interrupt INT1	4
1006	Watchdog timer	—
1008	Timer T0	6
100A	Timer T1	3
100C	Timer T2	3
100E	SIO	1
1012	A/D converter	2
1014	Servo controller	5
101E	Irregal instruction	—

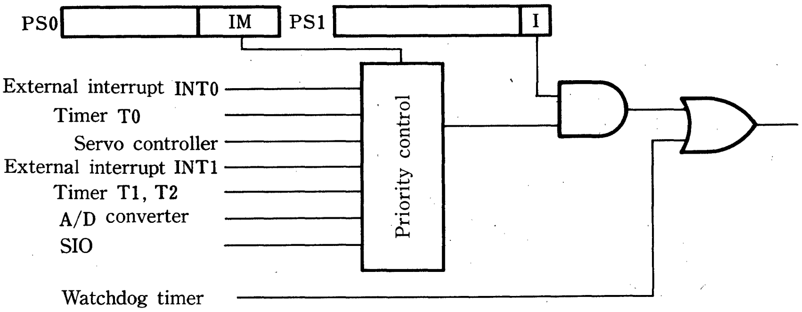


Fig. 6 Interrupt mask

(7) A/D converter

The control register ADC allows 8 bit input ports $P4_0$-$P4_7$ to be selected from the input ports, the A/D converters or the comparators.

- Input ports

The port status with a digital data may be read out from the input ports.

- A/D converter

The analog data of any one of channels $P4_0$-$P4_7$ specified by the ADC is converted into the digital data through the A/D converter, the result of A/D conversion may be read out as a data register of port P4. The A/D converter may be started with a bit manipulation of ADC. Then the CPU acknowledges the interrupt if it is enabled. The A/D conversion is executed by the comparison between the analog input and the voltage based upon the ladder resistor applied between V_{CC} and GND. The conversion time should be 68μs under 4MHz of the clock frequency.

- Comparator mode

Selecting the comparator mode writes data into the internal register instead of the data register of port P4. The D/A conversion of the contents of an internal register is executed to compare with an analog data. The corresponding 1 bit of the ADC is set or reset specify the result of comparison.

(8) Serial I/O (SIO)

The serial I/O port transfers and receives an 8-bit data in synchronization with the shift clock. The serial I/O consists of a couple of registers, an octal counter and some controllers.

(9) Timer

The device contains 4 timers including three 8-bit interval timers and a watchdog timer which may be selected by an 8-bit select register (TS).

A 14-bit prescaler commonly used for each timer has the output which becomes the input clock at each timer. The input timer is a clock (ϕ_{11}) equivalent to the reference clock divided into 2. The corresponding input clock at each timer may be set with a program.

Note that the timer 0 of the SM8203 can also be used as a counter which counts the input clock of the CTL signal.

(10) Sound output ($P3_2$/TO)

The $P3_2$/TO may be switched to either an I/O port or a prescaler output port through the select register (TS). A 4kHz clock for the sound output is output from the $P3_2/T_0$.

(11) Timing

The internal clock of the device is a half frequency of the reference clock. The read cycle of the internal ROM is generated with 2 clocks, and that of the internal RAM or I/O register with 1 clock.

A high speed operation is obtained from the function that the operation code fetch overlaps the execution cycle and the next instruction operation code is taken during execution of one instruction cycle.

Servo Control Function

The device is provided with the hard block for exclusive use of servo control, which offers flexible servo control with a software. The hard block requires less software and performs servo control by simple load instructions and arithmetic instructions. Figures 7 and 8 show the block diagram of a servo controller.

(1) Drum servo speed comparator (D-AFC)

The drum speed signal (D-FG) is compared to the reference clock to calculate the tolerance.

(2) Drum phase comparator (D-APC)

The drum phase comparator signal (D-PG) is compared to the V_{SYNC} or reference clock to calculate the tolerance.

(3) Capstan speed comparator (C-AFC)

The capstan speed signal (C-FG) is compared to the reference clock to calculate the tolerance.

(4) Capstan phase comparator (C-APC)

The capstan phase comparator signal (HSW) is compared to the divided signal C-PG of either the CTL or C-FG signal to calculate the tolerance.

(5) Head switching circuit (HSW)

The head switching circuit generates the head switching pulse (HSP) necessary for switching a drum head, based upon FG and PG signals of a drum. The mono multicircuit of a register allows a digital tracking control of a delay from the signal to the head with the head switching pulse to be delayed.

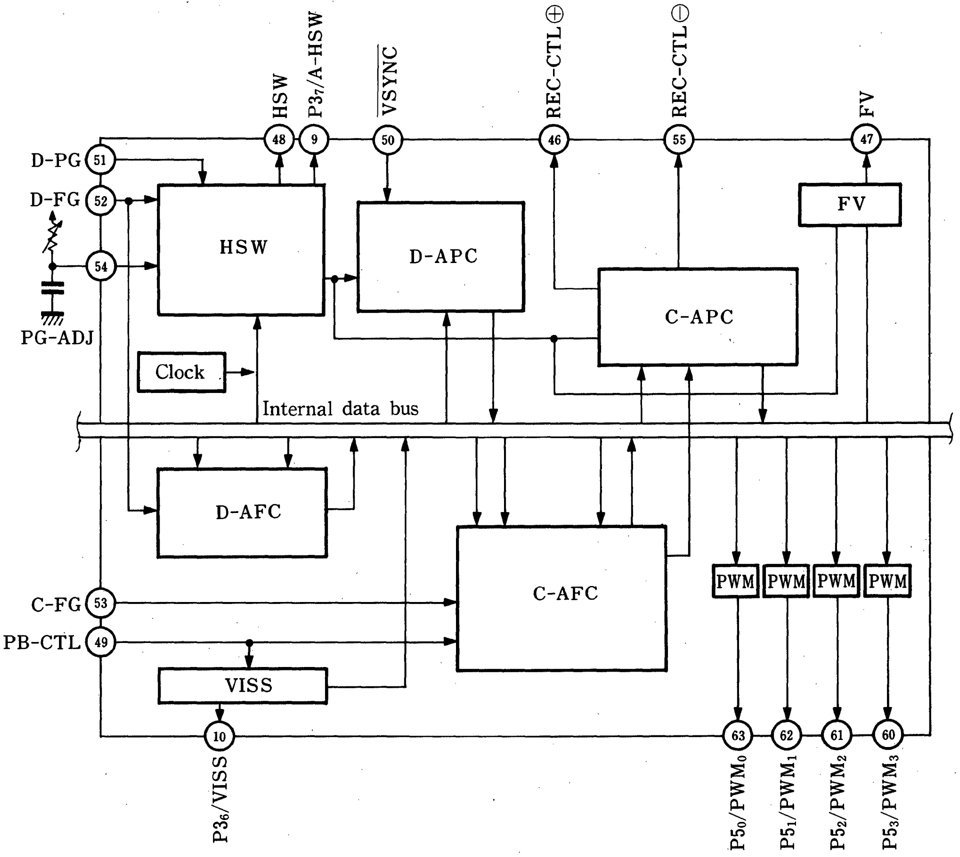


Fig. 7 SM8202 servo control block

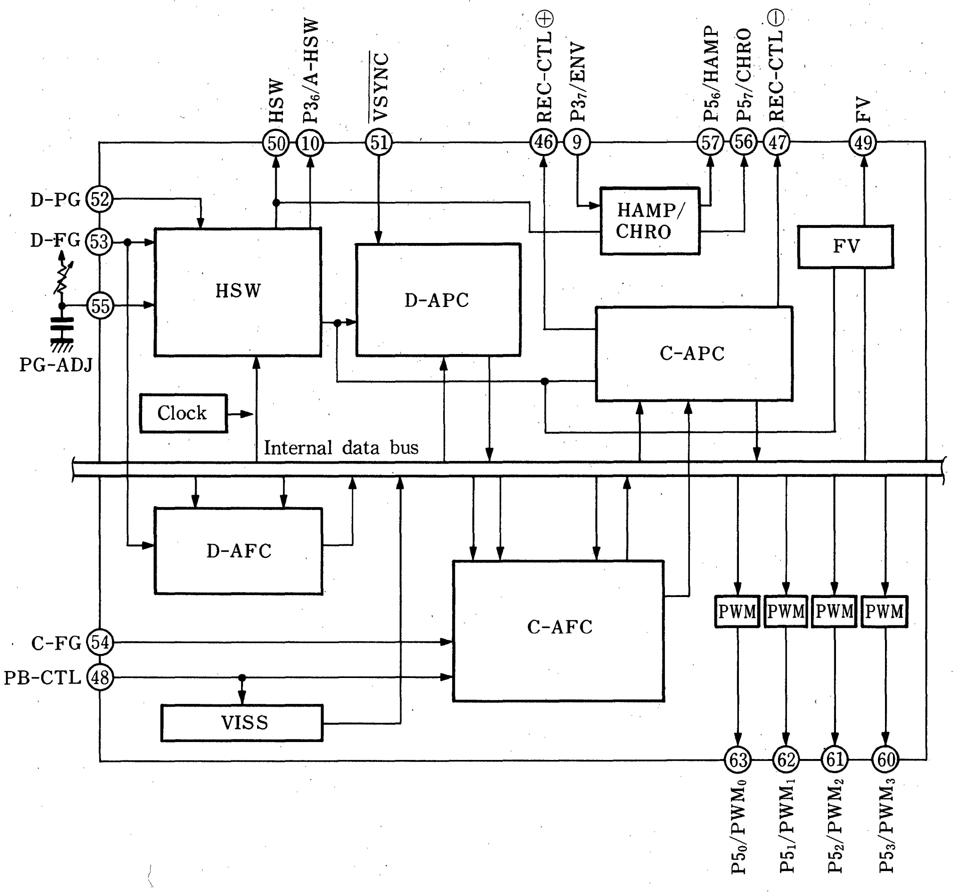


Fig. 8 SM8203 servo control block

(6) False synchronous circuit block (FV)

The false synchronous circuit block sets and outputs the intervals of programmable 3 levels of High, Low and High impedance, referenced to the rising edge and falling edge of a head switching pulse.

(7) Pulse width mudulation output block (PWM)

4 sets of the pulse width modulation (PWM) output circuits constitute a digital filter through an internal arithmetic operation, which may be allocated 2 sets for a drum and a capstan, or 4 sets for D-AFC, D-APC, C-AFC and C-APC.

The PWM signal performs a modulation by 1 bit rate of a 10 bit differential data divided by upper 7 bits and lower 3 bits. And 1 pulse is incremented to the PWM signal of upper 7 bits according to the lower 3-bit data. An 8 pulse is defined as one cycle. This function allows high resolution and high speed PWM features under the conditions of a 10-bit quantization and a 28kHz frequency (f_{SC}= 3.579545MHz).

(8) VISS circuit

The VISS circuit discriminates the CTL signal duty between "1" and "0", and counts the cascading data of "1" to be stored into the latch circuit. The contents of the latch circuit may be transferred to the CPU through the data bus line, which allows a simple detection of the VISS signal.

The SM8202 outputs the discrimination results of the CTL signal duty from the P3$_6$/VISS pin.

(9) HAMP/CHRO circuit (only for the SM8203)

The head amp./chroma-rotation circuit (HAMP/CHRO) outputs the switching signal for a head amp. and chroma-rotation when a special playback.

Provided with these features, this microcomputer is applicable to 2-head and HiFi 4-head camcoders, a variety of servo controllers of NTSC, PAL and VHS-C.

Typical features are mentioned below:

- Applicable to a variety of video head system of a 2-head, and a-double azimuth 4 head, recording systems of an NTSC and a PAL, and high resolution system of VHS-C.
- A variety of trick motion with a software control.
- Built-in a tracking circuit and a REC CTL signal delay circuit.
- CTL signal duty discriminating circuit and VISS counter circuit.
- Variable CTL output signal (REC CTL) duty ratio when recording.
- High resolution and 4 high frequency RWM outputs.
- 1PG system HSP signal generator circuit, and external HSP input capability.
- False synchronous signal generator circuit.
- Automatic mode discrimination with a software control.

Instruction set

(1) Load instructions

Instruction	Operand	Function
CLR	dst	Clear
MOV	dst, src	Move
MOVM	dst, IM, src	Move Under Mask
MOVW	dst, src	Move Word
POP	dst	Pop
POPW	dst	Pop Word
PUSH	src	Push
PUSHW	src	Push Word

(2) Arithmetic instructions

Instruction	Operand	Function
ADC	dst, src	Add With Carry
ADCW	dst, src	Add Word With Carry
ADD	dst, src	Add
ADDW	dst, src	Add Word
CMP	dst, src	Compare
CMPW	dst, src	Compare Word
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
DIV	dst, src	Divide
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst, src	Multiply
NEG	dst	Negate
SBC	dst, src	Subtract With Carry
SBCW	dst, src	Subtract Word With Carry
SUB	dst, src	Subtract
SUBW	dst, src	Subtract Word

(3) Logic instructions

Instruction	Operand	Function
AND	dst, src	Logical And
COM	dst	Complement
OR	dst, src	Logical Or
XOR	dst, src	Logical Exclusive Or

(4) Program control instructions

Instruction	Operand	Function
BBC	src, dst	Branch on Bit Clear
BBS	src, dst	Branch on Bit Set
BR	cc, dst	Branch
CALL	dst	Call Subroutine
CALS	dst	Short Call Subroutine
DBNZ	r, dst	Decrement and Branch on Non-Zero
IRET		Interrupt Return
JMP	cc, dst	Jump
RET		Return

(5) Bit manipulation instructions

Instruction	Operand	Function
BAND	BF, src	Bit And
BCLR	dst	Bit Clear
BCMP	BF, src	Bit Compare
BMOV	src, dst	Bit Move
BOR	BF, src	Bit Or
BTST	dst, src	Bit Test
BSET	dst	Bit Set
BXOR	BF, src	Bit Exclusive Or

(6) Rotate and shift instructions

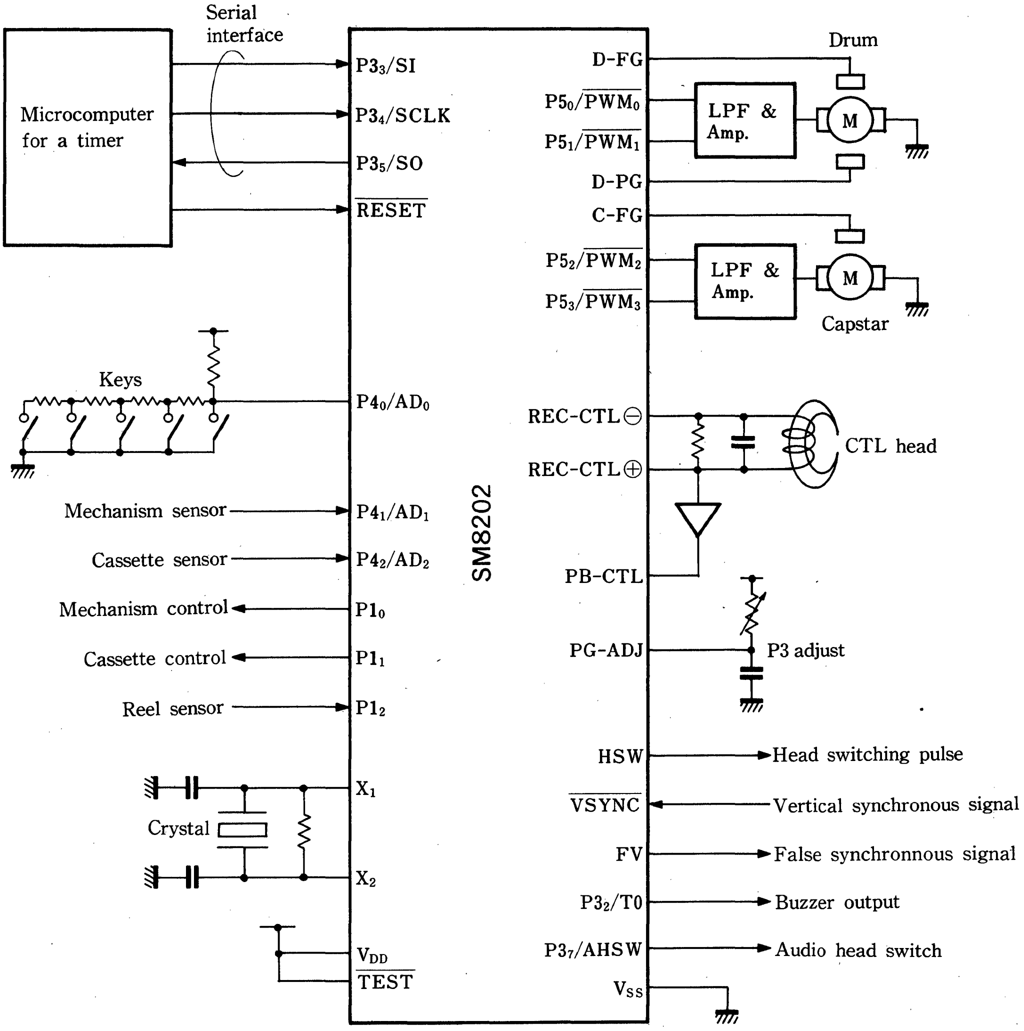
Instruction	Operand	Function
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SLL	dst	Shift Left Logical
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

(7) CPU control instructions

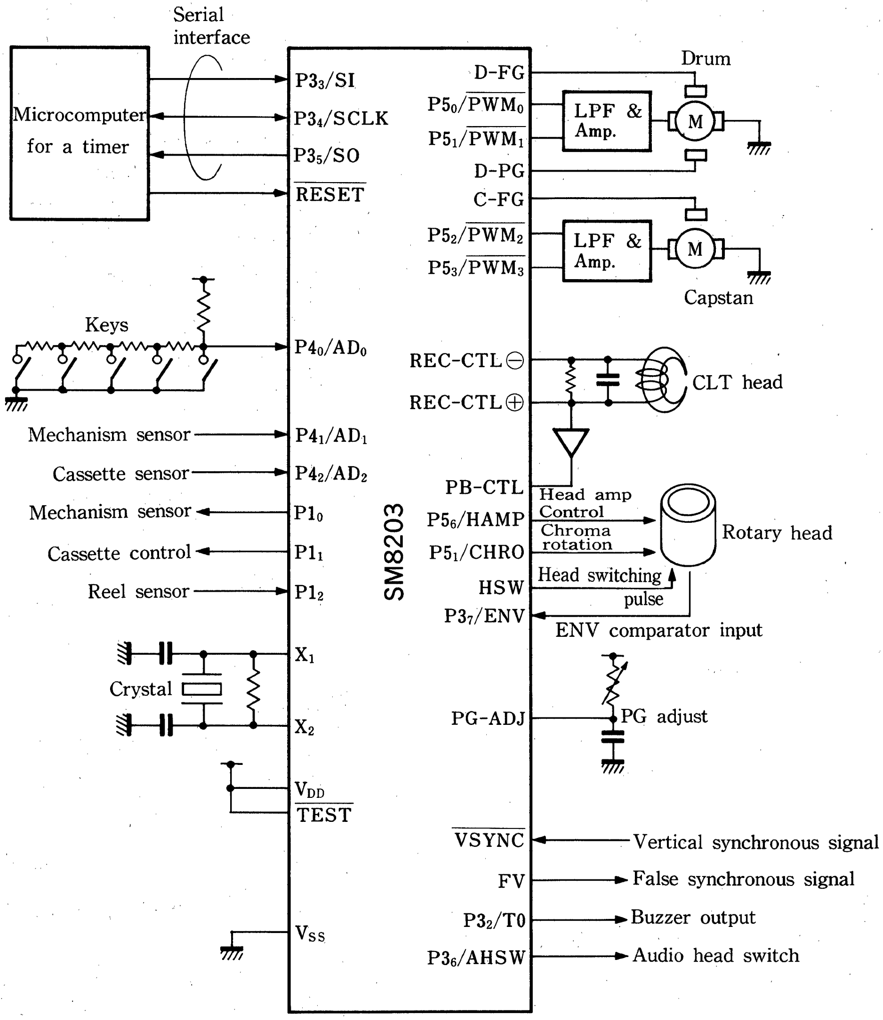
Instruction	Operand	Function
COMC		Complement Carry Flag
CLRC		Clear Carry Flag
DI		Disable Interrupt
DM	src	Data Memory Prefix
EI		Enable Interrupt
HALT		Halt CPU
NOP		No Operation
SETC		Set Carry Flag
STOP		Stop CPU

■ System Configuration Example (VCR)

(1) SM8202



(2) SM8203


Serial interface
Microcomputer for a timer
P3₃/SI
P3₄/SCLK
P3₅/SO
RESET
Keys
P4₀/AD₀
Mechanism sensor
P4₁/AD₁
Cassette sensor
P4₂/AD₂
Mechanism sensor
P1₀
Cassette control
P1₁
Reel sensor
P1₂
Crystal
X₁
X₂
V_DD
TEST
V_SS
SM8203
D-FG
P5₀/PWM₀
P5₁/PWM₁
D-PG
C-FG
P5₂/PWM₂
P5₃/PWM₃
LPF & Amp.
LPF & Amp.
M
M
Drum
Capstan
REC-CTL⊖
REC-CTL⊕
CLT head
PB-CTL
P5₆/HAMP
P5₁/CHRO
HSW
P3₇/ENV
Head amp Control
Chroma rotation
Head switching pulse
Rotary head
ENV comparator input
PG-ADJ
PG adjust
VSYNC
Vertical synchronous signal
FV
False synchronous signal
P3₂/T0
Buzzer output
P3₆/AHSW
Audio head switch

SM8320 8-Bit Microcomputer (Controller with An Inverter Drive Circuit)

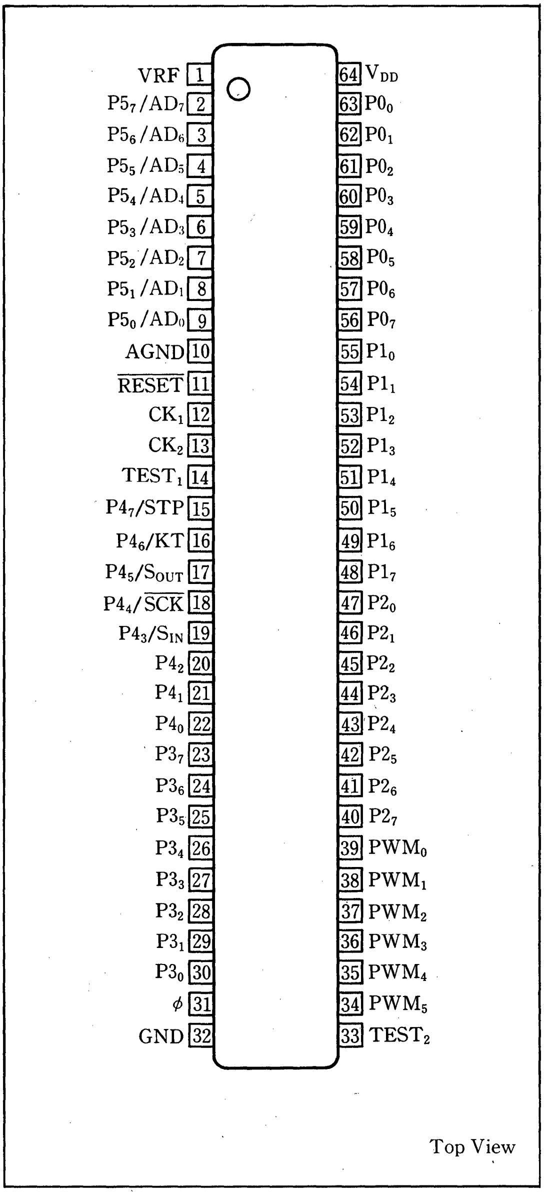
Description

The SM8320 is a CMOS 8-bit microcomputer which integrates an 8-bit core CPU, a ROM, a RAM, serial I/O, a timer/event counter, a watchdog timer, an A/D converter and a PWM waveform generator circuit. It is best suited to inverter air-conditioners required for an inverter drive capability of the 3-phase AC motor.

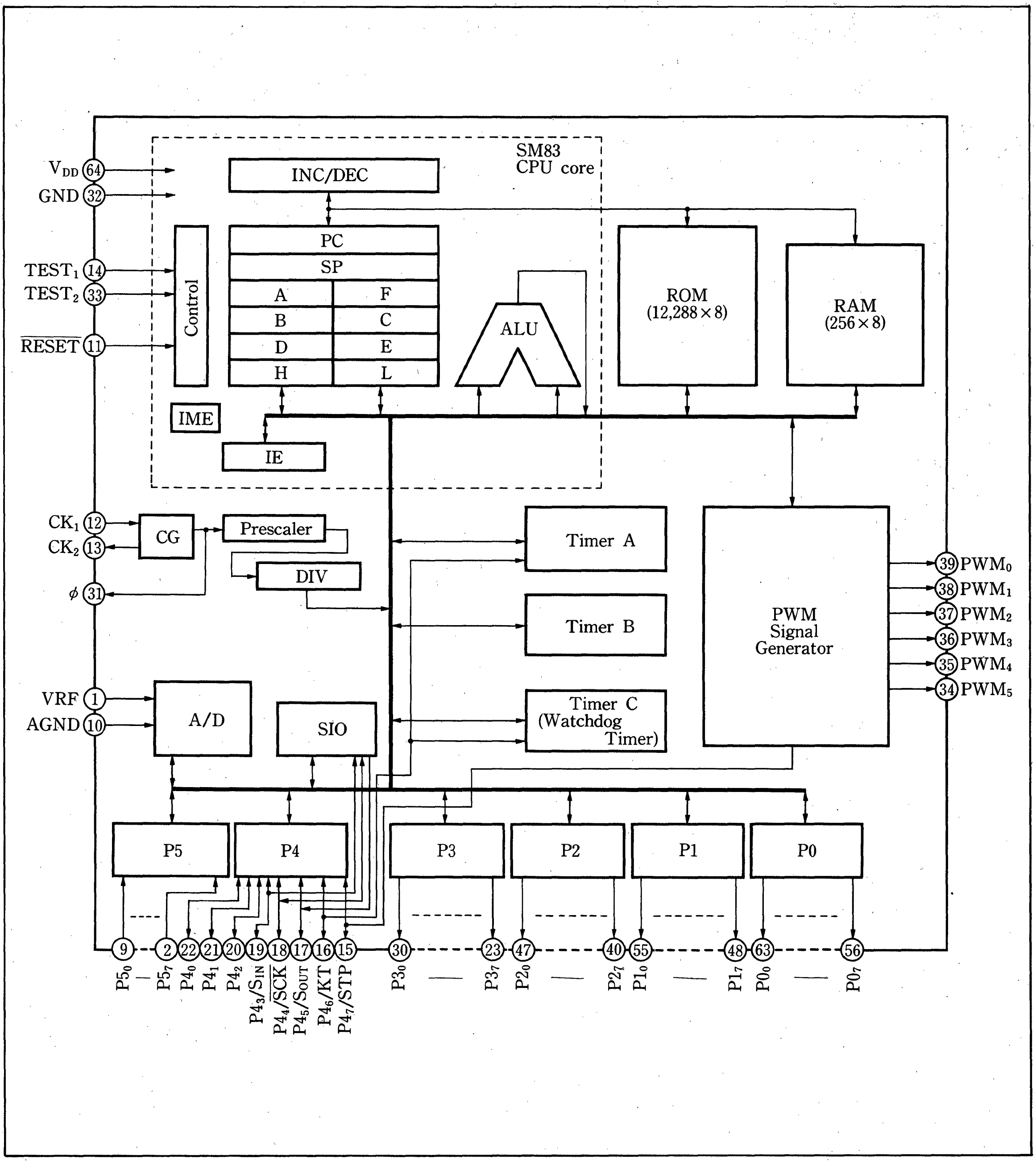
Features

1. CPU core: SM83
2. ROM capacity: 12,288 × 8 bits
3. RAM capacity: 256 × 8 bits
4. Instruction set: 81
5. Subroutine nesting: using RAM area
6. Instruction cycle time: 1 μs (MIN.)
7. Interrupts
   External interrupt: 1
   Internal interrupts: 7
8. Input/Output ports
   I/O ports: 40
   Input ports: 8 (for switching with A/D input)
   PWM output ports: 6
9. A/D converter: 8 bits (8 channels)
10. Counter/timer: 2 sets
11. PWM waveform generator circuit
12. Watchdog timer
13. Standby function: STOP/HALT mode
14. Crystal or ceramic oscillator circuit
15. Supply voltage: 5V±10%
16. Package
    64-pin SDIP (SDIP64-P-750)
    64-pin QFP (QFP64-P-1420)

Pin Conections



■ Block Diagram



Absolute Maximum Ratings

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		−0.3 to +7.0	V
Input voltage	V_I		−0.3 to V_{DD}+0.3	V
Output voltage	V_O		−0.3 to V_{DD}+0.3	V
Output HIGH current	I_{OH}	All output ports	−4	mA
Output LOW current	I_{OL1}	Output ports except for PWM_0-PWM_5, $P3_0$-$P3_3$	4	mA
	I_{OL2}	PWM_0-PWM_5	30	mA
	I_{OL3}	$P3_0$-$P3_3$	15	mA
Total output HIGH current	ΣI_{OH}	All output ports	−20	mA
Total output LOW current	ΣI_{OL1}	Output ports export for PWM_0-PWM_5, $P3_0$-$P3_3$	20	mA
	ΣI_{OL2}	PWM_0-PWM_5	80	mA
	ΣI_{OL3}	$P3_0$-$P3_3$	45	mA
Operating temperature	Topr		−20 to +70	℃
Storage temperature	Tstg		−55 to +150	℃

Recommended Operating Conditions

(Ta=−20 to +70℃)

Parameter	Symbol	Rating	Unit
Supply vortage	V_{DD}	+4.5 to +5.5	V
Instruction cycle	t_{SYS}	16.0 to 0.95	μs
Oscillating clock frequency	f	0.25 to 8.0	MHz

DC Characteristics

(V_{DD}=4.5 to 5.5V, Ta=−20 to +70℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	Note
Input voltage	V_{IH1}		0.7V_{DD}		V_{DD}	V	1
	V_{IL1}		0		0.3V_{DD}		
	V_{IH1}		V_{DD}−0.5		V_{DD}	V	2
	V_{IL1}		0		0.5		
Input current	I_{IH1}	V_{IN}=V_{DD}		50		μA	3
	I_{IL1}	V_{IN}=0V		−50		μA	4
	I_{IH2}	V_{IN}=V_{DD}			10	μA	5
	I_{IL2}	V_{IN}=0V			10		
Output voltage	V_{OH1}	I_{OH}=−1mA	V_{DD}−0.5			V	6
	V_{OL1}	I_{OL}=1mA			0.5		
	V_{OH2}	I_{OH}=−1mA	V_{DD}−0.5			V	7
	V_{OL2}	I_{OL}=10mA			2		
	V_{OH3}	I_{OH}=−400μA	V_{DD}−0.5			V	8
	V_{OL3}	I_{OL}=20mA			2		
Current consumption	I_{DD}	f_S=1MHz		6		mA	
	I_{DDH}	f_S=1MHz HALT		2		mA	
	I_{DDS}	Oscillation STOP		1	10	μA	

Note 1: Applied to pins $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, and $\overline{RESET}$.
Note 2: Applied to pins $TEST_1$, $TEST_2$, and CK_1.
Note 3: Applied to pins $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, and $P4_0$-$P4_7$ that have a pull-down resistor.
Note 4: Applied to pin SCK that has a pull up resistor.
Note 5: Applied to pins $P5_0$-$P5_7$, $\overline{RESET}$, $TEST_1$, $TEST_2$, and P0-P4 that have no pull down resistor (when in input mode).
Note 6: Applied to pins $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P4_0$-$P4_7$, and CK_2.
Note 7: Applied to pins $P3_0$-$P3_7$.
Note 8: Applied to pins PWM_0-PWM_5.

Pin Functions

(1) GND, V_{DD} (Power supply inputs)

The V_{DD} pin should be positive +5V (TYP.) with respect to GND.

(2) $TEST_1$, $TEST_2$ (Device test inputs)

The TEST pins must normally be connected to GND.

(3) RESET (System reset)

The RESET accepts an acvtive-Low system reset which initializes the internal logic of the device. It is internally connected to the positive supply V_{DD} with a pull-up resistor. Normally a capacitor is connected between this pin and GND to provide a power-on reset function.

(4) CK_1, CK_2 (System clock oscillator)

The CK_1 and CK_2 pins, in conjunction with an external ceramic or crystal oscillator, provide a system clock oscillator. An external clock must be input to the CK_1 pin.

(5) ϕ (Clock output)

ϕ, the system clock output pin, provides a clock frequency which is one eighth the master clock frequency (CK_1).

(6) VRF, AGND (Reference power for A/D converter)

The VRF and AGND pins are reference power supplies for A/D conversion. A High level of reference voltage (MAX. V_{DD}) should be input to the VRF with respect to AGND, and a minimum GND level of voltage should be input to the AGND.

(7) $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ (I/O ports)

Ports P0, P1, P2 and P3 may be independently set to Input or Output mode. These ports are all set to Input mode after reset. The P3 port can output a large drive current (sink current).

(8) $P4_0$-$P4_7$ (I/O ports)

Port P4 may be independently set to Input or Output mode. It serves for switching with a serial I/O, a timer, an input clock or a PWM output pause input.

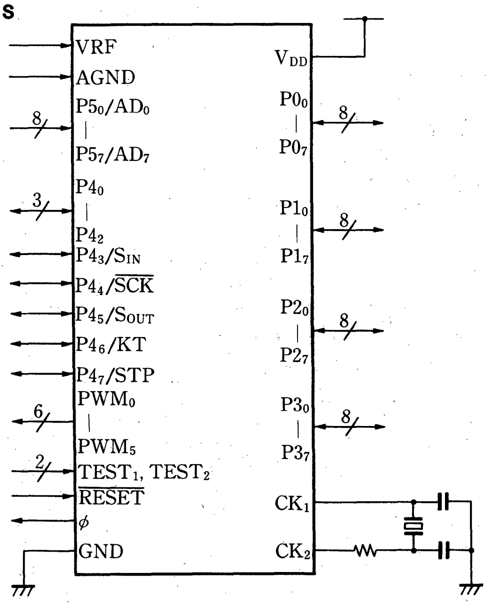
(9) $P5_0$-$P5_7$ (Input port)

The P5 is an input port, which can be used to input an analog data for A/D conversion.

(10) PWM_0-PWM_5 (PWM output ports)

The PWM output port is used to output the inverter drive signals for a 3-phase AC motor converted from the internal ROM data through an internal PWM generator circuit.

Functiond Connections



Hardware Configuration

(1) Address architecture

The on-chip ROM is allocated in the address at 0000-2FFF (12K bytes), a RAM at FE80-FF7E (256 bytes) and port register at FFD0-FFFF.

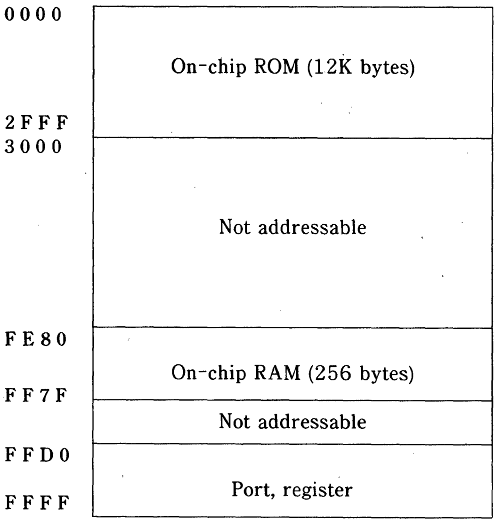


Fig. 1 Address architecture

(2) Program memory (ROM)

The program memory has 12K bytes and is allocated in the address at 0000-2FFF. 9 interrupt vectors and a jump destination of an RST instruction are allocated in the address shown in Fig. 2. Applying a Low level signal to the RESET pin starts excution of the user's program at address 0000.

(3) Data memory (RAM)

The data memory has 256 bytes and is allocated in the address at FE80-FF7F.

(4) Control register

The control registers including an I/O register and a mode register are allocated in the address at FFD0-FFFF.

(5) CPU core structure

The internal CPU core consists of an accumulator, a general-purpose register, a program counter, a stack pointer, an interrupt mask register, an interrupt master enable flag and an arithmetic logic unit (ALU).

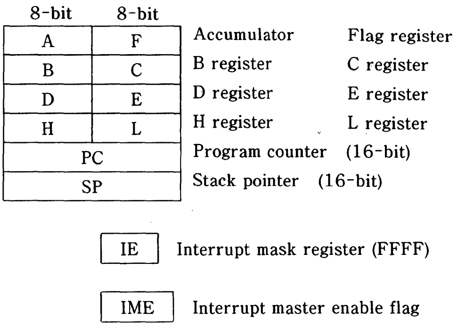


Fig. 3 SM83 CPU core internal register structure

Address	Vector
0000	Start address/RST0
0008	RST1
0010	RST2
0018	RST3
0020	RST4
0028	RST5
0030	RST6
0038	RST7
0040	INT0: External STP interrupt
0048	INT1: Start/End bit interrupt from a PWM generator circuit
0050	INT2: CNT12/CNTGG counter interrupt from a PWM generator circuit
0058	INT3: TIMA overflow interrupt
0060	INT4: Interrupts for the end of A/D conversion, comparison and SIO transfer
0068	INT5: TIMB overflow interrupt
0070	INT6: TIMC overflow interrupt
0078	INT7: DIV overflow interrupt
0080	NMI: Mask disable interrupt (watchdog timer overflow interrupt)

Fig. 2 Vector addresses

(6) P0, P1, P2 and P3 (I/O ports)

The P0, P1, P2 and P3, 8-bit I/O ports may be switched between Input (0) and Output (1) modes with a directional register. The contents of the output data register should also be transferred to the accumulator A_{CC}.

(7) P4 (I/O port)

The P4 is an 8-bit I/O port, and with a program, pins $P4_3$-$P4_5$ serve as serial I/O (S_{IN}, SCK, S_{OUT}), pin $P4_6$ serves as a counter/timer input (KT) and pin $P4_7$ as a stop signal input of a PWM output and an external interrupt input (STP).

(8) S_{IN}, SCK, S_{OUT} (Serial I/O)

The serial I/O ports consist of a couple of register, 8/4/2/1 counter and controllers, which are used to transmit and receive 8-bit data synchronized with the shift clock.

(9) P5 (Input port)

The P5, input port, can be set to the analog data input for an A/D converter with a program.

(10) AD_0-AD_7 (A/D converter)

The device contains an 8-bit A/D converter with 8-channel multiplexer analog inputs. The A/D converter inputs can be set to 3 modes including an automatic A/D conversion mode, a comparator mode between analog input value and internal register, and an input mode. The mode is normally set to the input mode.

- Input mode When in input mode, there is no data transfer from accumulator to port P5, and the current status (digital value) of the port should be loaded into the accumulator.
- A/D conversion mode In the A/D conversion mode, an analog data of selected channel (one of ports $P5_0$-$P5_7$) is converted into digital data which will be loaded into the accumulator. Then, if an interrupt is enabled, the CPU acknowledges the interrupt. The A/D conversion will be performed by comparing voltages determined by a ladder resistor placed between VRF and AGND with analog inputs. The A/D conversion cycle should be 68 μs at 8MHz of oscillation frequency (1 μs of system clock).
- Comparator mode In the comparator mode, one bit location of control register is determined according to large or small data obtained by the comparison between data registers and analog inputs. Upon completion of comparator operation, the port is used for switching with general-purpose input.

(11) Timer

The timer circuit consists of a 6-stage prescaler, an 8-bit divider (DIV), an 8-bit timer A, B, C, and a timer control register (TMODE). The timer C can be used as a watchdog timer. The prescaler input clock is used for a system clock ϕ (reference clock f/8).

(12) Interrupts

The interrupt functions include 8 kinds of maskable interrupts (internal: 7, external: 1) and a non-maskable interrupt (see table 1).

The interrupt request flag (IF_0-IF_7) is set if a maskable interrupt (IRQ_0-IRQ_7) occurs. Then, presetting the corresponding bit of interrupt masterenable flag (IME) and interrupt mask register issues the interrupt.

If more than one interrupt occurs simultaneously, all of the corresponding interrupt request flags will be set, but the CPU will only acknowledge that interrupt with the highest priority and other interrupts will be queued.

Table 1 Interrupt reguest

Priority	Address	Interrupt	Mask
1	0 0 8 0	NMI: Mask disable interrupt (Overflow interrupt of watchdog timer)	Disable
2	0 0 4 0	IRQ 0: External STP interrupt	Enable
3	0 0 4 8	IRQ 1: Start/End bit interrupt from a PWM generator circuit	Enable
4	0 0 5 0	IRQ 2: CNT12/CNTGG counter interrupt from a PWM generator circuit	Enable
5	0 0 5 8	IRQ 3: TIMA overflow interrupt	Enable
6	0 0 6 0	IRQ 4: Interrupts for the end of A/D conversion, comparison and SIO transfer	Enable
7	0 0 6 8	IRQ 5: TIMB overflow interrupt	Enable
8	0 0 7 0	IRQ 6: TIMC overflow interrupt	Enable
9	0 0 7 8	IRQ 7: DIV overflow interrupt	Enable

(13) Reset function

Applying a Low level signal (master clock period × 2) to the RESET pin resets the internal logic of the device and starts execution of the program at address 0000.

The following two reset functions are also available:

1) The interrupt master enable flag IME, interrupt mask register IE and interrupt request flag IF are reset to disable all maskable interrupts.

2) The port and mode registers are initialized. The other registers are indefinite. Applying a High level signal to the RESET pin with a double master clock frequency (32.768kHz at 8MHz master clock) starts execution of the program.

(14) Standby mode

The standby mode includes HALT mode and STOP mode. In each standby modes, output ports, internal registers and internal RAM remain operative.

- HALT mode Executing the HALT instruction places the device in HALT standby mode. To reduce power consumption, the system clock is inactivated. However, the oscillator circuit between CK_1 and CK_2 is operating, and the DIV, SIO and timer remain operative, provided their operations do not depend on the system clock. The PWM circuit is inactivated. While in HALT mode, if a Low level signal is applied to the RESET pin, or an interrupt enable flag is set, the device exists HALT mode.
- STOP mode Executing the STOP instruction places the device in STOP standby mode, and the entire system clock except for external clock of SIO and timers is inactivated. While in STOP mode, if a Low level signal is applied to the RESET pin, or an interrupt occurs such as a High level signal is applied to the SIO, timer A or STP pin, the device exists STOP mode.

■ PWM Waveform Generator Circuit

The PWM waveform generator circuit automatically outputs inverter drive PWM waveforms which are 3-phase AC waveforms required for driving compressors such as inverter air conditioners, from the PWM_3-PWM_5 pins. This circuit greatly reduces loads on software.

In addition, the lower 6 bits of the internal ROM data can be automatically output as they are from the PWM_3-PWM_5 pins. They can be used as drive waveform output for stepping motors.

(1) Basic operation of PWM waveform generator circuit

The PWM waveform data is written to the program area of ROM, and the start address and the waveform data output interval (sampling time) are specified on the register. Thus, every time waveform data is read from ROM, the waveform ROM address counter is automatically incremented or decremented, and the waveform data at the addressed area is directly transferred to the waveform data buffer.

Then, PWM waveforms are automatically generated via the data conversion circuit and the rise delay circuit, and output from the PWM_0-PWM_5 pins.

(2) PWM waveform data preparation

The waveform data provided in any way are generally obtained by comparing 3-phase AC waveforms (3 different sine curves having a phase difference of 120 degrees) with delta waveforms and converted into 3-bit data and then written to ROM.

Fig. 4 shows an example for PWM waveform data preparation.

It is not necessary to have data for one cycle (360 degrees). Only having data for 30, 60, 120 or 180 degrees allows waveforms of 360 degrees.

(3) Sampling time

The sampling time can be set in steps of 0.25 μs. in the range from 10 μs to 256 μs.

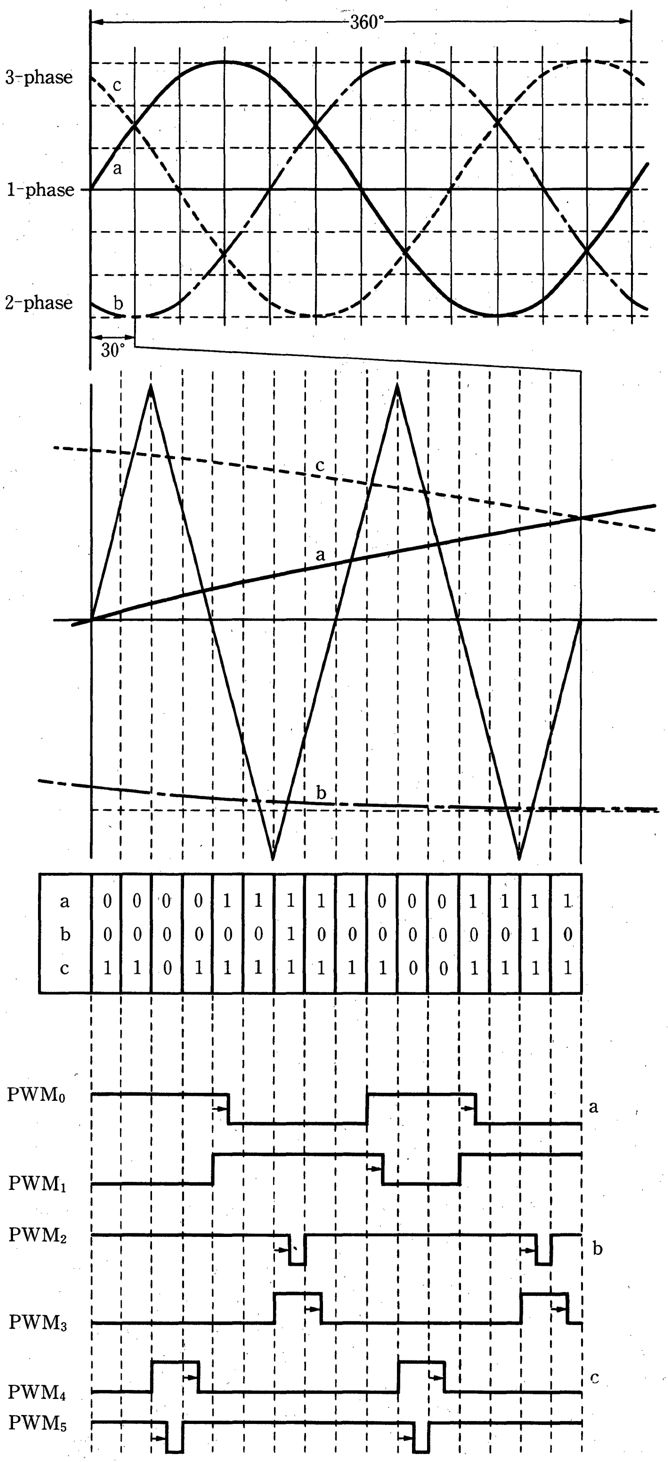
The frequency of the PWM waveform (sine curve) can be varied based on the waveform data written to ROM by changing the sampling time.

(4) Delay time setting

When external transistors are used as the drive elements for the inverter and a pair of transistors which are vertically connected in push-pull configuration are alternately switched, one transistor must be turned ON considering the delay time at the time when the other transistor is turned OFF in order to prevent short circuit in the vertical direction.

The delay time varies depending on tarnsistor types and must be changed in accordance with them.

This microcomputer allows this delay time to be freely set in steps of 2 μs in the range from 0 to 30 μs or in steps of 0.25 μs in the range from 0 to 3.85 μs.



a	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1
b	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
c	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1

(a) 3-phase AC waveform

a, b and c indicate the 30 degree sample waveform of one cycle. 360 degree waveform can be obtained by combining inverted waveforms of a, b and c with inverted-timeframe waveforms.

(b) Comparison with deltas

In comparing AC wave with daltas at certain interval (sampling time), if delta wave is positioned in upper part of AC wave, the binary code should be 1, and in lower part of AC wave, it should be 0.

(c) 30 degree PWM waveform data

One cycle of PWM waveform is automatically formed by an internal conversion circuit based upon 30 degree PWM waveform data.

(d) Output data from PWM

The ON signal is output with delay time not to simultaneously turn on the external power transistor composed of external push-pull type (see System Configuration Example).

→ : ON signal delay time

Fig. 4 Example of PWM wavefrom data

Instruction Set

The SM8320 has 77 instruction set.

(1) 8-bit transfer instructions, I/O instructions
(2 kinds, 19 instructions)
LD, LDX

(2) 16-bit transfer instructions
(4 kinds, 6 instructions)
LD, LDHL, POP, PUSH

(3) 8-bit arithmetic instructions, Logic instructions
(12 kinds, 12 instructions)
ADC, ADD, AND, CP, CPL, DAA, DEC, INC, OR, SBC, SUB, XOR

(4) 16-bit arithmetic instructions
(3 kinds, 4 instructions)
ADD, DEC, INC

(5) Rotate, shift instructions
(12 kinds, 12 instructions)
RLA, RLC, RLCA, RL, RRA, RRC, RRCA, RR, SLA, SRA, SRL, SWAP

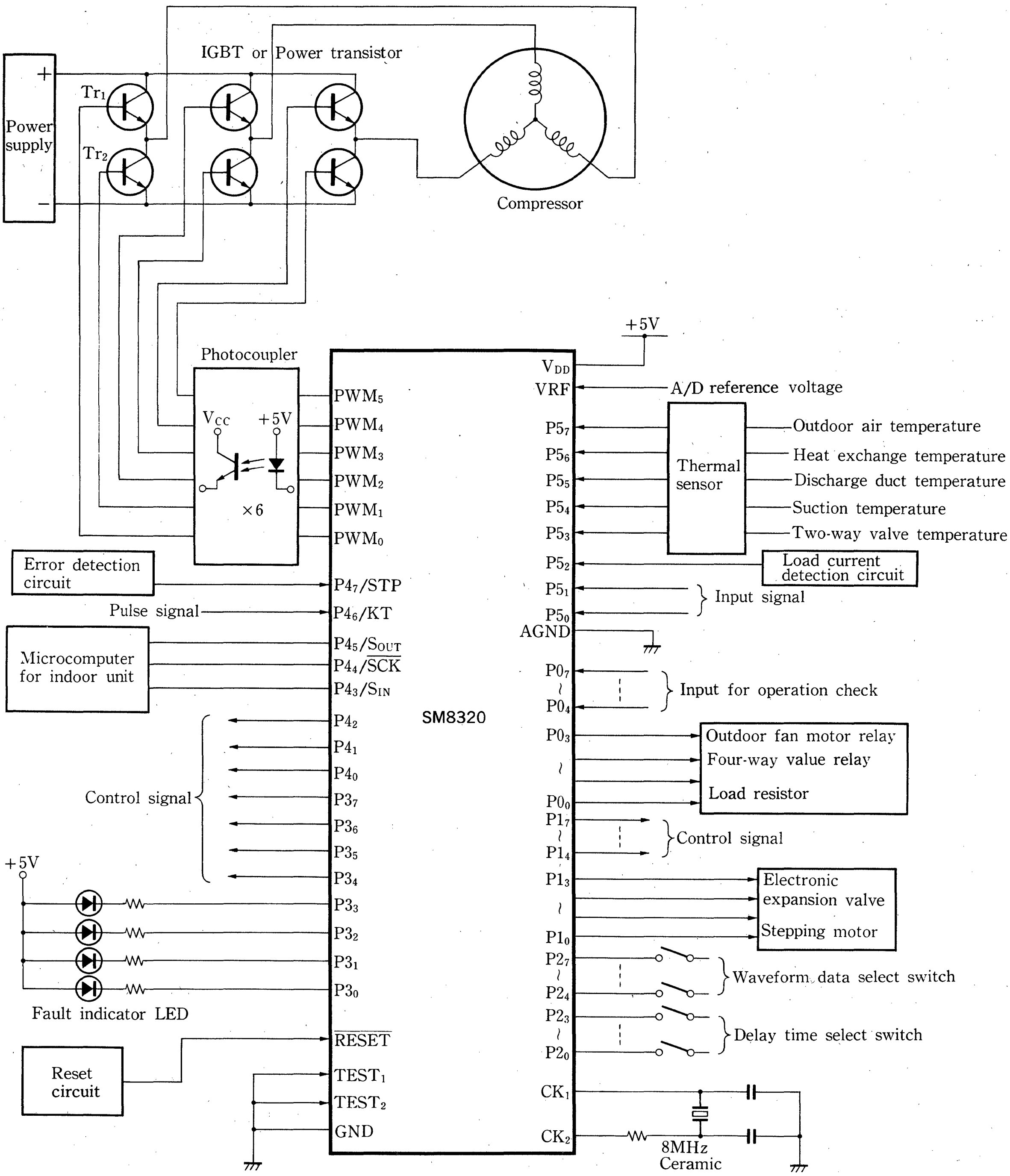
(6) Bit manipulation instructions
(3 kinds, 6 instructions)
BIT, RES, SET

(7) Jump instructions
(2 kinds, 5 instructions)
JP, JR

(8) Call, return instructions
(4 kinds, 6 instructions)
CALL, RET, RETI, RST

(9) CPU control instructions
(7 kinds, 7 instructions)
CCF, DE, EI, HALT, NOP, SCF, STOP

System Configuration Example (Outdoor unit of inverter air-conditioner)



Development Support Tools

LUXXXH2 Evaluation Board for SM Series

Description

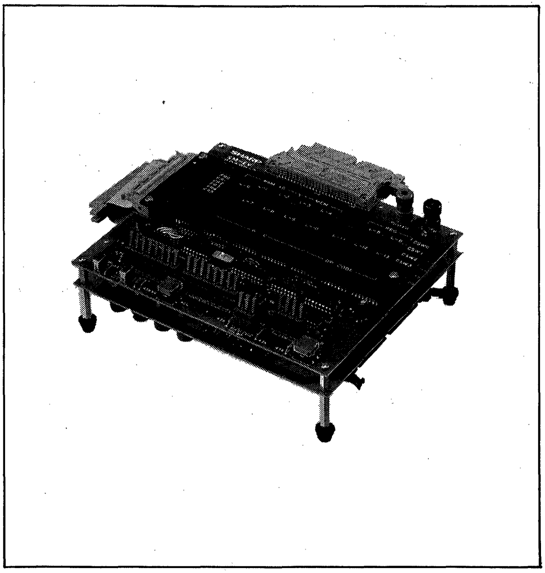
The LUXXXH2 is an evaluation board for use in developing programs of 4-bit single chip microcomputer SM series. It is available for any types of evaluation board applicable to each SM series.

The evaluation board is equivalent to the SM series with ROMless in functions and electrical characteristics. It is designed to develop programs together with an EPROM or the development support tool SME-30 emulator in RAM basis.

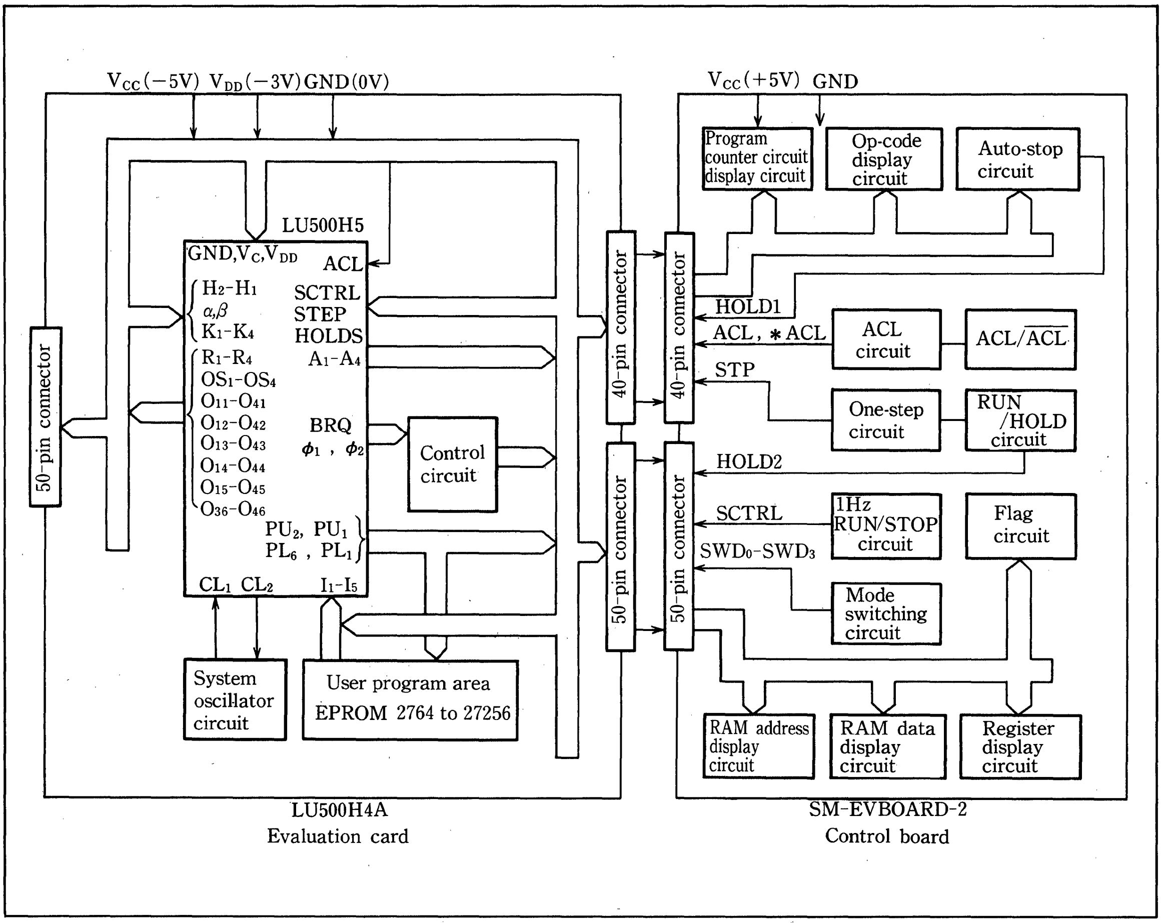
Features

1. System debug with EPROM
2. Debug in RAM basis in conjunction with an emulator SME-30
3. Typical functions of the evaluation board
   - Hold function
   - One-step function
   - Auto-stop function
   - Program counter indicator
   - Accumulator and carry F/F indicator
   - RAM address register and memory indicator
   - Instruction code indicator
   - PLA specification function

Outline



Block Diagram (LU500H2A)



Evaluation Board of SM Series

The evaluation board consists of a couple of printed circuit boards including a control board and an evaluation card integrating with an evaluation chip and an EPROM socket (see Fig. 1).

For example, the evaluation board of single chip microcomputer SM500 consists of an evaluation card (LU500H4A) and a control board (SM-EVBOARD-2).

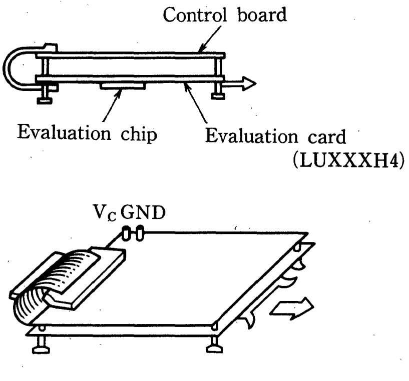


Fig. 1 Evaluation board structure

■ Components Layout on The Board

In explaining the LU500H2A, see Fig. 2 for the evaluation card (LU500H4A) and Fig. 3 for the control board (SM-EVBOARD).

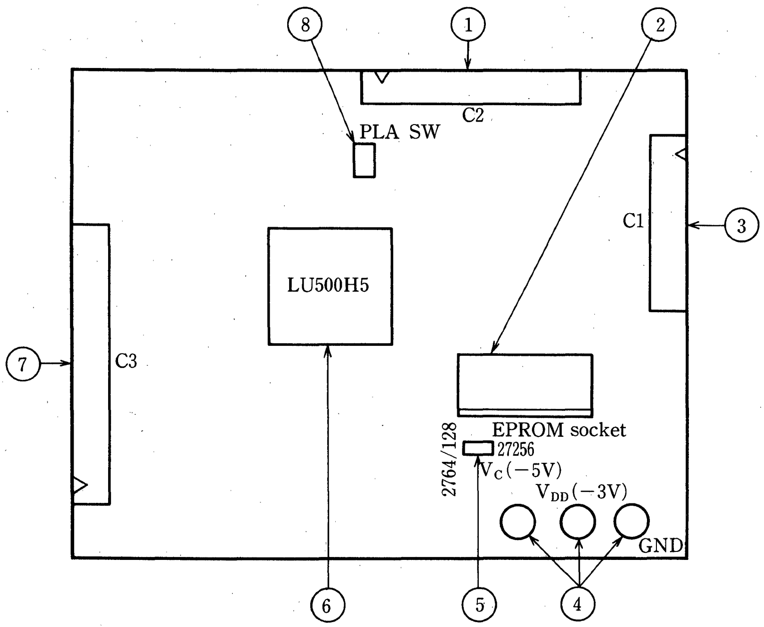


Fig. 2 LU500H4A component layout

Table 1 LU500H4A Component Description

No.	Components	Description
1	C2 connector (50-pin connector)	Inputs and outputs data.
2	IC socket for programmed EPROM	Adapts an EPROM written with user's program.
3	C1 connector (40-pin connector)	Inputs and outputs data.
4	Power supply terminal	Applies DC voltage.
5	EPROM select switch	Selects between 2764 and 27256 for EPROMs.
6	Evaluation chip	LU500H5, provides the logic function of SM500.
7	C3 connector (50-pin connector)	User's connector Connects with the cable from the user's system.
8	PLA SW	Specifies the PLA.

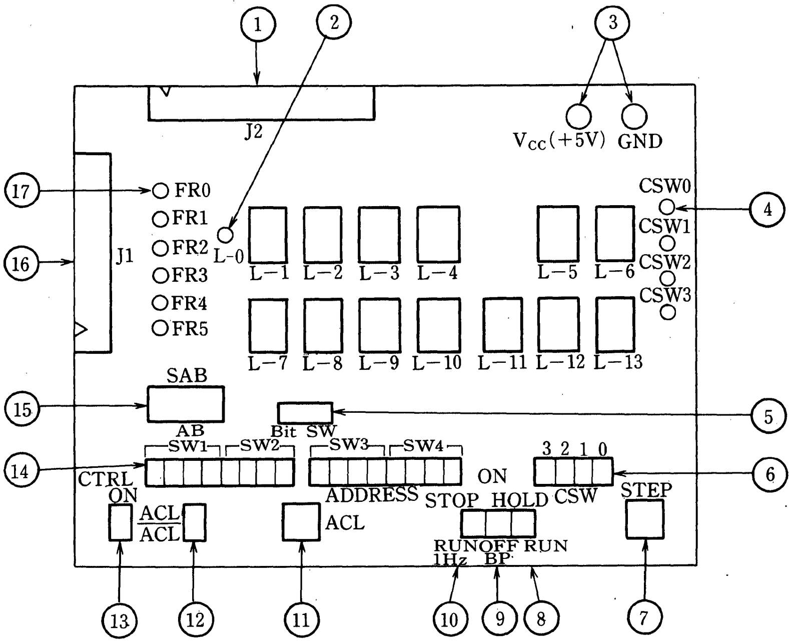


Fig. 3 SM-EVBOARD-2 component layout

Tble 2 SM-EVBOARD-2 Component Description

No.	Components	Description
1	50-pin connector	Inputs and outputs data
2	Data indicator LED	L-0 to L-13 indicate registers and operation codes. · L-1 and L-2 indicate the RAM address counters BM and BL. · L-4 indicates the RAM contents specified by BM and BL. · L-6 indicates the contents of the accumulator. · L-7 to L-9 indicate the contents of the program counter. · L-11 to L-13 indicate the operation codes.
3	Power supply terminal	Applies DC voltage (Applies +5V to the V_{CC} with respect to GND.)
4	CSW indicator LED	Turns on with the CSW switch ON, and turns off with the CSW switch OFF.
5	Bit SW	Bit numbers of the step PL for the program counter. (set to 6 in the LU500H2A)
6	CSW switch	Selects data output from the data pins D_0-D_7. (not used in the LU500H2A)
7	STEP switch	Turning on this switch checks the program at every one step.
8	RUN/HOLD switch	Executes program at the RUN side and stops at the HOLD side.
9	BP switch	Turning on this switch activates an auto-stop function.
10	1Hz switch	Controls the clock start with 1 sec. signal.
11	ACL switch	Turning on this switch enables the evaluation chip to activate the ACL operation.
12	ACL polarity select SW	Selects the ACL polarity. Set to the ACL in the LU500H2A.
13	CTRL switch	When developed using the LU500H2A only, turn this switch on.
14	ADDRESS switch (address specification)	When used with an auto-stop function, specify the stop address with this switch.
15	SAB AB connector	Sets the connector according to the field and page of a program counter.
16	40-pin connector	Inputs and outputs data
17	Flag indicator LED	The FR0 lights up with the C F/F to be set, and turns off with the C F/F to be reset.

System Configuration Example

Connection between the emulator SME-30 and the evaluation board LU500H2A.

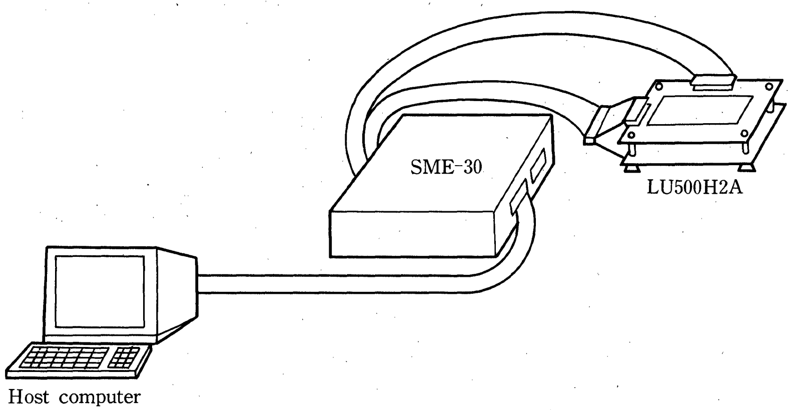


Fig. 4 System configuration

SME-30 (LU4DH300) Emulator for 4-Bit Microcomputers

Description

The SME-30 (LU4DH300) is an emulation system for programming a 4-bit single chip microcomputer SM series. This system is used to develop programs in combination with a PROM writer as well as each type of evaluation board available for any of the 4-bit microcomputer SM series.

Provided with the serial interface (RS-232C), this system, connecting to the host system applicable to the software such as cross-assembler, debugs the program.

The host system is a personal computer which can be driven under control of the MS-DOS^{TM} operation system.

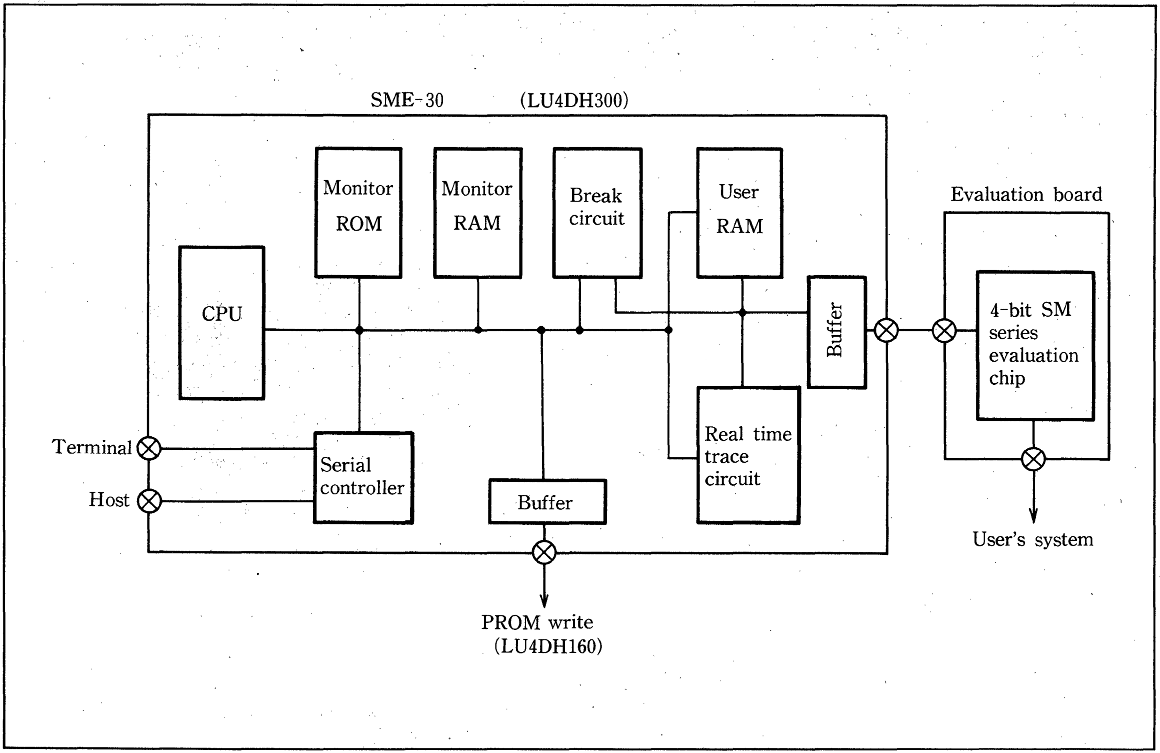
* MS-DOS^{TM} is a trademark of Microsoft Corporation.

Outline

Features

1. Exchanging the evaluation board and control software applies to any type of 4-bit microcomputer SM series.
2. User's program area up to 16K words (16 bits/word) for all of SM series can be supported.
   The user's program area is expandable up to 64K words at a 16K word unit.
3. A 16-bit step counter measures the program execution time.
4. Historical memory for tracing the program run.
5. Symbols defined by a cross-assembler for the operand.
6. Data rewrite with assembly languages
7. Symbolic reverse assembling
8. Allows transition of the register and flag contents.
9. Program execution from any addresses
10. Break point set conditions
    - Program counter: 2
    - RAM address: 1
    - Logical product of RAM address and RAM data: 1
    - External signal: 1
11. Operation mode
    - Real-time execution
    - Single-step execution
    - Trace execution
    - Dummy execution
    - Snap shot execution

Block Diagram



Specifications

Parameter	Specification	Remarks
Clock	Switchable between internal and user clocks	
Emulation CPU	Evaluation chip	4-bit single-chip microcomputer SM series
User's RAM	16K words (16K×16 words)	Expandable up to 64K words (MAX.)
History RAM	2K steps	
Dummy RAM	256 words	
Break point	5 circuits (hardware)	
Serial port	RS-232C×2	(110 to 9600BPS) at 8 levels
External current capacitance	1A (MAX.)	Current te be output
Power supply	AC100V±10%, 50/60Hz	
Operating temperature	0 to 40℃	
Outer dimensions	310×270×98	W×L×H (Unit: mm) Not including projections

Connection Method

Two connection methods between the SME-30 emulator and the host system are available depending upon the types of the host system.

(1) Local mode

The local mode is available for the CRT separation type of host system. In this mode, connection is possible only with the RS-232C interface between the emulator and the host system. Fig. 1 shows the connection diagram of the system in this mode.

(2) Remote mode

The remote mode is available for the host system embedded CRT. In this mode, the RS-232C interface and the software interface connect the emulator with the host system. Fig. 2 shows the connection diagram of the system in this mode.

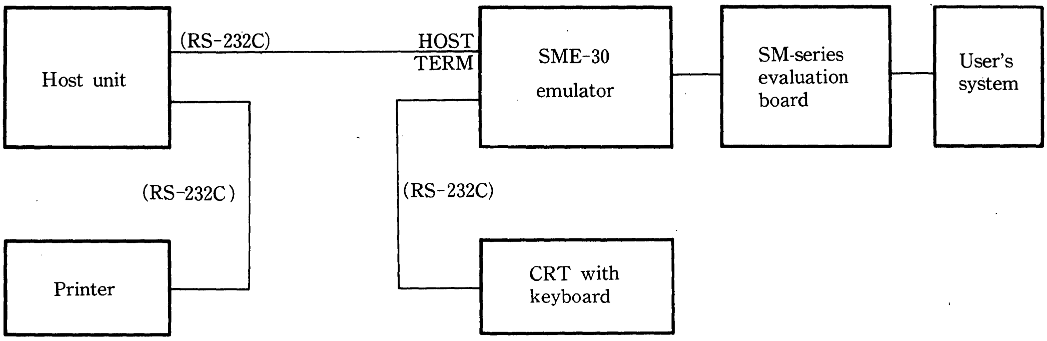
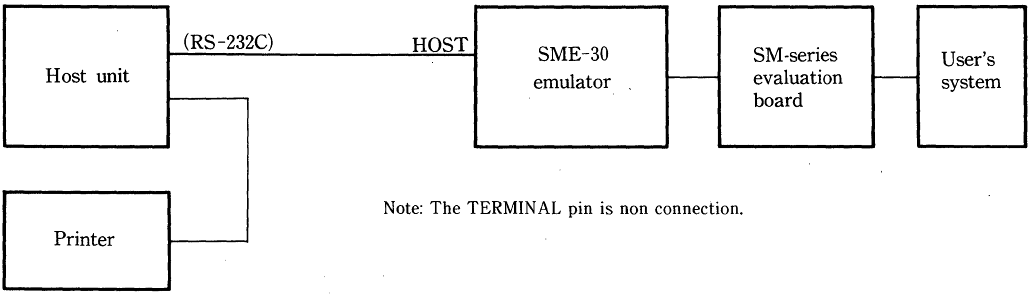


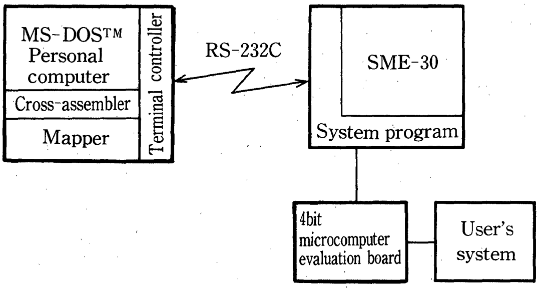
Fig. 1 Connection at the local mode



Note: The TERMINAL pin is non connection.

Fig. 2 Connection at the remote mode

Development Circumstances of MSDOS™ personal computer and SME-30



(1) System Program

The system program is given by the each MS-DOS™ disk which depends on models.

(2) Cross-assembler

The cross-assembler is used to assemble the source program provided by the editor, and make a list file and an objection file on request.

(3) Mapper

The objection mapper program is used to replace the objection file on the diskette assembled by a cross-assembler with the steps and pages for easy-to-read report.

(4) Terminal controller

The terminal controller is a program for the connection between the SME-30 emulator and the personal computer. The program depends on the personal computers.

LU8200H7/LU820XH4

SM82 Series In-circuit Emulator/Emulation Pod

■ Description

The SM82 in-circuit emulator is designed to effectively program an 8-bit single chip micro-computer SM82 series.

This system consists of an in-circuit emulator (LU8200H7) and a replaceable emulation pod (LU820XH4), which is available for any type of the SM82 series.

The host system must be controlled under the MS-DOS™ operation system as well as provided with the RS-232C interface unit or the Centronics interface unit which allows a high speed transfer of the program to the emulator unit.

* MS-DOS™ is a trademark of Microsoft Corporation.

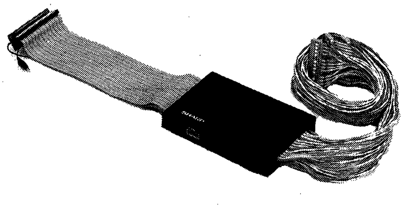
■ Features

1. 64K bytes of emulation memory
2. A variety of brake and trace functions
3. Real-time operation
4. Execution time scale
5. Reverse assembler and line assembler
6. Symbolic debugger
7. The coverage function checks nonaccessed area under programming
8. Centronics interface unit allows a high speed down load and a connection to the printer
9. Exchangeable emulation pod applicable to any type of the SM82 series

■ Outline

Emulator unit (LU8200H7)

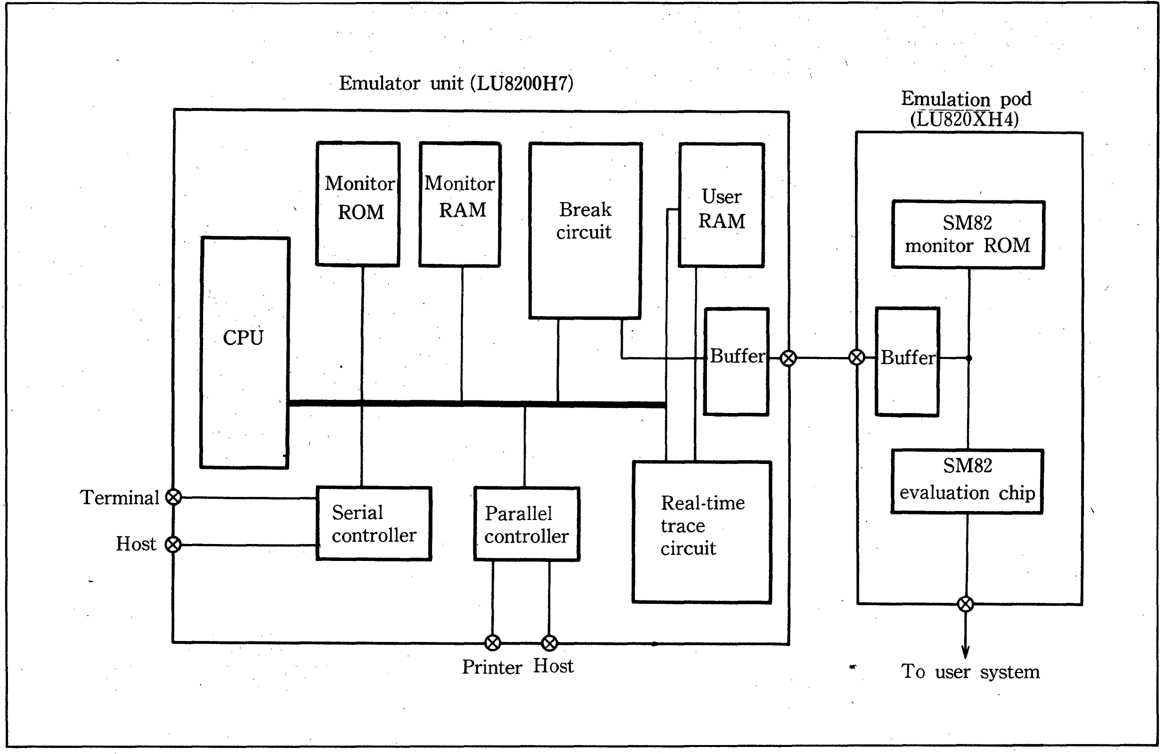
Emulation pod (LU8202H4)



■ Specifications

Parameter	Specification
Clock	On-chip/user clock switching
Emulation	SM82 evaluation-chip
Emulation memory	64K byte
Break point	2 (hardware)
Break counter	16-bit×2
Serial interface	RS-232C×2
Parallel interface	Centronics input×1 output×1
Power supply	AC100V±10% 50/60Hz
Operating temperature	0 to 40℃
Unit	330 (W)×315 (D)×150 (H)

Block Diagram



System Configuration

Below shows the development support system of an 8-bit single chip microcomputer SM82 series.

(1) SM82 in-circuit emulator

- Emulator unit
- SM82 emulation pod

(2) Host system

The host computer system operates under control of the MS-DOS™ operation system, which is designed to make a program of the SM82 with the cross-software mentioned below. The program should be down-loaded into the SM82 in-circuit emulator, and debugged with the emulator. The host system must be provided with the RS-232C interface unit.

(3) SM82 cross-software

The cross-software consists of an assembler, a linker, HEX dumper and an emulator control software, which is offered by SHARP.

Note: The emulator may have some different function from an actual device. The operation must finally be checked with a piggy-back device.

■ Connection Method

(1) Connection between the emulator unit and the emulation pod

Fig. 1 shows the emulator unit from a front view, and Fig. 2 shows from a back view. Fig. 3 shows the emulation pod. The connector of the emulation pod must be connected to the connector located in the front of the emulator. The side of the emulation pod is provided with a connector for eight external ternal probe cables which accept the maximum of 0 to 5.5V of input voltage.

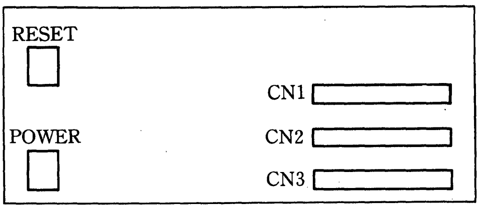


Fig. 1 Front side of emulator

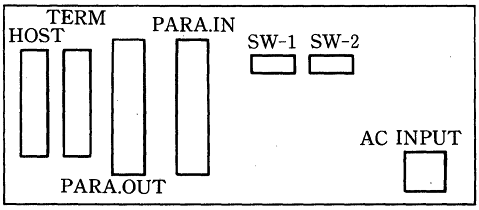


Fig. 2 Rear end of emulator

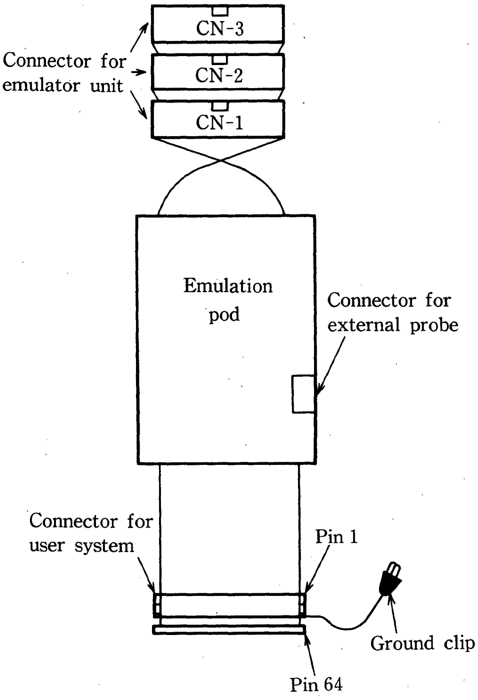

CN-3
Connector for emulator unit
CN-2
CN-1
Emulation pod
Connector for external probe
Connector for user system
Pin 1
Ground clip
Pin 64

Fig. 3 Emulation pod

probe cables which accept the maximum of 0 to 5.5V of input voltage.

(2) Connection between the emulation pod and the user's system

The connector of user's system must be inserted into the socket, adjusting the pin No. 1 (see Fig. 3). And at the same time, connect a ground clip to the ground of user's system. The external probes receive signals from user's system to store the signal levels into the trace memory and to be broken.

(3) Connection to the peripheral equipment

The connectors equipped with the back of the emulator unit may be used to connect with peripheral equipment (see Fig. 2). It is designed to connect to the host system in a serial interface cable (RS-232C) or in a parallel interface cable (Centronics).

The emulator makes a command control in a serial communication with the RS-232C interface. Be sure to connect the serial connector of a host system with the TERM connector of the emulator. A down-load interface type of user's program develped with a host system can be specified with a command control between the RS-232C serial interface and the Centronics parallel interface. When a high speed parallel down-load interface is selected, connect the printer connector of a host system with the PARA. IN connector of an emulator. When the display contents are printed using a printer, connect the Centronics interface connector of a printer with the PRINTER connector of an emulator. Fig. 4 shows the connection diagram between each equipment.

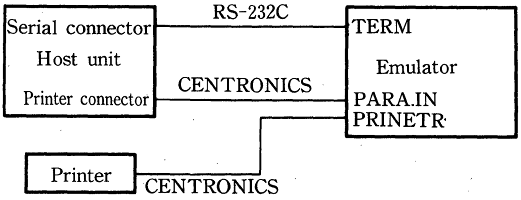


Fig. 4 Connection diagram between systems

SHARP offers the emulator control program together with the cross-assembler which enable the emulator command/data to be controlled by a keyboard/CRT of a host system.

■ Trigger Function of SM82 in-circuit emulator

(1) Trigger conditions

- Memory address
- Memory data
- Register address
- Register data
- Bus control signals (command fetch, memory read and write, register read and write)
- External probe signal
- Passing-through count

The above conditions can be set with the specifications for data area and "Don't care data" with a bit unit.

(2) Trigger mode

- 2-level of a sequencial trigger

  When a multiple nesting program is triggered, a passing point must be specified by a sequencial trigger. This function becomes effective only when it passed through two points in order.
- Pre-trigger, center trigger and post trigger functions
- Acquisition trigger

  The acquisition trigger function traces a necessary condition, and specifies the trigger conditions as well as the machine cycle count to be traced.
- Time count

  The execution time from the start condition to the stop condition is measured.

(3) Real-time trace

The trace capacity has a configuration of 64 bit words × 8,192 steps which stores:

- Memory address
- Memory data
- Register address
- Register data
- Bus control signals
- External probe signal

The pre-trigger and post-trigger conditions can also be checked.

8-bit Microprocessors/Peripherals

LH5080 Z80 CMOS CPU Central Processing Unit

Description

The LH5080 is a Z80 CPU fabricated with CMOS silicon-gate process technology and is compatible with the conventional Z80 NMOS CPU (LH0080)

Due to the CMOS static structure, it provides low power consumption and large operating margin.

The power save mode can be obtained with a software control on the models suffixed with "L".

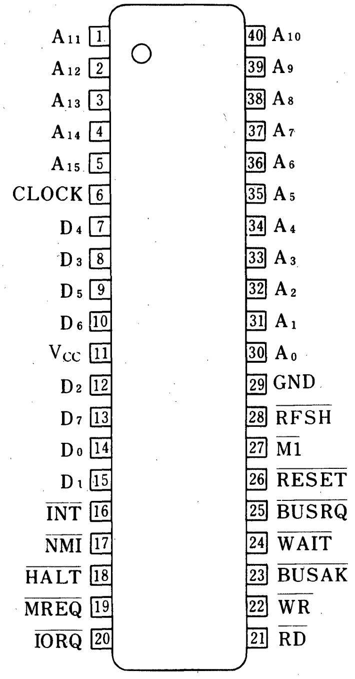
Features

1. Z80 CMOS CPU
2. Compatible with the Z80 NMOS CPU (LH0080)
3. 158 instructions
4. 22 registers
5. 3 modes of maskable interrupt and a nonmaskable interrupt
6. Instruction fetch cycle 1.6 μs/1.0 μs
7. Single +5V power supply and single phase clock
8. All inputs and outputs except clock input fully TTL compatible
9. Fully static operation DC to 2.5MHz/DC to 4MHz
10. Low power consumption
11. Power save mode (L suffix)
12. 40-pin DIP (DIP40-P-600)
    44-pin QFP (QFP44-P-1010A)

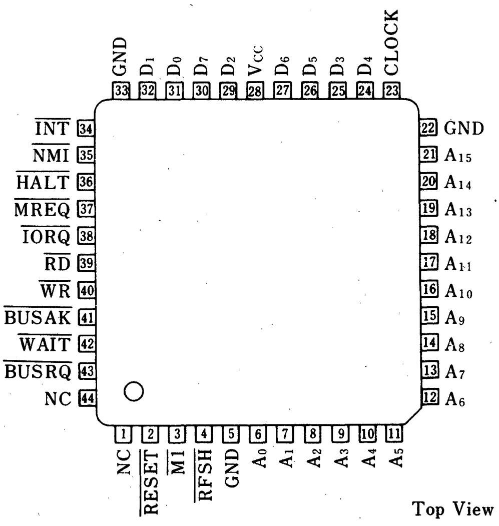
Note: The Z80 CMOS CPU (LH5080) is compatible with the Z80 NMOS CPU (LH0080). So there is no description here about the pins, CPU registers, architecture, interrupts, basic timings, and instruction sets. Refer back to the Z80 NMOS CPU described earlier.

Pin Connections

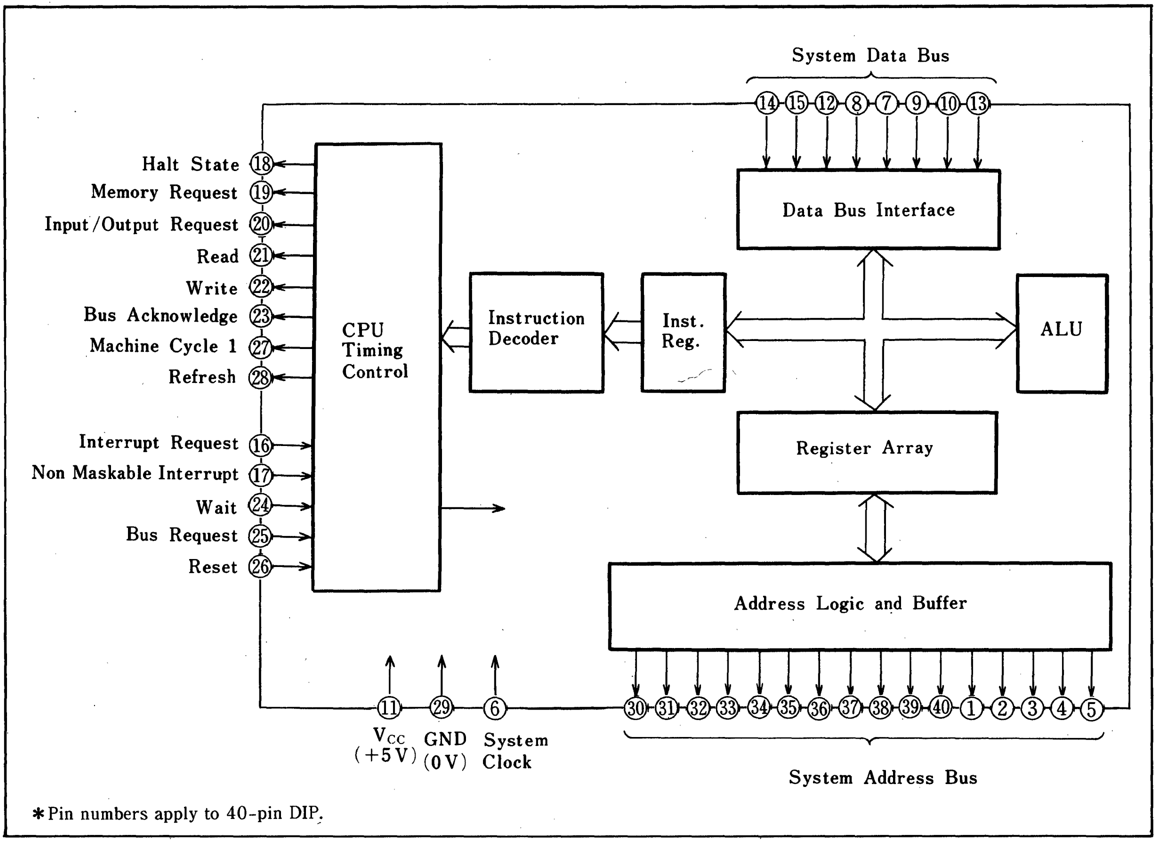
LH5080/LH5080L/LH5080A/LH5080AL



LH5080M/LH5080LM/LH5080AM/LH5080ALM



Block Diagram



*Pin numbers apply to 40-pin DIP.

Ordering Information

LH5080 X X X

- Package
  - Blank: 40-pin DIP (DIP40-P-600)
  - M: 44-pin QFP (QFP44-P-1010A)
- Power save mode
  - Blank: No power save
  - L: Power save
- Clock frequency
  - Blank: 2.5MHz
  - A: 4MHz
- Model No.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Supply voltage	V_{CC}	−0.3 to +7.0	V
Input voltage	V_{IN}	−0.3 to V_{CC}+0.3	V
Output voltage	V_{OUT}	−0.3 to V_{CC}+0.3	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃

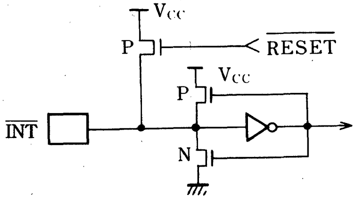
DC Characterisitics

(V_{CC}=5V±10%, Ta=0 to+70°C)

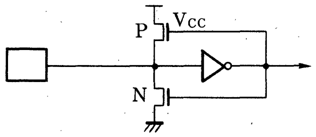
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Clock input low voltage	V_{ILC}			−0.3		0.45	V	
Clock input high voltage	V_{IHC}			V_{CC}−0.6		V_{CC}+0.3	V	
Input low voltage	V_{IL}			−0.3		0.8	V	
Input high voltage	V_{IH}			2.4		V_{CC}	V	
Output low voltage	V_{OL}	I_{OL}=1.8mA				0.4	V	
Output high voltage	V_{OH}	$I_{OH}=-250\mu A$		2.4			V	
		$I_{OH}=-50\mu A$		V_{CC}−0.4				
Current consumption	I_{CC}	V_{IL}=0.4V,V_{IH}=V_{CC}−0.4V	LH5080/L		10	15	mA	1
		Outputs open	LH5080A/AL		15	20		2
Input leakage current	$\mid I_{LI} \mid$	V_{IN}=0V, V_{CC}				10	μA	3
3-state output leakage current	$\mid I_{LOH} \mid$	V_{OUT}=V_{CC}				10	μA	4
3-state output leakage current	$\mid I_{LOL} \mid$	V_{OUT}=0V				10	μA	4
Data bus leakage current	$\mid I_{LD} \mid$	$0 \leqq V_{IN} \leqq V_{CC}$				10	μA	
Current consumption in PS mode (LH5080L/LH5080LM)	I_{CCPS}	V_{IH}=0V, V_{CC}	LH5080L		50	150	μA	1
		Outputs open	LH5080AL		80	200		2

Note 1: TcC=400ns
Note 2: TcC=250ns

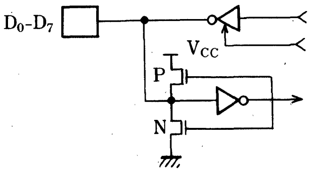
Note 3−(1): For | I_{LI} | Specification, see below circuits of $\overline{INT}$ pin.



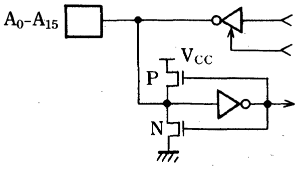
Note 3−(2): For | I_{LI} | Specification, see below circuits of $\overline{WAIT}$, $\overline{NMI}$ and $\overline{BUSRQ}$.



Note 3 and 4: For | I_{LI} |, | I_{LOH} | and | I_{LOL} | Specifications, see below circuit of D_0-D_7 pins.



Note 4: For | I_{LOH} | and | I_{LOL} | Specifications, see below circuit of A_0-A_{15} pins.



Capacititance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground	5	pF
Input capacitance	C_{IN}		6	pF
Output capacitance	C_{OUT}		10	pF

■ AC Characteristics

($V_{CC}=5V\pm10\%$, Ta=0 to+70°C)

No.	Parameter	Symbol	LH5080		LH5080A		Unit
			MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400*		250*		ns
2	Clock pluse high width	TwCh	180*		110*		ns
3	Clock pluse low width	TwCl	180		110		ns
4	Clock fall time	TfC		30		30	ns
5	Clock rise time	TrC		30		30	ns
6	Clock ↑ to address valid delay	TdCr (A)		145		110	ns
7	Address valid to $\overline{\text{MREQ}}$ ↓ delay	TdA (MREQf)	125*		65*		ns
8	Clock ↓ to $\overline{\text{MREQ}}$ ↓ delay	TdCf (MREQf)		100		85	ns
9	Clock ↑ to $\overline{\text{MREQ}}$ ↑ delay	TdCr (MREQr)		100		85	ns
10	$\overline{\text{MREQ}}$ pulse high width	TwMREQh	170*		110*		ns
11	$\overline{\text{MREQ}}$ pulse low width	TwMREQl	360*		220*		ns
12	Clock ↓ to $\overline{\text{MREQ}}$ ↑ delay	TdCf (MREQr)		100		85	ns
13	Clock ↓ to $\overline{\text{RD}}$ ↓ delay	TdCf (RDf)		130		95	ns
14	Clock ↑ to $\overline{\text{RD}}$ ↑ delay	TdCr (RDr)		100		85	ns
15	Data setup time to clock ↑	TsD (Cr)	50		35		ns
16	Data hold time after $\overline{\text{RD}}$ ↑	ThD (RDr)	15		15		ns
17	$\overline{\text{WAIT}}$ setup time to clock ↓	TsWAIT (Cf)	70		70		ns
18	$\overline{\text{WAIT}}$ hold time after clock ↓	ThWAIT (Cf)	15		15		ns
19	Clock ↑ to $\overline{\text{Ml}}$ ↓ delay	TdCr (Mlf)		130		100	ns
20	Clock ↑ to $\overline{\text{Ml}}$ ↑ delay	TdCr (Mlr)		130		100	np
21	Clock ↑ to $\overline{\text{RFSH}}$ ↓ delay	TdCr (RFSHf)		180		130	ns
22	Clock ↑ to $\overline{\text{RFSH}}$ ↑ delay	TdCr (RFSHr)		150		120	ns
23	Clock ↓ to $\overline{\text{RD}}$ ↑ delay	TdCf (RDr)		110		85	ns
24	Clock ↑ to $\overline{\text{RD}}$ ↓ delay	TdCr (RDf)		100		85	ns
25	Data setup to clock ↑ during M_2, M_3, M_4 or M_5 cycles	TsD (Cf)	60		50		ns
26	Address stable prior to $\overline{\text{IORQ}}$ ↓	TdA (IORQf)	320*		180*		ns
27	Clock ↑ to $\overline{\text{IORQ}}$ ↓ delay	TdCr (IORQf)		90		75	ns
28	Clock ↓ to $\overline{\text{IORQ}}$ ↑ delay	TdCf (IORQr)		110		85	ns
29	Data stable prior to $\overline{\text{WR}}$ ↓ (memory cycle)	TdDm (WRf)	190*		80*		ns
30	Clock ↓ to $\overline{\text{WR}}$ ↓ delay	TdCf (WRf)		90		80	ns
31	$\overline{\text{WR}}$ pulse width	TwWR	360*		220*		ns
32	Clock ↓ to $\overline{\text{WR}}$ ↑ delay	TdCf (WRr)		100		80	ns
33	Data stable prior to $\overline{\text{WR}}$ ↓ (I/O cycle)	TdDi (WRf)	20*		−10*		ns
34	Clock ↑ to $\overline{\text{WR}}$ ↓ delay	TdCr (WRf)		80		65	ns
35	Data stable from $\overline{\text{WR}}$ ↑	TdWRr (D)	120*		60*		ns
36	Clock ↓ to $\overline{\text{HALT}}$ ↑	TdCf (HALT)		300		300	ns
37	$\overline{\text{NMI}}$ pulse width	TwNMI	80		80		ns
38	$\overline{\text{BUSREQ}}$ setup time to clock ↑	TsBUSRQ (Cr)	80		50		ns
39	$\overline{\text{BUSREQ}}$ hold time after clock ↑	ThBUSRQ (Cr)	15		15		ns
40	Clock ↑ to $\overline{\text{BUSACK}}$ ↓ delay	TdCr (BUSAKf)		120		100	ns
41	Clock ↓ to $\overline{\text{BUSACK}}$ ↑ delay	TdCf (BUSAKr)		110		100	ns
42	Clock ↑ data float delay	TdCr (Dz)		90		90	ns
43	Clock ↑ to control output float delay ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$)	TdCr (CTz)		110		80	ns

↑ Rising edge, ↓ Falling edge.

5

No.	Parameter	Symbol	LH5080		LH5080A		Unit
			MIN.	MAX.	MIN.	MAX.	
44	Clock ↑ to address float delay	TdCr (Az)		110		90	ns
45	$\overline{\text{MREQ}}$ ↑, $\overline{\text{IORQ}}$ ↑, $\overline{\text{RD}}$ ↑, and $\overline{\text{WR}}$ ↑ to address hold time	TdCTr (A)	160*		80*		ns
46	$\overline{\text{RESET}}$ to clock ↑ setup time	TsRESET (Cr)	90		60		ns
47	Clock ↑ to $\overline{\text{RESET}}$ hold time	ThRESET (Cr)	15		15		ns
48	$\overline{\text{INT}}$ to clock ↑ setup time	TsINTf (Cr)	80		80		ns
49	Clock ↑ to $\overline{\text{INT}}$ hold time	ThINTr (Cr)	15		15		ns
50	$\overline{\text{Ml}}$ ↓ to $\overline{\text{IORQ}}$ ↓ delay	TdMlf (IORQf)	920*		565*		ns
51	Clock ↓ to $\overline{\text{IORQ}}$ ↓ delay	TdCf (IORQf)		110		85	ns
52	Clock ↑ to $\overline{\text{IORQ}}$ ↑ delay	TdCf (IORQr)		100		85	ns
53	Clock ↓ to data valid delay	TdCf (D)		230		150	ns

↑ Rising edge, ↓ Falling edge

* For clock periods other than the minimums shown in the table, calculate parameters using the expressions in the table below

Footnotes to AC Characteristics

No.	Symbol	Formula
1	TcC	TwCh+TwCl+TrC+TfC
2	TwCh	MAX. 200 μs
7	TdA (MREQf)	TwCh+TfC−75
10	TwMREQh	TwCh+TfC−30
11	TwMREQ1	TcC−40
26	TdA (IORQf)	TcC−80
29	TdD (WRf)	TcC−210
31	TwWR	TcC−40
33	TdD (WRf)	TwCl+TrC−180
35	TdWRr (D)	TwCl+TrC−80
45	TdCTr (A)	TwCl+TrC−40
50	TdM1f (IORQf)	2TcC+TwCh+TfC−80

AC Test Conditions

- ●Input voltage amplitude : 0.4V to 2.8V
- ●Clock input voltage amplitude : 0.4V to $V_{cc}-0.6$V
- ●Input signal rise and fall time : 10 ns
- ●Input judge level : 0.8V and 2.0V
- ●Output judge level : 0.8V and 2.0V
- ●Output load : ITTL+100 pF

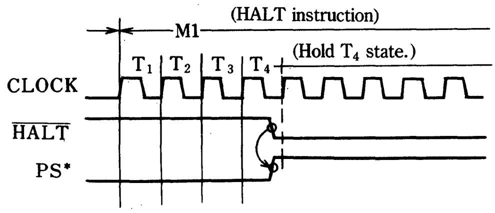
Power Save Function

The LH5080L series features the power save (PS) function. After a HALT instruction has been executed, the internal clock signal is automatically cut off to bring the CPU into the halt mode.

(1) PS mode setting

With a HALT instruction executed, the PS mode will be automatically established. In this mode, the internal clock signal is cut off to save the power consumed for the clock signal operation. Cutting an external clock signal does not give any problem inside, therefore, in this mode. To cut off the external clock, it is possible to utilize the rise timing of a $\overline{\text{HALT}}$ signal output. It should be noted, however, that this timing cannot be used to restart the external clock.

In the PS mode, the bus request ($\overline{\text{BUSRQ}}$) is not accepted and the memory refresh is not done, either.



* PS is internal signal, and not output externally.

PS mode setting

(2) PS mode clear

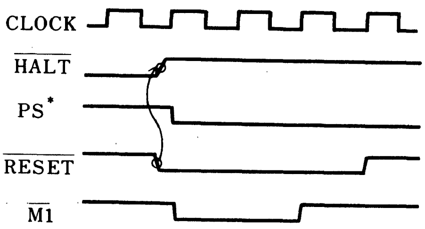
The PS mode is cleared by any of the following; reset ($\overline{\text{RESET}}$), non-maskable interrupt ($\overline{\text{NMI}}$) and maskable interrupt ($\overline{\text{INT}}$).

When the external clock is shut down in the PS mode, a stable clock signal must be input before clearing the PS mode.

(i) Clearing with $\overline{\text{RESET}}$: Input the $\overline{\text{RESET}}$ signal for more than 3 clock cycles. The PS mode is then cleared and the reset just as before is carried out.

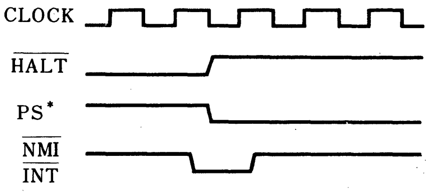
(ii) Clearing with $\overline{\text{NMI}}$: Input the $\overline{\text{NMI}}$ signal (edge trigger) to clear the PS mode and to carry out the instruction next to the HALT. Now the non-maskable interrupt processing routine will be introduced.

(iii) Clearing with $\overline{\text{INT}}$: Input the $\overline{\text{INT}}$ signal (level trigger) regardless of which state the interrupt enable flag is in. The PS mode is now cleared and the HALT instruction executed. If the interrupt enable flag is set up and the $\overline{\text{INT}}$ signal is "Low" at the clock pulse rise timing in the last clock cycle of the HALT instruction, the maskable interrupt processing routine will be introduced as the next machine cycle.



* PS is internal signal, and not output externally.

Halt clear by $\overline{\text{RESET}}$



* PS is internal signal, and not output externally.

PS mode clear by $\overline{\text{NMI}}$, $\overline{\text{INT}}$ signal

LH5081 Z80 CMOS PIO Parallel I/O Controller

Description

The LH5081 is a Z80 PIO fabricated with CMOS silicon gate process technology and is compatible with the conventional Z80 NMOS PIO (LH0081)

Due to the CMOS static structure, it provides low power consumption and large operating margin.

The power save mode can be obtained with a software control on the models suffixed with "L".

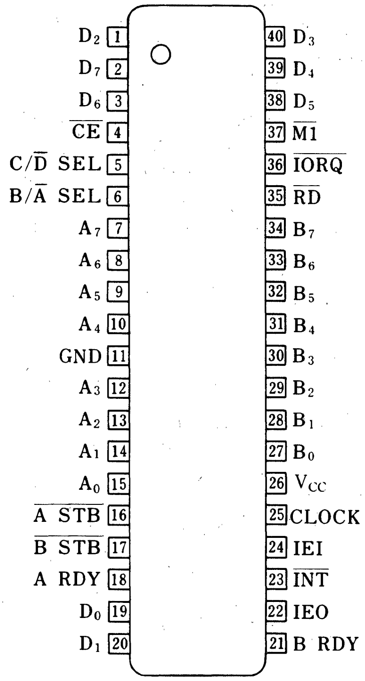
Features

1. Z80 CMOS PIO
2. Compatible with NMOS Z80 PIO (LH0081)
3. Tow independent 8-bit bidirectional peripheral interface ports with handshake data transfer control
4. 4 programmable operating modes
   - Byte input mode
   - Byte output mode
   - Byte bidirectional bus mode (Port A only)
   - Byte control mode
5. Programmable interrupt on peripheral status conditions
6. Vectored daisy chain priority interrupt
7. Darlington transistor drive capability (Port B output)
8. All inputs and outputs except clock input fully TTL compatible
9. Single +5V power supply and single phase clock
10. Fully static operation (DC to 2.5MHz/4MHz/6MHz)
11. Low power consumption
12. Power save mode (L suffix)
13. Status read mode (L suffix)
14. 40-pin DIP (DIP40-P-600)
    44-pin QFP (QFP44-P-1010A)

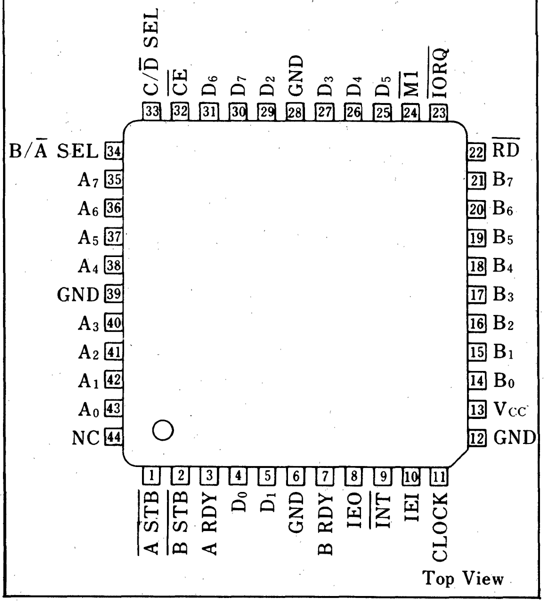
Note: The Z80 CMOS CPU (LH5081) is compatible with the Z80 NMOS PIO (LH0081). So there is no description here about the pins, programming, and basic timings waveforms. Refer back to the Z80 NMOS PIO described earlier.

Pin Connections

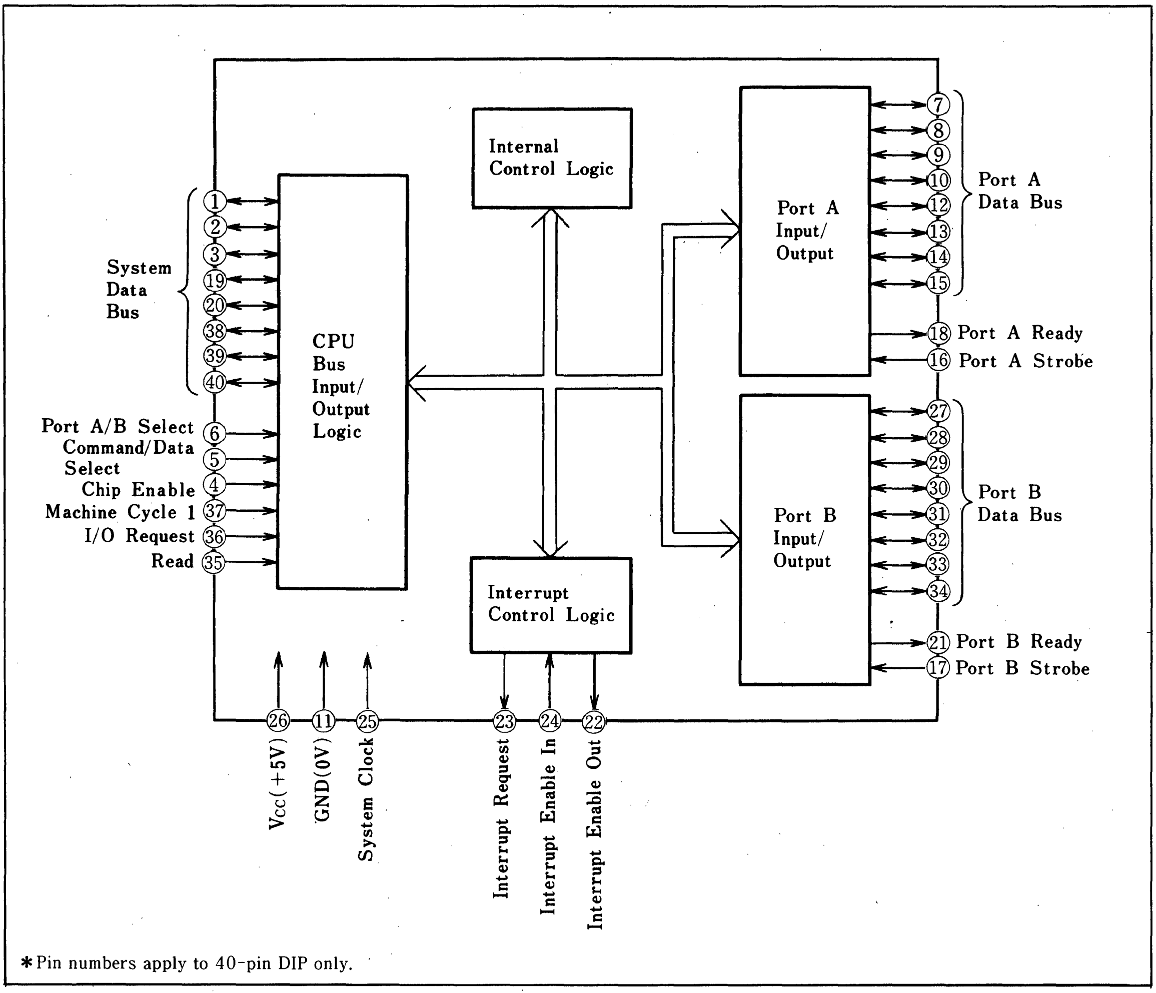
LH5081/LH5081A/LH5081B
LH5081L/LH5081AL/LH5081BL



LH5081M/LH5081AM
LH5081LM/LH5081ALM



Block Diagram



∗Pin numbers apply to 40-pin DIP only.

Ordering Information

LH5081 X X X

- Package ∗
  - Blank: 40-pin DIP (DIP40-P-600)
  - M: 44-pin QFP (QFP44-P-1010A)
- Power save mode
  - Blank: No power save
  - L: Power save
- Clock frequency
  - Blank: 2.5MHz
  - A: 4MHz
  - B: 6MHz
- Model No.

∗ The 6MHz type is packaged in 40-pin DIP only.

■ Absolute Maximum Ratings

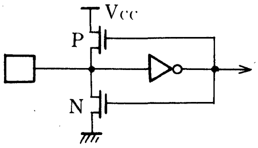
Parameter	Symbol	Ratings	Unit
Supply voltage	V_{CC}	−0.3 to 7.0	V
Input voltage	V_{IN}	−0.3 to V_{CC}+0.3	V
Output voltage	V_{OUT}	−0.3 to V_{CC}+0.3	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃

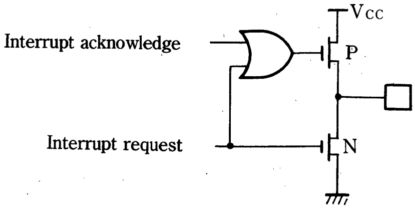
■ DC Characteristics

(V_{CC}=5V±10%, Ta=0 to +70℃)

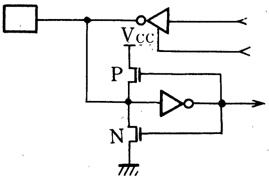
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Clock input low voltage	V_{ILC}			−0.3		0.45	V	
Clock input high voltage	V_{IHC}			V_{CC}−0.6		V_{CC}+0.3	V	
Input low voltage	V_{IL}			−0.3		0.8	V	
Input high voltage	V_{IH}			2.2		V_{CC}+0.3	V	
Output low voltage	V_{OL}	I_{OL}=2mA				0.4	V	
Output high voltage	V_{OH}	I_{OH}=−250μA		2.4			V	
		I_{OH}=−50μA		V_{CC}−0.4V			V	
Current consumption	I_{CC}	V_{IL}=0.4,V_{IH}=V_{CC}−0.4V, output open	LH5081/L		1	3	mA	1
			LH5081A/AL		4	6	mA	2
			LH5081B		6	8	mA	3
Input leakage current	$\mid I_{LI}\mid$	V_{IN}=0V, V_{CC}				10	μA	4
3-state output leakage current	$\mid I_{LOH}\mid$	V_{OUT}=V_{CC}				10	μA	5
3-state output leakage current	$\mid I_{LOL}\mid$	V_{OUT}=0V				10	μA	5
Data bus leakage current input	$\mid I_{LD}\mid$	0≦V_{IN}≦V_{CC}				10	μA	
Darlington drive current	I_{OHD}	V_{OH}=1.5V, Port B only		−1.5			mA	
Current consumption in PS mode	I_{CCPS}	V_{IN}=0V, V_{CC} Outputs open	LH5081L		1	100	μA	1
			LH5081AL		1	100		2

Note 1: TcC=400ns, V_{IL}=0.4V, V_{IH}=V_{CC}−0.4V, outputs open.
Note 2: TcC=250ns
Note 3: TcC=167ns
Note 4: (1) For $\mid I_{LI}\mid$ specification, see below circuit of $\overline{\text{A STB}}$ and $\overline{\text{B STB}}$.2978 (2) The $\overline{\text{INT}}$ pin is arraged as shown below.





Note 5: For $\mid I_{LOH}\mid$ and $\mid I_{LOL}\mid$ specifications, see below circuit of A_0-A_7 and B_0-B_7



■ Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground	7	pF
Input capacitance	C_{IN}		7	pF
Output capacitance	C_{OUT}		10	pF

■ AC Characteristics

(V_{CC}=5V±10%, Ta=0 to +70℃)

No.	Parameter	Symbol	LH5081		LH5081A		LH5081B		Unit	Note
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
1	Clock cycle time	TcC	400	(Note 1)	250	(Note 1)	165	(Note 1)	ns	
2	Clock high width	TwCh	170		105		65		ns	
3	Clock low width	TwCl	170		105		65		ns	
4	Clock fall time	TfC		30		30		20	ns	
5	Clock rise time	TrC		30		30		20	ns	
6	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORQ}$ ↓ setup time	TsCS (RI)	50		50		50		ns	6
7	Any hold times for specified setup time	Th	15		15		15		ns	
8	$\overline{RD}$, $\overline{IORQ}$ to clock ↑ setup time	TsRI (C)	115		115		70		ns	
9	$\overline{RD}$, $\overline{IORQ}$ ↓ to data output delay	TdRI (DO)		430		380		300	ns	2
10	$\overline{RD}$, $\overline{IORQ}$ ↑ to data output float delay	TdRI (DOs)		160		110		70	ns	
11	Data in to clock ↑ setup time	TsDI (C)	50		50		40		ns	C_L=50pF
12	$\overline{IORQ}$ ↓ to data out delay (INTACK cycle)	TdIO (DOI)		340		160		120	ns	3
13	$\overline{M1}$ ↓ to clock ↑ setup time	TsM1 (Cr)	210		90		70		ns	
14	$\overline{M1}$ ↑ to clock ↓ setup time ($\overline{M1}$ cycle)	TsM1 (Cf)	0		0		0		ns	8
15	$\overline{M1}$ ↓ to IEO ↓ delay (interrupt immediately precceding $\overline{M1}$ ↓)	TdM1 (IEO)		300		190		100	ns	5, 7
16	IEI to $\overline{IORQ}$ ↓ setup time (INTACK cycle)	TsIEI (IO)	140		140		100		ns	7
17	IEI ↓ to IEO ↓ delay	TdIEI (IEOf)		190		130		120	ns	5 C_L=50pF
18	IEI ↑ to IEO ↑ delay (after ED decode)	TdIEI (IEOr)		210		160		160	ns	5
19	$\overline{IORQ}$ ↑ to clock ↓ setup time (to activate READY on next clock cycle)	TcIO (C)	220		200		170		ns	
20	Clock ↓ to READY ↑ delay	TdC (RDYr)		200		190		170	ns	5 C_L=50pF
21	Clock ↓ to READY ↓ delay TdC (RDYf)	TdC (RDYr)		150		140		120	ns	5
22	$\overline{STROBE}$ pulse width	TwSTB	150		150		120		ns	4
23	$\overline{STROBE}$ ↑ to clock ↓ setup time (to activate READY on next clock cycle)	TsSTB (C)	220		220		150		ns	5
24	$\overline{IORQ}$ ↑ to PORT DATA stable delay (mode 0)	TdIO (PD)		200		180		160	ns	5
25	PORT DATA to $\overline{STROBE}$ ↑ setup time (mode 1)	TsPD (STB)	260		230		190		ns	
26	$\overline{STROBE}$ ↓ to PORT DATA stable (mode 2)	TdSTB (PD)		230		210		180	ns	5
27	$\overline{STROBE}$ ↑ to PORT DATA float delay (mode 2)	TdSTB (PDr)		200		180		160	ns	C_L=50pF
28	PORT DATA match to $\overline{INT}$ ↓ delay (mode 3)	TdPD (INT)		540		490		430	ns	
29	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ delay	TdSTB (INT)		490		440		350	ns	

↑ Rising edge, ↓ Falling edge

Note 1 : TcC=TwCh+TwCl+TrC+TfC.

Note 2 : Increase TdRI (DO) by 10 ns for each 50 pF increase in load up to 200 pF max.

Note 3 : Increase TdIO (DOI) by 10 ns for each 50 pF, increase in loading up to 200 pF max.

Note 4 : For Mode 2 : TwSTB>TsPD (STB).

Note 5 : Increase these values by 2 ns for each 10 pF increase in loading up to 100 pF max.

Note 6 : TsCS (RI) may be reduced. However, the time substracted from TsCS (RI) will be added to TdRI (DO).

Note 7 : 2.5 TcC > (N-2) TdIEI (IEOf)+TdM1 (IEO)+TsIEI (IO)+TTL Buffer Delay, if any.

Note 8 : $\overline{M1}$ must be active for a minimum of two clock cycles to reset the PIO.

AC Test Conditions :

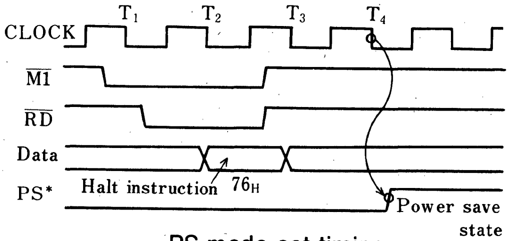
- Input voltage amplitude : 0.4V to 2.8V
- Clock input voltage amplitude : 0.4V to Vcc−0.6V
- Input signal rise and fall time : 10ns
- Input judge level : 0.8V and 2.0V
- Output judge level : 0.8V and 2.0V
- Output load : ITTL+100pF (unless otherwise specified)

■ Power Save and Status Information Read Function

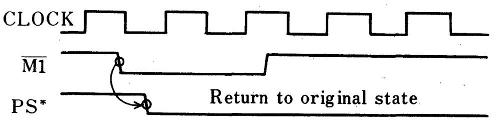
Unlike the LH0081/LH5081, the LH5081L series has the power save (PS) and status information read functions.

(1) Power save function

(i) PS mode setting When the CPU (LH5080L series) has executed an HALT instruction in the PS mode, the LH5081L series reads this HALT instruction to automatically go into the PS mode. Now the internal clock signal is cut off. Therefore, cutting an external clock input gives no problem inside in this mode.



PS mode set timing



* PS is internal signal and not output externally.

PS mode clear timing

(ii) PS mode clear The PS mode is cleared by detecting the fall of the $\overline{M1}$ signal. When the external clock is off in the PS mode, however, a stable clock signal must be input before clearing the PS mode.

When the CPU (LH5080L series) is cleared from the PS mode and comes into the next fetch cycle, therefore, the LH5081L series is also cleared from its PS mode at the fall of the first $\overline{M1}$ signal in this cycle.

The PS mode clearing can be done by issuing an interrupt request.

Set up the interrupt generate conditions in Mode 3 of the LH5081L series. By this, an interrupt request (INT) is issued even in the PS mode, the CPU (LH5080L series) is cleared from the PS mode, and thus LH5081L series is also cleared.

(2) Status information read

Under the following conditions, the mode setup bits and handshake signals of Port A and Port B are read from the data bus during the read cycle. See the chart below.

Conditions: $\overline{CE}$ = "Low", $\overline{RD}$ = "Low", $\overline{IORQ}$ = "Low", C/D = "High", B/$\overline{A}$ = X (undefined)

Bits	7	6	5	4	3	2	1	0
	AM1	AM0	ARDY	$\overline{ASTB}$	BM1	BM0	BRDY	$\overline{BSTB}$

Uppermost two bits of the A port mode control registor (Set mode bits)

Uppermost two bits of the B port mode control registor (Set mode bits)

LH5081L series status information words

LH5082 Z80 CMOS CTC Counter Timer Circuit

■ Description

The LH5082 is a Z80 CTC fabricated with CMOS silicon-gate process technology and is compatible with the conventional Z80 NMOS CTC (LH0082)

The LH5082 is designed with CMOS fully static circuits and so provides low power consumption and wide range power supply voltage operation.

The LH5082L/LH5082LM provides power save mode controlled by software.

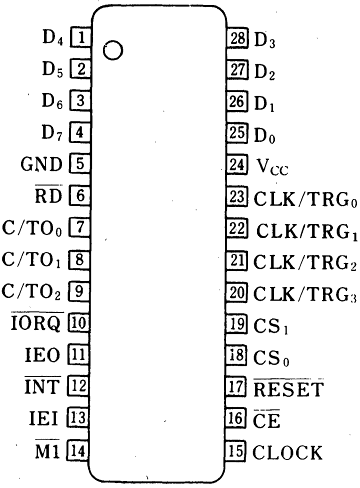
■ Features

1. Z80 CMOS CTC
2. Compatible with the Z80 NMOS CTC (LH0082)
3. 4 independent programmable 8-bit counter/ 16-bit timer channels
4. Selectable counter/timer mode for each channel
5. Programmable interrupt triggered by counter/ timer
6. Downcounters reloaded automatically at zero count
7. Readable downcounters
8. Selectable 16 or 256 prescaler (timer mode)
9. Selectable positive or negative triggers for timer and selectable positive or negative clock edge for counter
10. ZC/TO outputs of three channels capable of driving Darlington transistors
11. Vectored and daisy chain piority interrupt
12. Single +5V power supply and single phase clock
13. All inputs and outputs except clock input fully TTL compatible
14. Fully static operation (DC to 2.5MHz/4MHz/6MHz)
15. Low power consumption
16. Power save mode (L suffix)
17. 28-pin DIP (DIP28-P-600) 44-pin QFP (QFP44-P-1010A)

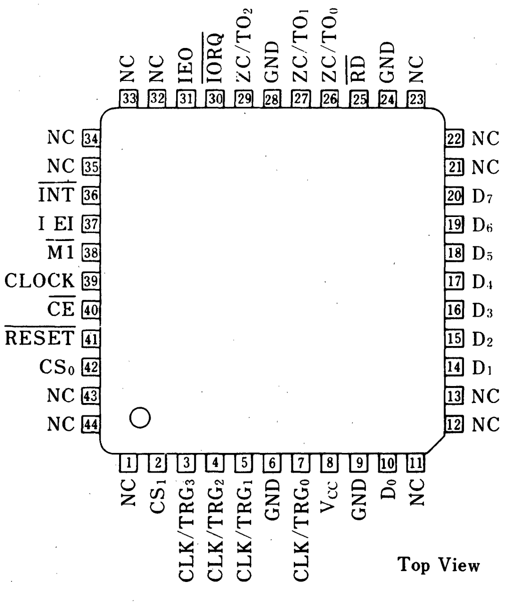
Note: The Z80 CMOS CTC (LH5082) is compatible with the Z80 NMOS CTC (LH0082). So there is no description here about the pins, programming, and basic timing waveform. Refer back to the Z80 NMOS CTC described earlier.

■ Pin Connections

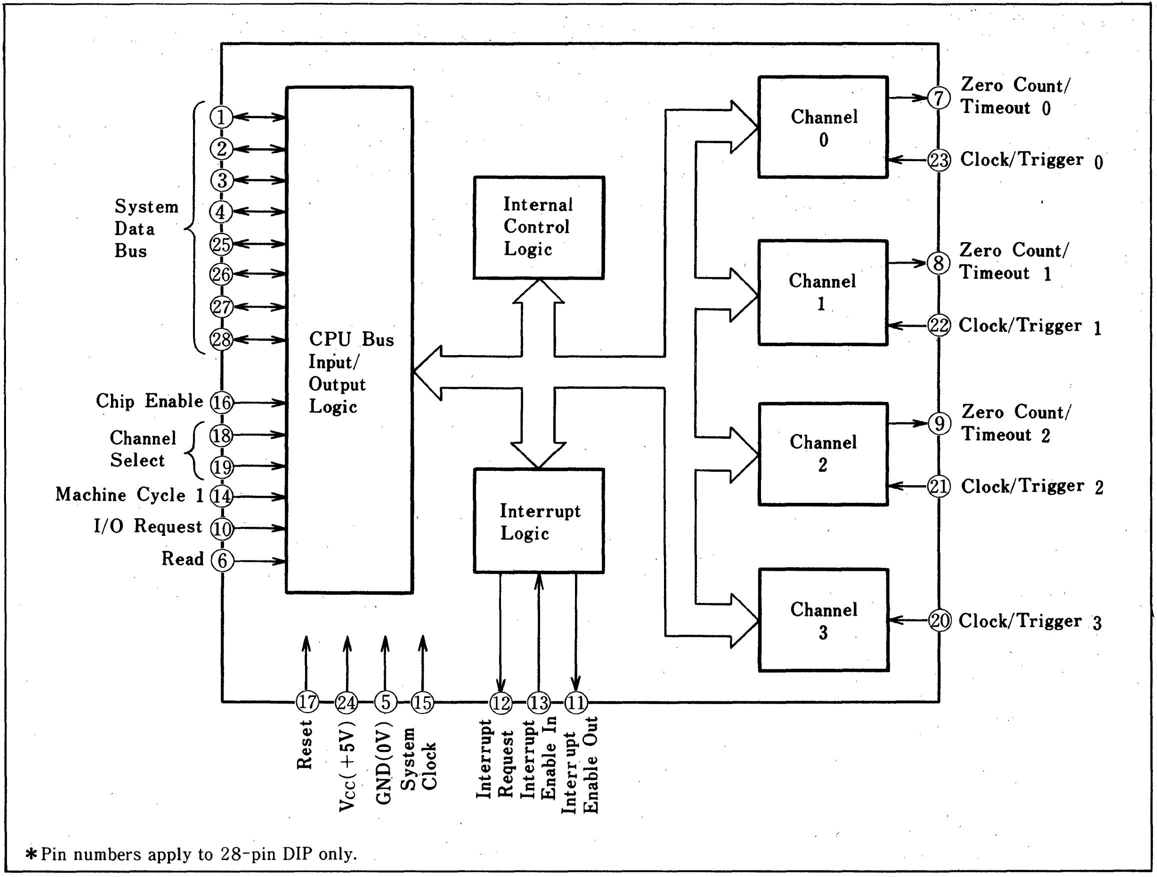
LH5082/LH5082A/LH5082B
LH5082L/LH5082AL/LH5082BL



LH5082M/LH5082AM
LH5082LM/LH5082ALM



Block Diagram



*Pin numbers apply to 28-pin DIP only.

Ordering Information

```
LH5082 X X X
 |     | | └ Package *
 |     | |     Blank: 28-pin DIP (DIP28-P-600)
 |     | |     M: 44-pin QFP (QFP44-P-1010A)
 |     | └──Power save mode
 |     |       Blank: No power save
 |     |       L: Power save
 |     └────Clock frequency
 |             Blank: 2.5MHz
 |             A: 4MHz
 |             B: 6MHz
 └──────────Model No.
```

* The 6MHz type is packaged in 28-pin DIP only.

Absolute Maximum Ratings

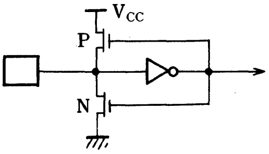
Parameter	Symbol	Ratings	Unit
Supply voltage	V_{CC}	−0.3 to +7.0	V
Input voltage	V_{IN}	−0.3 to V_{CC} +0.3	V
Output voltage	V_{OUT}	−0.3 to V_{CC} +0.3	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃

DC Characteristics

(V_{CC}=5V±10%, Ta=0 to +70℃)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	Note
Clock input low voltage	V_{ILC}			−0.3		0.45	V	
Clock input high voltage	V_{IHC}			V_{CC}−0.6		V_{CC}+0.3	V	
Input low voltage	V_{IL}			−0.3		0.8	V	
Input high voltage	V_{LH}			2.2		V_{CC}	V	
Output low voltage	V_{OL}	I_{OL}=2mA				0.4	V	
Output high voltage	V_{OH}	I_{OH}=−250μA		2.4			V	
		I_{OH}=−50μA		V_{CC}−0.4V			V	
Current consumption	$\|I_{CC}\|$	V_{IL}=0.4V,V_{IH}=V_{CC}−0.4V, outputs open	LH5082/L		2	4	mA	1
			LH5082A/AL		3	6	mA	2
			LH5082B/BL		5	8	mA	3
Input leakage current	$\|I_{LI}\|$	V_{IN}=0V, V_{CC}				10	μA	4
3-state output leakage current	$\|I_{LOH}\|$	V_{OUT}=V_{CC}				10	μA	
3-state output leakage current	$\|I_{IOL}\|$	V_{OUT}=0V				10	μA	
Darlington drive current	I_{OHD}	V_{OH}=1.5V Applicated to ZC/TO_0-ZC/TO_2		−1.5			mA	
Current consumption in PS mode	I_{CCPS}	V_{IN}=0V, V_{CC} Outputs open	LH5082L		1	100	μA	1
			LH5082AL		1	100		2
			LH5082BL		1	100		3

Note 1: TcC=400ns
Note 2: TcC=250ns
Note 3: TcC=167ns
Note 4: For $|I_{LI}|$ specification, see below circuit of CLK/TRG_0-CLK/TG_3.



Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground	5	pF
Input capacitance	C_{IN}		5	pF
Output capacitance	C_{OUT}		10	pF

■ AC Characteristics

(V_{CC}=5V±10%, Ta=0 to +70℃)

No.	Parameter	Symbol	LH5082		LH5082A		LH5082B		Unit	Note
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
1	Clock cycle time	TcC	400	(Note 1)	250	(Note 1)	165		ns	
2	Clock high width	TwCh	170		105		65		ns	
3	Clock low width	TwCl	170		105		65		ns	
4	Clock fall time	TlC		30		30		20	ns	
5	Clock rise time	TrC		30		30		20	ns	
6	All hold times	Th	15		15		15		ns	
7	$\overline{\text{CS}}$ to clock ↑ setup time	TcCS (C)	250		160		100		ns	
8	$\overline{\text{CE}}$ to clock ↑ setup time	TsCE (C)	200		150		100		ns	
9	$\overline{\text{IORQ}}$ ↓ to clock ↑ setup time	TsIO (C)	250		115		70		ns	
10	$\overline{\text{RD}}$ ↓ clock ↑ setup time	TsRD (C)	240		115		70		ns	
11	Clock ↑ to data out delay	TdC (DO)		240		200		130	ns	2
12	Clock ↓ to data out float dalay	TdC (DOz)		230		110		90	ns	
13	Data In to clock ↑ setup time	TsDI (C)	60		50		40		ns	
14	$\overline{\text{MI}}$ to clock ↑ setup time	TsM1 (C)	210		90		70		ns	
15	$\overline{\text{M1}}$ ↓ to IEO ↓ delay (interrupt immediately preceding $\overline{\text{M1}}$)	TdM1 (IEO)		300		190		130	ns	3
16	$\overline{\text{IORQ}}$ ↓ to data out delay (INTA cycle)	TdIO (DOI)		340		160		110	ns	2
17	IEI ↓ to IEO ↓ delay	TdIEI (IEOf)		190		130		100	ns	3
18	IEI ↑ to IEO ↑ delay ↑ (after ED decode)	TdIEI (IEOr)		220		160		110	ns	3
19	Clock ↑ to $\overline{\text{INT}}$ ↓ delay	TdC (INT)		TcC+200		TcC+140		TcC+120	ns	4
20	CLK/TRG ↑ to $\overline{\text{INT}}$ ↓ delay (tsCTR (C) satisfied)	TdClk (INT)		TcC+230		TcC+160		TcC+130	ns	5
	CLK/TRG ↑ to $\overline{\text{INT}}$ ↓ delay (tsCTR (C) not satisfied)	TdCLK (INT)		2TcC+530		2TcC+370		2TcC+280	ns	5
21	CLK/TRG cycte time	TcCTR	2TcC		2TcC		2TcC		ns	5
22	CLK/TRG rise time	TrCTR		50		50		40	ns	
23	CLK/TRG fall time	TfCTR		50		50		40	ns	
24	CLK/TRG low width	TwCTR1	200		200		120		ns	
25	CLK/TRG high width	TwCTRh	200		200		120		ns	
26	CLK/TRG ↑ to clock ↑ setup time for immediate count	TsCTR (Cf)	300		210		150		ns	5
27	CLK/TRG ↑ to clock ↑ setup time for enabling of prescaler on following clock ↑	TsCTR (Ct)	210		210		150		ns	4
28	Clock ↑ to ZC/TO ↑ delay	TdC (ZC/TOr)		260		190		140	ns	
29	Clock ↑ to ZC/$\overline{\text{TO}}$ ↓ delay	TdC (ZC/TOf)		190		190		140	ns	
30	IEI setup time $\overline{\text{IORQ}}$ ↓ (INTA cycle)	TsIEI (IO)	140		140		140		ns	

↑ Rising edge, ↓ Falling edge

[A] 2.5 TcC>(n-2) TdIEI (IEOf)+TdMI (IEO) +TsIEI (IO)+TTL buffer delay, if any.

[B] $\overline{\text{RESET}}$ must be active for minimum of 3 clock cycles

Note 1 : TcC=TwCh+TwCl+TrC+TfC.

Note 2 : Increase delay by 10 ns for each 50 pF increase in loading, 200 pF maximum for data lines, and 100 pF for control lines.

Note 3 : Increase delay by 2 ns for each 10 pF increase in loading, 100 pF maximum.

Note 4 : Timer mode.

Note 5 : Counter mode.

* All timing are preliminary and subject to change.

AC Test Conditions

- ●Input voltage amplitude: 0.4 V to 2.8 V
- ●Clock input voltage amplitude: 0.4V to V_{CC}-0.6V
- ●Input signal rise and fall time: 10 ns
- ●Input judge level: 0.8 V and 2.0 V
- ●Output judge level: 0.8 V and 2.0 V
- ●Output load: ITTL + 100 pF (unless otherwise specified)

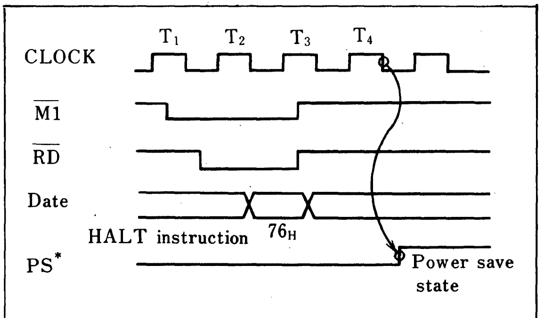
■ Power Save Function

The LH5082L series has the power save (PS) functions.

(1) PS mode setting

When the CPU (LH5080L series) has executed a HALT instruction, the LH5082L series reads this HALT instruction to autmatically go into the PS mode. In this mode, the internal clock signal is cut off. The external clock may be off during the PS mode.

About the external clock stop, the same is true as the power-saving CPU (LH5080L series)



* PS is an internal signal and not output externally.

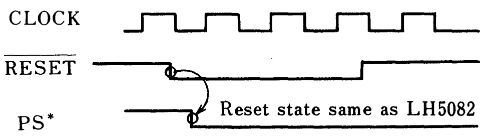
PS mode set timing

(2) PS mode clear

The PS mode is cleared by the fall of $\overline{M1}$ signal or the $\overline{RESET}$ signal.

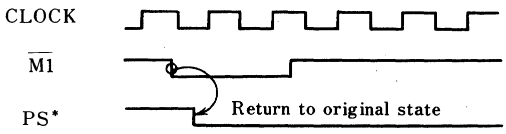
When the external clock is off the PS mode, however, a stable clock signal must be input before clearing the PS mode.

Once cleared from the PS mode, the power-saving CPU (LH5080L series) comes into the next fetch cycle. At the time when the first M1 signal during this cycle falls, the LH5082L series is also cleared from the PS mode.



* PS is an internal signal and not output externally.

PS mode clear by $\overline{RESET}$ signal



* PS is an internal signal and not output externally.

PS mode clear by $\overline{MI}$ signal

LH0080 Z80 CPU Central Processing Unit

Description

The LH0080 Z80 CPU (Z80 CPU for short below) is a general-purpose 8-bit microprocessor fabricated using an N-channel silicon-gate process.

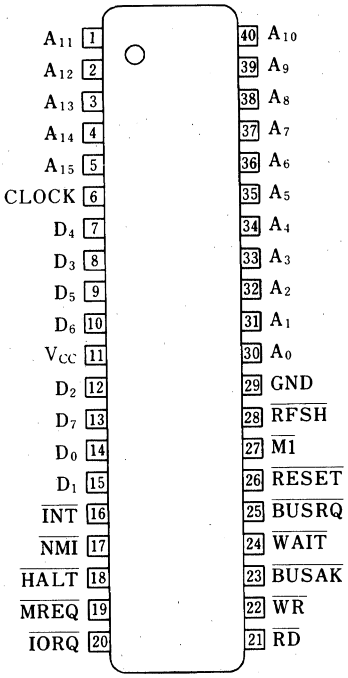
The LH0080A Z80A, LH0080B Z80B, LH0080E Z80E CPU are the high speed version which can operate at the 4MHz, 6MHz and 8MHz system clock, respectively.

Features

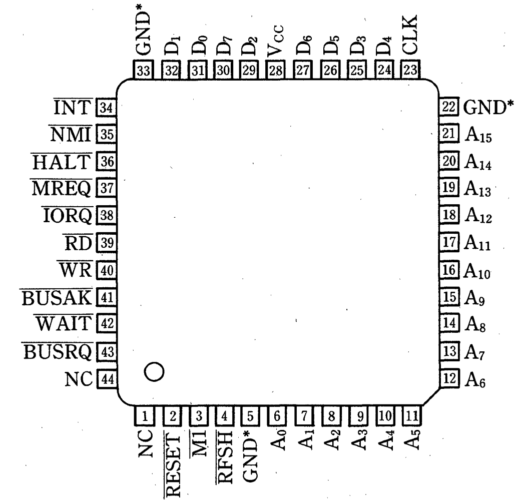
1. 8-bit parallel processing microprocessor
2. N-channel silicon-gate process
3. 158 instructions (The instruction of the 8080A are included as a subset ; 8080A software compatibility is maintained)
4. 22 registers
5. The capability of 3 modes maskable interrupt and non-maskable interrupt
6. On-chip dynamic memory refresh counter
7. Instruction fetch cycle : 1.6μs(Z80), 1.0μs (Z80A), 0.67 μs (Z80B), 0.5 μs (Z80E)
8. Single +5V power supply and single phase clock
9. All inputs and outputs fully TTL compatible
10. 40-pin DIP (DIP40-P-600)
    44-pin QFP (QFP44-P-1010A)
    44-pin QFJ (QFJ44-P-S650)

Pin Connections

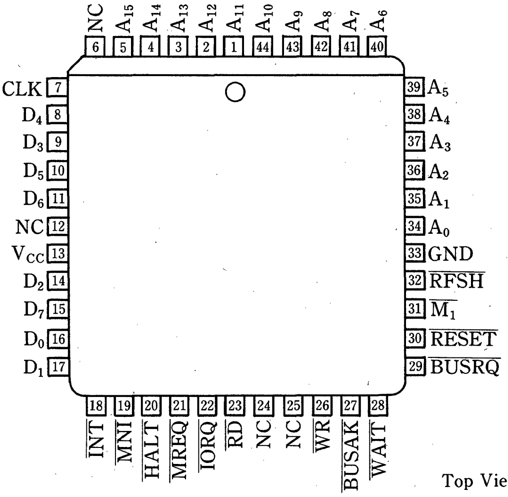
LH0080/LH0080A/LH0080B/LH0080E
LH0080H/LH0080AH



LH0080M/LH0080AM



LH0080U/LH0080AU/LH0080BU



Top View

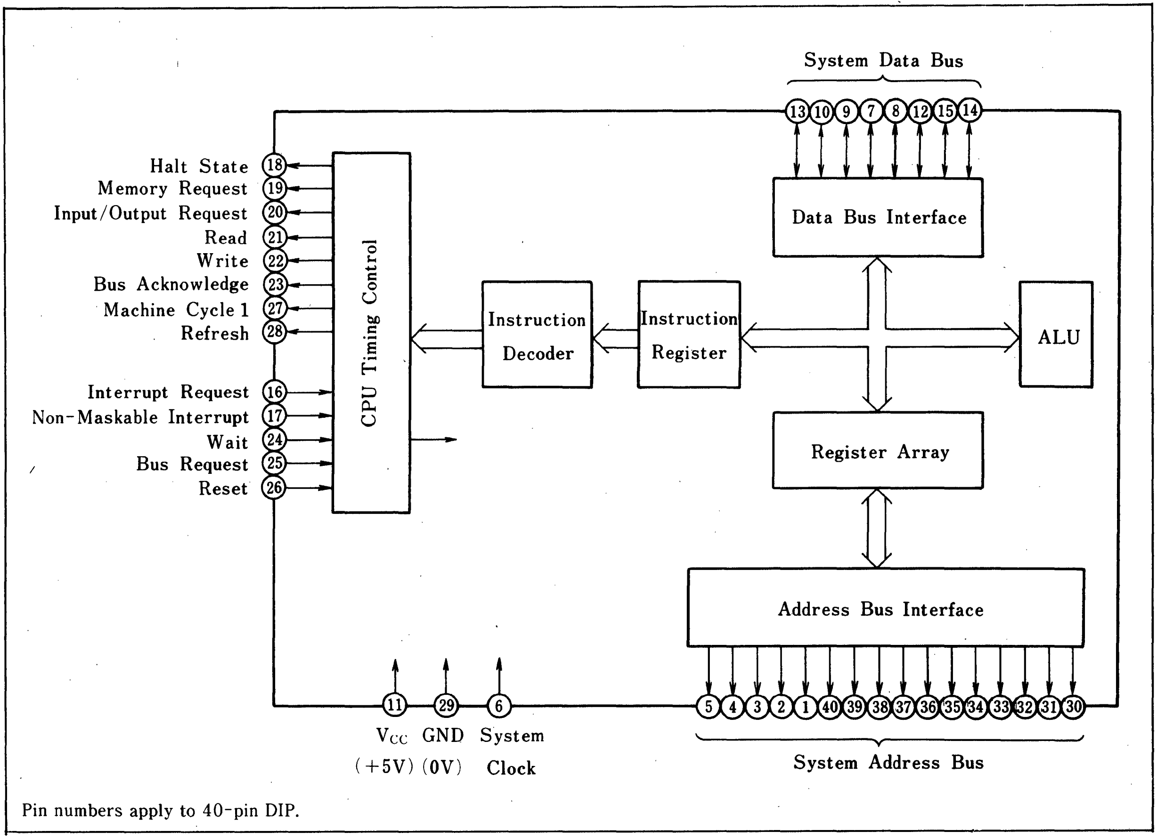
*The GND pins must be connected to the GND level.

Ordering Information

Product	Z80 CPU	Z80A CPU	Z80B CPU	Z80E CPU	Package	Operating temperature
Clock frequency	2.5MHz	4MHz	6MHz	8MHz		
Model No.	LH0080	LH0080A	LH0080B	LH0080E	40-pin DIP	0℃ to +70℃
	LH0080H*	LH0080AH*				−20℃ to +85℃
	LH0080M	LH0080AM			44-pin QFP	0℃ to +60℃
	LH0080U	LH0080AU	LH0080BU		44-pin QFJ	0℃ to +70℃

* H suffix is a wide temperature spec, packaged in 40-pin DIP.

Block Diagram



Pin numbers apply to 40-pin DIP.

Pin Description

Signal	Pin name	I/O	Function
A_0-A_{15}	Address bus	3-state O	System address bus
D_0-D_7	Data bus	Bidirectional 3-state	System data bus
$\overline{\mathrm{M1}}$	Machine cycle one	O	Active "Low". Indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.
$\overline{\mathrm{MREQ}}$	Memory request	3-state O	Active "Low". Indicates that the address bus holds a valid address for a memory read or memory write operation.
$\overline{\mathrm{IORQ}}$	I/O request	3-state O	Active "Low". Indicates that the lower 8 bits of the address bus holds a valid I/O address for an I/O read or write operation. Also generated concurrently with $\overline{\mathrm{M1}}$ during an interrupt acknowledge cycle to indicate an interrupt response.
$\overline{\mathrm{RD}}$	Memory read	3-state O	Active "Low". Indicates that the CPU wants to read data from memory or an I/O device.
$\overline{\mathrm{WR}}$	Memory write	3-state O	Active "Low". Indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.
$\overline{\mathrm{RFSH}}$	Refresh	O	Active "Low". Indicates that the lower 7 bits of the system address bus can be used as a refresh address to the system's dynamic memories. Together with $\overline{\mathrm{MREQ}}$ at "Low".
$\overline{\mathrm{HALT}}$	Halt state	O	Active "Low". Indicates that a Halt instruction is being executed. While halted, the CPU executes NOPs to maintain memory refresh. The Halt state is cleared with $\overline{\mathrm{RESET}}$, $\overline{\mathrm{NMI}}$, or $\overline{\mathrm{INT}}$ (when allowed).
$\overline{\mathrm{WAIT}}$	Wait	I	Active "Low". Indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a wait state as long as this signal is active.
$\overline{\mathrm{INT}}$	Maskable interrupt request	I	Active "Low". Generated by I/O devices. The CPU honors a request at the end of the current instruction if the interrupt enable flip-flop is enabled.
$\overline{\mathrm{NMI}}$	Non-maskable interrupt request	I	Active "Low". Has a higher priority than $\overline{\mathrm{INT}}$. Always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. Automatically forces the Z80 CPU to restart at location 0066H.
$\overline{\mathrm{RESET}}$	Reset	I	Active "Low". Resets the interrupt enable flip-flop, the program counter interrupt vector register and the memory refresh register, and sets the interrupt status to Mode O, in order to initialize the CPU.
$\overline{\mathrm{BUSRQ}}$	Bus request	I	Active "Low". Has a higher priority than NMI. Always recognized at the end of the current machine cycle. Activated to allow a bus master other than the CPU to control the system bus.
$\overline{\mathrm{BUSAK}}$	Bus acknowledge	O	Active "Low". Indicates to the requesting device that the external circuitry can control the system bus.
CLOCK	System clock	I	Inputs+5V single-phase clock.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	−0.3 to +7.0	V	
Output voltage	V_{OUT}	−0.3 to +7.0	V	
Operating temperature	Topr	0 to +70	℃	1
		0 to +60		2
		−20 to +85		3
Storage temperature	Tstg	−65 to +150	℃	

Note 1: 40-pin DIP and 44-pin QFJ
Note 2: 44-pin QFP
Note 3: 40-pin DIP with wide temperature spec.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

All ac parameters assume a load capacitance of 100 pF. Add 10 ns delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for address and control lines.

DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Clock input low voltage	V_{ILC}			−0.3		0.45	V
Clock input high voltage	V_{IHC}			$V_{CC}-0.6$		$V_{CC}+0.3$	V
Input low voltage	V_{IL}			−0.3		0.8	V
Input high voltage	V_{IH}			2.0		V_{CC}	V
Output low voltage	V_{OL}	$I_{OL}=1.8mA$				0.4	V
Output high voltage	V_{OH}	$I_{OH}=-250\mu A$		2.4			V
Current consumption	I_{CC}		LH0080			150	mA
			LH0080A			200	mA
			LH0080B			200	mA
			LH0080E			200	mA
Input leakage current	$\lvert I_{LI} \rvert$	$0 \leqq V_{IN} \leqq V_{CC}$				10	μA
3-state output leakage current in float	$\lvert I_{LEAK} \rvert$	$V_{OUT}=0.4V$ to V_{CC}				10	μA

Note 1: Ta=0 to +60℃ for 44-pin QFP
Ta=−20 to +85℃ for 40-pin DIP with wide temperature spec.

5

Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			35	pF
Input capacitance	C_{IN}				5	pF
Output capacitance	C_{OUT}				10	pF

AC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

No.	Parameter	Symbol	LH0080		LH0080A		LH0080B		LH0080E*		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400*		250*		165*		125*		ns
2	Clock pulse width (High)	TwCh	180*		110*		65*		55*		ns
3	Clock pulse width (Low)	TwCl	180	2000	110	2000	65	2000	55	2000	ns
4	Clock fall time	TfC		30		30		20		10	ns
5	Clock rise time	TrC		30		30		20		10	ns
6	Clock ↑ to address valid delay	TdCr (A)		145		110		90		80	ns
7	Addreess valid to $\overline{\text{MREQ}}$ ↓ delay	TdA (MREQf)	125*		65*		35*		20*		ns
8	Clock ↓ $\overline{\text{MREQ}}$ ↓ delay	TdCf (MREQf)		100		85		70		60	ns
9	Clock ↑ to $\overline{\text{MREQ}}$ ↑ delay	TdCr (MREQr)		100		85		70		60	ns
10	$\overline{\text{MREQ}}$ pulse width (High)	TwMREQh	170*		110*		65*		45*		ns
11	$\overline{\text{MREQ}}$ pulse width (Low)	TwMREQ1	360*		220*		135*		100*		ns
12	Clock ↓ to $\overline{\text{MREQ}}$ ↑ delay	TdCf (MREQr)		100		85		70		60	ns
13	Clock ↓ to $\overline{\text{RD}}$ ↓ delay	TdCf (RDf)		130		95		80		70	ns
14	Clock ↑ to $\overline{\text{RD}}$ ↑ delay	TdCr (RDr)		100		85		70		60	ns
15	Data setup time to clock ↑	TsD (Cr)	50		35		30		30		ns
16	Data hold time from $\overline{\text{RD}}$ ↑	ThD (RDr)	0		0		0		0		ns
17	$\overline{\text{WAIT}}$ setup time to clock ↓	TsWAIT (Cf)	70		70		60		50		ns
18	$\overline{\text{WAIT}}$ hold time after clock ↓	ThWAIT (Cf)	0		0		0		0		ns
19	Clock ↑ to $\overline{\text{M1}}$ ↓ delay	TdCr (M1f)		130		100		80		70	ns
20	Clock ↑ to $\overline{\text{M1}}$ ↑ delay	TdCr (Mlr)		130		100		80		70	ns
21	Clock ↑ to $\overline{\text{RFSH}}$ ↓ delay	TdCr (RFSHf)		180		130		110		95	ns
22	Clock ↑ to $\overline{\text{RFSH}}$ ↑ delay	TdCr (RFSHr)		150		120		100		85	ns
23	Clock ↓ to $\overline{\text{RD}}$ ↑ delay	TdCf (RDr)		110		85		70		60	ns
24	Clock ↑ to $\overline{\text{RD}}$ ↓ delay	TdCr (RDf)		100		85		70		60	ns
25	Data Setup to clock ↑ during M_2, M_3, M_4 or M_5 cycles	TsD (Cf)	60		50		40		30		ns
26	Address stable prior to $\overline{\text{IORQ}}$ ↓	TdA (IORQf)	320*		180*		110*		75*		ns
27	Clock ↑ $\overline{\text{IORQ}}$ ↓ delay	TdCr (IORQf)		90		75		65		55	ns
28	Clock ↓ to $\overline{\text{IORQ}}$ ↑ delay	TdCf (IORQr)		110		85		70		60	ns
29	Data stable prior to $\overline{\text{WR}}$ ↓	TdDm (WRf)	190*		80*		25*		5*		ns
30	Clock ↓ $\overline{\text{WR}}$ ↓ delay	TdCf (WRf)		90		80		70		60	ns
31	$\overline{\text{WR}}$ pulse width	TwWR	360*		220*		135*		100*		ns
32	Clock ↓ to $\overline{\text{WR}}$ ↑ delay	TdCr (WRr)		100		80		70		60	ns
33	Data stable prior to $\overline{\text{WR}}$ ↓	TdDi (WRf)	20*		−10*		−55*		−55*		ns
34	Clock ↑ to $\overline{\text{WR}}$ ↓ delay	TdCr (WRf)		80		65		60		55	ns
35	Data stable from $\overline{\text{WR}}$ ↑	TdWRr (D)	120*		60*		30*		15*		ns
36	Clock ↓ to $\overline{\text{HALT}}$ ↑ or ↓	TdCf (HALT)		300		300		260		225	ns
37	$\overline{\text{NMI}}$ pulse width	TwNMI	80		80		70		80		ns
38	$\overline{\text{BUSREQ}}$ setup time to clock ↑	TsBUSRQ (Cr)	80		50		50		40		ns
39	$\overline{\text{BUSREQ}}$ hold time after clock ↑	ThBUSRQ (Cr)	0		0		0		0		ns
40	Clock ↑ to $\overline{\text{BUSACK}}$ ↓ delay	TdCr (BUSAKf)		120		100		90		80	ns
41	Clock ↓ to $\overline{\text{BUSACK}}$ ↑ delay	TdCf (BUSAKr)		110		100		90		80	ns
42	Clock ↑ to data float delay	TdCr (Dz)		90		90		80		70	ns
43	Clock ↑ to control output float delay ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$)	TdCr (CTz)		110		80		70		60	ns
44	Clock ↑ to address float delay	TdCr (Az)		110		90		80		70	ns
45	$\overline{\text{MREQ}}$ ↑, $\overline{\text{IORQ}}$ ↑, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ ↑ to address hold time	TdCTr (A)	160*		80*		35*		20*		ns

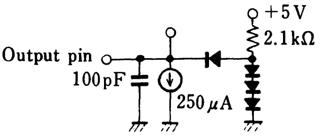
↑ Rising edge, ↓ Falling edge

Note 1: Ta=0 to +60℃ for 44-pin QFP.
Ta=−20 to +85℃ for 40-pin DIP with wide temperature spec.

No.	Parameter	Symbol	LH0080		LH0080A		LH0080B		LH0080E*		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
46	$\overline{\text{RESET}}$ ↓ to clock ↑ setup time	TsRESET (Cr)	90		60		60		45		ns
47	$\overline{\text{RESET}}$ from clock! ↑ hold time	ThRESET (Cr)	0		0		0		0		ns
48	$\overline{\text{INT}}$ to clock ↑ setup time	TsINTf (Cr)	80		80		70		55		ns
49	$\overline{\text{INT}}$ from clock ↑ hold time	ThINTr (Cr)	0		0		0		0		ns
50	$\overline{\text{M1}}$ ↓ to $\overline{\text{IORQ}}$ ↓ delay	TdM1f (IORQf)	920*		565*		365*		270*		ns
51	Clockk ↓ to $\overline{\text{IORQ}}$ ↓ delay	TdCf (IORQf)		110		85		70		60	ns
52	Clock ↑ to $\overline{\text{IORQ}}$ ↑ delay	TdCf (IORQr)		100		85		70		60	ns
53	Clock ↓ to data valid delay	TdCf (D)		230		150		130		115	ns

All ac parameters assume a load capacitance of 100 pF. Add 10 μs delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for address and control lines.

*For clock periods other than the minimums shown in the table, calculate parameters using the following expressions.



Footnotes to AC Characteristics

No.	Symbol	LH0080	LH0080A	LH0080B	LH0080E
1	TcC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC
2	TwCh	MAX. 200 μs	MAX. 200 μs	MAX. 200 μs	MAX. 200 μs
7	TdA (MREQf)	TwCh+TfC−75	TwCh+TfC−65	TwCh+TfC−50	TwCh+TfC−45
10	TwMREQh	TwCh+TfC−30	TwCh+TfC−20	TwCh+TfC−20	TwCh+TfC−20
11	TwMREQ1	TcC−40	TcC−30	TcC−30	TcC−25
26	TdA (IORQf)	TcC−80	TcC−70	TcC−55	TcC−50
29	TdD (WRf)	TcC−210	TcC−170	TcC−140	TcC−120
31	TwWR	TcC−40	TcC−30	TcC−30	TcC−25
33	TdD (WRf)	TwCl+TrC−180	TwCl+TrC−140	TwCl+TrC−140	TwCl+TrC−120
35	TdWRr (D)	TwCl+TrC−80	TwCl+TrC−70	TwCl+TrC−55	TwCl+TrC−50
45	TdCTr (A)	TwCl+TrC−40	TwCl+TrC−50	TwCl+TrC−50	TwCl+TrC−45
50	TdMlf (IORQf)	2Tch+TwCh+TfC−80	2TcC+TwCh+TfC−65	2TcC+TwCh+TfC−50	2TcC+TwCh+TfC−45

AC Test Conditions :

V_{IH}=2.0V V_{IHC}=V_{CC}−0.6V V_{OH}=2.0V FLOAT=±0.5

V_{IL}=0.8V V_{ILC}=0.45V V_{OL}=0.8V

■ CPU Timing

The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

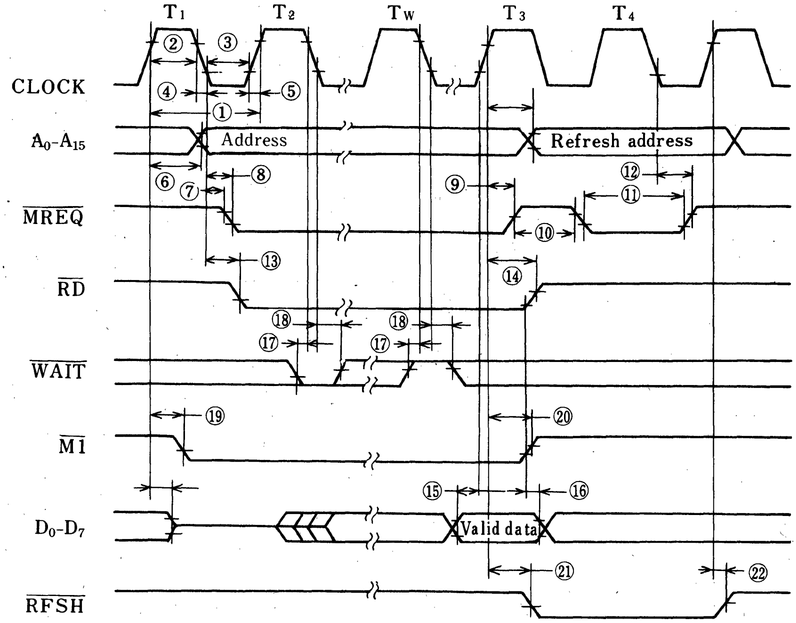
- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

(1) Instruction Opcode Fetch

The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Fig. 1). Approximately one-half clock cycle later, $\overline{\text{MREQ}}$ goes active. When active, $\overline{\text{RD}}$ indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the $\overline{\text{WAIT}}$ input with the falling edge of clock state T_2. During clock states T_3 and T_4 of an $\overline{\text{M1}}$ cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.



Note: T_W-Wait cycle added when necessary for slow ancilliary devices.

Fig. 1 Instruction opcode fetch

(2) Memory Read or Write Cycles

Fig. 2 shows the timing of memory read or write cycles other than an opcode fetch ($\overline{\text{M1}}$) cycle.
The $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$ signals function exactly as in the fetch cycle. In a memory write cycle, MREQ also becomes active when the address bus is stable. The $\overline{\text{WR}}$ line is active when the data bus is stable, so that it can be used directly as an R/$\overline{\text{W}}$ pulse to most semiconductor memories.

(3) Input or Output Cycles

Fig. 3 shows the timing for an I/O read or I/O write operation.
During I/O operations, the CPU automatically inserts a single wait state (T_W). This extra wait state allows sufficient time for an I/O port to decode the address from the port address lines.

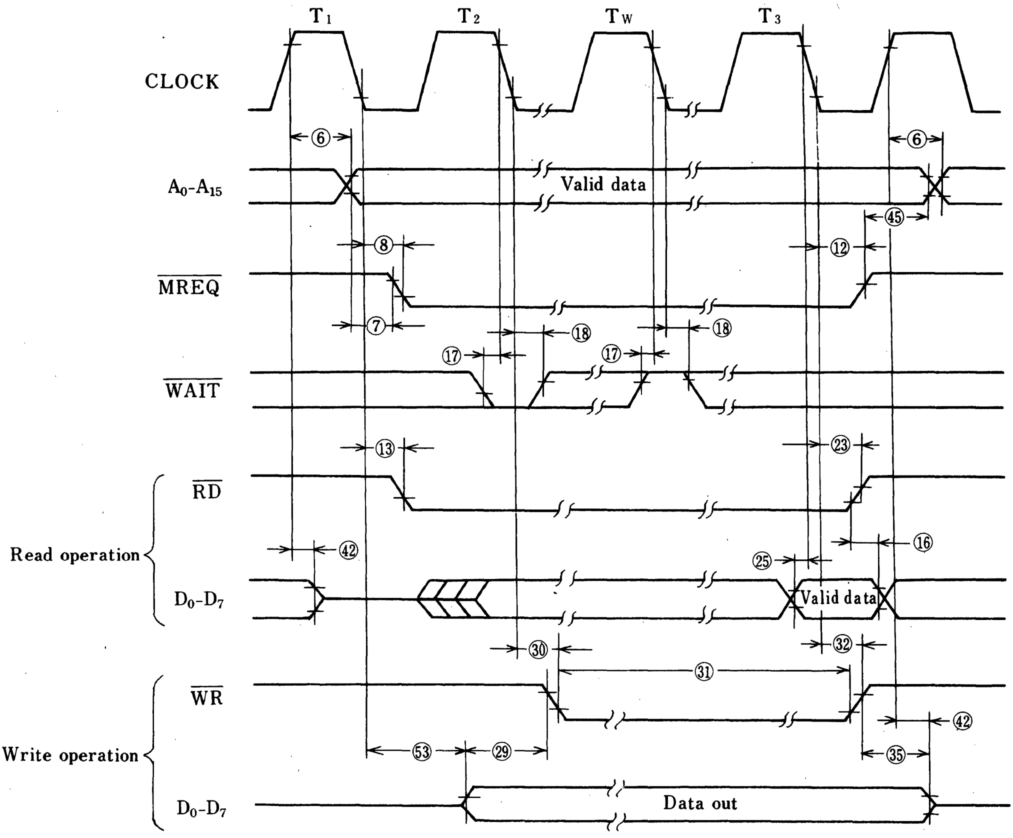
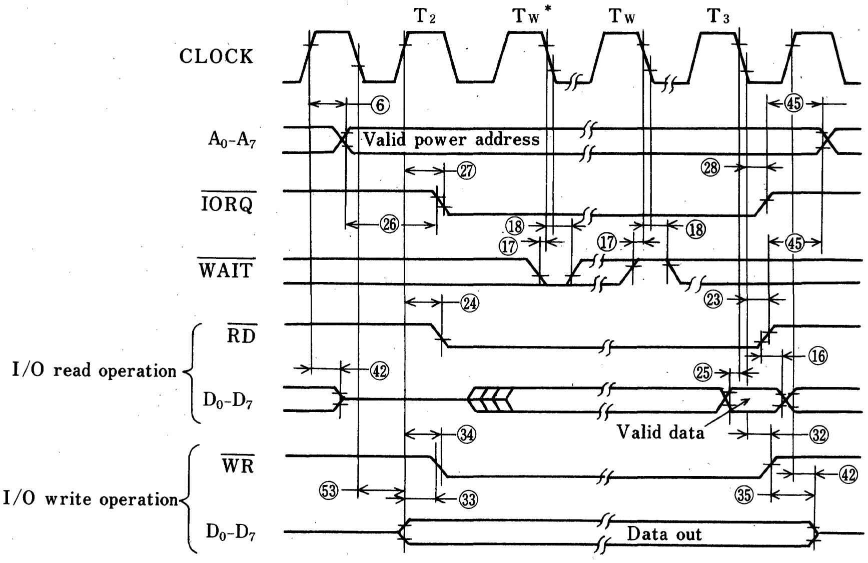


Fig. 2 Memory read or write cycles

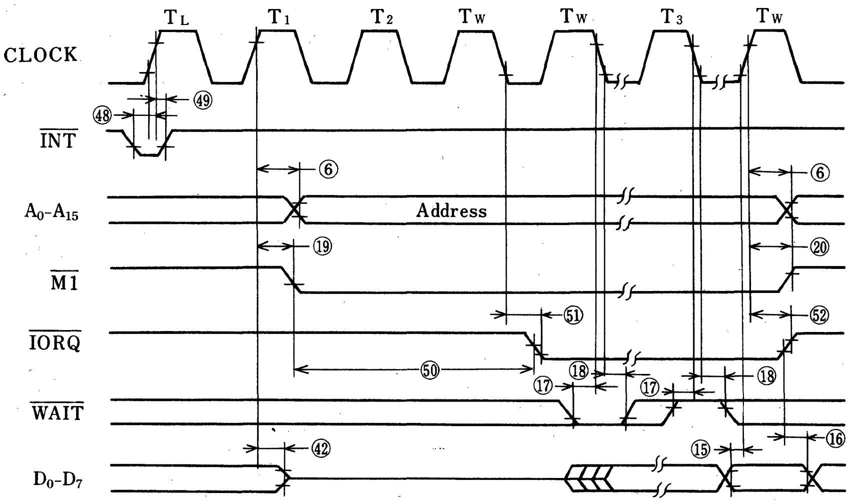


Note : Tw=One wait cycle automatically inserted by CPU.

Fig. 3 Input or output

(4) Interrupt request/acknowledge cycle

The CPU samples the interrupt signal with the rising edge of the last clock at the end of any instruction (Fig. 4). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, IORQ becomes active (instead of MREQ) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two wait states to this cycle.



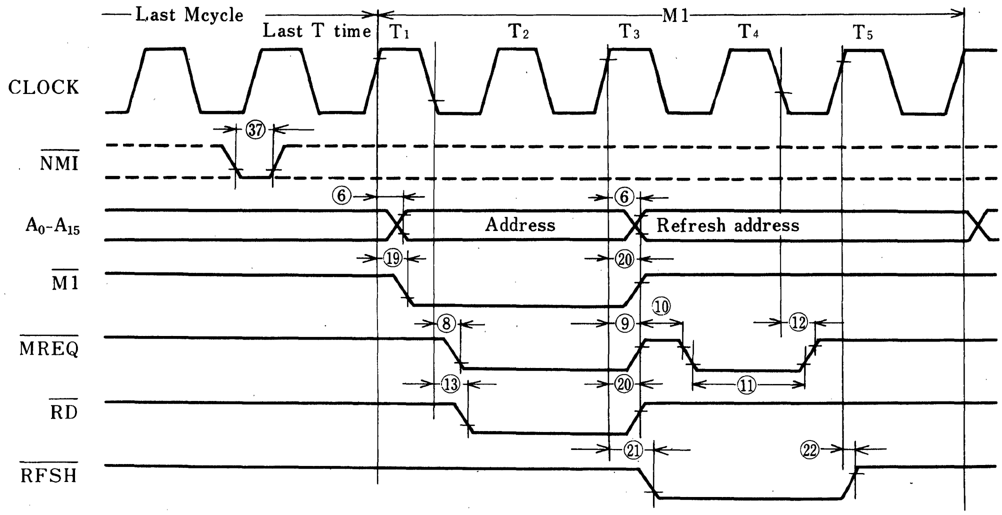
Note 1 : TL=Last state of previous instruction.
Note 2 : Two wait cycles automatically inserted by CPU (*).

Fig. 4 Interrupt request/acknowledge cycle

(5) Non-maskable interrupt request cycle

NMI is sampled at the same time as the maskable interrupt INT but hashigher priority and cannot be disabled under software control.

The subsequent timing is similar to that of a normal instruction fetch except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the NMI service routine located at address 0066H (Fig. 5).



*Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge must occur no later than rising edge of the clock cycle preceding T_{LAST}.

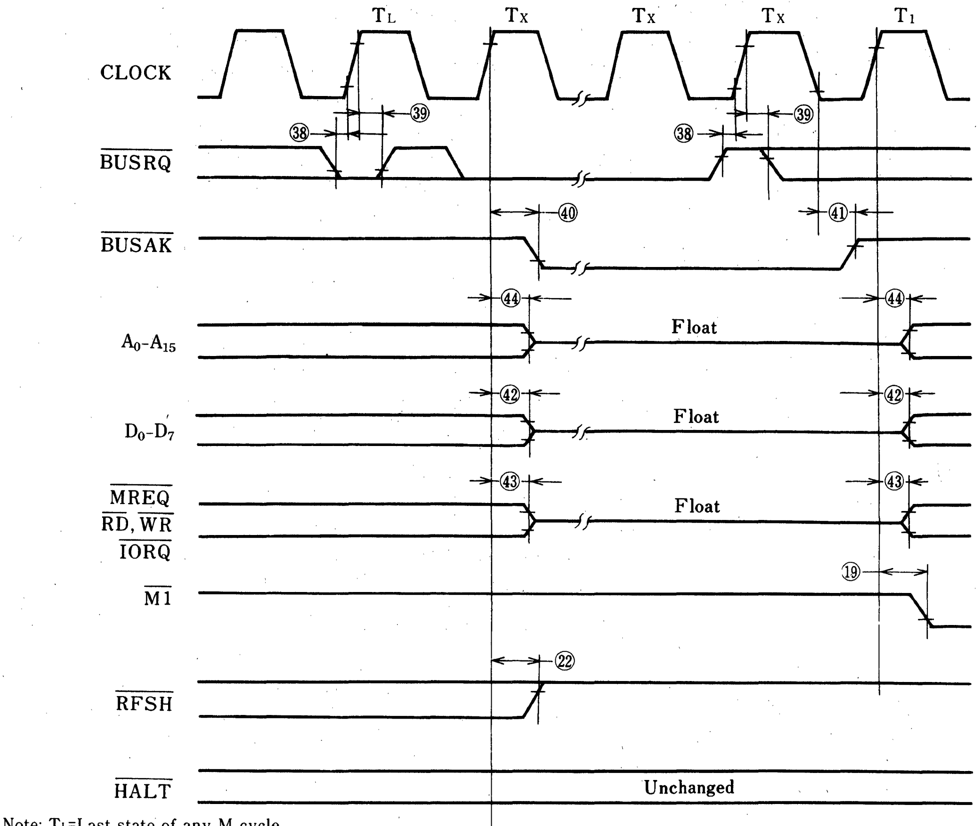
Fig. 5 Non-maskable interrupt request operation

(6) Bus request/acknowledge cycle

The CPU samples BUSREQ with the rising edge of the last clock period of any machine cycle (Fig. 6). If BUSREQ is active, the CPU sets its address, data, and MREQ, IORQ, RD, and WR lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.

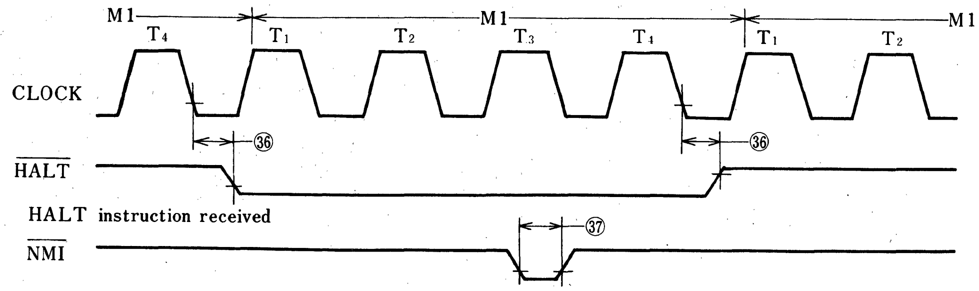
(7) Reset cycle

RESET must be active for at least three clock cycles for the CPU to properly accept it. As long as RESET remains active, the address and data buses float, and the control outputs are inactive. Once RESET goes inactive, three internal T cycles are consumed before the CPU resumes normal processing operation. RESET clears the PC register, so the first opcode fetch will be location 0000 (Fig. 8).



Note: T_L=Last state of any M cycle.
T_X=An arbitrary clock cycle used by requesting device.

Fig. 6 Z-bus request/acknowledge cycle



Note: $\overline{INT}$ will also force a Halt exit.

Fig. 7 Halt acknowledge cycle

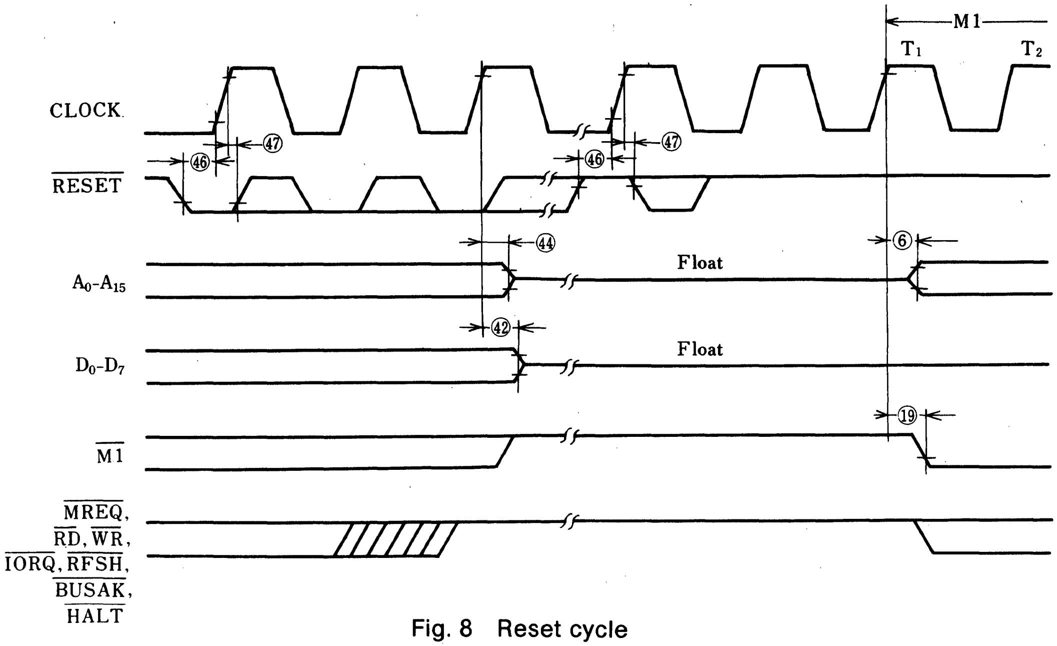


Fig. 8 Reset cycle

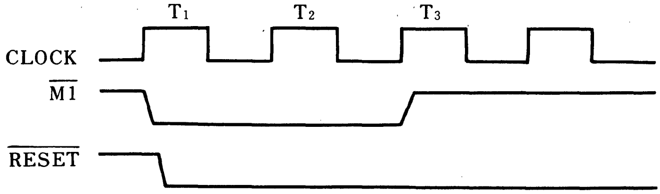


Fig. 9 Timing diagram when M1 cycle has no wait state

⟨Reference⟩

The RAM contents may be adversely affected by resetting the CPU while it is in operation.

To prevent this, a $\overline{\text{RESET}}$ signal should be input in the following timings.

(1) No walt state in the M1 cycle

Input a $\overline{\text{RESET}}$ signal to start sampling this signal at the clock rising in the M1 cycle's T_2 state. (See Fig. 9.)

(2) A walt state in the M1 cycle

Input a $\overline{\text{RESET}}$ signal to start sampling this signal at the clock rising in the M1 cycle's T_3 state. (See Fig. 10.)

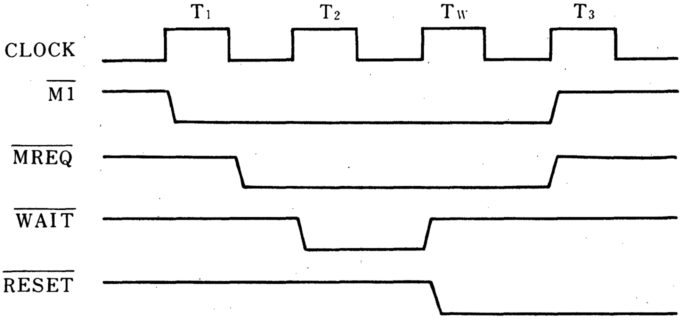


Fig. 10 Reset circuit and timing diagram when M1 cycle has a wait state

CPU Registers

A Accumulator	F Flag Register	A′ Accumulator	F′ Flag Register
B General Purpose	C General Purpose	B′ General Purpose	C′ General Purpose
D General Purpose	E General Purpose	D′ General Purpose	E′ General Purpose
H General Purpose	L General Purpose	H′ General Purpose	L′ General Purpose

← 8 bits →

I Interrupt Vector	R Memory Refresh
IX Index Register	
IY Index Register	
SP Stack pointer	
PC Program Counter	

← 16 bits →

Architecture

(1) CPU Registers

(i) **Program Counter (PC)** The program counter holds the 16 bits memory address of a current instruction. The CPU fetches the contents from memory address specified by the PC.

The PC feeds the data to the address line, automatically setting the PC value to +1. When a program jump takes place, a new value is directly set to the PC.

(ii) **Stack Pointer (SP)** The stack pointer holds the top 16-bit address of the stack with an external RAM. An external file is based on LIFO (Last-In, First-Out).

The data are transferred between a CPU-specified register and the stack by a PUSH or POP instruction. The last-pushed data are first popped from the stack.

(iii) **Index Register (IX & IY)** For index mode addressing, there are independent index registers IX and IY, each of which holds 16-bit reference address.

In the index mode, the index registers are used to designate the memory area for data input/output.

With an INDEX ADDRESSING instruction, an effective address comes by adding a one-byte displacement to the register content. This displacement is an integral signed two's complement number

(iv) **Interrupt Register (I)** The Z80 CPU has indirect subroutine call mode for any memory area according to an interrupt. For this purpose, this register stores the upper 8 bits of memory address for vectored interrupt processing and the lower 8 bits for the interrupting device.

(v) **Refresh Register (R)** The built-in refresh register provides user-transparent dynamic memory refresh. Its lower 7 bits are automatically incremented during each instruction fetch cycle.

While the CPU records a fetched instruction and executes the instruction, the refresh register data are placed on the address bus by a REFRESH control signal.

(vi) **Accumulator and Flag Register (A & F)** The CPU has also two independent 8-bit accumulators in combination with two 8-bit flag registers.

The accumulators store an operand or the results of an 8-bit operation. The flag registers, on the other hand, deal with the results of an 8-bit or 16-bit operation; for example, seeing if the result is equal to 0 or not.

(vii) **General-Purpose Registers** There are several pairs of general-purpose registers. In each pair, they can be used separately or as a 16-bit paired register. The paired registers are BC, DE, HL, as well as BC′ DE′ HL′. Either of these sets can work by an "Exchange" instruction at any time on a program.

(2) Arithmetic/Logical Unit (ALU)

An 8-bit arithmetic/logical operation instruction is executed by the ALU inside the CPU. The ALU connects to each register through the internal bus for data transfer between them.

(3) Instruction Register, CPU Control

Each instruction is read out of the memory, held in the instruction register, and decoded. The control unit controls this action and gives control signals necessary to read and write data from and to the registers.

The control unit also makes ALU control signal and other external control signals.

〈Interrupts : General Operation〉 The Z80 CPU accepts two interrupt input signals: $\overline{\text{NMI}}$ and $\overline{\text{INT}}$. The $\overline{\text{NMI}}$ is a non-maskable interrupt and has the highest priority. $\overline{\text{INT}}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate.

(1) Non-Maskable Interrupt ($\overline{\text{NMI}}$)

The non-maskable interrupt will be accepted at all times by the CPU.

After recognition of the $\overline{\text{NMI}}$ signal, the CPU jumps to restart location 0066H.

(2) Maskable Interrupt ($\overline{\text{INT}}$)

The maskable interrupt, $\overline{\text{INT}}$, has three programmable response modes available.

(i) **Mode 0 Interrupt Operation.** This mode is similar to the 8080A microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is a Restart instruction or a Call instruction.

(ii) **Mode 1 Interrupt Operation.** Mode 1 operation is very similar to that for the $\overline{\text{NMI}}$. The principal difference is that the Mode 1 interrupt has a restart location of 0038H only.

(iii) **Mode 2 Interrupt Operation.** This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address (16 bits) of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8-bits and the contents of the I register as the upper 8-bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address.

All the Z80 peripheral devices have the interrupt priority circuit with a daisy-chain configuration. During an interrupt acknowledge cycle, vectors are automatically fed. For more details, refer to the Z80 PIO description.

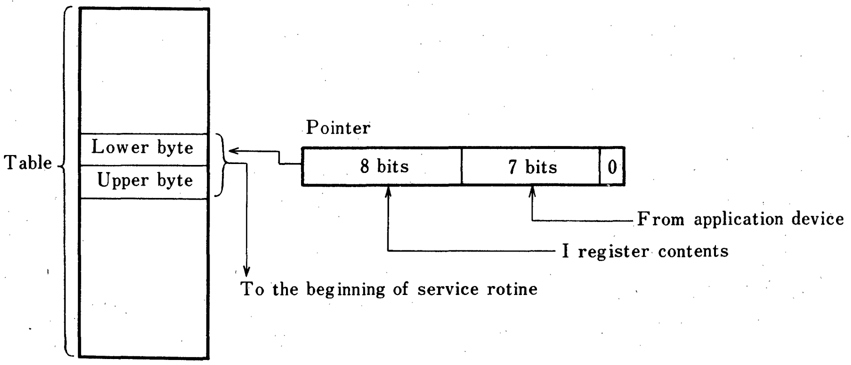


Fig. 1 Mode 2 interrupt diagram

Instruction Set

Table 1 8-bit load group

Mnemonic	Symbolic	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
	operation	76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
LD r, r′	r←r′	01 r r′	40+	●	●	●	●	●	●	1	1	4	
LD r, n	r←n	00 r 110 ← n →	06+	●	●	●	●	●	●	2	2	7	
LD r, (HL)	r← (HL)	01 r 110	46+	●	●	●	●	●	●	1	2	7	
LD r, (IX+d)	r← (IX+d)	11 011 101 01 r 110 ← d →	DD 46+	●	●	●	●	●	●	3	5	19	
LD r, (IY+d)	r← (IY+d)	11 111 101 01 r 110 ← d →	FD 46	●	●	●	●	●	●	3	5	19	
LD (HL), r	(HL)←r	01 110 r	70+	●	●	●	●	●	●	1	2	7	
LD (IX+d), r	(IX+d)←r	11 011 101 01 110 r ← d →	DD 70+	●	●	●	●	●	●	3	5	19	
LD (IY+d), r	(IY+d)←r	11 111 101 01 110 r ← d →	FD 70+	●	●	●	●	●	●	3	5	19	
LD (HL), n	(HL)←n	00 110 110 ← n →	36	●	●	●	●	●	●	2	3	10	
LD (IX+d), n	(IX+d)←n	11 011 101 00 110 110 ← d → ← n →	DD 36	●	●	●	●	●	●	4	5	19	
LD (IY+d), n	(IY+d)←n	11 111 101 00 110 110 ← d → ← n →	FD 36	●	●	●	●	●	●	4	5	19	
LD A, (BC)	A ← (BC)	00 001 010	0A	●	●	●	●	●	●	1	2	7	
LD A, (DE)	A ← (DE)	00 011 010	1A	●	●	●	●	●	●	1	2	7	
LD A, (nn)	A ← (nn)	00 111 010 ← n → ← n →	3A	●	●	●	●	●	●	3	4	13	
LD (BC), A	(BC) ← A	00 000 010	02	●	●	●	●	●	●	1	2	7	
LD (DE), A	(DE) ← A	00 010 010	12	●	●	●	●	●	●	1	2	7	
LD (nn), A	(nn) ← A	00 110 010 ← n → ← n →	32	●	●	●	●	●	●	3	4	13	
LD A, I	A ← I	11 101 101 01 010 111	ED 57	●	↕	IFF	↕	0	0	2	2	9	
LD A, R	A ← R	11 101 101 01 011 111	ED 5F	●	↕	IFF	↕	0	0	2	2	9	
LD I, A	I ← A	11 101 101 01 000 111	ED 47	●	●	●	●	●	●	2	2	9	
LD R, A	R ← A	11 101 101 01 001 111	ED 4F	●	●	●	●	●	●	2	2	9	

r,r	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

Notes : r, r′ means any of the registers A, B, C, D, E, H, L, IFF the content of the interrupt enable flip-flop, (IFF) is copied into the P/V flag.
Flags : C (carry), Z (zero), S (sign), P/V (parity/overflow), H (half carry), N (add/substract).
: ● =unchanged, 0=reset, 1=set, X=undefined.
: ↕ set or reset according to the result of the operation.

Table 2 16-bit load group

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
LD dd, nn	dd ← nn	00 dd0 001 ← n → ← n →	01+	●	●	●	●	●	●	3	3	10	dd Reg. 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	11 011 101 00 100 001 ← n → ← n →	DD 21	●	●	●	●	●	●	4	4	14	
LD IY, nn	IY ← nn	11 111 101 00 100 001 ← n → ← n →	FD 21	●	●	●	●	●	●	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	00 101 010 ← n → ← n →	2A	●	●	●	●	●	●	3	5	16	nn : 2-byte number. Lower byte just after opcode.
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	11 101 101 01 dd1 011 ← n → ← n →	ED 4B+	●	●	●	●	●	●	4	6	20	Upper byte comes next.
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	11 011 101 00 101 010 ← n → ← n →	DD 2A	●	●	●	●	●	●	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	11 111 101 00 101 010 ← n →	FD 2A	●	●	●	●	●	●	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	00 100 010 ← n → ← n →	22	●	●	●	●	●	●	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	11 101 101 01 dd0 011 ← n → ← n →	ED 43+	●	●	●	●	●	●	4	6	20	
LD (nn), IX	(nn+1)←IX_H (nn) ← IX_L	11 011 101 00 100 010 ← n → ← n →	DD 22	●	●	●	●	●	●	4	6	20	
LD (nn), IY	(nn+1)←IY_H (nn) ← IY_L	11 111 101 00 100 010 ← n → ← n →	FD 22	●	●	●	●	●	●	4	6	20	
LD SP, HL	SP ← HL	11 111 001	F9	●	●	●	●	●	●	1	1	6	
LD SP, IX	SP ← IX	11 011 101 11 111 001	DD F9	●	●	●	●	●	●	2	2	10	
LD SP, IY	SP ← IY	11 111 101 11 111 001	FD F9	●	●	●	●	●	●	2	2	10	
PUSH qq	(SP−2)←qq_L (SP−1)←qq_H	11 qq0 101	C5+	●	●	●	●	●	●	1	3	11	
PUSH IX	(SP−2)←IX_L (SP−1)←IX_H	11 011 101 11 100 101	DD E5	●	●	●	●	●	●	2	4	15	qq Reg. 00 BC 01 DE 10 HL 11 AF
PUSH IY	(SP−2)←IY_L (SP−1)←IY_H	11 111 101 11 100 101	FD E5	●	●	●	●	●	●	2	4	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP)	11 qq0 001	C1+	●	●	●	●	●	●	1	3	10	
POP IX	IX_H←(SP+1) IX_L ← (SP)	11 011 101 11 100 001	DD E1	●	●	●	●	●	●	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	11 111 101 11 100 001	FD E1	●	●	●	●	●	●	2	4	14	

Notes : dd is any of the register pairs BC, DE, HL, SP.
qq is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively, e.g., BC_L=C, AF_H=A.

Flags : ● =unchanged, 0=reset, 1=set, X=undefined, ↕ =set or reset according to the result of the operation

Table 3 Exchange, block transfer, block search groups

Mnemonic	Symbolic	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
	operation	76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
EX DE, HL	DE ↔ HL	11 101 011	EB	●	●	●	●	●	●	1	1	4	
EX AF, AF'	AF ↔ AF'	00 001 000	08	●	●	●	●	●	●	1	1	4	
EXX	BC DE HL ↔ BC' DE' HL'	11 011 001	D9	●	●	●	●	●	●	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	11 100 011	E3	●	●	●	●	●	●	1	5	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	11 011 101 11 100 011	DD E3	●	●	●	●	●	●	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	11 111 101 11 100 011	FD E3	●	●	●	●	●	●	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1	11 101 101 10 100 000	ED A0	●	●	↕ ①	●	0	0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1	11 101 101 10 110 000	ED B0	●	●	0	●	0	0	2	5	21	If BC≠0
	If BC=0 end									2	4	16	If BC=0
LDD	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1	11 101 101 10 101 000	ED A8	●	●	↕ ①	●	0	0	2	4	16	
LDDR	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1	11 101 101 10 111 000	ED B8	●	●	0	●	0	0	2	5	21	If BC≠0
	If BC=0 end									2	4	16	If BC=0
CPI	A− (HL) HL ← HL+1 BC ← BC−1	11 101 101 10 100 001	ED A1	●	↕ ②	↕ ①	↕	1	↕	2	4	16	
CPIR	A− (HL) HL ← HL+1 BC ← BC−1	11 101 101 10 110 001	ED B1	●	↕ ②	↕ ①	↕	1	↕	2	5	21	If BC≠0 and A≠(HL)
	If A= (HL) or BC=0 end									2	4	16	If BC=0 or A= (HL)
CPD	A− (HL) HL ← HL−1 BC ← BC−1	11 101 101 10 101 001	ED A9	●	↕ ②	↕ ①	↕	1	↕	2	4	16	
CPDR	A− (HL) HL ← HL−1 BC ← BC−1	11 101 101 10 111 001	ED B9	●	↕ ②	↕ ①	↕	1	↕	2	5	21	If BC≠0 and A≠(HL)
	If A= (HL) or BC=0 end									2	4	16	If BC=0 or A= (HL)

Note: ①P/V flag is 0 if the result of BC=0, otherwise P/V=1
②Z flag is 1 if A= (HL), otherwise Z=0

Flags : ● = unchanged
0 = set, 1 = reset
↕ = set or reset according to the result of the operation

Table 4 8-bit arithmetic and logical group

Mnemonic	Symbolic	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
	operation	76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
ADD A, r	A ← A+r	10 k r	80+	↕	↕	V	↕	0	↕	1	1	4	
ADD A, n	A ← A+n	11 k 110 ← n →	C6+	↕	↕	V	↕	0	↕	2	2	7	
ADD A, (HL)	A ← A+ (HL)	10 k 110	86+	↕	↕	V	↕	0	↕	1	2	7	
ADD A, (IX+d)	A ← A+ (IX+d)	11 011 101 10 k 110 ← d →	DD 86+	↕	↕	V	↕	0	↕	3	5	19	
ADD A, (IY+d)	A ← A+ (IY+d)	11 111 101 10 k 110 ← d →	FD 86+	↕	↕	V	↕	0	↕	3	5	19	
ADC A, s	A ← A+s+C	4 types available based on the above ADD instruction (see Comments)		↕	↕	V	↕	0	↕	1※1 2 1 3	1※1 2 2 5	4※1 7 7 19	
SUB s	A ← A−s			↕	↕	V	↕	1	↕				
SBC A, s	A ← A−s−C			↕	↕	V	↕	1	↕				
AND s	A ← A∧s			0	↕	P	↕	0	1				
OR s	A ← A∨s			0	↕	P	↕	0	0				
XOR s	A ← A⊕s			0	↕	P	↕	0	0				
CP s	A−s			↕	↕	V	↕	1	↕				
INC r	r ← r+1	00 r ℓ	00+	●	↕	V	↕	0	↕	1	1	4	
INC (HL)	(HL) ← (HL)+1	00 110 ℓ	30+	●	↕	V	↕	0	↕	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	11 011 101 00 110 ℓ ← d →	DD 30+	●	↕	V	↕	0	↕	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	11 111 101 00 110 ℓ ← d →	FD 30+	●	↕	V	↕	0	↕	3	6	23	
DEC m	m ← m−1	4 types available based on the above INC instruction		●	↕	V	↕	1	↕	1※2 1 3 3	1※2 3 6 6	4※2 11 23 23	

r	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

Mnemonic	k
ADD	000
ADC	001
SUB	010
SBC	011
AND	100
OR	110
XOR	101
CP	111

S=r, n, (HL), (IX+d), (IY+d)

Mnemonic	ℓ
INC	100
DEC	101

m=r, (HL), (IX+d), (IY+d)

Note : V and P mean overflow and parity, respectively.
Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

※1: depends on s.
※2: depends on m.

Table 5 General purpose arithmetic and CPU control groups

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
DAA	Decimal adjustment (add/subtract)	00 100 111	27	↕	↕	P	↕	●	↕	1	1	4	Decimal adjust accumulator.
CPL	A ← $\overline{A}$	00 101 111	2F	●	●	●	●	1	1	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0−A	11 101 101 01 000 100	ED 44	↕	↕	V	↕	1	↕	2	2	8	Negate acc. (two's complement).
CCF	C ← $\overline{C}$	00 111 111	3F	↕	●	●	●	0	X	1	1	4	Complement carry flag.
SCF	C ← 1	00 110 111	37	1	●	●	●	0	0	1	1	4	Set carry flag.
NOP	No operation	00 000 000	00	●	●	●	●	●	●	1	1	4	
HALT	CPU halted	01 110 110	76	●	●	●	●	●	●	1	1	4	
DI	IFF ← 0	11 110 011	F3	●	●	●	●	●	●	1	1	4	Interrupt not enable
EI	IFF ← 1	11 111 011	FB	●	●	●	●	●	●	1	1	4	Interrupt enable
IM 0	Set interrupt mode 0	11 101 101 01 000 110	ED 46	●	●	●	●	●	●	2	2	8	Set interrupt mode.
IM 1	Set interrupt mode 1	11 101 101 01 010 110	ED 56	●	●	●	●	●	●	2	2	8	
IM 2	Set interrupt mode 2	11 101 101 01 011 110	ED 5E	●	●	●	●	●	●	2	2	8	

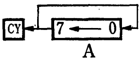
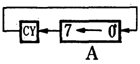
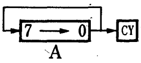
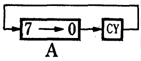
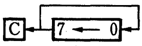
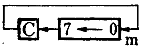
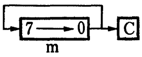
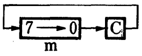
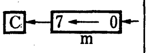
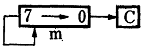
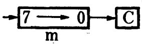
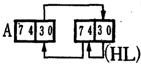
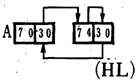
Note : IFF indicates the interrupt enable flip-flop, CY indicates the carry flip-flop.
Flags : ● =unchanged, 0=reset, 1=set, X=undefined, ↕ =set or reset according to the result of the operation

Table 6 16-bit arithmetic group

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
ADD HL, ss	HL ← HL +ss	00 ss1 001	09+	↕	●	●	●	0	X	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL +ss+C	11 101 101 01 ssl 010	ED 4A+	↕	↕	V	↕	0	X	2	4	15	
SBC HL, ss	HL ← HL −ss−C	11 101 101 01 ss0 010	ED 42+	↕	↕	V	↕	1	X	2	4	15	
ADD IX, pp	IX ← IX+pp	11 011 101 00 pp1 001	DD 09+	↕	●	●	●	0	X	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY+rr	11 111 101 00 rr1 001	FD 09+	↕	●	●	●	0	X	2	4	15	
INC ss	ss ← ss+1	00 ss0 011	03+	●	●	●	●	●	●	1	1	6	
INC IX	IX ← IX+1	11 011 101 00 100 011	DD 23	●	●	●	●	●	●	2	2	10	
INC IY	IY ← IY+1	11 111 101 00 100 011	FD 23	●	●	●	●	●	●	2	2	10	rr Reg. 00 BC 01 DE 10 IY 11 SP
DEC ss	ss ← ss−1	00 ss1 011	0B+	●	●	●	●	●	●	1	1	6	
DEC IX	IX ← IX−1	11 011 101 00 101 011	DD 2B	●	●	●	●	●	●	2	2	10	
DEC IY	IY ← IY−1	11 111 101 00 101 011	FD 2B	●	●	●	●	●	●	2	2	10	

Note : ss is any of the register pairs BC, DE, HL, SP.
pp is any of the register pairs BC, DE, IX, SP.
rr is any of the register pairs BC, DE, IY, SP.
Flags : ● =unchanged, 0=reset, 1=set, X=undefinede, ↕ =set or reset according to the result of the operation

Table 7 Rotate and shift groups

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
RLCA		00 000 111	07	↕	●	●	●	0	0	1	1	4	Rotate left circular accumulator.
RLA		00 010 111	17	↕	●	●	●	0	0	1	1	4	Rotate left accumulator.
RRCA		00 001 111	0F	↕	●	●	●	0	0	1	1	4	Rotate right circular accumulator.
RRA		00 011 111	1F	↕	●	●	●	0	0	1	1	4	Rotate right accumulator.
RLCr		11 001 011 00 k r	CB 00+	↕	↕	P	↕	0	0	2	2	8	Rotate left circular register r.
RLC (HL)		11 001 011 00 k 110	CB 06+	↕	↕	P	↕	0	0	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r, (HL), (IX+d), (IY+d)	11 011 101 11 001 011 ← d → 00 k 110	DD CB 06+	↕	↕	P	↕	0	0	4	6	23	
RLC (IY+d)		11 111 101 11 001 011 ← d → 00 k 110	FD CB 06+	↕	↕	P	↕	0	0	4	6	23	
RL m				↕	↕	P	↕	0	0				Mnemonic k RLC 000 RRC 001 RL 010 RR 011 SLA 100 SRA 101 SRL 111
RRC m				↕	↕	P	↕	0	0				
RR m				↕	↕	P	↕	0	0	2* 2 4 4	2* 4 6 6	8* 15 23 23	
SLA m				↕	↕	P	↕	0	0				
SRA m				↕	↕	P	↕	0	0				m=r, (HL), (IX+d), (IY+d)
SRL m				↕	↕	P	↕	0	0				*depends on m.
RLD		11 101 101 01 101 111	ED 6F	●	↕	P	↕	0	0	2	5	18	Rotate digit left and right between the accumulator and location (HL).
RRD		11 101 101 01 100 111	ED 67	●	↕	P	↕	0	0	2	5	18	The content of the upper half of the accumulator is unaffected.

Flags : ● = unchanged
0 = reset
1 = set
× = undefined
↕ = set or reset according to the result of the operation

Table 8 Bit set, reset and test group

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
BIT b, r	$Z \leftarrow \overline{r}_b$	11 001 011 01 b r	CB 40+	●	↕	X	X	0	1	2	2	8	
BIT b, (HL)	$Z \leftarrow \overline{(HL)}_b$	11 001 011 01 b 110	CB 46+	●	↕	X	X	0	1	2	3	12	
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	11 011 101 11 001 011 ← d → 01 b 110	DD CB 46+	●	↕	X	X	0	1	4	5	20	
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	11 111 101 11 001 011 ← d → 01 b 110	FD CB 46+	●	↕	X	X	0	1	4	5	20	
SET b, r	$r_b \leftarrow 1$	11 001 011 a b r	CB	●	●	●	●	●	●	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	11 001 011 a b 110	CB 06+	●	●	●	●	●	●	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	11 011 101 11 001 011 ← d → a b 110	DD CB 06+	●	●	●	●	●	●	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	11 111 101 11 001 011 ← d → a b 110	FD CB 06+	●	●	●	●	●	●	4	6	23	
RES b, m	$m_b \leftarrow 0$									2* 2 4 4	2* 4 6 6	8* 15 23 23	m=r, (HL), (IX+d), (IY+d) *depends on m

r	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

b	Bit Tested
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Mnemonic	a
SET	11
RES	10

Note : The notation m_b indicates bit b (0 to 7) or location m.
Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

Table 9 Jump group

Mnemonic	Symbolic	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
	operation	76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
JP nn	PC ← nn	11 000 011 ← n → ← n →	C3	●	●	●	●	●	●	3	3	10	
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	11 cc 010 ← n → ← n →	C2+	●	●	●	●	●	●	3	3	10	
										3	3	10	
JR e	PC ← PC+e	00 011 000 ← e-2 →	18	●	●	●	●	●	●	2	3	12	
JR C, e	If C=1 PC ← PC+e	00 111 000 ← e-2 →	38	●	●	●	●	●	●	2	3	12	
	If C=0 continue									2	2	7	
JR NC, e	If C=0 PC ← PC+e	00 110 000 ← e-2 →	30	●	●	●	●	●	●	2	3	12	
	If C=1 continue									2	2	7	
JR Z, e	If Z=1 PC ← PC+e	00 101 000 ← e-2 →	28	●	●	●	●	●	●	2	3	12	
	If Z=0 continue									2	2	7	
JR NZ, e	If Z=0 PC ← PC+e	00 100 000 ← e-2 →	20	●	●	●	●	●	●	2	3	12	
	If Z=1 continue									2	2	7	
JP (HL)	PC ← HL	11 101 001	E9	●	●	●	●	●	●	1	1	4	
JP (IX)	PC ← IX	11 011 101 11 101 001	DD E9	●	●	●	●	●	●	2	2	8	
JP (IY)	PC ← IY	11 111 101 11 101 001	FD E9	●	●	●	●	●	●	2	2	8	
DJNZ, e	If B ← B−1 B≠0 PC ← PC+1	00 010 000 ← e-2 →	10	●	●	●	●	●	●	2	3	13	
	If B=0 continue									2	2	8	

cc	Condition
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

NZ : non-zero
Z : zero
C : carry
PO : parity odd
PE : parity even
P : sign positive
M : sign negative

Note : e represents the extension in the relative addressing mode.
e is a signed two's complement number in the range <−126, 129>
e −2 in the opcode provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.
e itself is obtained from opcode position.

Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

Table 10 Call and return group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	(SP−1) ← PCH (SP−2) ← PCL PC ← nn	11 001 101 ← n → ← n →	CD	●	●	●	●	●	●	3	5	17	
CALL cc, nn	If condition cc is false continue,	11 cc 100 ← n →	C4+	●	●	●	●	●	●	3	5	17	
	otherwise same as CALL nn	← n →								3	3	10	
RET	PCL ← (SP) PCH ← (SP+1)	11 001 001	C9	●	●	●	●	●	●	1	3	10	
RET cc	If condition cc is false continue,	11 cc 000	C0+	●	●	●	●	●	●	1	3	11	
	otherwise same as RET									1	1	5	
RETI	Return from interrupt	11 101 101 01 001 101	ED 4D	●	●	●	●	●	●	2	4	14	
RETN	Return from non-maskable interrupt	11 101 101 01 000 101	ED 45	●	●	●	●	●	●	2	4	14	
RST p	(SP−1) ← PCH (SP−2) ← PCL PCH ← 0 PCL ← p	11 t 111	C7+	●	●	●	●	●	●	1	3	11	

cc	Condition
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

r	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

Flags : ● =unchanged
0=reset
1=set
X=undefined
‡ =set or reset according to the result of the operation

Table 11 Input and output group

Mnemonic	Symbolic operation	OP code	HEX code	Flags						No. of	No. of	No. of	Comments
		76 543 210	(Basic)	C	Z	P/V	S	N	H	Bytes	M Cycles	T States	
IN A, (n)	A ← (n)	11 011 011 ← n →	DB	●	●	●	●	●	●	2	3	11	n → A_0-A_7 Acc → A_8-A_{15}
IN r, (C)	r ← (C)	11 101 101 01 r 000	ED 40+	●	↕	P	↕	0	↕	2	3	12	
INI	(HL) ← (C) B ← B−1 HL ← HL+1	11 101 101 10 100 010	ED A2	X	↕ ①	X	X	1	X	2	4	16	C → A_0-A_7 B → A_8-A_{15}
INIR	(HL) ← (C) B ← B−1 HL ← HL+1 Repeat until B=0	11 101 101 10 110 010	ED B2	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	
IND	(HL) ← (C) B ← B−1 HL ← HL−1	11 101 101 10 101 010	ED AA	X	↕ ①	X	X	1	X	2	4	16	
INDR	(HL) ← (C) B ← B−1 HL ← HL−1 Repeat until B=0	11 101 101 10 111 010	ED BA	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	
OUT (n), A	(n) ← A	11 010 011 ← n →	D3	●	●	●	●	●	●	2	3	11	n → $(A\text{-}BUS)_{0\text{-}7}$ Acc → $(A\text{-}BUS)_{8\text{-}15}$
OUT (C), r	(C) ← r	11 101 101 01 r 001	ED 41+	●	●	●	●	●	●	2	3	12	
OUTI	(C) ← (HL) B ← B−1 HL ← HL+1	11 101 101 10 100 011	ED A3	X	↕ ①	X	X	1	X	2	4	16	C → A_0-A_7
OTIR	(C) ← (HL) B ← B−1 HL ← HL+1 Repeat until B=0	11 101 101 10 110 011	ED B3	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	B → A_8-A_{15}
OUTD	(C) ← (HL) B ← B−1 HL ← HL−1	11 101 101 10 101 011	ED AB	X	↕ ①	X	X	1	X	2	4	16	
OTDR	(C) ← (HL) B ← B−1 HL ← HL−1 Repeat until B=0	11 101 101 10 111 011	ED BB	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	

r	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

Note : ①If the result of B−1 is zero the Z flag is set, otherwise it is reset.
②Z flag is set upon instruction completion only.

Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

LH0081 Z80 PIO Parallel Input/Output Controller

■ Description

The Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The LH0081 Z80 PIO (Z80 PIO for short below) is a programmable two port device which provides TTL compatible interfacing between peripheral devices and the Z80 CPU. The Z80 CPU configures Z80 PIO to interface with standard peripheral devices such as printers, keyboards, etc.

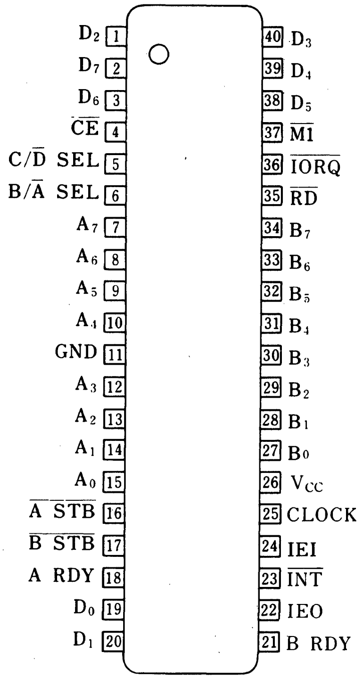
The LH0081A Z80A and LH0081B Z80B PIO are the high speed version which can opeate at the 4MHz and 6MHz system clock, respectively.

■ Features

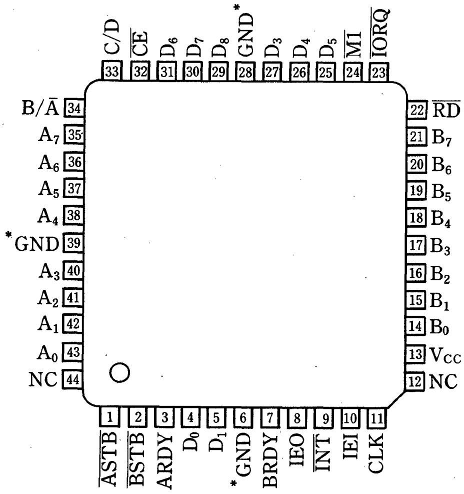
1. Two independent 8-bit bidirectional peripheral interface ports with "handshake" data transfer control
2. N-channel silicon-gate process

■ Pin Connections

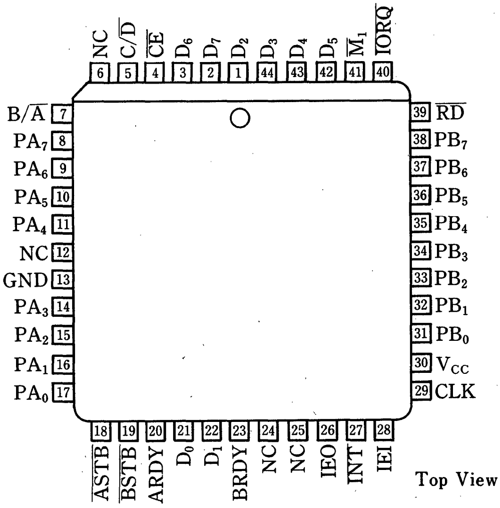
LH0081/LH0081A/LH0081B/LH0081E



LH0081M/LH0081AM



LH0081U/LH0081AU



Top View

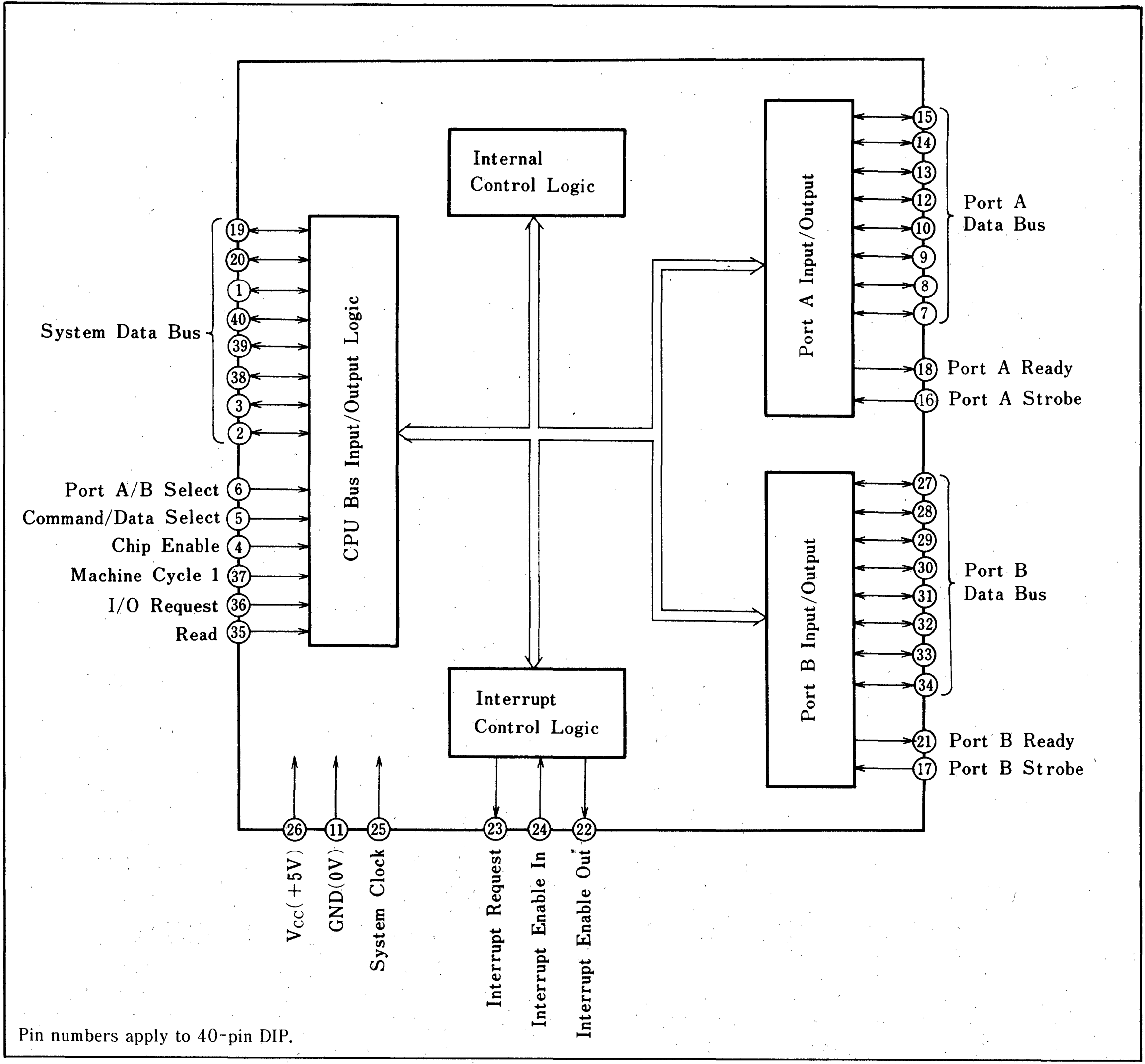
*The GND pins must be connected to the GND level.

3. Any one of the following four modes of operation may be selected.
   - Byte output mode
   - Byte input mode
   - Byte bidirectional bus (available on Port A only)
   - Bit mode
4. Programmable interrupt
5. Vectored daisy chain priority interrupt logic included
6. The port B outputs can drive Darlington transistors
7. All inputs and outputs fully TTL compatible
8. Single +5V power supply and single phase clock
9. 40-pin DIP (DIP40-P-600)
   44-pin QFP (QFP44-P-1010A)
   44-pin QFJ (QFJ44-P-S650)

Ordering Information

Product	Z80 PIO	Z80A PIO	Z80B PIO	Z80E PIO	Package	Operating temperature
Clock frequency	2.5MHz	4MHz	6MHz	8MHz		
Model No.	LH0081	LH0081A	LH0081B	LH0081E	40-pin DIP	0°C to+70°C
	LH0081M	LH0081AM			40-pin QFP	0°C to+60°C
	LH0081U	LH0081AU	LH0081BU		40-pin QFJ	0°C to+70°C

Block Diagram



Pin numbers apply to 40-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
D_0-D_7	Data bus	Bidirectional 3-state	System data bus.
B/$\overline{\text{A}}$ SEL	Port B or A select	I	Defines which port is accessed. A high selects port B, and a low port A.
C/$\overline{\text{D}}$ SEL	Control or data select	I	Defines the type of data transfer on the data bus. A high selects control, and a low data.
$\overline{\text{CE}}$	Chip enable	I	Active "Low". A low enables the CPU to transmit and receive control words and data.
CLOCK	System clock	I	Standard Z80 system clock used for internal synchronization signals.
$\overline{\text{M1}}$	Machine cycle one	I	Active "Low". Indicates that the CPU is acknowledging an interrupt, when both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active.
$\overline{\text{IORQ}}$	Input/output request	I	Active "Low". Read operation when $\overline{\text{RD}}$ is active, and write operation when it is not active. Indicates that the CPU is acknowledging an interrupt, when both $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are active.
$\overline{\text{RD}}$	Read cycle status	I	Active "Low". Read operation when active.
IEI	Interrupt enable in	I	Active "High". Forms a priority-interrupt daisy-chain.
IEO	Interrupt enable out	O	Active "High". Forms a priority-interrupt daisy-chain.
$\overline{\text{INT}}$	Interrupt request	Open drain, O	Active "Low". Active when requesting an interrupt.
A_0-A_7	Port A bus	Bidirectional 3-state	Transfers information between port A and a peripheral device.
$\overline{\text{A STB}}$	Port A strobe	I	Active "Low". Used as a handshake line for data transfer synchronization on port A. Not used in the bit control mode.
A RDY	Port A ready	O	Active "High". Used as a handshake line for data transfer synchronization on port A. Not used in the bit control mode.
B_0-B_7	Port B bus	Bidirectional 3-state	Transfers information between port B and a peripheral device.
$\overline{\text{B STB}}$	Port B strobe	I	Active "Low". Used as a handshake line for data transfer synchronization on port B. Not used in the bit control mode.
B RDY	Port B ready	O	Active "High". Used as a handshake line for data transfer synchronization on port B. Not used in the bit control mode.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Input voltage	V_{IN}	−0.3 to +7.0	V
Output voltage	V_{OUT}	−0.3 to +7.0	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃

■ DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃[Note 1])

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock input low voltage	V_{ILC}		−0.3		0.45	V
Clock input high voltage	V_{IHC}		V_{CC}−0.6		V_{CC}+0.3	V
Input low voltage	V_{IL}		−0.3		0.8	V
Input high voltage	V_{IH}		2.0		5.5	V
Output low voltage	V_{OL}	I_{OL}=2mA			0.4	V
Output high voltage	V_{OH}	I_{OH}=−250μA	2.4			V
Supply current	I_{CC}	V_{OH}=1.5V			100	mA
Input leakage current	$\vert I_{LI} \vert$	$0 \leqq V_{IN} \leqq V_{CC}$			10	μA
3 state output/data bus input leakage current	$\vert I_z \vert$	$0 \leqq V_{IN} \leqq V_{CC}$			10	μA
Darlington drive current	I_{OHD}	R_{EXT}=390Ω	−1.5			mA

Note 1: Ta=0 to +60℃ for 44-pin QFP.

■ Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			12	pF
Input capacitance	C_{IN}				7	pF
Output capacitance	C_{OUT}				10	pF

■ AC Characteristics

(V_{CC}=5V±5%, Ta=0 to+70°C)

No.	Parameter	Symbol	LH0081		LH0081A		LH0081B		LH0081E		Unit	Note
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
1	Clock Cycle time	TcC	400	(Note 1)	250	(Note 1)	165	(Note 1)	125	(Note 1)	ns	
2	Clock width (high)	TwCh	170	2000	105	2000	65	2000	55	2000	ns	
3	Clock width (low)	TwCl	170	2000	105	2000	65	2000	55	2000	ns	
4	Clock fall time	TfC		30		30		20		10	ns	
5	Clock rise time	TrC		30		30		20		10	ns	
6	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORO}$ ↓ Setup time	TsCS (RI)	50		50		50		50		ns	6
7	Any hold times for specified setup time	Th	0		0		0		0		ns	
8	$\overline{RD}$, $\overline{IORQ}$ to clock ↑ setup time	TsRI (C)	115		115		70		60		ns	
9	$\overline{RD}$, $\overline{IORQ}$ ↓ to deta out delay	TdRI (DO)		430		380		300		210	ns	2
10	$\overline{RD}$, $\overline{IORQ}$ ↑ to deta out float delay	TdRI (DOs)		160		110		70		60	ns	
11	Data in to clock ↑ setup time	TsRI (C)	50		50		40		30		ns	C_L=50pF
12	$\overline{IORQ}$ ↓ to vector out delay (INTACK cycle)	TdIO (DOI)	340		160		120		90		ns	3
13	$\overline{MI}$ ↓ to clock ↑ setup time	TsMI (Cr)	210		90		70		55		ns	
14	$\overline{MI}$ ↑ to clock ↓ setup time ($\overline{MI}$ cycle)	TsMI (Cf)	0		0		0		0		ns	8
15	$\overline{MI}$ ↓ to IEO ↓ delay (interrupt immedietely preceding $\overline{MI}$ ↓)	TdMI (IEO)		300		190		100		85	ns	5, 7
16	IEI to $\overline{IORQ}$ ↓ setup time (INTACK cycle)	TsIEI (IO)	140		140		100		100		ns	7
17	IEI ↓ to IEO ↓ delay	TdIEI (IEOf)		190		130		120		110	ns	5 C_L=50pF
18	IEI ↑ to IEO ↑ delay (After ED decode)	TdIEI (IEOr)		210		160		160		150	ns	5

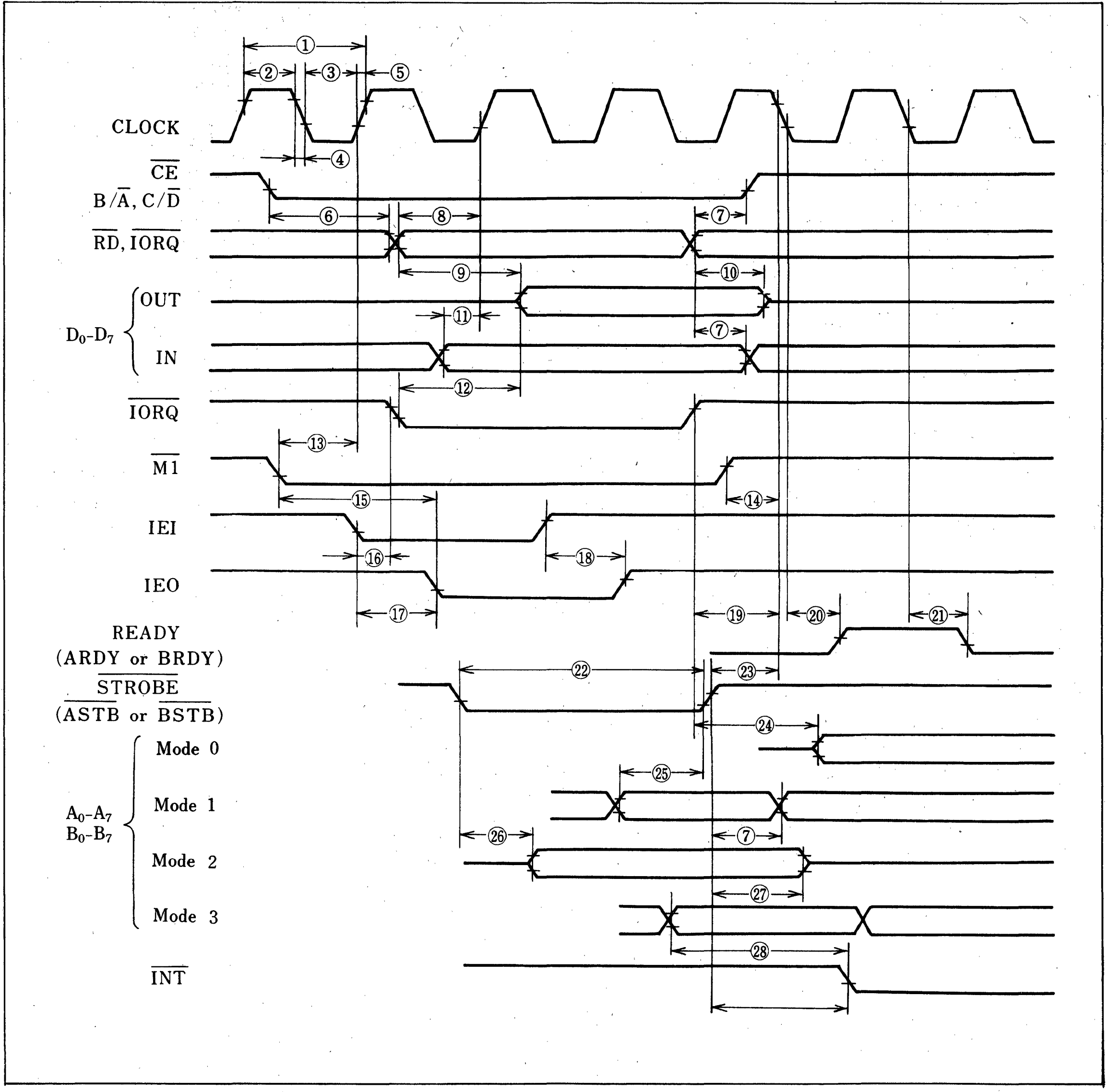
No.	Parameter	Symbol	LH0081		LH0081A		LH0081B		LH0081E		Unit	Note
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
19	$\overline{\text{IORQ}}$ ↑ to clock ↓ setup time (to activate READY to next clock cycle)	TcIO (C)	220		200		170		160		ns	
20	Clock ↓ to READY ↑ delay	TdC (RDYr)	200		190		170		160		ns	5 C_L=50pF
21	Clock ↓ to READY ↓ delay	TdC (RDYf)	150		140		120		110		ns	5
22	$\overline{\text{STROBE}}$ pulse width	TwSTB	150		150		120		100		ns	4
23	$\overline{\text{STROBE}}$ ↑ to clock ↓ setup time (to activate READY on next clock cycue)	TsSTB (C)	220		220		150		130		ns	5
24	$\overline{\text{IORQ}}$ ↑ to PORT DATA stable delay (Mode 0)	TdIO (PD)		200		180		160		150	ns	5
25	PORT DATA to $\overline{\text{STROBE}}$ ↑ setup time (mode 1)	TsPD (STB)	260		230		190		170		ns	
26	$\overline{\text{STROBE}}$ ↓ to PORT DATA stable (mode 2)	TdSTB (PD)		230		210		180		160	ns	5
27	$\overline{\text{STROBE}}$ ↑ to PORT DATA float delay (mode 2)	TdSTB (PDr)		200		180		160		140	ns	C_L=50pF
28	PORT DATA match to $\overline{\text{INT}}$ ↓ delay (mode 3)	TdPD (INT)		540		490		430		380	ns	
29	$\overline{\text{STROBE}}$ ↑ to $\overline{\text{INT}}$ ↓ delay	TdSTB (INT)		490		440		350		300	ns	

↑ Rising edge, ↓ Falling edge

Ta=0 to+60°C for 44-pin QFP.

Note 1: TcC=TwCh+TwCl+TrC+TfC.
Note 2: Increase TdRI (DO) by 10 ns for each 50 pF increase in load up to 200 pF max.
Note 3: Increase TdIO (DOI) by 10 ns for each 50 pF, increase in load up to 200 pF max.
Note 4: For Mode 2 : TwSTB>TsPD (STB).
Note 5: Increase these values by 2 ns for each 10 pF increase in load up to 100 pF max.
Note 6: TsCS (RI) may be reduced. However, the time subtracted from TsCS (RI) will be added to TdRI (DO).
Note 7: 2.5 TcC>(N−2) TdIEI (IEOf)+TdM1 (IEO)+TsIEI (IO)+TTL buffer delay, if any.
Note 8: $\overline{\text{M1}}$ must be active for a minimum of two clock cycles to reset the PIO.
Note 9: Z80B PIO numbers are preliminary and subject to change.

AC Timing Chart



Programming

(1) Interrupt vector read

An interrupting device needs giving an 8-bit interrupt vector to the CPU. Using this vector, the CPU forms an interrupt service routine address.

D7	D6	D5	D4	D3	D2	D1	D0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	0

Indicates an interrupt vector

(2) Operation mode select

An operation mode is selected by writing data to the 2-bit mode control register in the following manner.

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

Mode Words

Indicates Mode Words

X means they are not used

Mode	M1	M0
Byte output mode	0	0
Byte input mode	0	1
Bidirectional byte bus mode	1	0
Bit control mode	1	1

In selecting the bit control mode, an input/output direction should be set later.

D7	D6	D5	D4	D3	D2	D1	D0
I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0

I/O=1:Input; I/O=0:Output

(3) Interrupt control

The interrupt control words are as follows.

D7	D6	D5	D4	D3	D2	D1	D0
Interrupt enable flag	AND /OR	High /Low	Mask word	0	1	1	1

Effective in bit control mode

Indicates that the information is interrupt control words

Bit7=1: Interrupt enable flag is set to enable an interrupt.

Bit7=0: Interrupt enable flag is reset to disable an interrupt.

Bit6 to 4: Defines interrupt conditions in the bit mode. Ignored in other modes.

Bit3 to 0: Indicates interrupt control words.

If bit4=1, the following control words are supposed to be written in the mask register.

D7	D6	D5	D4	D3	D2	D1	D0
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

Only the port data line with MB=0 is monitored. When the interrupt conditions are satisfied, an interrupt takes place.

D7	D6	D5	D4	D3	D2	D1	D0
Interrupt enable flag	×	×	×	0	0	1	1

Not affected by these bits

Discriminator code

Timing

(1) Output mode (Mode 0)

An output cycle is always started by the execution of an output instruction by the CPU. The $\overline{WR}$* pulse from the CPU latches the data from the CPU data bus into the selected port's output register. The $\overline{WR}$* pulse sets the Ready flag after a Low-going edge of CLK, indicating data is available. Ready stays active until the positive edge of the strobe line is received, indication that data was taken by the peripheral. The positive edge of the strobe pulse generates an $\overline{INT}$ if the interrupt enable flipflop has been set and if this device has the highest priority.

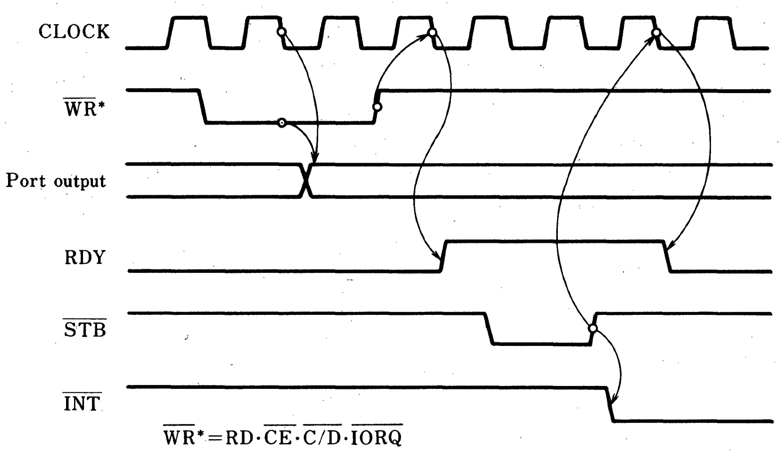


Fig. 1 Byte output mode timing

(2) Input mode (Mode 1)

When STROBE goes Low, data is loaded into the selected port input register. The next rising edge of strobe activates INT, if Interrupt Enable is set and this is the highest-priority requesting device. The following falling edge of CLK resets Ready to an inactive state, indicating that the input register is full and cannot accept any more data until the CPU completes a read. When a read is complete, the positive edge of RD sets Ready at the next Low-going transition of CLK. At this time new data can be loaded into the PIO.

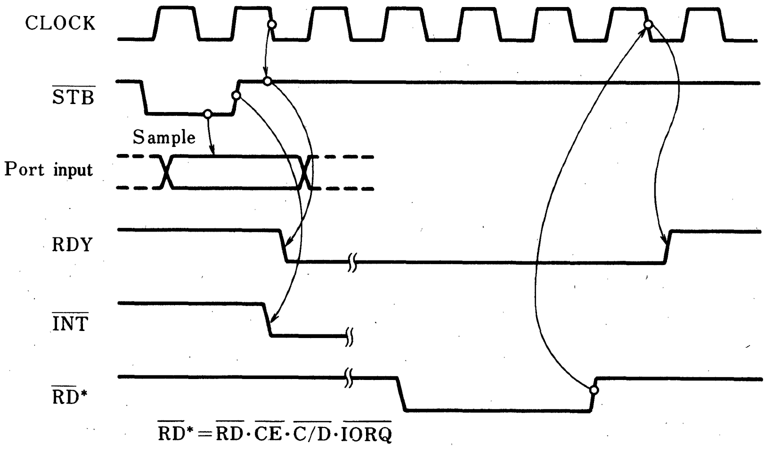


Fig. 2 Byte input mode timing

(3) Bidirectional mode (Mode 2)

This is a combination of Modes 0 and 1 using all four handshake lines and the eight Port A I/O lines. Port B must be set to the bit mode and its inputs must be masked. The Port A handshake lines are used for output control and the Port B lines are used for input control. If interrupts occur, Port A's vector will be used during port output and Port B's will be used during port input. Data is allowed out onto the Port A bus only when ASTB is Low. The rising edge of this strobe can be used to latch the data into the peripheral.

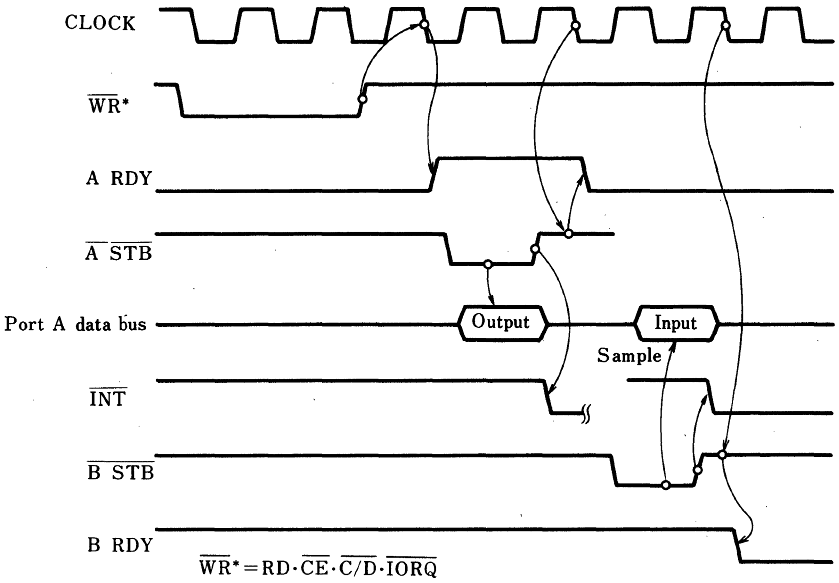


Fig. 3 Byte bidirectional bus mode timing

(4) Bit mode (Mode 3)

The bit mode does not utilize the handshake signals, and a normal port write or port read can be executed at any time. When writing , the data is latched into the output registers with the same timing as the output mode.

When reading the PIO, the data returned to the CPU is composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as inputs. The input register contains data that was present immediately prior to the falling edge of RD. An interrupt is generated if interrupts from the port are enabled and the data on the port data lines satisfy the logical equation defined by the 8-bit mask and 2-bit mask control registers. However, if Port A is programmed in bidirectional mode, Port B does not issue an interrupt in bit mode and must therefore be polled.

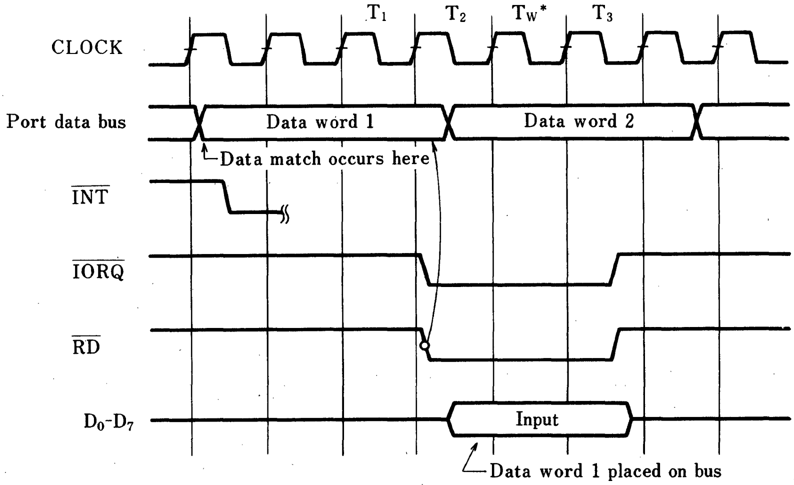


Fig. 4 Bit mode timing

(5) Interrupt acknowledge timing

During $\overline{M1}$ time, peripheral controllers are inhibited from changing their interrupt enable status, permitting the Interrupt Enable signal to ripple through the daisy chain. The peripheral with IEI High and IEO Low during $\overline{INTACK}$ places a preprogrammed 8-bit interrupt vector on the data bus at this time. IEO is held Low until a Return From Interrupt (RETI) instruction is executed by the CPU while IEI is High. The 2-byte RETI instruction is decoded internally by the PIO for this purpose.

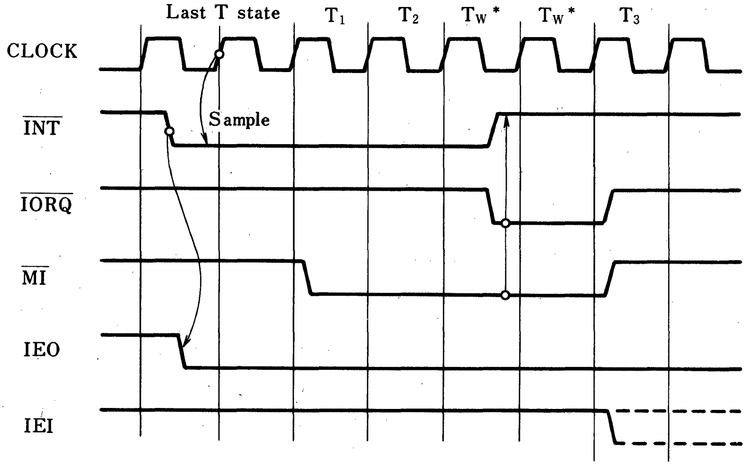


Fig. 5 Interrupt acknowledge timing

(6) Return from interrupt cycle

If a Z-80 peripheral has no interrupt pending and is not under service, then its IEO=IEI. If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its IEO is always Low, inhibiting lower priority devices from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO is Low unless an "ED" is decoded as the first byte of a 2-byte opcode. In this case, IEO goes High until the next opcode byte is decoded, whereupon it goes Low again. If the second byte of the opcode was a "4D", then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service has its IEI High and its IEO Low. This device is the highest-priority device in the daisy chain that has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D". this peripheral device resets its "interrupt under service" condition.

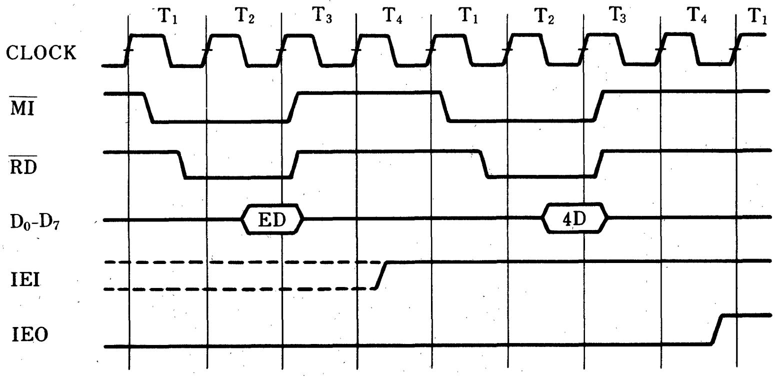


Fig. 6 Return from interrupt cycle timing

LH0082 Z80 CTC Counter Timer Circuit

Description

The Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The LH0082 Z80 CTC (Z80 CTC for short below) is a programmable, four channel device that provides counting and timing functions for the Z80 CPU. The Z80 CPU configures the Z80 CTC's four independent channels to operate under various modes and conditions as required.

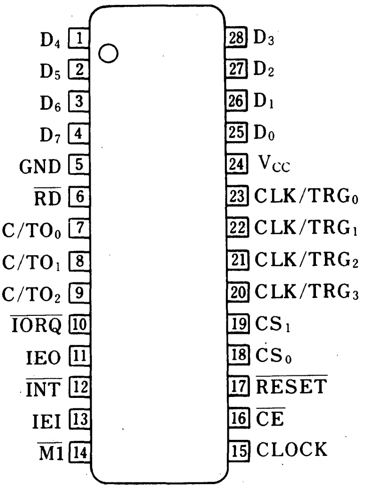
The LH0082A Z80A and LH0082B Z80B CTC are the high speed version which can operate at the 4MHz and 6MHz system clock, respectively.

Features

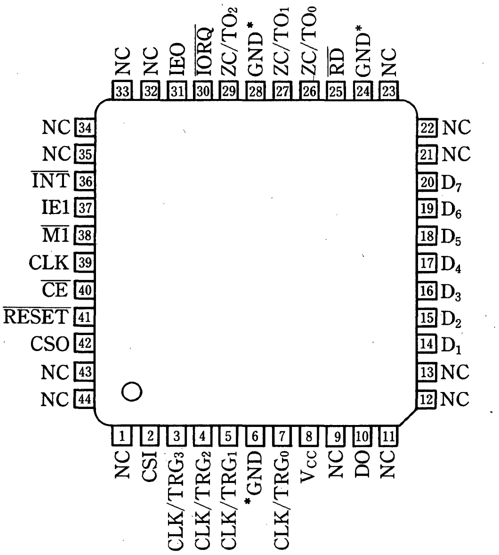
1. Four independent programmable 8-bit counter/16-bit timer channels
2. N-channel silicon gate process
3. Each channel may be selected to operate in either a counter mode or timer mode

Pin Connections

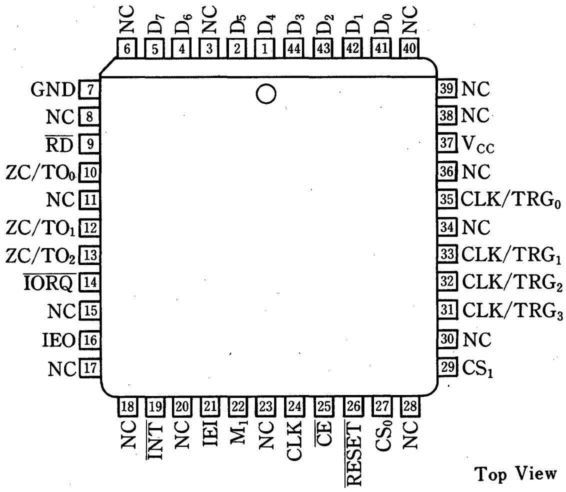
LH0082/LH0082A/LH0082B/LH0082E



LH0082M/LH0082AM



LH0082U/LH0082AU/LH0082BU



Top View

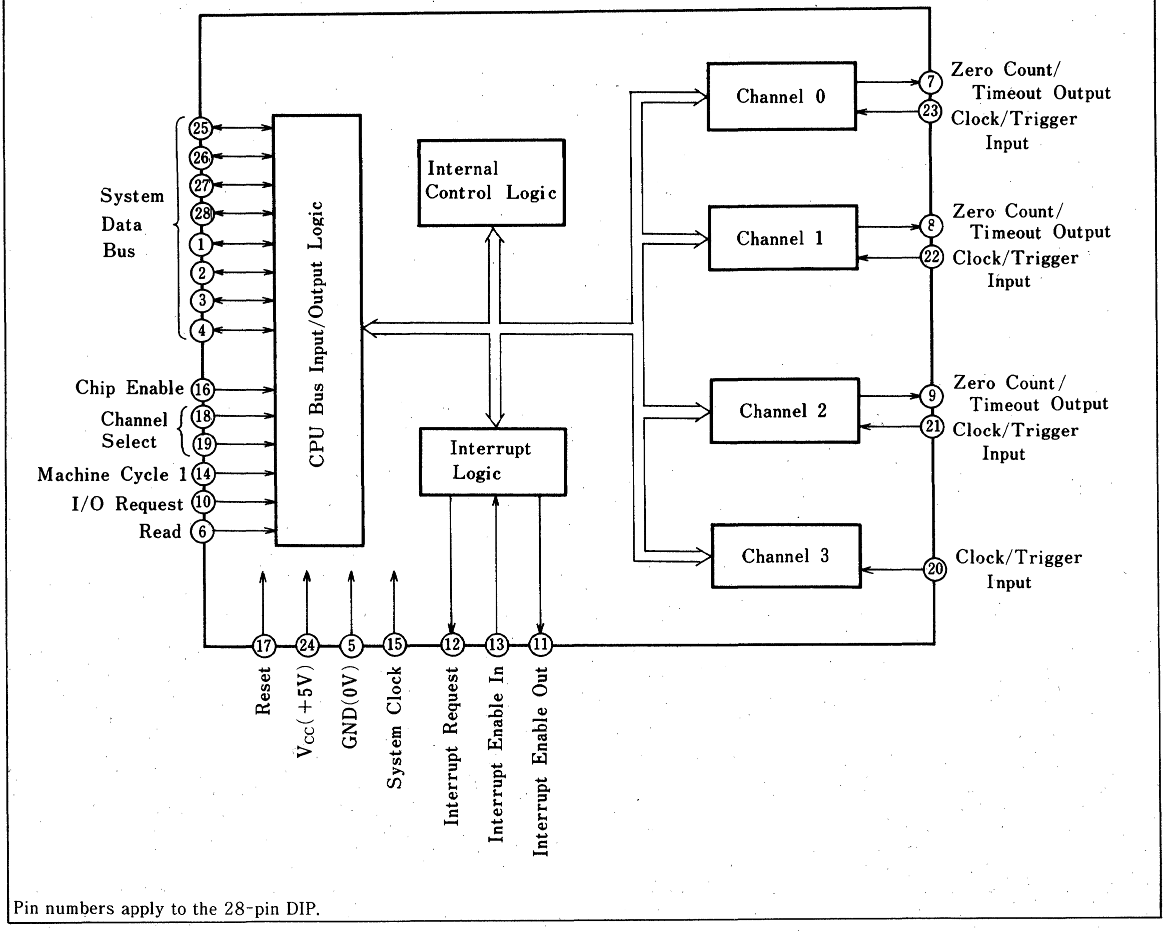
*The GND pins must be connected to the GND level.

4. Programmable interrupts on counter or timer states
5. When the down-counter reaches the zero count the CTC reloads its time constant automatically and continues it's channel operation
6. Readable down counter
7. Selectable 16 or 256 clock prescaler for each timer channels
8. Selectable positive or negative trigger may initiate timer or counter operation
9. Three channels have ZC/TO outputs capable of driving Darlington transistors
10. Vectored daisy chain priority interrupt logic included
11. Single +5V power supply and single phase clock
12. All inputs and outputs fully TTL compatible
13. 28-pin DIP (DIP28-P-600)
    44-pin QFP (QFP44-P-1010A)
    44-pin QFJ (QFJ44-P-S650)

Ordering Information

Product	Z80 CTC	Z80A CTC	Z80B CTC	Z80E CTC	Package	Operating temperature
Clock frequency	2.5MHz	4MHz	6MHz	8MHz		
Model No.	LH0082	LH0082A	LH0082B	LH0082E	28-pin DIP	0°C to+70°C
	LH0082M	LH0082AM			44-pin QFP	0°C to+60°C
	LH0082U	LH0082AU	LH0082BU		44-pin QFJ	0°C to+70°C

Block Diagram



Pin numbers apply to the 28-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
D_0-D_7	Data bus	Bidirectional 3-state	System data bus.
CS_0, CS_1	Channel select	I	Selects one of the four independent channels.
$\overline{CE}$	Chip enable	I	Active "Low". A Low enables the CPU to transmit and receive control words and data.
CLOCK	System clock	I	Standard Z80 system clock used for internal synchronization signals.
$\overline{M1}$	Machine cycle one	I	Active "Low". Indicates that the CPU is acknowledging an interrupt, when both $\overline{M1}$ and $\overline{IORQ}$ are active.
$\overline{IORQ}$	I/O request	I	Active "Low". Read operation when RD is active, and write operation when it is not active. Indicates the CPU is acknowledging an interrupt, when both $\overline{IORQ}$ and $\overline{M1}$ are active.
$\overline{RD}$	Read cycle status	I	Active "Low". Read operation when active.
IEI	Interrupt enable in	I	Active "High". Forms a priority-interrupt daisy-chain.
IEO	Interrupt enable out	O	Active "High". Forms a priority-interrupt daisy-chain.
$\overline{INT}$	Interrupt request	Open drain, O	Active "Low". Active when requesting an interrupt.
$\overline{RESET}$	Reset	I	Active "Low". Resets the interrupt bits.
CLK/TRG_0-CLK/TRG_3	External clock/timer trigger input	I	Counter/timer input for four independent channels.
ZC/TO_0-ZC/TO_2	External clock/timer trigger out	O	Active "High". 0/1/2 output for four independent channels. No output terminal at channel 3.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	−0.3 to +7.0	V	
Output voltage	V_{OUT}	−0.3 to +7.0	V	
Operating temperature	Topr	0 to +70	℃	1
Storage temperature	Tstg	−65 to +150	℃	

Note 1: Topr=0 to +60℃ for 44-pin QFP.

DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃$^{\text{Note 1}}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock input low voltage	V_{ILC}		−0.3		0.45	V
Clock input high voltage	V_{IHC}		$V_{CC}-0.6$		$V_{CC}+0.3$	V
Input low voltage	V_{IL}		−0.3		0.8	V
Input high voltage	V_{IH}		2.0		V_{CC}	V
Output low voltage	V_{OL}	I_{OL}=2mA			0.4	V
Output high voltage	V_{OH}	$I_{OH}=-250\mu A$	2.4			V
Supply current	I_{CC}	t_c=400ns			120	mA
Input leakage current	I_{LI}	$0\leqq V_{IN}\leqq V_{CC}$			10	μA
3-state output leakage current	$\mid I_{LOH}\mid$	$2.4V\leqq V_{OUT}\leqq V_{CC}$			10	μA
3-state output leakage current	$\mid I_{LOL}\mid$	V_{OUT}=0.4V			10	μA
Darlington drive current	I_{OHD}	V_{OH}=1.5V, R_{EXT}=390Ω	−1.5			mA

Note 1: Ta=0 to +60℃ for 44-pin QFP.

Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			20	pF
Input capacitance	C_{IN}				5	pF
Output capacitance	C_{OUT}				10	pF

AC Characteristics

(V_{CC}=5V ± 5%, Ta=0 to+70°C)

No.	Parameter	Symbol	LH0082		LH0082A		LH0082B		LH0082E		Unit	Note
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
1	Clock cycle time	TcC	400	(Note 1)	250	(Note 1)	165	(Note 1)	125	(Note 1)	ns	
2	Clock width (high)	TwCh	170	2000	105	2000	65	2000	55	2000	ns	
3	Clock width (low)	TwCl	170	2000	105	2000	65	2000	55	2000	ns	
4	Clock fall time	TfC		30		30		20		10	ns	
5	Clock rise time	TrC		30		30		20		10	ns	
6	All hold times	Th	0		0		0		0		ns	
7	CS to clock ↑ setup time	TsCS (C)	250		160		100		80		ns	
8	$\overline{CE}$ to clock ↑ setup time	TsCE (C)	200		150		100		75		ns	
9	$\overline{IORQ}$ ↑ to clock ↑ setup time	TsIO (C)	250		115		70		60		ns	
10	$\overline{RD}$ ↓ to clock ↑ setup time	TsRD (C)	240		115		70		60		ns	
11	Clock ↑ to data out delay	TdC (DO)		240		200		130		100	ns	2
12	Clock ↓ to data out float delay	TdC (DOz)		230		110		90		75	ns	
13	Data in to clock ↑ setup time	TsDI (C)	60		50		40		30		ns	
14	$\overline{Ml}$ to clock ↑ setup time	TsMl (C)	210		90		70		55		ns	
15	$\overline{Ml}$ ↓ to IEO ↓ delay (interrupt immediately preceding $\overline{Ml}$)	TdMl (IEO)		300		190		130		90	ns	3
16	$\overline{IORQ}$ ↓ to data out delay (INTA cycle)	TdIO (DOI)		340		160		110		80	ns	2
17	$\overline{IEI}$ ↓ to IEO ↓ delay	TdIEI (IEOf)		190		130		100		80	ns	3
18	$\overline{IEI}$ ↑ to IEO ↑ delay (after ED decode)	TdIEI (IEOr)		220		160		110		80	ns	3
19	Clock ↑ to $\overline{INT}$ ↓ delay	TdC (INT)		TcC+200		TcC+140		TcC+120		TcC+100	ns	4
20	CLK/TRG ↑ to $\overline{INT}$ ↓ delay tsCTR (C) satisfied	TdCLK (INT)		TcC+230		TcC+160		TcC+130		TcC+110	ns	5
	CLK/TRG ↑ to $\overline{INT}$ ↓ delay tsCTR (C) not satisfied			2TcC+530		2TcC+370		2TcC+280		2TcC+190	ns	5
21	CLK/TRG cycle time	TsCTR	2TcC		2TcC		2TcC		2TcC		ns	5
22	CLK/TRG rise time	TrCTR		50		50		40		35	ns	
23	CLK/TRG fall time	TfCTR		50		50		40		35	ns	
24	CLK/TRG width (low)	TwCTRl	200		200		120		100		ns	
25	CLK/TRG width (high)	TwCTRh	200		200		120		100		ns	
26	CLK/TRG ↑ to clock ↑ setup time for immediate count	TsCTR (Cs)	300		210		150		110		ns	5
27	CLK/TRG ↑ to clock ↑ setup time for enabling of prescaler on following clock ↑	TsCTR (Ct)	210		210		150		110		ns	4
28	Clock ↑ to ZC/TO ↑ delay	TdC (ZC/TOr)		260		190		140		110	ns	
29	Clock ↓ to ZC/TO ↓ delay	TdC (ZC/TOf)		190		190		140		110	ns	

↑ Rising edge, ↓ falling edge

Ta=0 to+60°C for 44-pin QFP

[A] 2.5 TcC> (n-2) TdIEI (IEOf) + TdMI (IEO) + TsIEI (IO) + TTL buffer delay, if any.

[B] $\overline{RESET}$ must be active for a minimum of 3 clock cycles.

Note 1 : TcC=TwCh+TwCl+TrC+TfC.

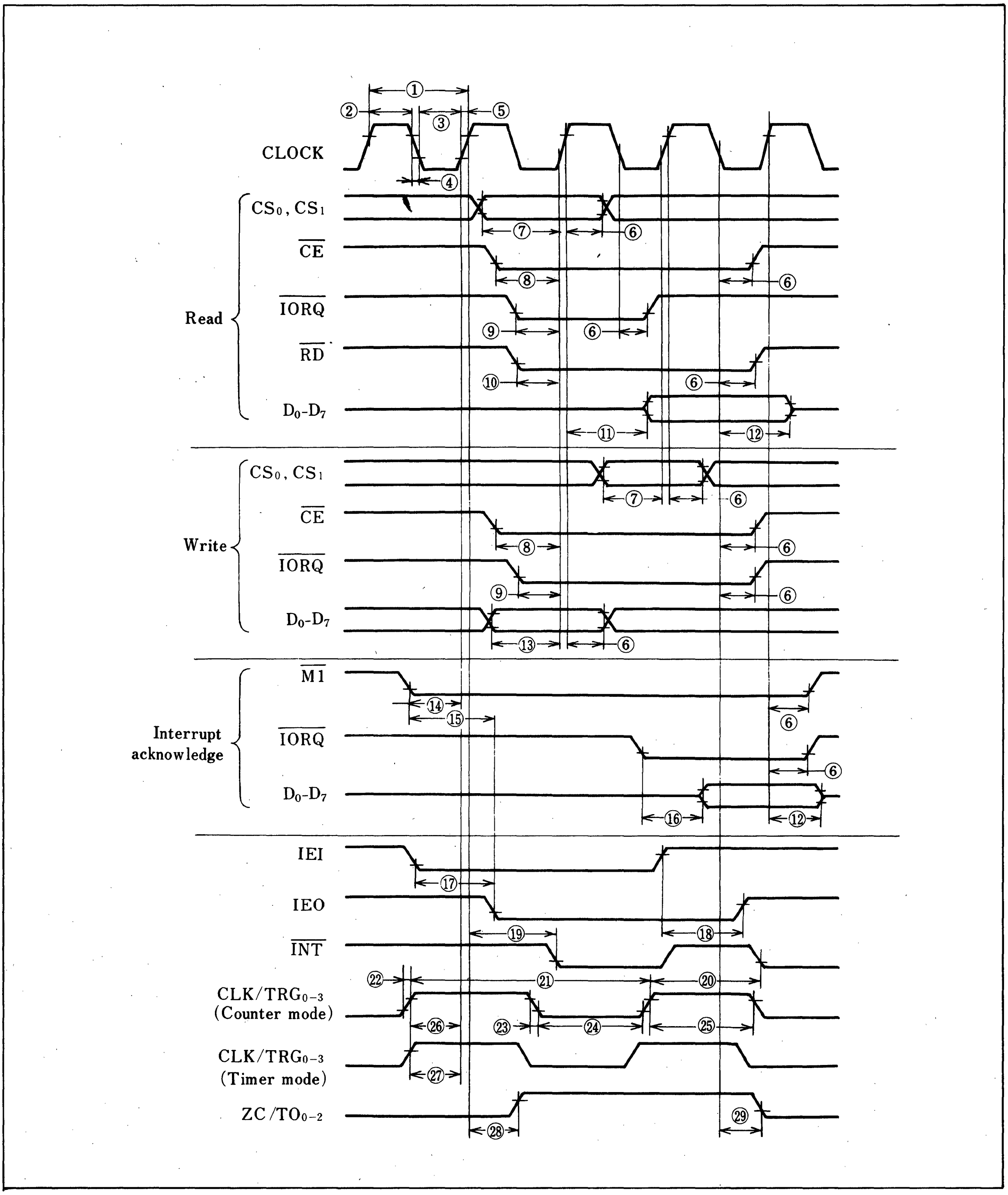
Note 2 : Increase delay by 10 ns for each 50 pF increase in loading, 200 pF maximum for data lines, and 100 pF for control lines.

Note 3 : Increase delay by 2 ns for each 10 pF increase in loading, 100 pF maximum.

Note 4 : Timer mode.

Note 5 : Counter mode.

AC Timing Chart



■ Programming

(1) Operation mode select

To select a channel operating mode, write a channel control word having bit 0 changed to 1 in the channel control register.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Interrupt enable	Mode	Prescaler value	CLK/TRO edge selection	Trigger mode	Time constant mode	Reset	1

D_3 and D_5 are used in timer mode only.

- ●Bit 7 = 0: Disables a channel interrupt.
- ●Bit 7 = 1: Enables a channel interrupt each time the down-counter counts down to zero. No interrupt is produced even with bit 7 as 1, after the counter has counted down to zero with bit 7 as 0.
- ●Bit 6 = 0: Selects the timer mode, having the prescaler output as the down-counter clock. The timer's period comes in t_C. P. TC. Where t_C represents system clock period, P has 16 or 256 (divisional scale by the prescaler), and TC means an 8-bit programmable time constant (max. 256).
- ●Bit 6 = 1: Selects the counter mode, having the external clock (CLK input) signal as the down-counter clock. The prescaler is not used.
- ●Bit 5 = 0: Used for the timer mode only. The prescaler divides the system clock into 16 sections.
- ●Bit 5 = 1: Used for the timer mode only. The prescaler divides the system clock into 256 sections.
- ●Bit 4 = 0: Starts the timer operation at the trigger input falling edge in the timer mode. In the counter mode, the down-counter comes on at the clock input rising edge.
- ●Bit 4 = 1: Starts the timer operation at the trigger input rising edge in the timer mode. In the counter mode, the down-counter comes on at the clock input rising edge.
- ●Bit 3 = 0: Effective in the timer mode only. With bit 1=1, the timer starts at the rising edge of the machine cycle T_2 which is next to the write cycle of a time constant. With bit 1=0, the timer starts at the rising edge of the machine cycle T_1 which is next to the write cycle of this control information.
- ●Bit 3 = 1: Effective in the timer mode only. The timer starts by an external trigger input that is given after the rising of the machine cycle T_2 next to the write cycle of a time constant. The operation starts at the second clock rising if the trigger input meets the set-up time, and at the third clock rising if it does not. If an external trigger input is given before writing a time constant the condition of bit 3 = 0 is caused.
- ●Bit 2 = 0: Indicates that there is no time constant written after the channel control word. This bit cannot be 0 for the channel control word to be immediately given when the channel is reset.
- ●Bit 2 = 1: Indicates that there is a time constant written after the channel control word. When a time constant is written during a down-counter operation, the new constant is set into the time constant register. But the counter keeps on counting. Once the counter counts zero, the new constant is available to use.
- ●Bit 1 = 0: The channel acts as a down-counter.
- ●Bit 1 = 1: Stops the operation as a down-counter. With bit 2 = 1, the operation restarts after a time contant is written. With bit 2 = 0, the channel does not act until a new control word is given.

(2) Time constant programming

An 8-bit time constant is written into the time constant register, following the channel control word with bit 2 = 1. "00" (hexadecimal) indicates the time constant 256.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0

(3) Interrupt vector programming

If the Z-80 CTC has one or more interrupts enabled, it can supply interrupt vectors to the Z-80 CPU. To do so, the Z-80 CTC must be pre-programmed with the most-significant five bits of the interrupt vector. Programming consists of writing a vector word to the I/O port corresponding to the Z-80 CTC Channel 0. Note that D_0 of the vector word is always zero, to distinguish the vector from a channel control word. D_1 and D_2 are not used in programming the vector word. These bits are supplied by the interrupt logic to identify the channel requesting interrupt service with a unique interrupt vector. Channel 0 has the highest priority.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

D_2	D_1	Channel
0	0	0
0	1	1
1	0	2
1	1	3

Timing

(1) Write cycle timing

Fig. 1 shows write cycle timing for loading control, time constant or vector words.

The CTC does not have a write signal input, so it generates one internally when the read ($\overline{RD}$) input is High during T_1. During T_2 $\overline{IORQ}$ and $\overline{CE}$ inputs are Low. $\overline{M1}$ must be High to distinguish a write cycle from an interrupt acknowledge. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be addressed, and the word being written is placed on the Z-80 data bus. The data word is latched into the appropriate register with the rising edge of clock cycle T_3.

(2) Read cycle timing

Fig. 2 shows read cycle timing. This cycle reads the contents of a down-counter without disturbing the count.

During clock cycle T_2, the Z-80 CPU initiates a read cycle by driving the following inputs Low: $\overline{RD}$, $\overline{IORQ}$, and $\overline{CE}$. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be read. $\overline{M1}$ must be High to distinguish this cycle from an interrupt acknowledge. No additional wait states are allowed.

(3) Interrupt acknowledge timing

Fig. 3 shows interrupt acknowledge timing. After an interrupt request, the Z-80 CPU sends an interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$). All channels are inhibited from changing their interrupt request status when $\overline{M1}$ is active—about two clock cycles earlier than $\overline{IORQ}$. $\overline{RD}$ is High to distinguish this cycle from an instruction fetch.

The CTC interrupt logic determines the highest priority channel requesting an interrupt. If the CTC interrupt enable input. (IEI) is High, the highest proiority interrupting channel within the CTC places its interrupt vector on the data bus when $\overline{IORQ}$ goes Low. Two wait states (T_{WA}) are automatically inserted at this time to allow the daisy chain to stabilize. Additional wait states may be added.

(4) Return from interrupt cycle

If a Z-80 peripheral has no interrupt pending and is not under service, then its IEO=IEI. If it has an interrupt under service (i.e.,it has already interrupted and received an interrupt acknowledge) then its IEO is always Low, inhibiting lower priority devices from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO is Low unless an "ED" is decoded as the first byte of a 2-byte opcode. In this case, IEO goes High until the next opcode byte is decoded, whereupon it goes Low again. If the second byte of the opcode

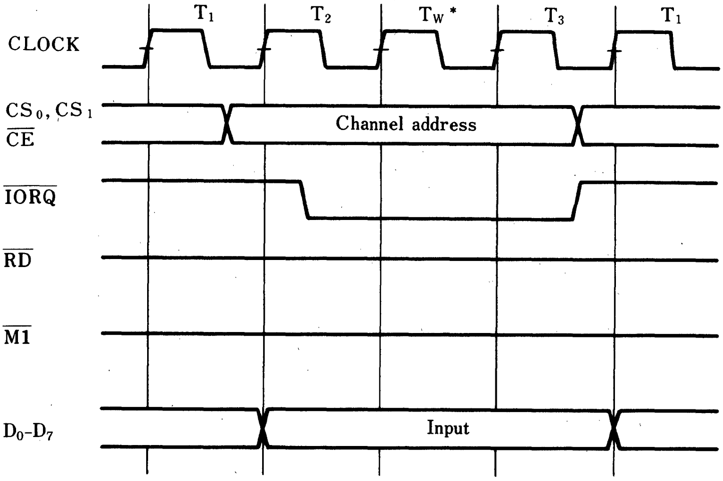


Fig. 1 Write cycle timing

was a "4D," then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service has its IEI High and its IEO Low. This device is the highest-priority device in the daisy chain that has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D", this peripheral device resets its "interrupt under service" condition.

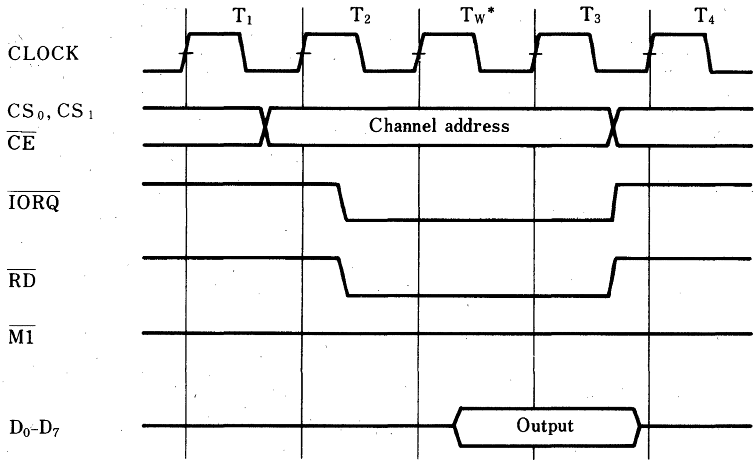


Fig. 2 Read cycle timing

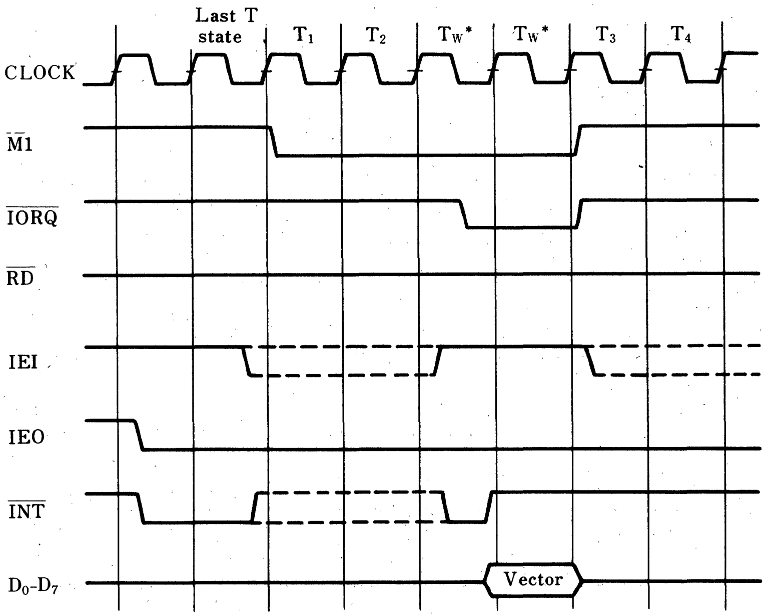


Fig. 3 Interrupt acknowledge timing

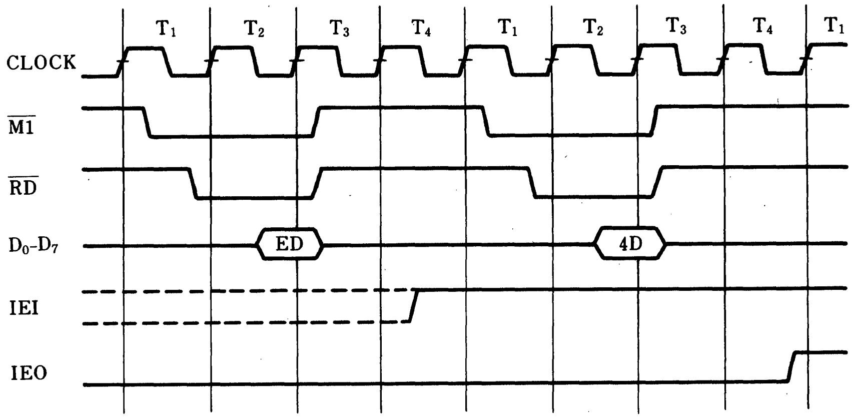
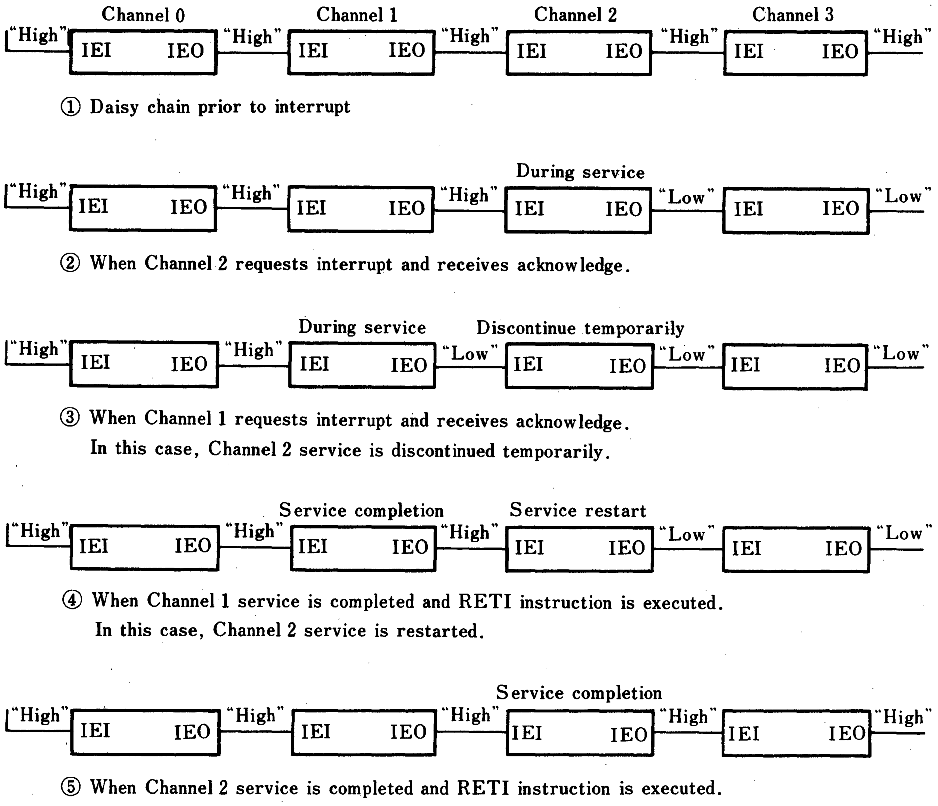


Fig. 4 Return from interrupt cycle



① Daisy chain prior to interrupt

② When Channel 2 requests interrupt and receives acknowledge.

③ When Channel 1 requests interrupt and receives acknowledge.
In this case, Channel 2 service is discontinued temporarily.

④ When Channel 1 service is completed and RETI instruction is executed.
In this case, Channel 2 service is restarted.

⑤ When Channel 2 service is completed and RETI instruction is executed.

Fig. 5 Daisy-chain interrupt service

(5) Daisy-chain interrupt service

Fig. 5 shows a typical nested interrupt order with the CTC. Channel 2 first requests an interrupt to be serviced. If the higher-priority Channel 1 requests an interrupt while Channel 2 is in service, the Channel 2 service is interrupted and Channel 1 is serviced instead. Now the Channel 1 service routine has been completely executed, an RETI instruction can be given to indicate that Channel 1 has been serviced. At this moment, Channel 2 will be in service again.

(6) Counter operation/timer operation

In the counter mode, the CLK/TRG pulse input decrements the down-counter. The trigger is asynchronous, but the count is synchronized with CLK. For the decrement to occur on the next rising edge of CLK, the trigger edge must precede CLK by a minimum lead time. In the timer mode, a CLK/TRG pulse input starts the timer on the second succeeding rising edge of CLK. The trigger pulse is asynchronous, and it must have a minimun width. A minimun lead time (210 ns) is required between the active edge of the CLK/TRG and the next rising edge of CLK to enable the prescaler on the following clock edge.

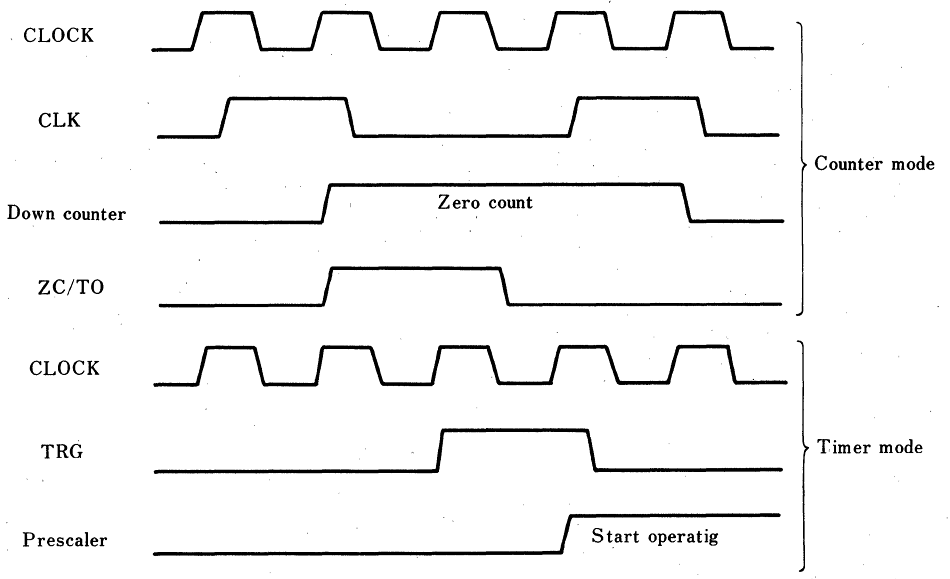


Fig. 6 Counter operation/timer operation

LH0083 Z80 DMA Direct Memory Access

Description

The LH0083 Z80 DMA (Z80 DMA for short below) is a powerful and versatile device for controlling and processing of data transfers. Its basic function of managing CPU-independent transfers between two ports is augmented by an array of features that optimize transfer speed and control with little or no external logic in systems using an 8-or 16-bit data bus and a 16-bit address bus.

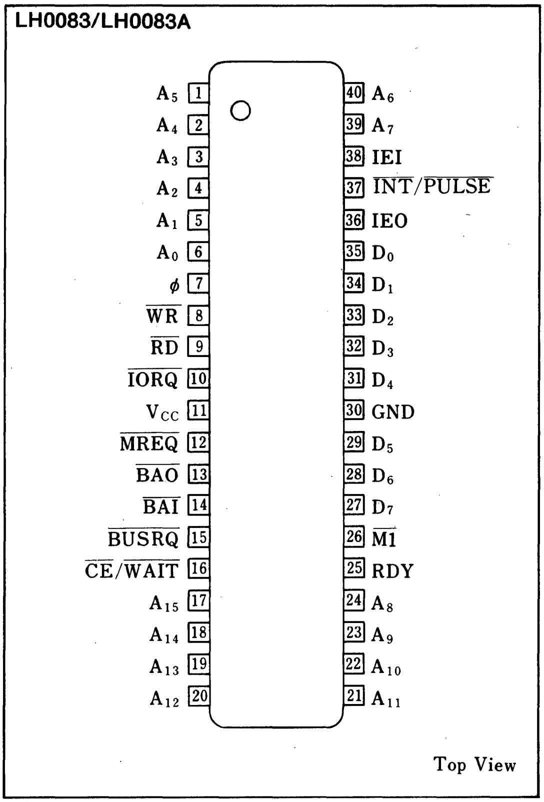
Transfers can be done between any two ports (source and destination), including memory-to-I/O, memory-to-memory, and I/O-to-I/O. Dual port addresses are automatically generated for each transaction and may be either fixed or incrementing/decrementing. In addition, bitmaskable byte searches can be performed either concurrently with transfers or as an operation in itself.

The LH0083A Z80A DMA is a high speed version which can operate at the 4MHz system clock.

Features

1. Transfers, searches and search/transfers in byte-at-a-time, burst or continuous modes
2. Cycle length and edge timing can be programmed
3. Dual port addresses generated for memory-to-I/O, memory-to-memory, or I/O-to-I/O operations Adderess may be fixed or automatically incremented/decremented
4. Next-operation loading without disturbing current operations via buffered starting-address registers and an entire previous sequence can be repeated automatically
5. Extensive programmability of functions CPU can read complete channel status
6. Vectored daisy chain priority interrupt logic
7. Single +5V power supply and single phase clock
8. TTL compatible inputs and outputs
9. N-channel silicon-gate process
10. 40-pin DIP (DIP40-P-600)

Pin Connections

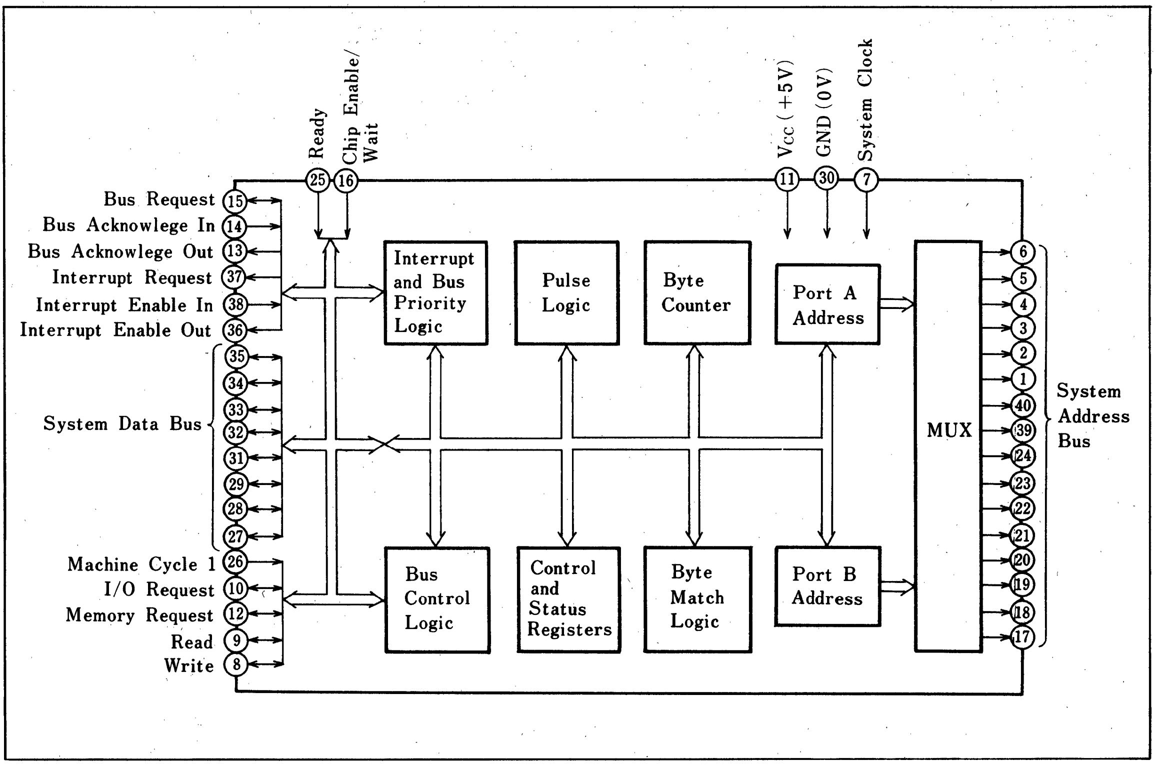


5

Ordering Information

LH0083 X
└ Clock frequency
Blank: 2.5MHz
A: 4MHz
└ Model No.

Block Diagram



Pin Description

Pin	Meaning	I/O	Function
A_0-A_{15}	Address bus	3-state O	System address bus.
D_0-D_7	Data bus	Bidirectional 3-state	System data bus.
$\overline{\text{BAI}}$	Bus acknowledge in	I	Active "low". Used to form a bus priority-interrupt daisy-chain.
$\overline{\text{BAO}}$	Bus acknowledge out	O	Active "low". Used to form a bus priority-interrupt daisy-chain.
$\overline{\text{BUSRQ}}$	Bus request	Open drain, O	Active "low". Active when controlling the bus.
$\overline{\text{CE}}/\overline{\text{WAIT}}$	Chip enable	I	Active "low". Acts as $\overline{\text{CE}}$ when the CPU accesses the DAM, and as $\overline{\text{WAIT}}$ when the DAM is the bus master.
CLOCK	System clock	I	Standard Z80 system clock used for internal synchronization signals.
$\overline{\text{M1}}$	Machine cycle one	I	Active "low". Indicates that CPU is acknowledging an interrupt, when both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active.
$\overline{\text{IORQ}}$	Input/output request	Bidirectional 3-state	Active "low". Transmits and receives data from the CPU as an input line. Acts as $\overline{\text{IORQ}}$ for another device as an output line. Indicates that the CPU is acknowledging an interrupt, when both $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are active.
$\overline{\text{MREQ}}$	Memory request	3-state O	Active "low". Requests a transfer from or to memory with the DMA as a bus master.
IEI	Interrupt enable in	I	Active "high". Used to form a priority-interrupt daisy-chain.
IEO	Interrupt enable out	O	Active "low". Used to form a priority-interrupt daisy-chain.
$\overline{\text{INT}}/\overline{\text{PULSE}}$	Interrupt request/pulse	Open drain, O	Active "low". Active when requesting an interrupt. Can also generate pulses.
$\overline{\text{RD}}$	Read	Bidirectional 3-state	Active "low". Reads data from the CPU as an input line. Acts as $\overline{\text{RD}}$ for another device as an output line.
$\overline{\text{WR}}$	Write	Bidirectional 3-state	Active "low". Writes data from the CPU as an input line. Acts as $\overline{\text{WR}}$ for another device as an output line.
RDY	Ready	I	With the DMA as a bus master, starts DMA operation when active, and stops it when not active.

■ Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Input voltage	V_{IN}	−0.3 to +7.0	V
Output voltage	V_{OUT}	−0.3 to +7.0	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃

■ DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock input low voltage		V_{ILC}		−0.3		0.45	V
Clock input high voltage		V_{IHC}		$V_{CC}-0.6$		5.5	V
Input low voltage		V_{IL}		−0.3		0.8	V
Input high voltage		V_{IH}		2.0		5.5	V
Output low voltage		V_{OL}	I_{OL}=3.2mA for $\overline{\text{BUSREQ}}$ I_{OL}=2.0 mA for all others			0.4	V
Output high voltage		V_{OH}	I_{OH}=−250μA	2.4			V
Current consumption	LH0083	I_{CC}	t_c=400ns			150	mA
	LH0083A		t_c=250ns			200	mA
Input leakage current		$\mid I_{L1} \mid$	$0 \leqq V_{IN} \leqq V_{CC}$			10	μA
3-state output leakage current		$\mid I_{LOH} \mid$	V_{OUT}=2.4V			10	μA
3-state output leakage current		$\mid I_{LOL} \mid$	V_{OUT}=0.4V			10	μA
Data bus leakage current in input mode		$\mid I_{LD} \mid$	$0V \leqq V_{IN} \leqq V_{CC}$			10	μA

■ Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			35	pF
Input capacitance	C_{IN}				5	pF
Output capacitance	C_{OUT}				10	pF

■ AC Characteristics

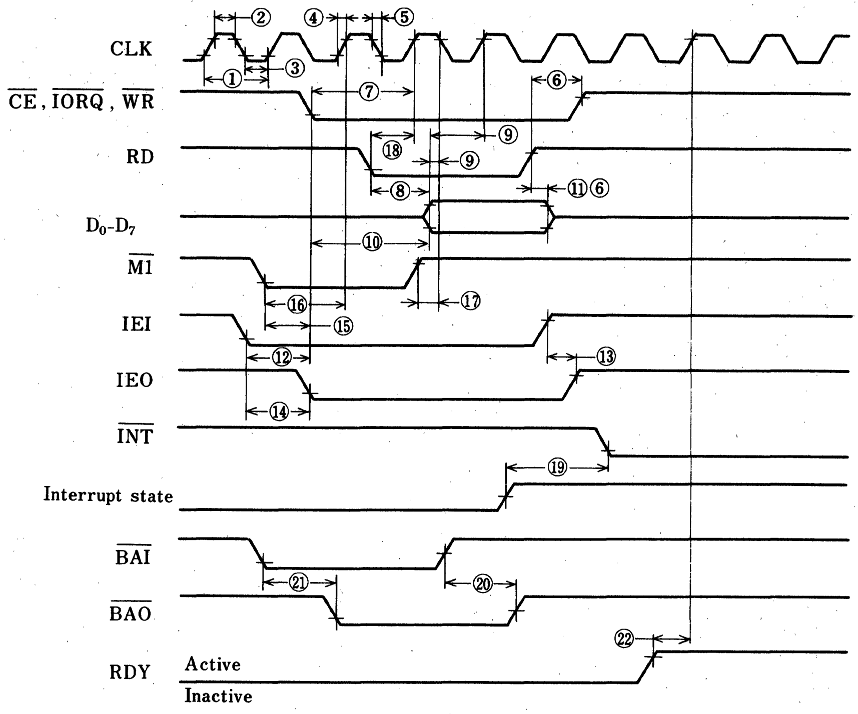
(1) Acting as CPU peripheral (inactive state)

(V_{CC}=5V±5%, Ta=0 to +70℃)

No.	Parameter	Symbol	LH0083		LH0083A		Unit
			MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400	4000	250	4000	ns
2	Clock width (high)	TwCh	170	2000	110	2000	ns
3	Clock width (low)	TwCl	170	2000	110	2000	ns
4	Clock rise time	TrC		30		30	ns
5	Clock fall time	TfC		30		30	ns
6	Hold time for any specified setup time	Th	0		0		ns
7	$\overline{\text{IORQ}}$, $\overline{\text{WR}}$, $\overline{\text{CE}}$ ↓ to clock ↑ setup time	TsC(Cr)	280		145		ns
8	$\overline{\text{RD}}$ ↓ to data output delay	TdDO(RDf)		500		380	ns
9	Data in to clock ↑ setup ($\overline{\text{WR}}$ or $\overline{\text{M1}}$)	TsWM(Cr)	50		50		ns
10	$\overline{\text{IORQ}}$ ↓ to data out delay (INTA cycle)	TdCf(DO)		340		160	ns
11	$\overline{\text{RD}}$ ↑ to data float delay (output buffer disable)	TdRD(DZ)		160		110	ns
12	IEI ↓ to $\overline{\text{IORQ}}$ ↓ setup (INTA cycle)	TsIEI(IORQ)	140		140		ns
13	IEI ↑ to IEO ↑ delay	TdIEOr(IEIr)		210		160	ns
14	IEI ↓ to IEO ↓ delay	TdIEOf(IEIf)		190		130	ns
15	$\overline{\text{M1}}$ ↓ to IEO ↓ delay (interrupt just prior to $\overline{\text{M1}}$ ↓)	TdM1(IEO)		300		190	ns
16	$\overline{\text{M1}}$ ↓ to clock ↑ setup	TsMlf(Cr)	210		90		ns
17	$\overline{\text{M1}}$ ↑ to clock ↓ setup	TsMlr(Cf)	20		−10		ns
18	$\overline{\text{RD}}$ ↓ to clock ↑ setup ($\overline{\text{M1}}$ cycle)	TsRD(Cr)	240		115		ns
19	Interrupt cause to $\overline{\text{INT}}$ ↓ delay ($\overline{\text{INT}}$ generated only when DMA is inactive)	TdI(INT)		500		500	ns
20	$\overline{\text{BAI}}$ ↑ to $\overline{\text{BAO}}$ ↑ delay	TdBAIr(BAOr)		200		150	ns
21	$\overline{\text{BAI}}$ ↓ to $\overline{\text{BAO}}$ ↓ delay	TdBAIf(BAOf)		200		150	ns
22	RDY active to clock ↑ setup time	TsRDY(Cr)	150		100		ns

Note : ↑ Rising edge, ↓ Falling edge.

Note 1: Negative minimum setup values mean that the first-mentioned event can come after the second-mentioned event.



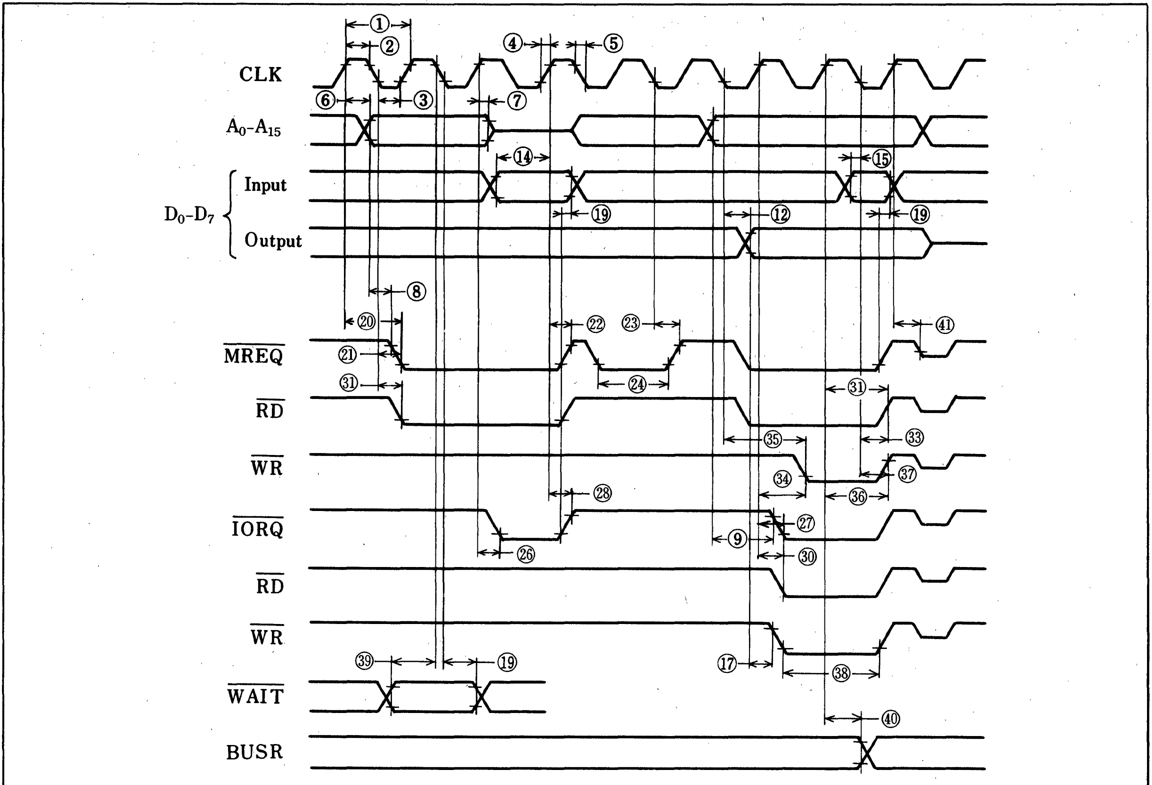
Acting as CPU peripheral (Inactive state)

(2) Acting as bus controller (active state)

(V_{CC}=5V±5%, Ta=0 to +70℃)

No.	Parameter	Symbol	LH0083		LH0083A		Unit
			MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400		250		ns
2	Clock width (high)	TwCh	180	2000	110	2000	ns
3	Clock width (low)	TwCl	180	2000	110	2000	ns
4	Clock rise time	TrC		30		30	ns
5	Clock fall time	TfC		30		30	ns
6	Address output delay	TdA		145		110	ns
7	Clock ↑ to address float delay	TdC(AZ)		110		90	ns
8	Address to MREQ ↓ setup (memory cycle)	TsA(MREQ)	(2)+(5)−75		(2)+(5)−75		ns
9	Address stable to IORQ, RD, WR ↓ setup (I/O cycle)	TsA(IRW)	(1)−80		(1)−70		ns
*10	RD, WR ↑ to addr. stable delay	TdRW(A)	(3)+(4)−40		(3)+(4)−50		ns
*11	RD, WR ↑ to addr. float delay	TdRW(AZ)	(3)+(4)−60		(3)+(4)−45		ns
12	Clock ↓ to data out delay	TdCf(DO)		230		150	ns
*13	Clock ↑ to data float delay (write cycle)	TdCr(Dz)		90		90	ns
14	Data in to clock ↑ setup (read cycle when rising edge ends read)	TsDI(Cf)	50		35		ns
15	Data in to clock ↓ setup (read cycle when falling edge ends read)	TsDO(WfM)	60		50		ns
*16	Data out to WR ↓ setup (memory clcle)	TsDO(WPI)	(1)−210		(1)−170		ns
17	Data out to WR ↓ setup (I/O cycle)	TsDO(WPI)	100		100		ns
*18	WR ↑ to data out hold time	TdWr(DO)	(3)+(4)−80		(3)+(4)−70		ns
19	Hold time for any specified setup time	Th	0		0		ns
20	Clock ↑ to MREQ ↓ delay	TdCr (Mf)		100		85	ns
21	Clock ↓ to MREQ ↓ delay	TdCf(Mf)		100		85	ns
22	Clock ↑ to MREQ ↑ delay	TdCf(Mr)		100		85	ns
23	Clock ↓ to MREQ ↑ delay	TdCf(Mr)		100		85	ns
24	MREQ low pulse width	TwMl	(1)−40		(1)−30		ns
*25	MREQ high pulse width	TwMh	(2)+(5)−30		(2)+(3)−20		ns
26	Clock ↓ to MREQ ↓ delay	TdCf(Mf)		110		85	ns
27	Clock ↑ to IORQ ↓ delay	TdCr(If)		90		75	ns
28	Clock ↑ to IORQ ↑ delay	TdCr(Ir)		100		85	ns
*29	Clock ↓ to IORQ ↑ delay	TdCr(Ir)		110		85	ns
30	Clock ↑ to RD ↓ delay	TdCr(Rf)		100		85	ns
31	Clock ↓ to RD ↓ delay	TdCr(Rf)		130		95	ns
32	Clock ↑ to RD ↑ delay	TdCr(Rr)		100		85	ns
33	Clock ↓ to RD ↑ delay	TdCr(Rr)		110		85	ns
34	Clock ↑ to WR ↓ delay	TdCr(Wf)		80		65	ns
35	Clock ↓ to WR ↓ delay	TdCf(Wf)		90		80	ns
36	Clock ↑ to WR ↑ delay	TdCr(Wr)		100		80	ns
37	Clock ↓ to WR ↑ delay	TdCf(Wr)		100		80	ns
38	WR Low pulse width	TwWl	(1)−40		(1)−30		ns
39	WAIT to clock ↓ setup	TsWA(Cf)	70		70		ns
40	Clock ↑ to BUSREQ delay	TdCr(B)		150		100	ns
41	Clock ↑ to IORQ, MREQ, RD, WR float delay	TdCr(Iz)		100		80	ns

Note : ↑ Rising edge, ↓ Falling edge
Note 1: Numbers in parentheses are other parameter numbers in this table; their values should be substituted in equations.
Note 2: All equations imply DMA default (standard) timing.
Note 3: Data must be enabled onto data bus when RD is active.
Note 4: Asterisk (*) before parameter number means the parameter is not illustrated in the AC Timing Diagrams.



Acting as bus controller (active state)

Programming

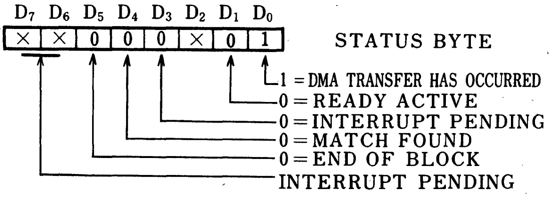
The Z-80 DMA has two programmable fundamental states.

- an enabled state, in which it can gain control of the system buses and direct the transfer of data between ports, and
- a disabled state, in which it can initiate neither bus requests nor data transfers. When the DMA is powered up or reset by any means, it is automatically placed into the disabled state.

(1) Reading

The Read Registers (RR0-RR6) are read by the CPU by addressing the DMA as an I/O port using an Input instruction. The readable bytes contain DMA status, byte-counter values, and port addresses since the last DMA reset. The registers are always read in a fixed sequence beginning with RR0 and ending with RR6.

- **Read Register 0**



- **Read Register 1**

BYTE COUNTER (LOW BYTE)

- **Read Register 2**

BYTE COUNTER (HIGH BYTE)

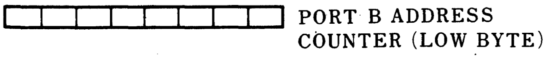
- **Read Register 3**

PORT A ADDRESS COUNTER (LOW BYTE)

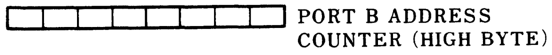
- **Read Register 4**

PORT A ADDRESS COUNTER (HIGH BYTE)

- **Read Register 5**



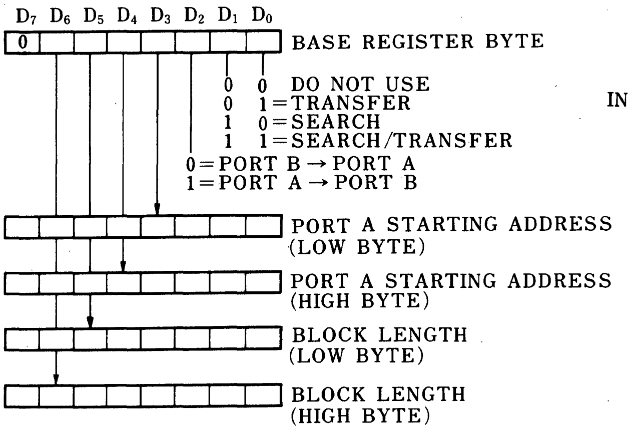
- **Read Register 6**



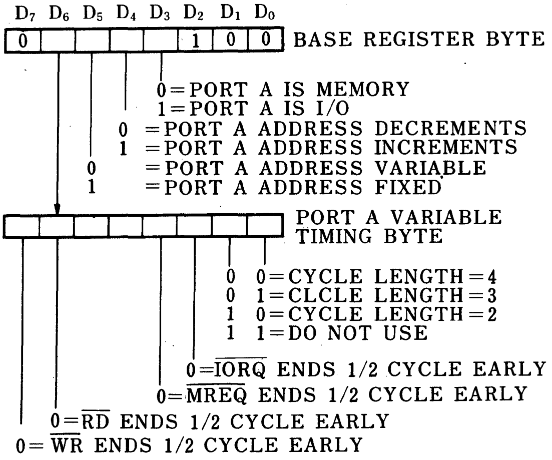
(2) **Writing**

Control or command bytes are written into one or more of the Write Register groups (WR0-WR6) by first writing to the base register byte in that group. All groups have base registers and most groups have additional associated registers. The associated registers in a group are sequentially accessed by first writing a byte to the base register containing register-group identification and pointer bits (I, s) to one or more of that base register's associated registers.

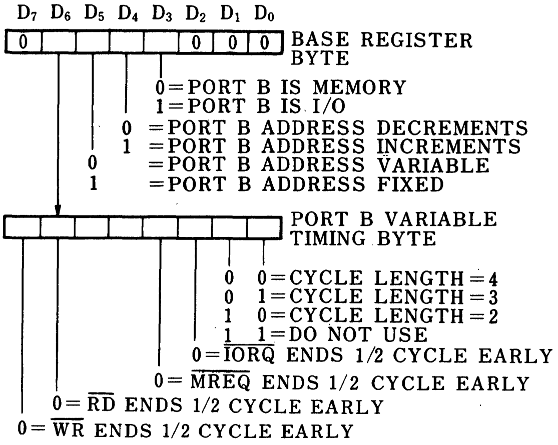
- **Write Register 0**



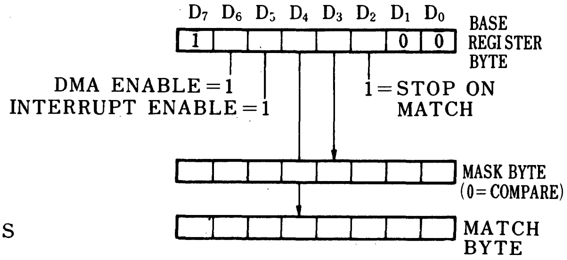
- **Write Register 1**


D7 D6 D5 D4 D3 D2 D1 D0
0 1 0 0
BASE REGISTER BYTE
0=PORT A IS MEMORY
1=PORT A IS I/O
0 =PORT A ADDRESS DECREMENTS
1 =PORT A ADDRESS INCREMENTS
0 =PORT A ADDRESS VARIABLE
1 =PORT A ADDRESS FIXED
PORT A VARIABLE TIMING BYTE
0 0=CYCLE LENGTH=4
0 1=CLCLE LENGTH=3
1 0=CYCLE LENGTH=2
1 1=DO NOT USE
0=IORQ ENDS 1/2 CYCLE EARLY
0=MREQ ENDS 1/2 CYCLE EARLY
0=RD ENDS 1/2 CYCLE EARLY
0=WR ENDS 1/2 CYCLE EARLY

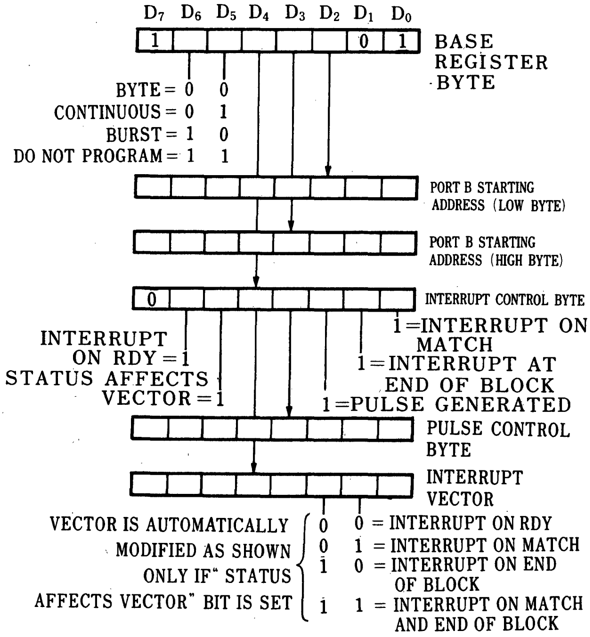
- **Write Register 2**


D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0
BASE REGISTER BYTE
0=PORT B IS MEMORY
1=PORT B IS I/O
0 =PORT B ADDRESS DECREMENTS
1 =PORT B ADDRESS INCREMENTS
0 =PORT B ADDRESS VARIABLE
1 =PORT B ADDRESS FIXED
PORT B VARIABLE TIMING BYTE
0 0=CYCLE LENGTH=4
0 1=CYCLE LENGTH=3
1 0=CYCLE LENGTH=2
1 1=DO NOT USE
0=IORQ ENDS 1/2 CYCLE EARLY
0=MREQ ENDS 1/2 CYCLE EARLY
0=RD ENDS 1/2 CYCLE EARLY
0=WR ENDS 1/2 CYCLE EARLY

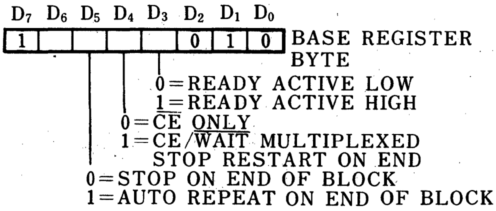
- **Write Register 3**



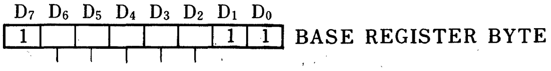
- **Write Register 4**



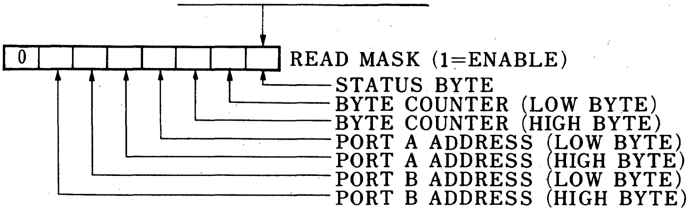
- **Write Register 5**



- **Write Register 6**



C3	1	0	0	0	0	INTERRUPT LINE RESET, INTERRUPT REQUEST AND BUS REQUEST DISABLE, INTERNAL READY STATE CLEAR, CE MULTIPLEX DISABLE, AUTOMATIC REPEAT STOP
C7	1	0	0	0	1	PORT A TIMING TO Z80 STANDARD TIMING
CB	1	0	0	1	0	PORT B TIMING TO Z80 STANDARD TIMING
CF	1	0	0	1	1	BOTH PORTS START ADDRESS LOAD, BYTE CONTER CLEAR
D3	1	0	1	0	0	ADDRESS CONTINUE FROM CURRENT VALUE, BYTE COUNTER CLEAR
AB	0	1	0	1	0	INTERRUPT ENABLE
AF	0	1	0	1	1	INTERRUPT DISABLE
A3	0	1	0	0	0	INTERRUPT CIRCUIT RESET AND DISABLE (SAME AS RETI), INTERNAL READY STATE CLEAR
87	0	0	0	0	1	DMA ENABLE } EFFECTIVE FOR ALL SECTIONS BUT INTERRUPT.
83	0	0	0	0	0	DMA DISABLE } NOT ALL FUNCTIONS RESETTABLE, HOWEVER
A7	0	1	0	0	1	READ SEQUENCE START FOR 1ST REGISTER DESIGNATED BY READ MASTER REGISTER
BF	0	1	1	1	1	STATUS REGISTER READ SETUP. FROM STATUS REGISTER FOR NEXT READ
B3	0	1	1	0	0	INTERNAL READY STATE TO BE FORCEDLY CLEAR OF "RDY" PIN (USED FOR DMA BETWEEN MEMORIES NEEDING NO RDY SIGNAL. NOT OPERATIVE IN "BYTE MODE")
88	0	0	0	1	0	REINITIALIZE. END-OF-BLOCK BIT CLEAR
B7	0	1	1	0	1	ENABLE AFTER RETI. BUS REQUEST ONLY AFTER RETI EXECUTION
BB	0	1	1	1	0	READ MASK FOLLOWS


0
READ MASK (1=ENABLE)
STATUS BYTE
BYTE COUNTER (LOW BYTE)
BYTE COUNTER (HIGH BYTE)
PORT A ADDRESS (LOW BYTE)
PORT A ADDRESS (HIGH BYTE)
PORT B ADDRESS (LOW BYTE)
PORT B ADDRESS (HIGH BYTE)

Timing

(1) Inactive state timing (DMA as CPU Peripheral)

In its disabled or inactive state, the DMA is addressed by the CPU as an I/O peripheral for write and read (control and status) operations. Write timing is illustrated in Fig. 1.

Reading of the DMA's status byte, byte counter or port address counters is illustrated in Fig. 2.

(2) Active state timing (DMA as BUS Controller)

(i) Default read and write cycles

By default, and after reset, the DMA's timing of read and write operations is exactly the same as the Z-80 CPU's timing of read and write cycles for memory and I/O peripherals, with one exception: during a read cycle, data is latched on the falling edge of T_3 and held on the data bus across the boundary between read and write cycles, through the end of the following write cycle.

Fig. 3 illustrates the timing for memory to-I/O port transfers and Fig. 4 illustrates I/O-to-memory transfers. Memory-to-memory and I/O-to-I/O transfer timings are simply permutations of these diagrams.

The default timing uses three T-cycles for memory transactions and four T-cycles for I/O transactions, which include one automatically inserted wait cycle between T_2 and T_3. If the $\overline{\text{CE}}/\overline{\text{WAIT}}$ line is programmed to act as a $\overline{\text{WAIT}}$ line during the DMA's active state, it is sampled on the falling edge of T_2 for memory transactions and the falling edge of T_W for I/O transactions. If $\overline{\text{CE}}/\overline{\text{WAIT}}$ is Low during this time another T-cycle is added, during which the $\overline{\text{CE}}/\overline{\text{WAIT}}$ line will again be sampled. The duration of transactions can thus be indefinitely extended.

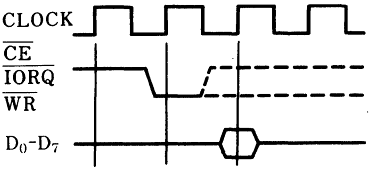


Fig. 1 CPU-DMA write cycle timing

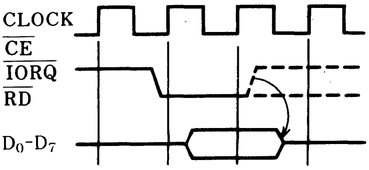


Fig. 2 CPU-DMA read cycle timing

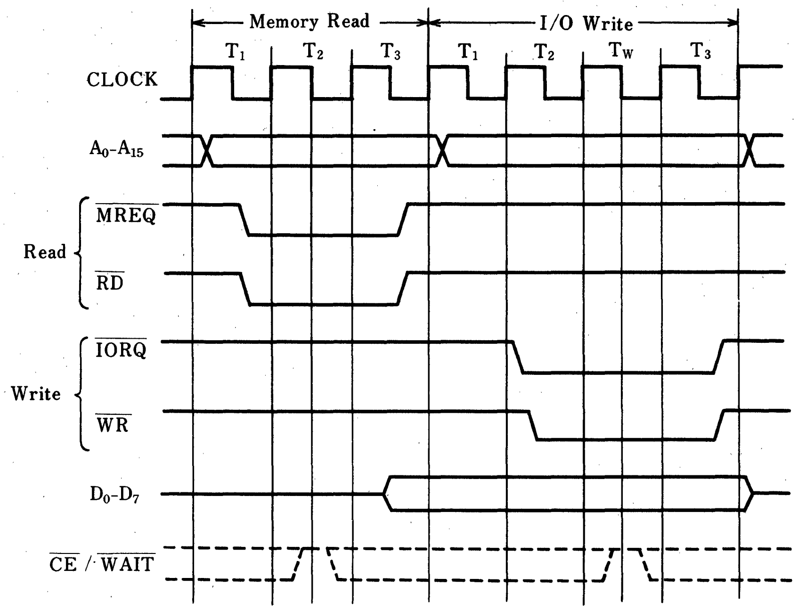


Fig. 3 Transfer from memory to I/O device

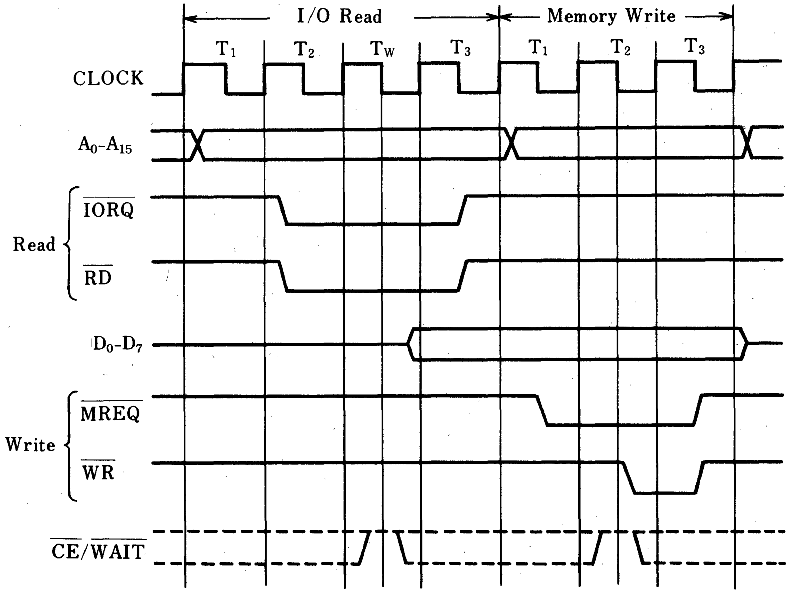


Fig. 4 Transfer from I/O device to memory

(ii) **Variable cycle and edge timing** The Z-80 DMA's default operation-cycle length for the source (read) port and destination (write) port can be independently programmed. This variable-cycle feature allows read or write cycles consisting of two, three or four T-cycles (more if Wait cycles are inserted), thereby increasing or decreasing the speed of all signals generated by the DMA. In addition, the trailing edges of the $\overline{\text{IORQ}}$, $\overline{\text{MREQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals can be independently terminated one-half cycle early. Fig. 5 illustrates this.

In the variable-cycle mode, unlike default timing, $\overline{\text{IORQ}}$ comes active one-half cycle before $\overline{\text{MREQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$. $\overline{\text{CE/WAIT}}$ can be used to extend only the 3 or 4 T-cycle variable memory cycles and only the 4-cycle variable I/O cycle. The $\overline{\text{CE/WAIT}}$ line is sampled at the falling edge of T_2 for 3-or 4-cycle memory cycles, and at the falling edge of T_3 for 4-cycle I/O cycles.

During transfers, data is latched on the clock edge causing the rising edge of $\overline{\text{RD}}$ and held through the end of the write cycle.

(iii) **Bus requests** Fig. 6 illustrates the bus request and acceptance timing. The RDY line, which may be programmed active High or Low, is sampled on every rising edge of CLK.
If it is found to be active, and if the bus is not in use by any other device, the following rising edge of CLK drives $\overline{\text{BUSREQ}}$ low. After receiving $\overline{\text{BUSREQ}}$ the CPU acknowledges on the $\overline{\text{BAI}}$ input either directly or through a multiple-DMA daisy chain. When a Low is detected on $\overline{\text{BAI}}$ for two consecutive rising edges of CLK, the DMA will begin transferring data on the next rising edge of CLK.

(iv) **Bus release byte-at-a-time** In Byte at a Time mode, $\overline{\text{BUSREQ}}$ is brought High on the rising edge of CLK prior to the end of each read cycle (search-only) or write cycle (transfer and transfer/search) as illustrated in Fig. 7. This is done regardless of the state of RDY.
The next bus request for the next byte will come after both $\overline{\text{BUSREQ}}$ and $\overline{\text{BAI}}$ have returned High.

(v) **Bus release at end of block** In Burst and Continuous modes, an end of block causes $\overline{\text{BUSREQ}}$ to go High usually on the same rising edge of CLK in which the DMA completes the transfer of the data block (Fig. 8). The last byte in the block is transferred even if RDY goes inactive before completion of the last byte transfer.

(vi) **Bus release on not ready** In Burst mode, when RDY goes inactive it causes $\overline{\text{BUSREQ}}$ to go High on the next rising edge of CLK after the completion of its current byte operation (Fig. 9). The action on $\overline{\text{BUSREQ}}$ is thus somewhat delayed from action on the RDY line. The DMA always completes its current byte operation in an orderly fashion before releasing the bus.

By contrast, $\overline{\text{BUSREQ}}$ is not released in Continuous mode when RDY goes inactive.
Instead, the DMA idles after completing the current byte operation, awaiting an active RDY again.

(vii) **Bus release on match** If the DMA is programmed to stop on match in Burst or Continuous modes, a match causes $\overline{\text{BUSREQ}}$ to go inactive on the next DMA operation, i.e., at the end of the next read in a search or at the end of the following write in a transfer (Fig. 10). Due to the pipelining scheme, matches are determined while the next DMA read or write is being performed.

The RDY line can go inactive after the matching operation begins without affecting this bus-release timing.

(viii) **Interrupts** Timings for interrupt acknowledge and return from interrupt are the same as timings for these in other Z-80 peripherals. (Refer to the Z80 PIO.)

Interrupt on RDY (interrupt before requesting bus) does not directly affect the $\overline{\text{BUSREQ}}$ line. Instead, the interrupt service routine must handle this by issuing the following commands.

a. Enable after return from interrupt (RETI) (Command code 87$_{\text{H}}$)

b. Enable DMA (Command code 87$_{\text{H}}$)

c. An RETI instruction

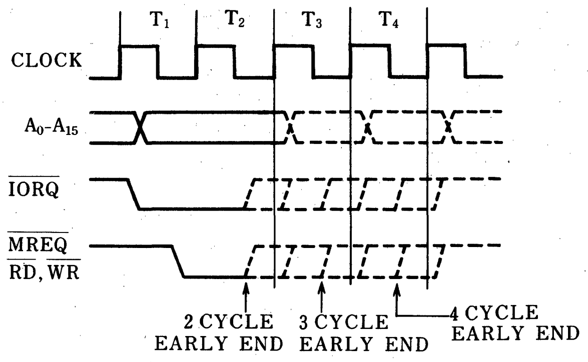


Fig. 5 Variable cycle and edge timing

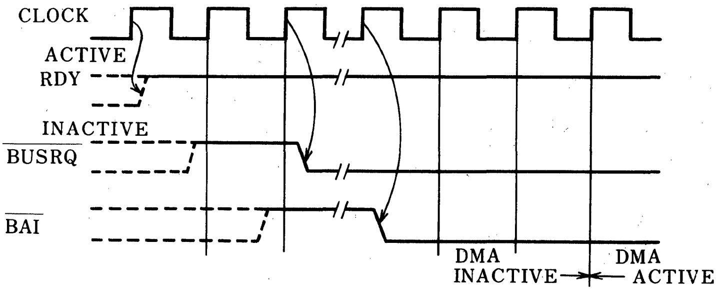


Fig. 6 Bus request and acknowledgement

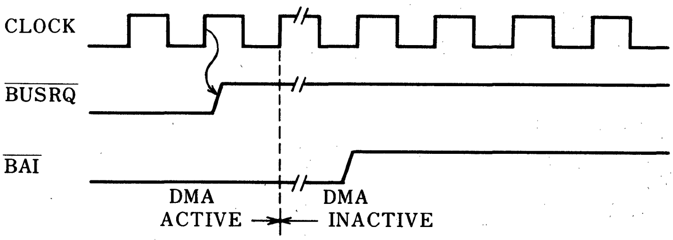


Fig. 7 Bus clear (byte mode)

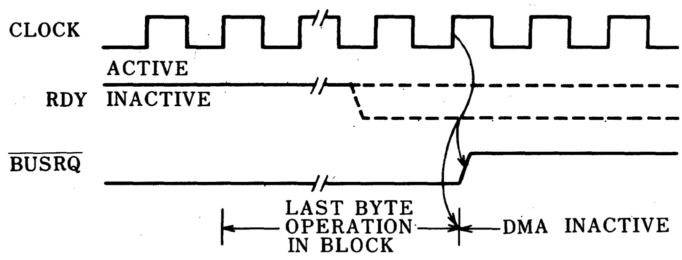


Fig. 8 End of block bus clear (burst, continuous mode)

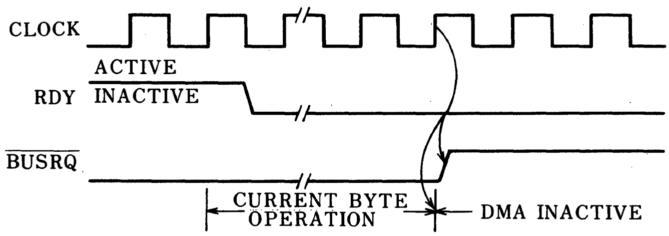


Fig. 9 No READY bus clear (burst mode)

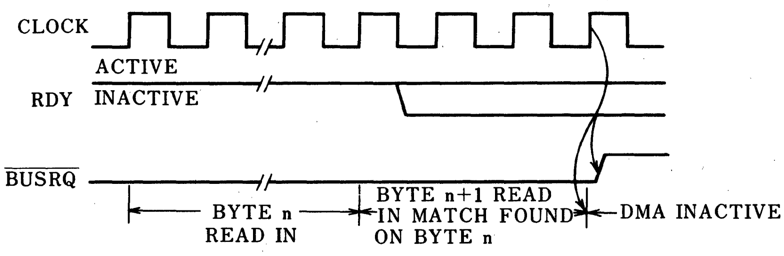


Fig. 10 Mating bus clear (burst, continuous mode)

LH0084/LH0085 LH0086/LH0087

Z80 SIO Serial Input/Output Controller

■ Description

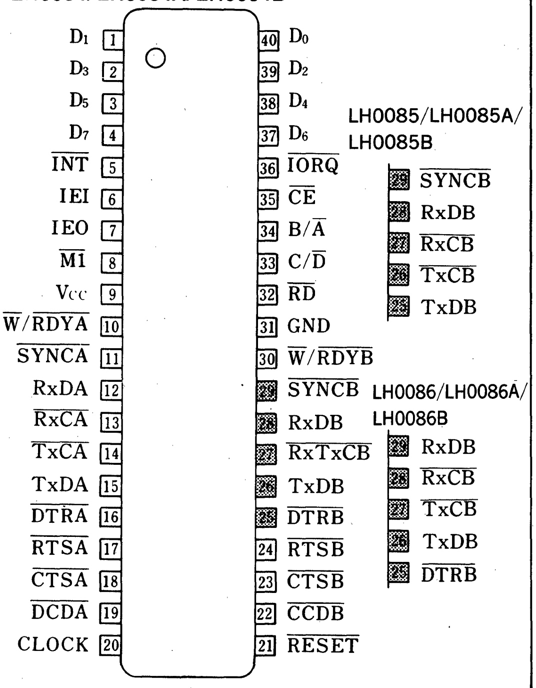
The LH0084/85/86/87, Z80 SIO (Z80 SIO for short below) is a dual-channel multi-function peripheral component designed to satisfy a wide variety of serial data communications requirements in microcomputer systems. Its basic function is a serial-to-parallel, parallel-to-serial converter/ controller, but–within that role–it is configurable by systems software so its "personality" can be optimized for a given serial data communications application.

The Z80 SIO is capable of handling asynchronous and synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device can also be used to support virtually any other serial protocol for applications other than data communications (cassette or floppy disk interfaces, for example).

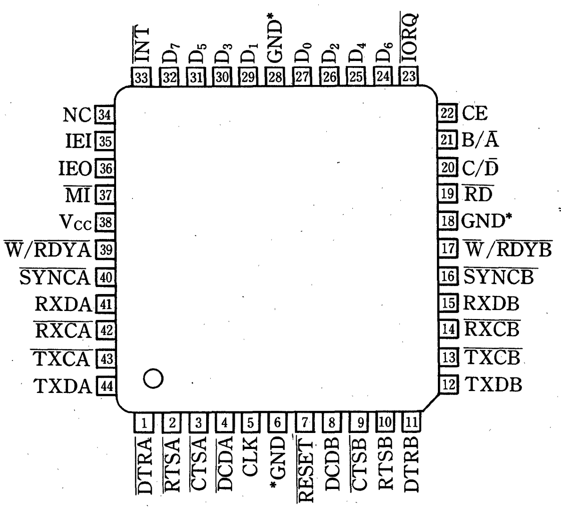
The Z80 SIO can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The device also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for

■ Pin Connections

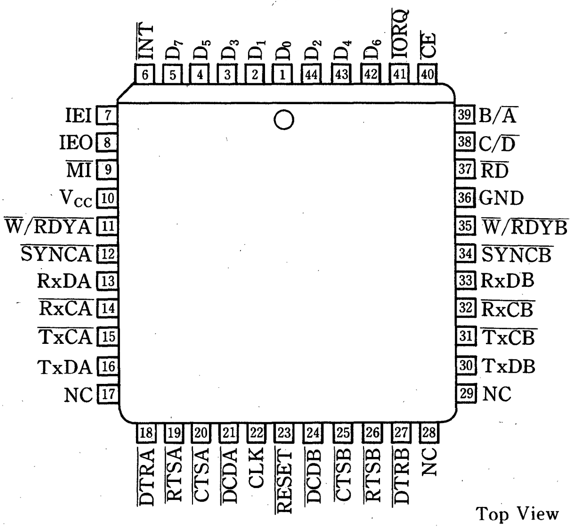
LH0084/LH0084A/LH0084B



LH0087M/LH0087AM



LH0087U/LH0087AU/LH0087BU



Top View

∗The GND pins must be connected to the GND level.

general-purpose I/O.

The Z80 SIO has six types as below according it's system clock and bonding option. The Z80A SIO and the Z80B SIO are a high speed version which can operate at the 4MHz and 6MHz system clock, respectively.

- LH0084 Z80 SIO/0
- LH0085 Z80 SIO/1
- LH0086 Z80 SIO/2
- LH0087 Z80 SIO
- LH0084A Z80A SIO/0
- LH0085A Z80A SIO/1
- LH0086A Z80A SIO/2
- LH0087A Z80A SIO
- LH0084B Z80B SIO/0
- LH0085B Z80B SIO/1
- LH0086B Z80B SIO/2
- LH0087B Z80B SIO

■ Features

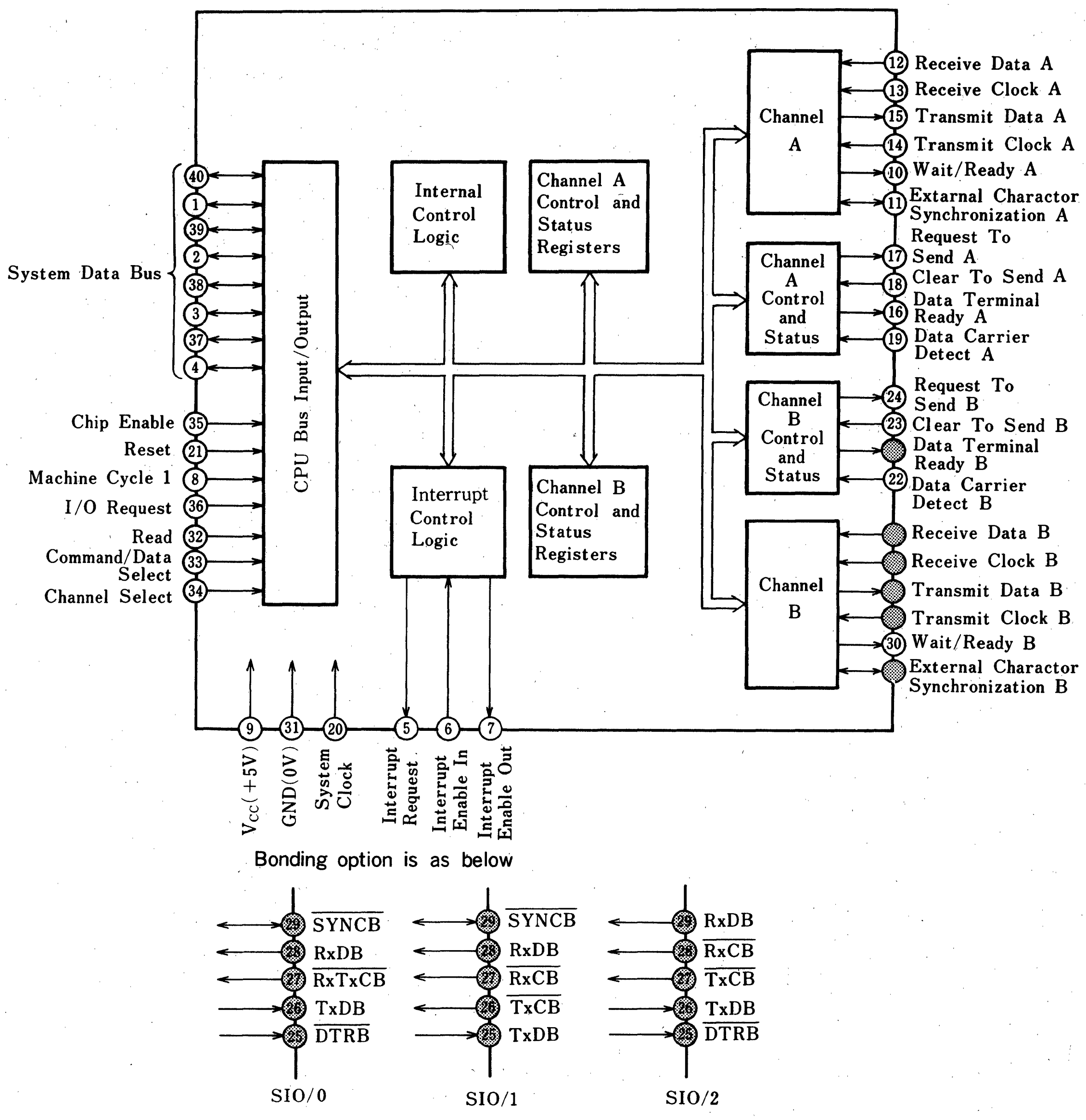
1. N-channel silicon-gate process
2. Single +5V power supply and single phase clock
3. Two independent full duplex channels
4. Data rates : 0 to 500K bits/second (at 2.5 MHz system clock)
   : 0 to 800K bits/second (at 4MHz system clock)
   : 0 to 1200K bits/second (at 6MHz system clock)
5. Asynchronous operation
   - 5, 6, 7 or 8 bits/character
   - 1, 1½ or 2 stop bits/character
   - Even, odd or no parity
   - ×1, ×16, ×32 and ×64 clock modes
   - Break generation and detection
   - Parity, Overrun and Framing error detection
6. Binary synchronous operation
   - Internal or external character synchronization
   - One or two Sync characters in separate registers
   - Automatic Sync character insertion
   - CRC generation and checking
7. HDLC or IBM SDLC operation
   - Abort sequence generation and detection
   - Automatic zero insertion and detection
   - Automatic flag insertion
   - Address field recognition
   - I-field residue handling
   - Valid receive messages protected from overrun
   - CRC generation and checking
8. Vectored daisy chain priority interrupt logic
9. CRC-16 or CRC-CCITT block check
10. Separate modem control inputs and outputs for both channels
11. Modem status can be monitored
12. 40-pin DIP (DIP40-P-600)
    44-pin QFP (QFP44-P-1010A)
    44-pin QFJ (QFJ44-P-S650)

■ Ordering Information

Product	Z80 SIO	Z80A SIO	Z80B SIO	Package	Operating temperature
Clock frequency	2.5MHz	4MHz	6MHz		
Model No.	LH008X	LH008XA	LH008XB	40-pin DIP	0°C to +70°C
	LH008XH	LH008XAH			−20°C to+85°C
	LH0087M	LH0087AM		44-pin QFP	0°C to +60°C
	LH0087U	LH0087AU	LH0087BU	44-pin QFJ	0°C to+70°C

X: It is the bonding option to select one of SIO/0, SIO/1 and SIO/2 on 40-pin DIP.
X=4: SIO/0
X=5: SIO/1
X=6: SIO/2
H: H affix indicates a wide temperature spec, packaged in 40-pin DIP.

Block Diagram



Pin numbers apply to the 40-pin DIP.

Pin Description

Pin	Meaning	I/O	Function
D_0-D_7	Data bus	Bidirectional 3-state	System data bus
B/$\overline{A}$	Channel A or B select	I	Defines which channel is accessed. Channel B at "High", channel A at "Low".
C/$\overline{D}$	Control or data select	I	Defines the type of information transfer on the data bus. Control word at "High", data at "Low".
$\overline{CE}$	Chip enable	I	Active "Low". A Low enables the CPU to transmit and receive control words and data.
CLOCK	System clock	I	Standard Z80 system clock used for internal synchronization signals.
$\overline{M1}$	Machine cycle one	I	Active "Low". Indicates that the CPU is acknowledging an interrupt, when both $\overline{M1}$ and $\overline{IORQ}$ are active.
$\overline{IORQ}$	Input/output request	I	Active "Low". Read operation when $\overline{RD}$ is active, and write operation when it is not active. Indicates that the CPU is acknowledging an interrupt, when both $\overline{IORQ}$ and $\overline{M1}$ are active.
$\overline{RD}$	Read cycle status	I	Active "Low". Read operation when active.
$\overline{RESET}$	Reset	I	Active "Low". Resets the interrupt bits.
IEI	Interrupt enable in	I	Active "High". Used to form a priority-interrupt daisy chain.
IEO	Interrupt enable out	O	Active "High". Used to form a priority-interrupt daisy chain.
$\overline{INT}$	Interrupt request	Open drain, O	Active "Low". Active when requesting an interrupt.
W/$\overline{RDYA}$ W/$\overline{RDYB}$	Wait/ready	Open drain, O	Active "Low". $\overline{READY}$ when the DMA is a bus master, $\overline{WAIT}$ when the CPU is a bus master.
$\overline{CTSA}$, $\overline{CTSB}$	Clear to send	I	Active "Low". Enables the respective transmitters. Also applicable as general-purpose input pins.
$\overline{DCDA}$, $\overline{DCDB}$	Data carrier detect	I	Active "Low". Enables the respective receivers. Also applicable as general-purpose input pins.
RxDA, RxDB	Receive data	I	Active "Low". Data line for receiving
TxDA, TxDB	Transmit data	O	Active "Low". Data line for transmitting.
$\overline{RxCA}$, $\overline{RxCB}$	Receiver clock	I	Active "Low". Receiving synchronization clock.
$\overline{TxCA}$, $\overline{TxCB}$	Transmitter clock	I	Active "Low". Transmitting synchronization clock.
$\overline{RTSA}$, $\overline{RTSB}$	Request to send	O	Active "Low". Indicates that the transmitter is empty during transfer. Also applicable as general-purpose output pins.
$\overline{DTRA}$, $\overline{DTRB}$	Data terminal ready	O	Active "Low". Also applicable as general-purpose output pins.
$\overline{SYNCA}$, $\overline{SYNCB}$	External character synchronization	I	Active "Low". Acts the same way as $\overline{CTS}$ and $\overline{DCD}$ in the asynchronous mode. Driven "Low" in the synchronous mode when a synchronizing pattern is achieved.

■ Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	−0.3 to +7.0	V	
Output voltage	V_{OUT}	−0.3 to +7.0	V	
Operating temperature	Topr	0 to +70	℃	1
		0 to +60		2
		−20 to +85		3
Storage temperature	Tstg	−65 to +150	℃	

Note 1: Specified for 40-pin DIP and 44-pin QFJ
Note 2: Specified for 44-pin QFP
Note 3: Specified for wide temperature type

■ DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock input low voltage	V_{ILC}		−0.3		0.45	V
Clock input high voltage	V_{IEC}		Vcc−0.6		5.5	V
Input low voltage	V_{IL}		−0.3		0.8	V
Input high voltage	V_{IH}		2.0		5.5	V
Output low voltage	V_{OL}	I_{OL}=2.0mA			0.4	V
Output high voltage	V_{OH}	I_{OH}=−250μA	2.4			V
Input leakage current	$\vert I_{LI}\vert$	$0<V_{IN}<V_{CC}$			10	μA
3-state output/data bus input leakage curreut	$\vert I_Z\vert$	$0<V_{IN}<V_{CC}$			10	μA
$\overline{\text{SYNC}}$ pin leakage current	$I_{L(SY)}$	$0<V_{IN}<V_{CC}$	−40		10	μA
Current consumption	I_{CC}				100	mA

Note 1: Ta=0 to +60℃ for 44-pin QFP, Ta=−20 to +85℃ for wide temperature types.

■ Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			40	pF
Input capacitance	C_{IN}				5	pF
Output capacitance	C_{OUT}				10	pF

■ AC Characteristics

(1) AC characteristics (I)

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

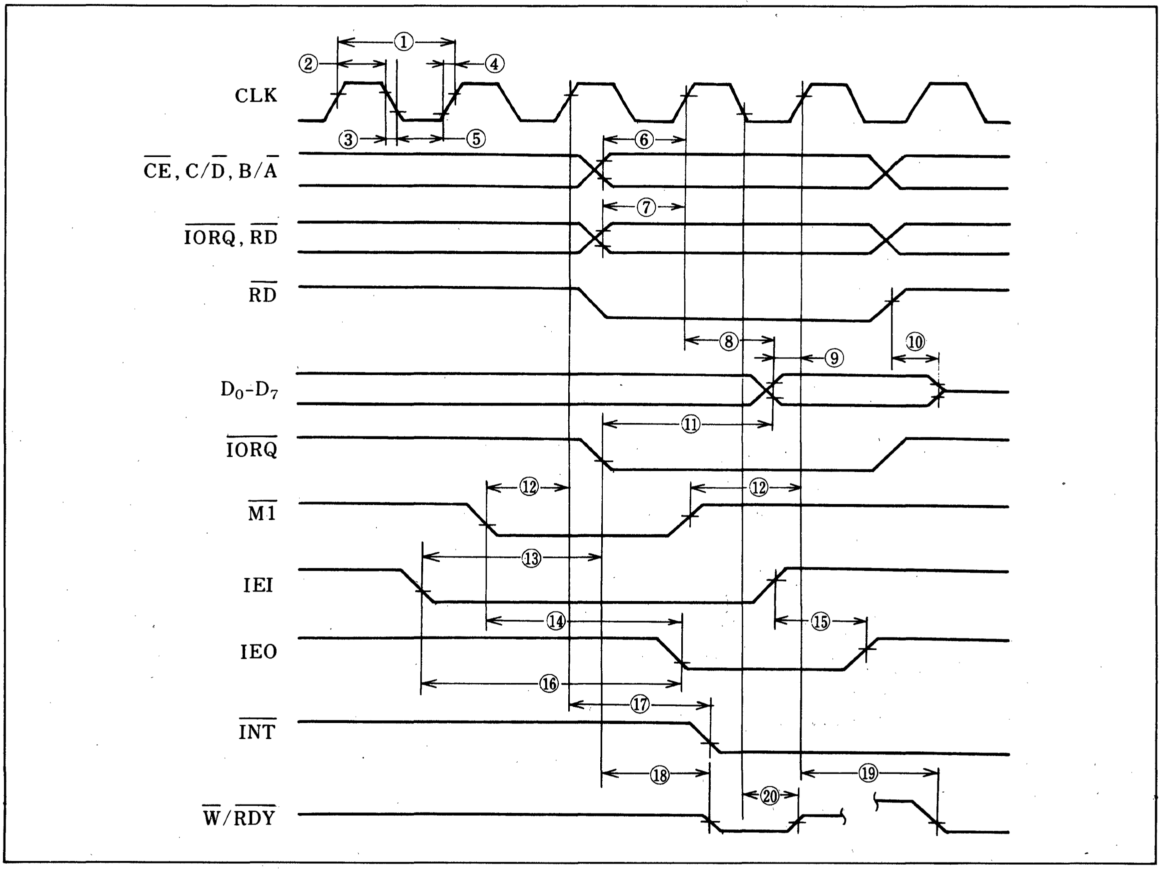
No.	Parameter	Symbol	LH0084/5/6/7		LH0084A/5A/6A/7A		LH0084B/5B/6B/7B		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400	4000	250	4000	165	4000	ns
2	Clock high width	TwCh	170	2000	105	2000	70	2000	ns
3	Clock fall time	TfC		30		30		15	ns
4	Clock rise time	TrC		30		30		15	ns
5	Clock low width	TwCl	170	2000	105	2000	70	2000	ns
6	$\overline{\text{CE}}$, C/$\overline{\text{D}}$, B/$\overline{\text{A}}$ to clock ↑ setup time	TsAD(C)	160		145		60		ns
7	$\overline{\text{IORQ}}$, $\overline{\text{RD}}$ to clock ↑ setup time	TsCS(C)	240		115		60		ns
8	Clock ↑ to data out delay	TdC(DO)		240		220		150	ns
9	Data in to clock ↑ setup (Write or $\overline{\text{M1}}$ cycle)	TsDI(C)	50		50		30		ns
10	$\overline{\text{RD}}$ ↑ to data out float delay	TdRD(DOz)		230		110		90	ns
11	$\overline{\text{IORQ}}$ ↓ to data out delay (INTACK cycle)	TdIO(DOI)		340		160		100	ns

No.	Parameter	Symbol	LH0084/5/6/7		LH0084A/5A/6A/7A		LH0084B/5B/6B/7B		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
12	$\overline{\mathrm{M1}}$ to clock ↑ setup time	TsM1(C)	210		90		75		ns
13	IEI to $\overline{\mathrm{IORQ}}$ ↓ setup time (INTACK cycle)	TsIEI(IO)	200		140		120		ns
14	$\overline{\mathrm{M1}}$ ↓ to IEO ↓ delay (interrupt before $\overline{\mathrm{M1}}$)	TdM1(IEO)		300		190		160	ns
15	IEI ↑ to IEO ↑ delay (after ED decode)	TdIEI(IEOr)		150		100		70	ns
16	IEI ↓ to $\overline{\mathrm{INT}}$ ↓ delay	TdIEI(IEOf)		150		100		70	ns
17	Clock ↑ to $\overline{\mathrm{INT}}$ ↓ delay	TdC(INT)		200		200		150	ns
18	$\overline{\mathrm{IORQ}}$ ↓ or $\overline{\mathrm{CE}}$ ↓ to$\overline{\mathrm{W/RDY}}$ ↓ delay (wait mode)	TdIO(W/RWf)		300		210		175	ns
19	Clock ↑ to $\overline{\mathrm{W}}$/$\overline{\mathrm{RDY}}$ ↓ delay (ready mode)	TdC(W/PR)		120		120		100	ns
20	Clock ↓ to $\overline{\mathrm{W}}$/$\overline{\mathrm{RDY}}$ float delay (wait mode)	TdC(W/RWz)		150		130		110	ns
21	Any unspecified hold when setup is specified	Th	0		0		0		ns

↑ Rising edge, ↓ Falling edge.

Note 1: Ta=0 to +60℃ for 44-pin QFP
Ta=−20 to +85℃ for wide temperature types

(2) AC timing chart (I)



CLK
$\overline{\mathrm{CE}}$, C/$\overline{\mathrm{D}}$, B/$\overline{\mathrm{A}}$
$\overline{\mathrm{IORQ}}$, $\overline{\mathrm{RD}}$
$\overline{\mathrm{RD}}$
D_0-D_7
$\overline{\mathrm{IORQ}}$
$\overline{\mathrm{M1}}$
IEI
IEO
$\overline{\mathrm{INT}}$
W/$\overline{\mathrm{RDY}}$
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑫
⑬
⑭
⑮
⑯
⑰
⑱
⑲
⑳

(3) AC characteristics (Ⅱ)

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

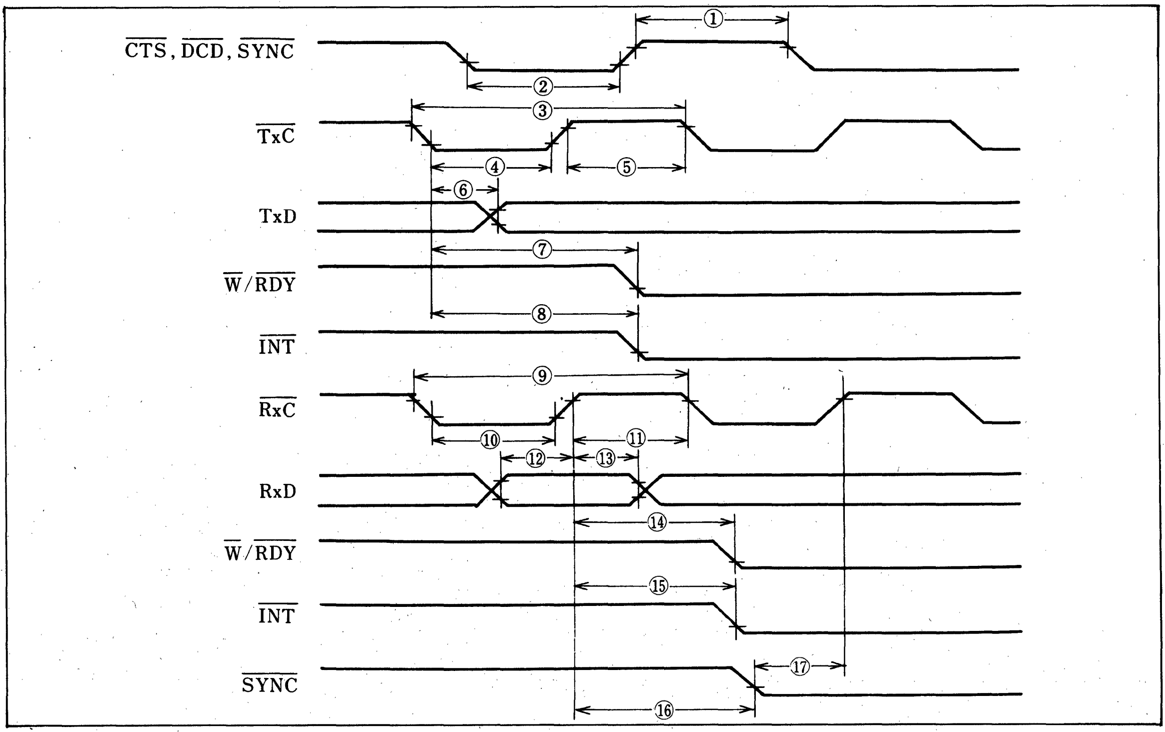
No.	Parameter	Symbol	LH0084/5/6/7		LH0084A/5A/6A/7A		LH0084B/5B/6B/7B		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
1	Pulse high width	TwPh	200		200		200		ns
2	Pulse low width	TwPl	200		200		200		ns
3	$\overline{\mathrm{TxC}}$ clock time	TcTxC	400	∞	400	∞	330	∞	ns
4	$\overline{\mathrm{TxC}}$ low width	TwTxCl	180	∞	180	∞	100	∞	ns
5	$\overline{\mathrm{TxC}}$ high width	TwTxCh	180	∞	180	∞	100	∞	ns
6	$\overline{\mathrm{TxC}}$ ↓ to TxD delay (xl mode)	TdTxC(TxD)		400		300		220	ns
7	$\overline{\mathrm{TxC}}$ ↓ to $\overline{\mathrm{W}}/\overline{\mathrm{RDY}}$ ↓ delay (ready mode)	TdTxC(W/RRf)	5	9	5	9	5	9	Clock period
8	$\overline{\mathrm{TxC}}$ ↓ to $\overline{\mathrm{INT}}$ ↓ delay	TdTxC(INT)	5	9	5	9	5	9	Clock period
9	$\overline{\mathrm{RxC}}$ cycle time	TdRxC	400	∞	400	∞	330	∞	ns
10	$\overline{\mathrm{RxC}}$ low width	TwRxCl	180	∞	180	∞	100	∞	ns
11	$\overline{\mathrm{RxC}}$ high width	TwRxCh	180	∞	180	∞	100	∞	ns
12	$\overline{\mathrm{RxD}}$ to $\overline{\mathrm{RxC}}$ ↑ setup time (xl mode)	TsRxD(RxC)	0		0		0		ns
13	RxC ↑ to RxD hold time (xl mode)	ThRxD(RxC)	140		140		100		ns
14	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{W}}/\overline{\mathrm{RDY}}$ ↓ delay (ready mode)	TdRxC(W/RRf)	10	13	10	13	10	13	Clock period
15	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{INT}}$ ↓ delay	TdRxC(INT)	10	13	10	13	10	13	Clock period
16	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{SYNC}}$ ↓ delay (output modes)	TdRxC(SYNC)	4	7	4	7	4	7	Clock period
17	$\overline{\mathrm{SYNC}}$ ↓ to $\overline{\mathrm{RxC}}$ ↓ setup (external sync modes)	TsSYNC(RxC)	−100		−100		100		ns

↑ Rising edge, ↓ Falling edge

Note 1: Ta=0 to +60℃ for 44-pin QFP
Ta=−20 to +85℃ for wide temperature types

Note 2: In all mode, the System Clock rate must be at least five times the maximum data rate.

(4) AC timing chart (Ⅱ)



$\overline{\mathrm{CTS}}$, $\overline{\mathrm{DCD}}$, $\overline{\mathrm{SYNC}}$
$\overline{\mathrm{TxC}}$
TxD
$\overline{\mathrm{W}}/\overline{\mathrm{RDY}}$
$\overline{\mathrm{INT}}$
$\overline{\mathrm{RxC}}$
RxD
$\overline{\mathrm{W}}/\overline{\mathrm{RDY}}$
$\overline{\mathrm{INT}}$
$\overline{\mathrm{SYNC}}$
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
⑭
⑮
⑯
⑰

■ Transmit and Receive Data Path

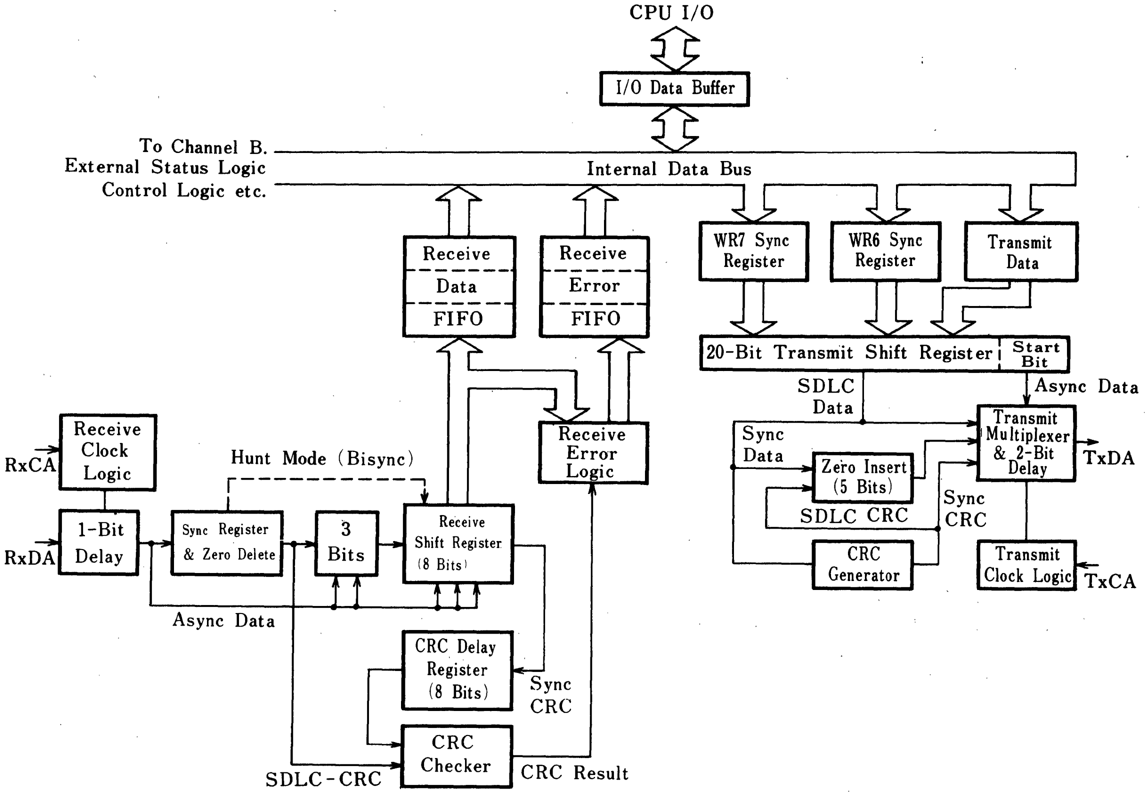


Fig. 1 Transmit and receive data path

■ Programming

The system program first issues a series of commands that initialize the basic mode of operation and then other commands that qualify conditions within the selected mode.

Both channels contain registers that must be programmed via the system program prior to operation.

(1) Read Registers

The SIO contains three read registers for Channel B and three read registers for Channel A (RR0-RR2) that can be read to obtain the status information. The status information includes error conditions, interrupt vector and standard communications-interface signals

● Read Register 0 (RR 0)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Break /abort	Tx under-run /EOM	CTS	Sync /hunt	DCD	Tx buffer empty	INT pending (ch.A only)	Rx character available

● Read Register 1 (RR 1)

The RR1 contains the status bits for specific receiving coditions as well as the one-field fraction codes for the SDLC receive mode.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
End of frame	CRC/ framing error	Rx overrun error	Parity error	Fraction code 2	Fraction code 1	Fraction code 0	All sent

● Read Register 2 (RR 2)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

$\underbrace{V_3 \; V_2 \; V_1 \; V_0}_{}$ Valiable if "status affects vector" is programmed

(2) Write Registers

The SIO contains eight write registers for Channel B and eight write registers for Channel A (WR0-WR7) that are programmed separately to configure the functional personality of the channels.

●Write Register 0 (WR 0)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CRC reset code 1	CRC reset code 0	Command bit 2	Command bit 1	Command bit 0	Pointer bit 2	Pointer bit 1	Pointer bit 0
		Control words			Register pointers		

●Write Register 1 (WR 1)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Wait/ ready enable	Wait/ ready function	Wait/ ready onR/T	Receive inter-rupt mode 1	Receive inter-rupt mode 0	Status affects vector	Tx INT enable	Ext INT enable

●Write Register 2 (WR 2)

The WR2 contains the interrupt vector for both channels and is only in the Channel B. When the status affected vector (WR1, D_2) is 1, the vector from the SIO during the interrupt acknowledge cycle varies (V_3 - V_1) depending on the interrupt conditions. The WR2 contents do not vary then.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

●Write Register 3 (WR 3)

The WR 3 contains the bits and parameters to control the receivers.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Rx bits char-acter 1	Rx bits char-acter 0	Auto enable	Enter hunt phase	Rx CRC enable	Address search mode	Sync charac-ter load inhibit	Rx enable

●Write Register 4 (WR 4)

The WR4 has the bits control both receivers and transmitters.

In initializing for transmitting and receiving, these bits must be set up before the WR1, WR3, WR5, WR6, and WR7.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Clock rate 1	Clock rate 0	Sync mode 1	Sync mode 0	Stop bit 1	Stop bit 0	Parity Even /$\overline{\text{ODD}}$	Parity enable

●Write Register 5 (WR 5)

The WR5 contains the bits (except for D_2) to control the transmitters.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
DTR	Tx bits /char-acter 1	Tx bits /char-acter 0	Send break	Tx enable	CRC16 /$\overline{\text{SDLC}}$	RTS	Tx CRC enable

●Write Register 6 (WR 6)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SYNC 7	SYNC 6	SYNC 5	SYNC 4	SYNC 3	SYNC 2	SYNC 1	SYNC 0

●Write Register 7 (WR 7)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SYNC 15	SYNC 14	SYNC 13	SYNC 12	SYNC 11	SYNC 10	SYNC 9	SYNC 8

Timing

(1) Read cycle

The timing signals generated by a Z-80 CPU input instruction to read a data or status byte from the SIO are illustrated in Fig. 2.

(2) Write cycle

Fig. 3 illustrates the timing and data signals generated by a Z-80 CPU output instruction to write a data or control byte into the SIO.

(3) Interrupt cycle

The interrupt-acknowledging and return-from-interrupt cycles are of the same timing as for other Z80 peripherals. (Refer to the Z80 PIO.)

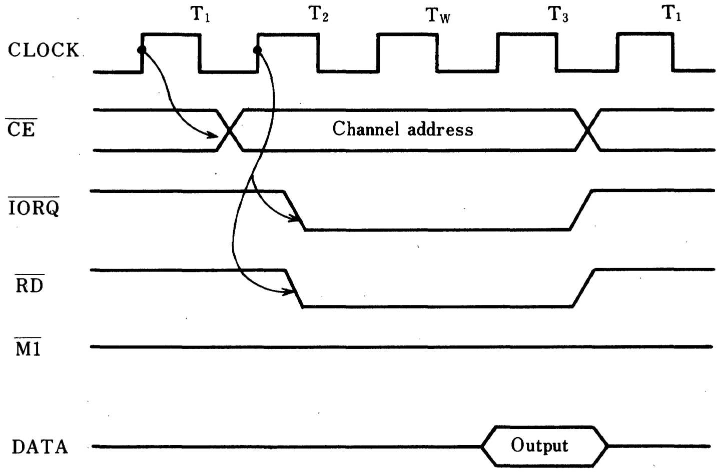


Fig. 2 Read cycle timing

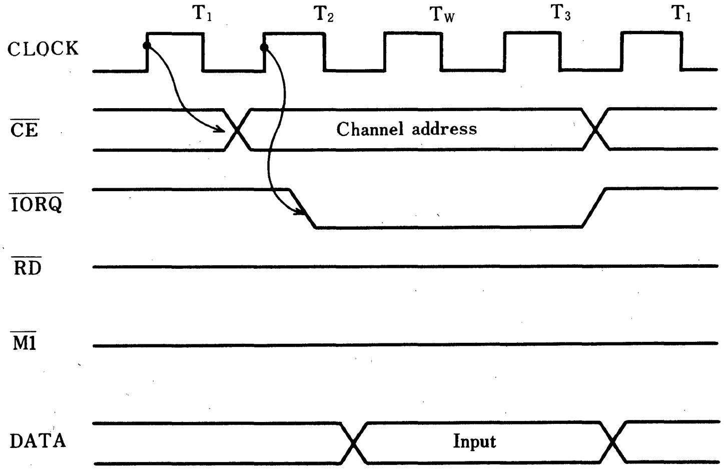


Fig. 3 Write cycle timing

LH8530 Z8530™ SCC Serial Communications Controller

■ Description

The LH8530 Z8530 SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses. The LH8530 functions as a serial-to-parallel, parallel-to-serial converter controller. The LH8530 can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital Phase-Locked Loops, and crystal oscillators that dramatically reduce the need for external logic.

The LH8530 handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial date transfer application (cassette, disk tape drives, etc).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The LH8530 also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The daisy-chain interrupt hierarchy is also supported by the LH8530.

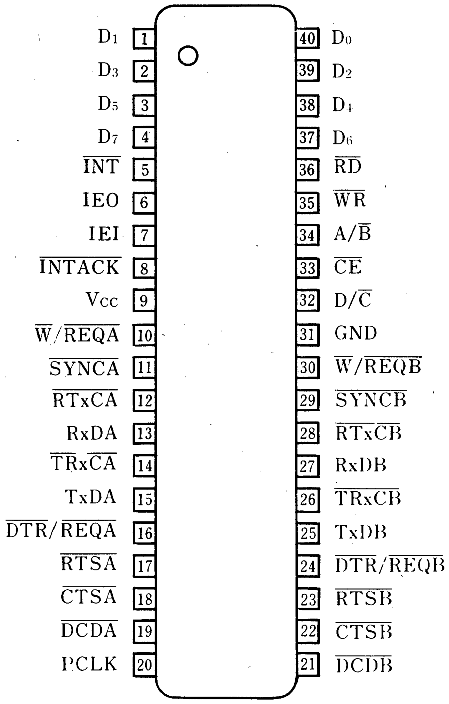
The LH8530A Z8530A SCC is the high speed version which can operate at 6MHz system clock.

■ Features

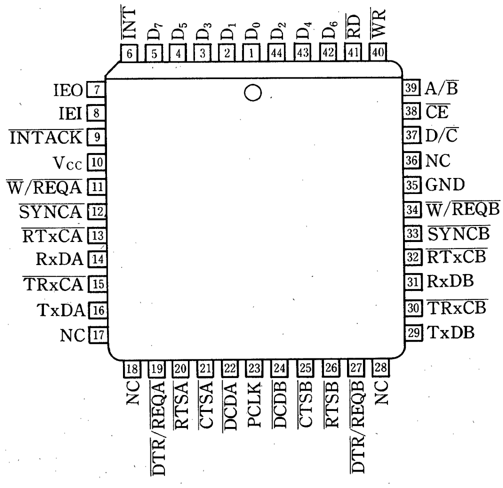
1. Two independent, 0 to 1.5M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
2. Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
3. Asychronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor, break detection and generation; parity, overrun, and framing error detection.

■ Pin Connections

LH8530P/LH8530AP



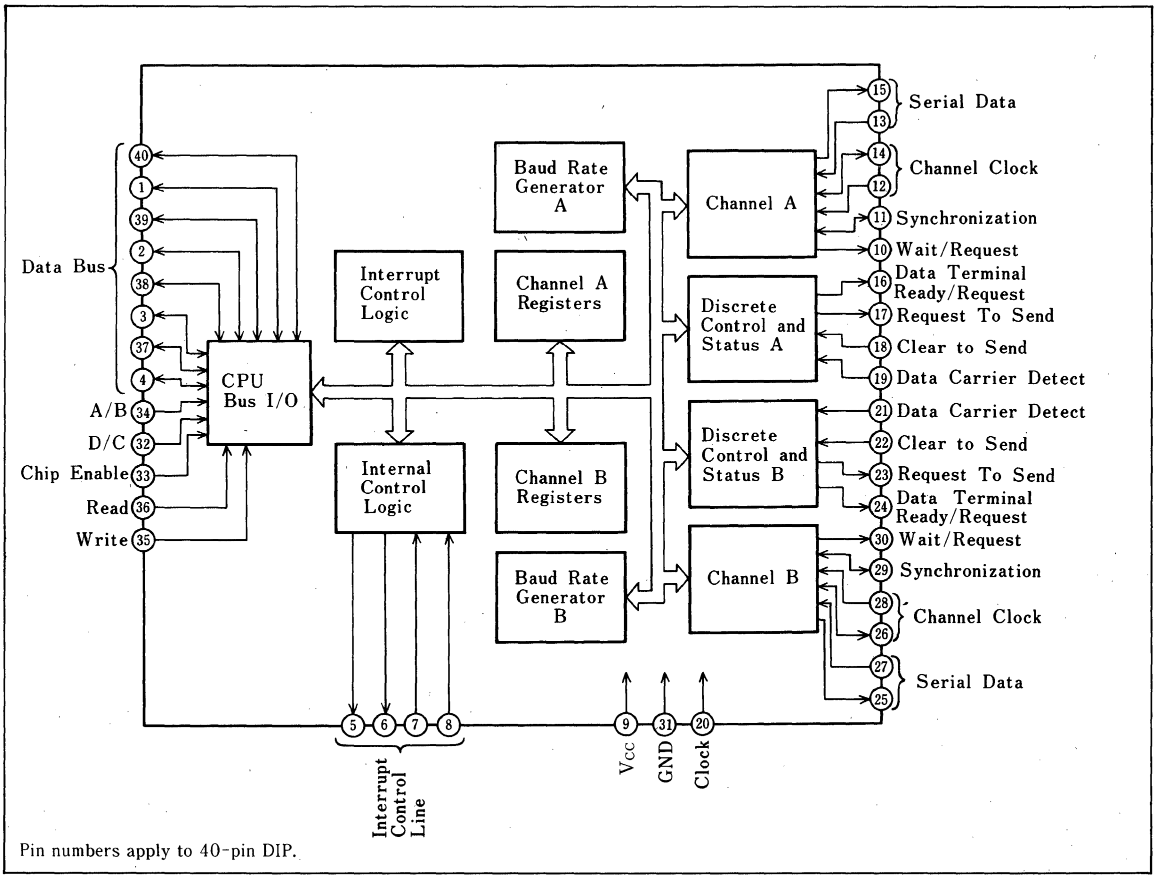
LH8530U/LH8530AU



Top View

4. Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
5. SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion. I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
6. Local Loopback and Auto Echo modes.
7. 40-pin DIP (DIP40-P-600)
   44-pin QFJ (QFJ44-P-S650)

Block Diagram



Pin numbers apply to 40-pin DIP.

Ordering Information

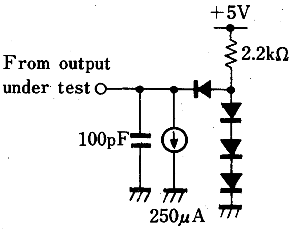
```
LH8530 X X
 │     │ └ Package
 │     │      P: 40-pin DIP (DIP40-P-600)
 │     │      U: 44-pin QFJ (QFJ44-P-S650)
 │     └── Clock frequency
 │            Blank: 4MHz
 │            A: 6MHz
 └──────── Model No.
```

Pin Description

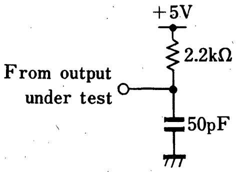
Pin	Meaning	I/O	Function
A/$\overline{B}$	Channel A/ Channel B select	I	Channel select signal.
$\overline{CE}$	Chip enable	I	Active low. Enables the CPU to transmit and receive command and data when low.
$\overline{CTSA}$ $\overline{CTSB}$	Clear to send	I	Active low. Enables the respective transmitters.
D/$\overline{C}$	Data/control select	I	This signal defines the type of information on the data bus. High means data ; Low indicates a command.
$\overline{DCDA}$ $\overline{DCDB}$	Data carrier detect	I	Active low. Enables the respective receivers.
D_0-D_7	Data bus	Bidirectional 3-state	System data bus.
$\overline{DTR}/\overline{REQA}$ $\overline{DTR}/\overline{REQB}$	Data terminal ready/request	O	Active low. These outputs follow the state programmed into the DTR bit.
IEI	Interrupt enable input	I	Active high. IEI is used to form a daisy chain that determines the interrupt priority order.
IEO	Interrupt enable output	O	Active high. IEO is used to form a daisy chain that determines the interrupt priority order.
$\overline{INT}$	Interrupt request	Open-drain	Active low, open-drain. Indicates an interrupt request to the CPU.
$\overline{INTACK}$	Interrupt acknowledge	I	Active low. This signal indicates an active interrupt acknowledge cycle.
$\overline{RD}$	Read	I	Active low. This signal indicates a read operation.
R×DA R×DB	Receive data	I	Active high. These are receive data lines.
$\overline{RT}$×$\overline{CA}$ $\overline{RT}$×$\overline{CB}$	Receive/transmit clocks	I	Active low. These are communication clock lines.
$\overline{RTSA}$ $\overline{RTSB}$	Request to send	O	Active low. Goes high after the transmitter is empty.
$\overline{SYNCA}$ $\overline{SYNCB}$	Synchronization	I/O	Active low. Indicates that a synchronization pattern has been recognized.
T×DA T×DB	Transmit data	O	Active high. These are transmit data lines.
$\overline{TR}$×$\overline{CA}$ $\overline{TR}$×$\overline{CB}$	Transmit/receive clocks	I/O	These are communication clocks.
$\overline{WR}$	Write	I	Active low. This signal indicates a write operation.
$\overline{W}/\overline{REQA}$ $\overline{W}/\overline{REQB}$	Wait/request	Open-drain	Active low. Operate as request lines when the DMA is the bus master or as wait lines when the CPU is the bus master.
PCLK	Clock	I	Single-phase clock. It does not have to be the CPU clock.

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Input voltage	V_{IN}	−0.3 to +7.0	V
Output voltage	V_{OUT}	−0.3 to +7.0	V
Operating temperature	Topr	0 to +70	℃
Storage temperature	Tstg	−65 to +150	℃



Standard test load



Open-drain test load

■ DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Input high voltage	V_{IH}		2	V_{CC}+0.3	V
Input low voltage	V_{IL}		−0.3	0.8	V
Output high voltage	V_{OH}	I_{OH}=−250 μA	2.4		V
Output low voltage	V_{OL}	I_{OL}=+2mA		0.4	V
Input leakage current	$\mid I_{IL}\mid$	0.4 ≦ V_{IN} ≦ 2.4V		10	μA
Output leakage current	$\mid I_{OL}\mid$	0.4 ≦ V_{OUT} ≦ 2.4V		10	μA
Current consumption	I_{CC}	LH8530		250	mA
		LH8530A		280	

■ Capacitance

(f=1MHz, Ta=0 to +70℃)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Input capacitance	C_{IN}	Unmeasured Pins Returned to Ground		10	pF
Output capacitance	C_{OUT}			15	pF
Bidirectional capacitance	$C_{I/O}$			20	pF

■ AC Characteristics

(1) CPU interface timing, interrupt timing, and interrupt acknowledge timing

No.	Symbol	Parameter	LH8530		LH8530A		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
1	TwPCl	PCLK low width	105	2000	70	1000	ns	
2	TwPCh	PCLK high width	105	2000	70	1000	ns	
3	TfPC	PCLK fall time		20		10	ns	
4	TrPC	PCLK rise time		20		15	ns	
5	TcPC	PCLK cycle time	250	4000	165	2000	ns	
6	TsA(WR)	Address to $\overline{\text{WR}}$ ↓ setup time	80		80		ns	
7	ThA(WR)	Address to $\overline{\text{WR}}$ ↑ hold time	0		0		ns	
8	TsA(RD)	Address to $\overline{\text{RD}}$ ↓ setup time	80		80		ns	
9	ThA(RD)	Address to $\overline{\text{RD}}$ ↑ hold time	0		0		ns	
10	TsIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ setup time	0		0		ns	
11	TsIAi(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↓ setup time	200		160		ns	1
12	ThIA(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↑ hold time	0		0		ns	
13	TsIAi(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ setup time	200		160		ns	1
14	ThIA(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↑ hold time	0		0		ns	
15	ThIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ hold time	100		100		ns	
16	TsCEl(WR)	$\overline{\text{CE}}$ low to $\overline{\text{WR}}$ ↓ setup time	0		0		ns	
17	ThCE(WR)	$\overline{\text{CE}}$ to $\overline{\text{WR}}$ ↑ hold time	0		0		ns	
18	TsCEh(WR)	$\overline{\text{CE}}$ high to $\overline{\text{WR}}$ ↓ setup time	100		70		ns	
19	TsCEl(RD)	$\overline{\text{CE}}$ low to $\overline{\text{RD}}$ ↓ setup time	0		0		ns	1
20	ThCE(RD)	$\overline{\text{CE}}$ to $\overline{\text{RD}}$ ↑ hold time	0		0		ns	1
21	TsCEh(RD)	$\overline{\text{CE}}$ high to $\overline{\text{RD}}$ ↓ setup time	100		70		ns	1
22	TwRDl	$\overline{\text{RD}}$ low width	390		250		ns	1
23	TdRD(DRA)	$\overline{\text{RD}}$ ↓ to read data active delay	0		0		ns	
24	TdRDr(DR)	$\overline{\text{RD}}$ ↑ to read data not valid delay	0		0		ns	
25	TdRDf(DR)	$\overline{\text{RD}}$ ↓ to read data valid delay		250		180	ns	
26	TdRD(DRz)	$\overline{\text{RD}}$ ↑ to read data float delay		70		45	ns	2
27	TdA(DR)	Address required valid to read data valid delay		590		420	ns	
28	TwWRl	$\overline{\text{WR}}$ low width	390		250		ns	
29	TsDW(WR)	Write data to $\overline{\text{WR}}$ ↓ setup time	0		0		ns	

No.	Symbol	Parameter	LH8530		LH8530A		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
30	ThDW(WR)	Write data to $\overline{\text{WR}}$ ↑ hold time	0		0		ns	
31	TdWR(W)	$\overline{\text{WR}}$ ↓ to wait valid delay		240		200	ns	4
32	TdRD(W)	$\overline{\text{RD}}$ ↓ to wait valid delay		240		200	ns	4
33	TdWRf(REQ)	$\overline{\text{WR}}$ ↓ to $\text{W}/\overline{\text{REQ}}$ not valid delay		240		200	ns	
34	TdRDf(REQ)	$\overline{\text{RD}}$ ↓ to $\text{W}/\overline{\text{REQ}}$ not valid delay		240		200	ns	
35	TdWRr(REQ)	$\overline{\text{WR}}$ ↑ to $\overline{\text{DTR}}/\overline{\text{REQ}}$ not valid delay		5TcPC+300		5TcPC+250	ns	
36	TdRDr(REQ)	$\overline{\text{RD}}$ ↑ to $\overline{\text{DTR}}/\overline{\text{REQ}}$ not valid delay		5TcPC+300		5TcPC+250	ns	
37	TdPC(INT)	PCLK ↓ to $\overline{\text{INT}}$ valid delay		500		500	ns	4
38	TdIAi(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ (acknowledge) delay					ns	5
39	TwRDA	$\overline{\text{RD}}$ (acknowledge) width	285		250		ns	
40	TdRDA(DR)	$\overline{\text{RD}}$ ↓ (acknowledge) to read data valid delay		190		180	ns	
41	TsIEI(RDA)	IEI to $\overline{\text{RD}}$ ↓ (acknowledge) setup time	120		100		ns	
42	ThIEI(RDA)	IEI to $\overline{\text{RD}}$ ↑ (acknowledge) hold time	0		0		ns	
43	TdIEI(IEO)	IEI to IEO delay time		120		100	ns	
44	TdPC(IEO)	PCLK ↑ to IEO delay		250		250	ns	
45	TdRDA(INT)	$\overline{\text{RD}}$ ↓ to $\overline{\text{INT}}$ inactive delay		500		500	ns	4
46	TdRD(WRQ)	$\overline{\text{RD}}$ ↑ to $\overline{\text{WR}}$ ↓ delay for no reset	30		15		ns	
47	TdWRQ(RD)	$\overline{\text{WR}}$ ↑ to $\overline{\text{RD}}$ ↓ delay for no reset	30		30		ns	
48	TwRES	$\overline{\text{WR}}$ and $\overline{\text{RD}}$ coincident low for reset	250		250		ns	
49	Trc	Valid access recovery time	6TcPC+200		6TcPC+130		ns	3

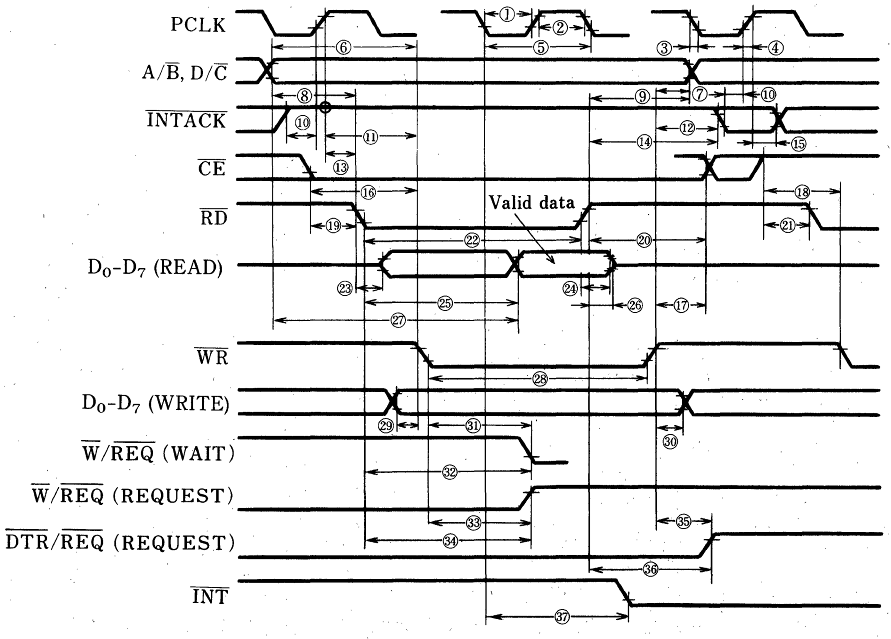
Note 1: Parameter does not apply to Interrupt Acknowledge transactions.

Note 2: Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and minimum AC load.

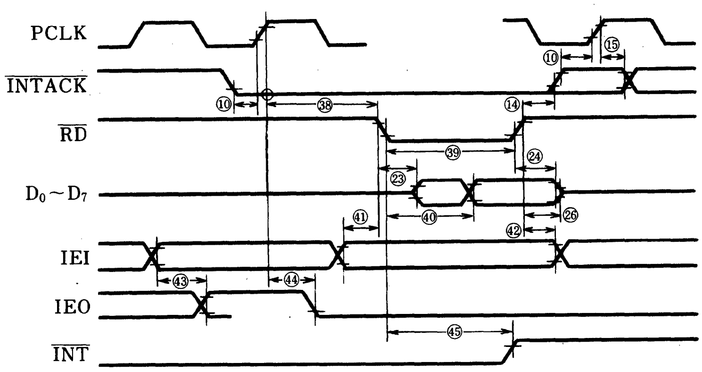
Note 3: Parameter applies only between transactions involving the SCC.

Note 4: Open-drain output, measured with open-drain test load.

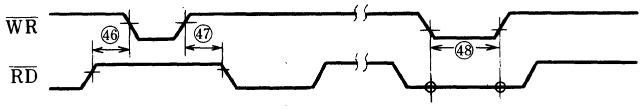
Note 5: Parameter is system dependent. For any SCC in the daisy chain, TdIAi (RD) must be greater than the sum of TdPC (IEO) for the highest priority device in the daisy chain, TsIEI (RDA) for the SCC, and TdIEIf (IEO) for each device separating them in the daisy chain.



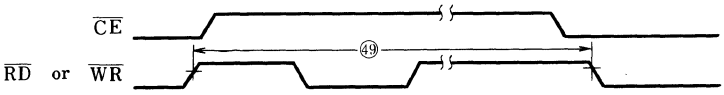
Read and write timing



Interrupt acknowledge timing



Reset timing

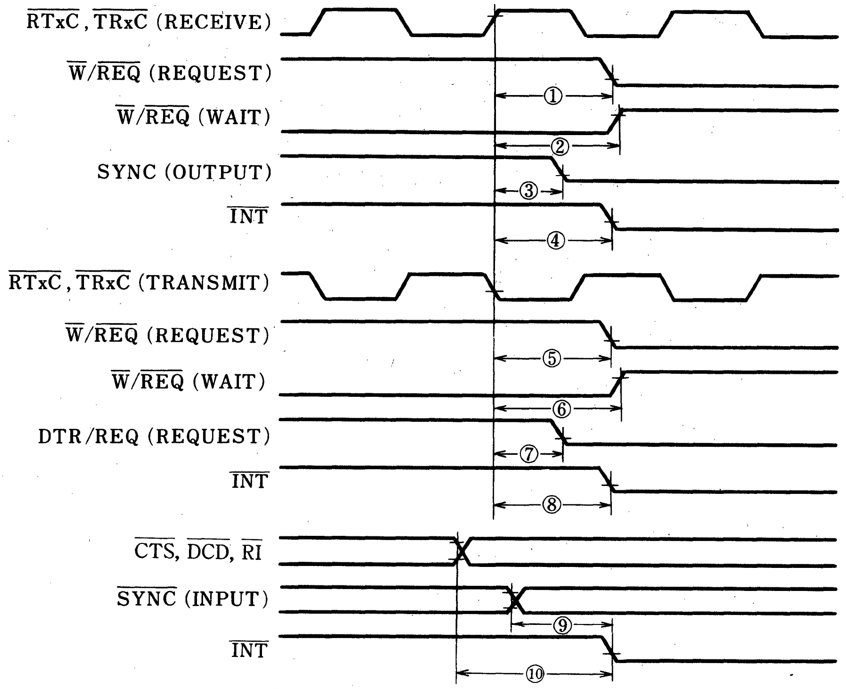


Cycle timing

(2) System timing

No.	Symbol	Parameter	LH8530		LH8530A		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
1	TdRxC(REQ)	RxC ↑ to W/REQ valid delay	8	12	8	12	TcPC	2
2	TdRxC(W)	RxC ↑ to wait inactive delay	8	12	8	12	TcPC	1,2
3	TdRxC(SY)	RxC ↑ to SYNC valid delay	4	7	4	7	TcPC	2
4	TdRxC(INT)	RxC ↑ to INT valid delay	10	16	10	16	TcPC	1,2
5	TdTxC(REQ)	TxC ↓ to W/REQ valid delay	5	8	5	8	TcPC	3
6	TdTxC(W)	TxC ↓ to wait inactive delay	5	8	5	8	TcPC	1,3
7	TdTxC(DRQ)	TxC ↓ to DTR/REQ valid delay	4	7	4	7	TcPC	3
8	TdTxC(INT)	TxC ↓ to INT valid delay	6	10	6	10	TcPC	1,3
9	TdSx(INT)	SYNC transition to INT valid delay	2	6	2	6	TcPC	1
10	TdExT(INT)	DCD or CTS transition to INT valid delay	2	6	2	6	TcPC	1

Note 1: Open-drain output, measured with open-drain test load.
Note 2: RxC is RTxC or TRxC, whichever is supplying the receive clock.
Note 3: TxC is TRxC or RTxC, whichever is supplying the transmit clock.


$\overline{RTxC}$, $\overline{TRxC}$ (RECEIVE)
$\overline{W}/\overline{REQ}$ (REQUEST)
$\overline{W}/\overline{REQ}$ (WAIT)
SYNC (OUTPUT)
$\overline{INT}$
$\overline{RTxC}$, $\overline{TRxC}$ (TRANSMIT)
$\overline{W}/\overline{REQ}$ (REQUEST)
$\overline{W}/\overline{REQ}$ (WAIT)
DTR/REQ (REQUEST)
$\overline{INT}$
$\overline{CTS}$, $\overline{DCD}$, $\overline{RI}$
$\overline{SYNC}$ (INPUT)
$\overline{INT}$
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩

System timing

(3) General timing

No.	Symbol	Parameter	LH8530		LH8530A		Unit	Note
			MIN.	MAX.	MIN.	MAX.		
1	TdPC(REQ)	PCLK ↓ to $\overline{W}/\overline{REQ}$ valid delay		250		250	ns	
2	TdPC(W)	PCLK ↓ to wait inactive delay		350		350	ns	
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ setup time (PCLK ÷ 4 case only)	80	TwPC1	70	TwPC1	ns	1,4
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ setup time (X1 mode)	0		0		ns	1
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ hold time (X1 mode)	150		150		ns	1
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ setup time (X1 mode)	0		0		ns	1,5
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ hold time (X1 mode)	150		150		ns	1,5
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ setup time	−200		−200		ns	1
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ hold time	3TcPC+200		3TcPC+200		ns	1
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ setup time	0		0		ns	2,4
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD delay (X1 mode)		300		300	ns	2
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD delay (X1 mode)		300		300	ns	2,5
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ delay (send clock echo)		200		200	ns	
14	TwRTXh	$\overline{RTxC}$ high width	180		180		ns	6
15	TwRTXl	$\overline{RTxC}$ low width	180		180		ns	6
16	TcRTX	$\overline{RTxC}$ cycle time	400		400		ns	6
17	TcRTXX	Crystal oscillator period	250	1000	250	1000	ns	3
18	TwTRXh	$\overline{TRxC}$ high width	180		180		ns	3
19	TwTRXl	$\overline{TRxC}$ low width	180		180		ns	6
20	TcTRX	$\overline{TRxC}$ cycle time	400		400		ns	6
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ pulse width	200		200		ns	
22	TwSY	$\overline{SYNC}$ pulse width	200		200		ns	

Note 1: RxC is RTxC or TRxC, whichever is supplying the receive clock.

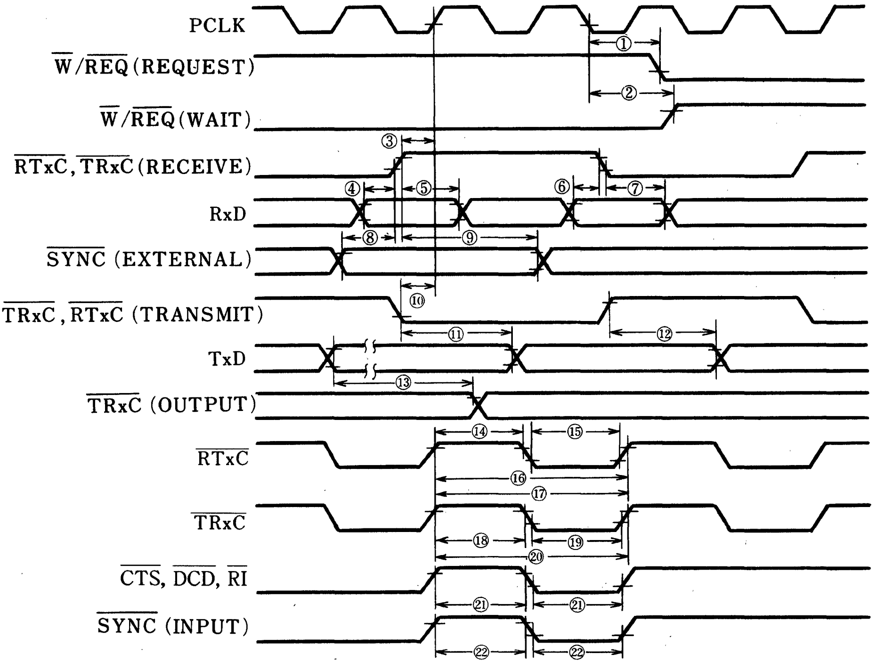
Note 2: TxC is TRxC or RTxC, whichever is supplying the transmit clock.

Note 3: Both RTxC and SYNC have 30 pF capacitors to ground connected to them.

Note 4: Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.

Note 5: Parameter applies only to FM encoding/decoding.

Note 6: Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.



General timing

Data Communications Capabilities

The LH8530 provides two independent fullduplex channels programmable for use in any common Asynchronous or Synchronous data communication protocol. Fig. 1 illustrates these protocols.

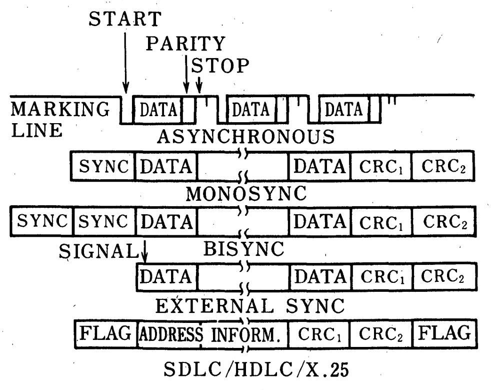


Fig. 1 Some SCC protocols

SDLC Loop Mode

The LH8530 supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the LH8530 performs the functions of a secondary station while an LH8530 operating in regular SDLC mode can act as a controller (Fig. 2).

A secondary station in an SDLC Loop is alway listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the LH8530. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

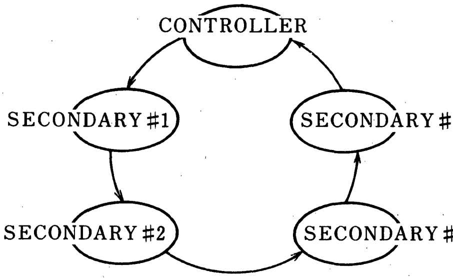


Fig. 2 An SDLC loop

Data Encoding

The LH8530 may be programmed to encode and decode the serial data in four different ways (Fig. 3). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the LH8530 can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

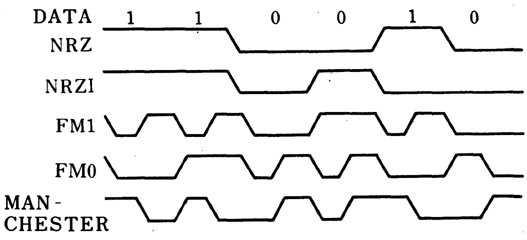


Fig. 3 Data encoding methods

Auto Echo and Local Loopback

The LH8530 is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{\text{CTS}}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and $\overline{\text{WAIT}}/\overline{\text{REQUEST}}$ on transmit.

The LH8530 is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

Baud Rate Generator

Each channel in the LH8530 contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time , but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{\text{TRxC}}$ pin, the output of the baud rate generator may be echoed out via the $\overline{\text{TRxC}}$ pin.

The following formula relates the time constant to the baud rate (the baud rate is in bits/second and the BR clock period is in seconds).

$$\text{baud rate}=\frac{1}{2(\text{time constant}+2)\times(\text{BR clock period})}$$

Digital Phase-Locked Loop

The LH8530 contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

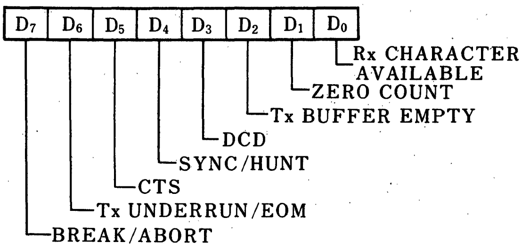
For NRZI encoding, the DPLL counts the 32X clock 60 create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

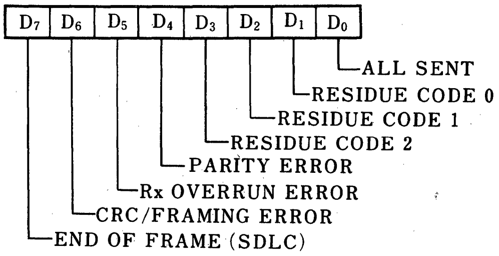
The 32X clock for the DPLL can be programmed to come from either the $\overline{\text{RTxC}}$ input or the output of the baud rate generator. The DPLL output may be programmed to be enchoed out of the LH8530 via the $\overline{\text{TRxC}}$ pin (if this pin is not being used as an input).

Read Registers

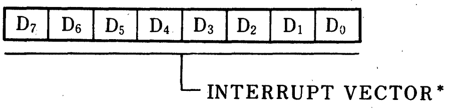
• Read Register 0



• Read Register 1

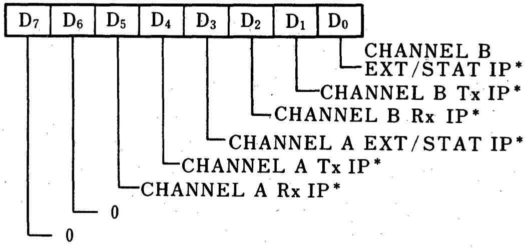


• Read Register 2



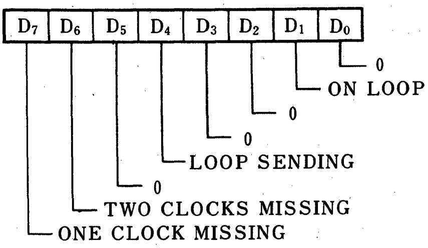
*MODIFIED IN CHANNEL B

• Read Register 3*

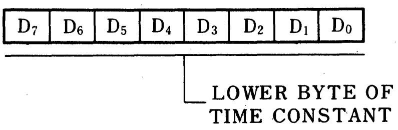


*ALWAYS 0 IN CHANNEL B

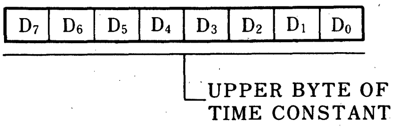
• Read Register 10



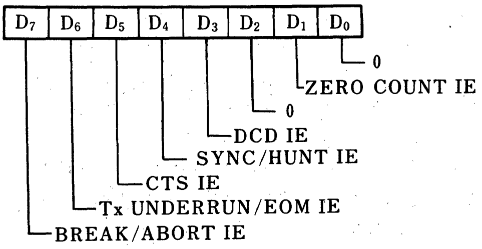
• Read Register 12



• Read Register 13

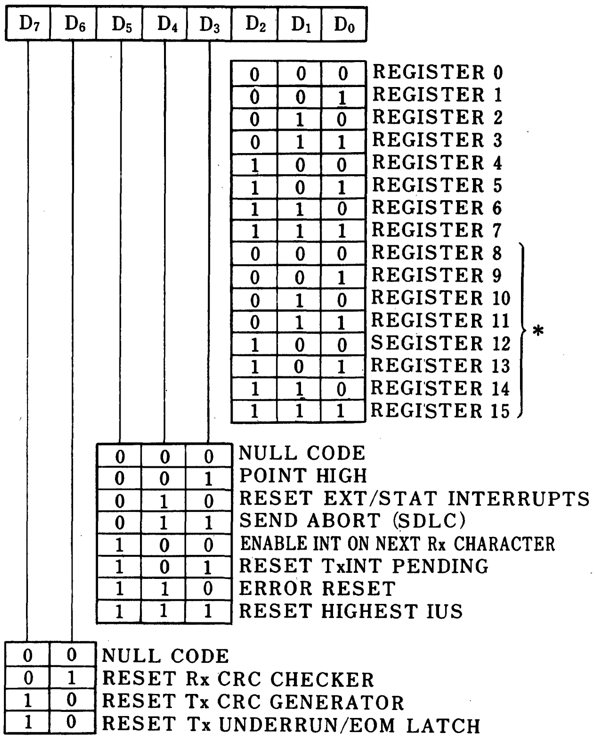


• Read Register 15



Write Registers

• Write Register 0



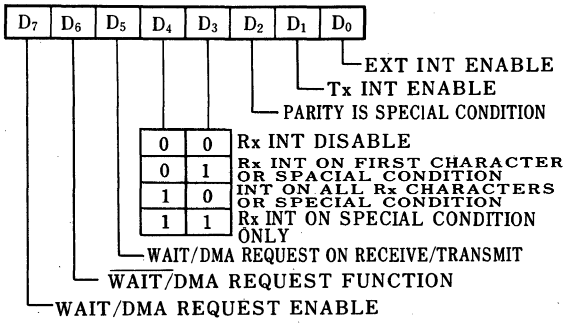
D2	D1	D0	
0	0	0	REGISTER 0
0	0	1	REGISTER 1
0	1	0	REGISTER 2
0	1	1	REGISTER 3
1	0	0	REGISTER 4
1	0	1	REGISTER 5
1	1	0	REGISTER 6
1	1	1	REGISTER 7
0	0	0	REGISTER 8 *
0	0	1	REGISTER 9 *
0	1	0	REGISTER 10 *
0	1	1	REGISTER 11 *
1	0	0	SEGISTER 12 *
1	0	1	REGISTER 13 *
1	1	0	REGISTER 14 *
1	1	1	REGISTER 15 *

D5	D4	D3	
0	0	0	NULL CODE
0	0	1	POINT HIGH
0	1	0	RESET EXT/STAT INTERRUPTS
0	1	1	SEND ABORT (SDLC)
1	0	0	ENABLE INT ON NEXT Rx CHARACTER
1	0	1	RESET TxINT PENDING
1	1	0	ERROR RESET
1	1	1	RESET HIGHEST IUS

D7	D6	
0	0	NULL CODE
0	1	RESET Rx CRC CHECKER
1	0	RESET Tx CRC GENERATOR
1	0	RESET Tx UNDERRUN/EOM LATCH

*WITH POINT HIGH COMMAND ($D_5D_4D_3$=001)

• Write Register 1



D0 — EXT INT ENABLE

D1 — Tx INT ENABLE

D2 — PARITY IS SPECIAL CONDITION

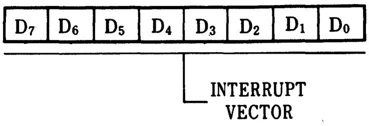
D4	D3	
0	0	Rx INT DISABLE
0	1	Rx INT ON FIRST CHARACTER OR SPACIAL CONDITION
1	0	INT ON ALL Rx CHARACTERS OR SPECIAL CONDITION
1	1	Rx INT ON SPECIAL CONDITION ONLY

D5 — WAIT/DMA REQUEST ON RECEIVE/TRANSMIT

D6 — $\overline{\text{WAIT}}$/DMA REQUEST FUNCTION

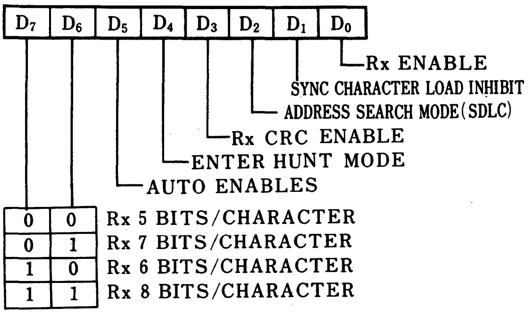
D7 — WAIT/DMA REQUEST ENABLE

• Write Register 2



D7–D0 — INTERRUPT VECTOR

• Write Register 3



D0 — Rx ENABLE

D1 — SYNC CHARACTER LOAD INHIBIT

D2 — ADDRESS SEARCH MODE (SDLC)

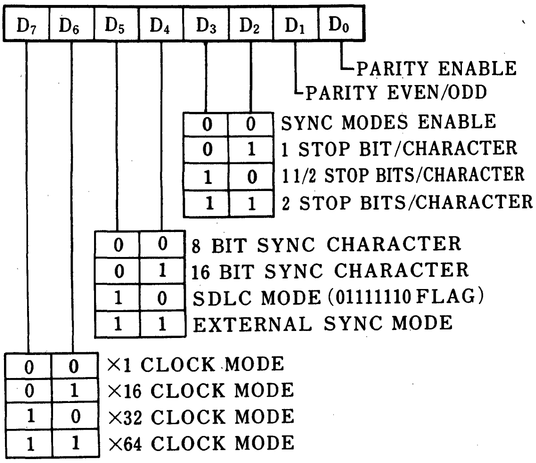
D3 — Rx CRC ENABLE

D4 — ENTER HUNT MODE

D5 — AUTO ENABLES

D7	D6	
0	0	Rx 5 BITS/CHARACTER
0	1	Rx 7 BITS/CHARACTER
1	0	Rx 6 BITS/CHARACTER
1	1	Rx 8 BITS/CHARACTER

• Write Register 4



D0 — PARITY ENABLE

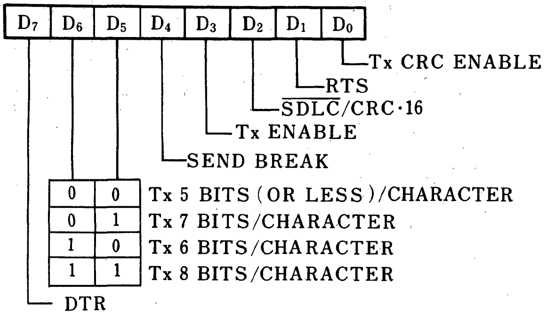
D1 — PARITY EVEN/ODD

D3	D2	
0	0	SYNC MODES ENABLE
0	1	1 STOP BIT/CHARACTER
1	0	11/2 STOP BITS/CHARACTER
1	1	2 STOP BITS/CHARACTER

D5	D4	
0	0	8 BIT SYNC CHARACTER
0	1	16 BIT SYNC CHARACTER
1	0	SDLC MODE (01111110 FLAG)
1	1	EXTERNAL SYNC MODE

D7	D6	
0	0	×1 CLOCK MODE
0	1	×16 CLOCK MODE
1	0	×32 CLOCK MODE
1	1	×64 CLOCK MODE

- **Write Register 5**



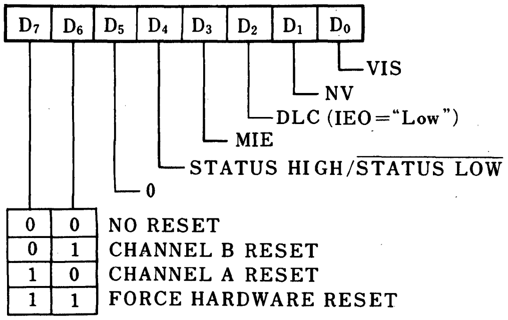
- **Write Register 6**

D7	D6	D5	D4	D3	D2	D1	D0	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_1$	SYNC$_0$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 6 BITS
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	BISYNC, 16 BITS
SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	1	1	1	1	BISYNC, 12 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	ADR$_3$	ADR$_2$	ADR$_1$	ADR$_0$	SDLC, 8 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	×	×	×	×	SDLC, 4 BITS (ADDRESS RUN)

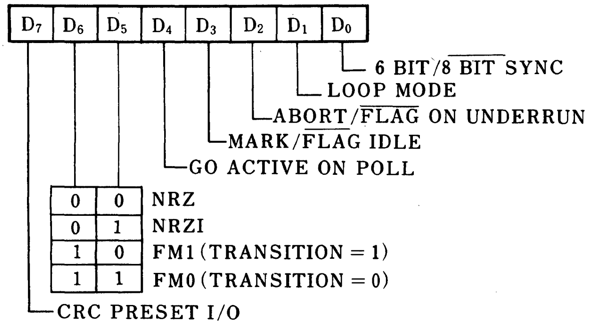
- **Write Register 7**

D7	D6	D5	D4	D3	D2	D1	D0	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	×	×	MONOSYNC, 6 BITS
SYNC$_{15}$	SYNC$_{14}$	SYNC$_{13}$	SYNC$_{12}$	SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	BISYNC, 16 BITS
SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	BISYNC, 12 BITS
0	1	1	1	1	1	1	0	SDLC FLAG

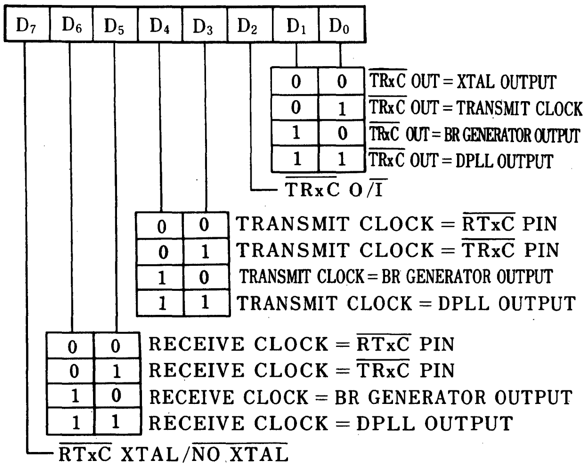
- **Write Register 9**



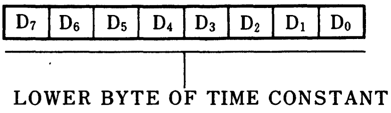
- **Write Register 10**



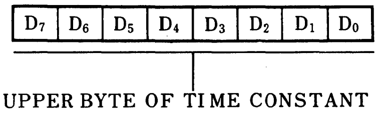
- **Write Register 11**


D7 D6 D5 D4 D3 D2 D1 D0
0 0 $\overline{\text{TRxC}}$ OUT = XTAL OUTPUT
0 1 $\overline{\text{TRxC}}$ OUT = TRANSMIT CLOCK
1 0 $\overline{\text{TRxC}}$ OUT = BR GENERATOR OUTPUT
1 1 $\overline{\text{TRxC}}$ OUT = DPLL OUTPUT
TRxC O/$\overline{\text{I}}$
0 0 TRANSMIT CLOCK = $\overline{\text{RTxC}}$ PIN
0 1 TRANSMIT CLOCK = $\overline{\text{TRxC}}$ PIN
1 0 TRANSMIT CLOCK = BR GENERATOR OUTPUT
1 1 TRANSMIT CLOCK = DPLL OUTPUT
0 0 RECEIVE CLOCK = $\overline{\text{RTxC}}$ PIN
0 1 RECEIVE CLOCK = $\overline{\text{TRxC}}$ PIN
1 0 RECEIVE CLOCK = BR GENERATOR OUTPUT
1 1 RECEIVE CLOCK = DPLL OUTPUT
$\overline{\text{RTxC}}$ XTAL/$\overline{\text{NO XTAL}}$

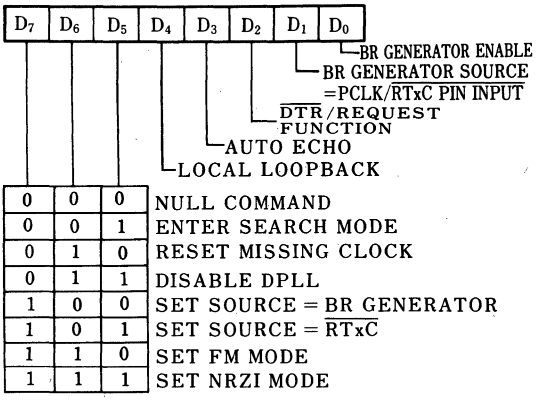
- **Write Register 12**



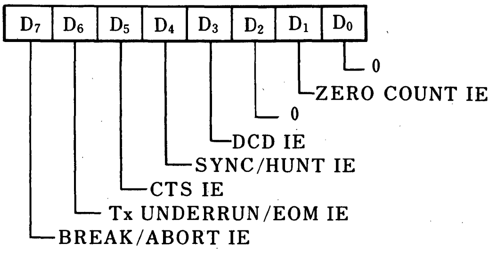
- **Write Register 13**



- **Write Register 14**


D7 D6 D5 D4 D3 D2 D1 D0
BR GENERATOR ENABLE
BR GENERATOR SOURCE =PCLK/$\overline{\text{RTxC}}$ PIN INPUT
$\overline{\text{DTR}}$/REQUEST FUNCTION
AUTO ECHO
LOCAL LOOPBACK
0 0 0 NULL COMMAND
0 0 1 ENTER SEARCH MODE
0 1 0 RESET MISSING CLOCK
0 1 1 DISABLE DPLL
1 0 0 SET SOURCE = BR GENERATOR
1 0 1 SET SOURCE = $\overline{\text{RTxC}}$
1 1 0 SET FM MODE
1 1 1 SET NRZI MODE

- **Write Register 15**



16-bit Microprocessors

LH70108(V20) High-Performance 16-Bit Microprocessor

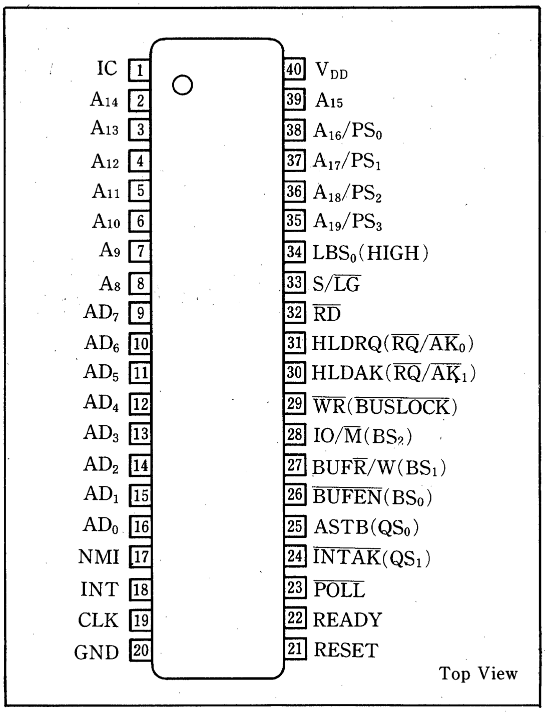
Description

The LH70108(V20) is a CMOS 16-bit microprocessor with internal 16-bit architecture and an 8-bit external data bus. The LH70108 additionally has a powerful instruction set including bit processing, packed BCD operations, and high-speed multiplication/division operations. The LH70108 can also execute the entire 8080 instruction set comes with a standby mode that significantly reduces power consumption. It is software-compatible with the LH70116 16-bit microprocessor.

Features

1. Minimum instruction execution time: 250ns (at 8MHz)
2. Maximum addressable memory: 1M byte
3. Abundant memory addressing modes
4. 14×16-bit register set
5. 101 instructions
6. Bit, byte, word, and block operations
7. Bit field operation instructions
8. Packed BCD instructions
9. Multiplication/division instruction execution time: 2.4μs to 7.1μs (at 8MHz)
10. High-speed block transfer instructions: 1M byte/s (at 8MHz)
11. Hign-speed calculation of effective addresses: 2 clock cycles in any addressing mode
12. Maskable (INT) and nonmaskable (NMI) interrupt inputs
13. IEEE-796 bus compatible interface 8080 emulation mode
14. CMOS technology
15. Low-power consumption
16. Low-power standby mode
17. Single power supply
18. 5MHz or 8MHz clock
19. 40-pin DIP (DIP40-P-600)

Pin Connections



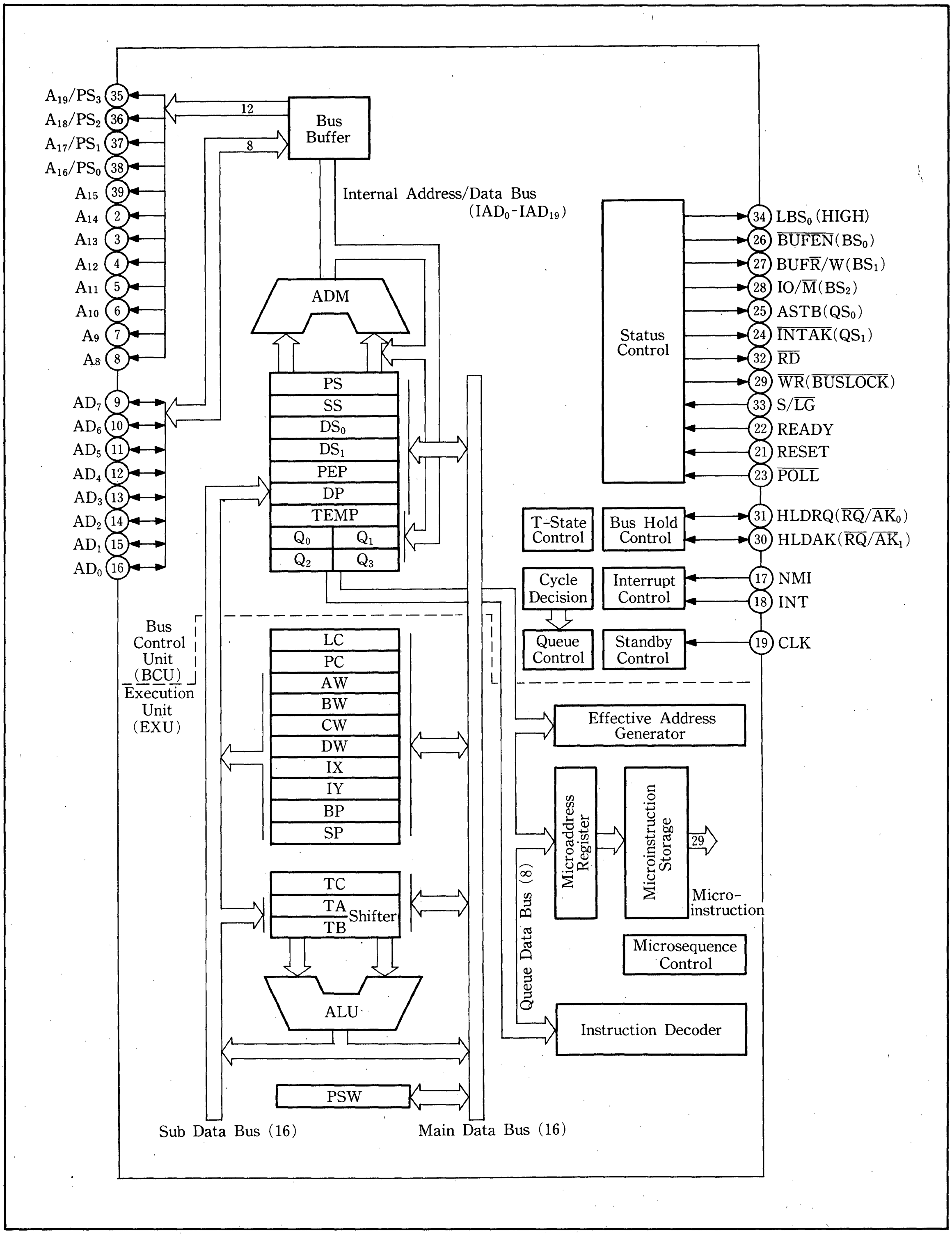
Ordering Information

LH70108-X

- X — Frequency
  - 5: 5MHz
  - 8: 8MHz
- LH70108 — Model No.

*V20 is a trademark of NEC corporation.

Block Diagram



Pin Identification

No.	Symbol	Direction	Function
1	IC*		Internally connected
2-8	A_{14}-A_8	Out	Address but, middle bits
9-16	AD_7-AD_0	In/Out	Address/data bus
17	NMI	In	Nonmaskable interrupt input
18	INT	In	Maskable interrupt input
19	CLK	In	Clock input
20	GND		Ground potential
21	RESET	In	Reset input
22	READY	In	Ready input
23	$\overline{\text{POLL}}$	In	Poll input
24	$\overline{\text{INTAK}}$ (QS_1)	Out	Interrupt acknowledge output (queue status bit 1 output)
25	ASTB (QS_0)	Out	Address strobe output (queue status bit 0 output)
26	$\overline{\text{BUFEN}}$ (BS_0)	Out	Buffer enable output (bus status bit 0 output)
27	$\text{BUF}\overline{\text{R}}\text{/W}$ (BS_1)	Out	Buffer read/write output (bus status bit 1 output)

No.	Symbol	Direction	Function
28	$\text{IO/}\overline{\text{M}}$ (BS_2)	Out	Access is I/O or memory (bus status bit 2 output)
29	WR ($\overline{\text{BUSLOCK}}$)	Out	Write strobe output (bus lock output)
30	HLDAK ($\overline{\text{RQ}}/\overline{\text{AK}_1}$)	Out (In/Out)	Hold acknowledge output, (bus hold request input/acknowledge output 1)
31	HLDRQ ($\overline{\text{RQ}}/\overline{\text{AK}_0}$)	In (In/Out)	Hold request input (bus hold request input/acknowledge output 0)
32	$\overline{\text{RD}}$	Out	Read strobe output
33	$\text{S/}\overline{\text{LG}}$	In	Small-scale/large-scale system input
34	LBS_0 (HIGH)	Out	Latched bus status output 0 (always high in large-scale systems)
35-38	A_{19}/PS_3- A_{16}/PS_0	Out	Address bus, high bits or processor status output
39	A_{15}	Out	Address bus, bit 15
40	V_{DD}		Power supply

Notes: * IC should be connected to ground.
Where pins have different functions in small-and large-scale systems, the large-scale system pin symbol and function are in parentheses.
Unused input pins should be tied to ground or V_{DD} to minimize power dissipation and prevent the flow of potentially harmful currents.

■ Absolute Maximum Ratings (Ta=+25℃)

Parameter	Symbol	Ratings	Units
Supply voltage	V_{DD}	−0.5 to+7.0	V
Input voltage	V_I	−0.5 to V_{DD}+0.3	V
CLK input voltage	V_K	−0.5 to V_{DD}+1.0	V
Output voltage	V_O	−0.5 to V_{DD}+0.3	V
Operating temperature	Topr	−40 to +85	℃
Storage temperature	Tstg	−65 to +150	℃

■ Capacitance (Ta=+25℃, V_{DD}=OV)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Input capacitance	C_I	fc=1MHz Unmeasured pins returned to OV		15	pF
I/O capacitance	C_{IO}			15	pF

■ DC Characteristics

(LH70108-5, Ta=−40℃ to+85℃, V_{pp}=+5V±10%)
(LH70108-8, Ta=−10℃ to+70℃, V_{pp}=+5V±5%)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input HIGH voltage	V_{IH}			2.2		V_{DD}+0.3	V
Input LOW voltage	V_{IL}			−0.5		0.8	V
CLK input HIGH voltage	V_{KH}			3.9		V_{DD}+1.0	V
CLK input LOW voltage	V_{KL}			−0.5		0.6	V
Output HIGH voltage	V_{OH}	I_{OH}=−400 μA		0.7V_{DD}			V
Output LOW voltage	V_{OL}	I_{OL}=2.5mA				0.4	V
Input leakage HIGH current	I_{LIH}	V_I=V_{DD}				10	μA
Input leakage LOW current	I_{LIL}	V_I=0V				−10	μA
Output leakage HIGH current	I_{LOH}	V_O=V_{DD}				10	μA
Output leakage LOW current	I_{LOL}	V_O=0V				−10	μA
HLDRQ input HIGH current	I_{HQH}	V_I=V_{DD}				10	μA
HLDRQ input LOW current	I_{HQL}	V_I=0V				−0.5	mA
Supply current	I_{DD}	Normal operation	70108-5		30	60	mA
		Standby mode	5MHz		5	10	mA
		Normal operation	70108-8		45	80	mA
		Standby mode	8MHz		6	12	mA

■ AC Characteristics

(LH70108-5, Ta=−40℃ to+85℃, V_{DD}=+5V±10%)
(LH70108-8, Ta=−10℃ to+70℃, V_{DD}=+5V±5%)

Parameter	Symbol	Conditions	LH70108-5		LH70108-8		Unit
			MIN.	MAX.	MIN.	MAX.	
Small/Large Scale							
Clock cycle	t_{CYK}		200	500	125	500	ns
Clock pulse HIGH width	t_{KKH}	V_{KH}=3.0V	69		44		ns
Clock pulse LOW width	t_{KKL}	V_{KL}=1.5V	90		60		ns
Clock rise time	t_{KR}	1.5V to 3.0V		10		10	ns
Clock fall time	t_{KF}	3.0V to 1.5V		10		10	ns
READY inactive setup to CLK ↓	t_{SRYLK}		−8		−8		ns
READY inactive hold after CLK ↑	t_{HKRYH}		30		20		ns
READY active setup to CLK ↑	t_{SRYHK}		$t_{KKL}-8$		$t_{KKL}-8$		ns
READY active hold after CLK ↑	t_{HKRYL}		30		20		ns
Data setup time to CLK ↓	t_{SDK}		30		20		ns
Data hold time after CLK ↓	t_{HKD}		10		10		ns
NMI, INT, $\overline{POLL}$ setup time to CLK ↑	t_{SIK}		30		15		ns
Input rise time (except CLK)	t_{IR}	0.8V to 2.2V		20		20	ns
Input fall time (except CLK)	t_{IF}	2.2V to 0.8V		12		12	ns
Output rise time	t_{OR}	0.8V to 2.2V		20		20	ns
Output fall time	t_{OF}	2.2V to 0.8V		12		12	ns
Small Scale							
Address delay time from CLK ↓	t_{DKA}	C_L=100pF	10	90	10	60	ns
Address hold time from CLK ↓	t_{HKA}		10		10		ns
PS delay time from CLK ↓	t_{DKP}		10	90	10	60	ns
PS float delay time from CLK ↑	t_{FKP}		10	80	10	60	ns
Address setup time to ASTB ↓	t_{SAST}		$t_{KKL}-60$		$t_{KKL}-30$		ns
Address float delay time from CLK ↓	t_{FKA}		t_{HKA}	80	t_{HKA}	60	ns
ASTB ↑ delay time from CLK ↓	t_{DKSTH}			80		50	ns
ASTB ↓ delay time from CLK ↑	t_{DKSTL}			85		55	ns
ASTB HIGH width	t_{STST}		$t_{KKL}-20$		$t_{KKL}-10$		ns
Address hold time from ASTB ↓	t_{HSTA}		$t_{KKH}-10$		$t_{KKH}-10$		ns
Control delay time from CLK	t_{DKCT}		10	110	10	65	ns
Address float to $\overline{RD}$ ↓	t_{AFRL}		0		0		ns
$\overline{RD}$ ↓ delay time from CLK ↓	t_{DKRL}		10	165	10	80	ns
$\overline{RD}$ ↑ delay time from CLK ↓	t_{DKRH}		10	150	10	80	ns
Address delay time from $\overline{RD}$ ↑	t_{DRHA}		$t_{CYK}-45$		$t_{CYK}-40$		ns
$\overline{RD}$ LOW width	t_{RR}		$2t_{CYK}-75$		$2t_{CYK}-50$		ns
Data output delay time from CLK ↓	t_{DKD}		10	90	10	60	ns
Data float delay time from CLK ↓	t_{FKD}		10	80	10	60	ns
$\overline{WR}$ LOW width	t_{WW}		$2t_{CYK}-60$		$2t_{CYK}-40$		ns
HLDRQ setup time to CLK ↑	t_{SHQK}		35		20		ns
HLDAK delay time from CLK ↓	t_{DKHA}		10	160	10	100	ns
$\overline{BUFEN}$ ↑ from $\overline{WR}$ ↑	t_{WCT}		$t_{KKL}-20$		$t_{KKL}-20$		ns

■ AC Characteristics (cont)

(LH70108-5, Ta=−40℃ to+85℃, V_{DD}=−5V±10%)
(LH70108-8, Ta=−10℃ to 70℃, V_{DD}=+5V±5%)

Parameter	Symbol	Conditions	LH70108-5		LH70108-8		Unit
			MIN.	MAX.	MIN.	MAX.	
Large Scale							
Address delay time from CLK ↓	t_{DKA}	C_L=100pF	10	90	10	60	ns
Address hold time from CLK ↓	t_{HKA}		10		10		ns
PS delay time from CLK ↓	t_{DKP}		10	90	10	60	ns
PS float delay time from CLK ↑	t_{FKP}		10	80	10	60	ns
Address float delay time from CLK ↓	t_{FKA}		t_{HKA}	80	t_{HKA}	60	ns
Address delay time from $\overline{RD}$ ↑	t_{DRHA}		$t_{CYK}-45$		$t_{CYK}-40$		ns
ASTB delay time from BS ↓	t_{DBST}			15		15	ns
BS ↓ delay time from CLK ↑	t_{DKBL}		10	110	10	60	ns
$\overline{BS}$ ↑ delay time from CLK ↓	t_{DKBH}		10	130	10	65	ns
$\overline{RD}$ ↓ delay time from address float	t_{DAFRL}		0		0		ns
$\overline{RD}$ ↓ delay time from CLK ↓	t_{DKRL}		10	165	10	80	ns
$\overline{RD}$ ↑ delay time from CLK ↓	t_{DKRH}		10	150	10	80	ns
$\overline{RD}$ LOW width	t_{RR}		$2t_{CYK}-75$		$2t_{CYK}-50$		ns
Data output delay time from CLK ↓	t_{DKD}		10	90	10	60	ns
Data float delay time from CLK ↑	t_{FKD}		10	80	10	60	ns
$\overline{AK}$ delay time from CLK ↓	t_{DKAK}			70		50	ns
$\overline{RQ}$ setup time to CLK ↑	t_{SRQK}		20		10		ns
$\overline{RQ}$ hold time after CLK ↓	t_{HKRQ1}		0		0		ns
$\overline{RQ}$ hold time after CLK ↑	t_{HKRQ2}		40		30		ns

Pin Functions

Some pins of the LH70108 have different functions according to whether the microprocessor is used in a small- or large-scale system. Other pins function the same way in either type of system.

A_{15}-A_8 (Address Bus)

For small- and large-scale systems.

The CPU uses these pins to output the middle 8 bits of the 20-bit address data. They are three-state outputs and become high impedance during hold acknowledge.

AD_7-AD_0 (Address/Data Bus)

For small- and large-scale systems.

The CPU uses these pins as the time-multiplexed address and data bus. When high, and AD bit is a one; when low, an AD bit is a zero. This bus contains the lower 8 bits of the 20-bit address during T1 of the bus cycle and is used as an 8-bit data bus during T2, T3, and T4 of the bus cycle.

Sixteen-bit data I/O performed in two steps. The low byte is sent first, followed by the high byte. The address/data bus is a three-state bus and can be at a high or low level during standby mode. The bus will be high impedance during hold and interrupt acknowledge.

NMI (Nonmaskable Interrupt)

For small- and large-scale systems.

This pin is used to input nonmaskable interrupt requests. NMI cannot be masked by software. This input is positive edge triggered and must be held high for five clocks to guarantee recognition. Actual interrupt processing begins, however, after completion of the instruction in progress.

The contents of interrupt vector 2 determine the starting address for the interrupt-servicing routine. Note that a hold request will be accepted even during NMI acknowledge.

This interrupt will cause the LH70108 to exit the standby mode.

INT (Maskable Interrupt)

For small- and large-scale systems.

This pin is an interrupt request that can be masked by software.

INT is active high level and is sensed during the last clock of the instruction. The interrupt will be accepted if the interrupt enable flag IE is set. The CPU outputs the $\overline{\text{INTAK}}$ signal to inform external devices that the interrupt request has been granted. INT must be asserted until the interrupt acknowledge is returned.

If NMI and INT interrupts occur at the same time, NMI has higher priority than INT and INT cannot be accepted. A hold request will be accepted during INT acknowledge.

This interrupt causes the LH70108 to exit the standby mode.

CLK (Clock)

For small- and large-scale systems.

This pin is used for external clock input.

RESET (Reset)

For small- and large-scale systems.

This pin is used for the CPU reset signal. It is an active high level. Input of this signal has priority over all other operations. After the reset signal input returns to a low level, the CPU begins execution of the program starting at address FFFF0H.

In addition to causing normal CPU start, RESET input will cause the LH70108 to exit the standby mode.

READY (Ready)

For small- and large-scale systems.

When the memory or I/O device being accessed cannot complete data read or write within the CPU basic access time, it can generate a CPU wait state (Tw) by setting this signal to inactive (low level) and requesting a read/write cycle delay.

If the READY signal is active (high level) during either the T3 or Tw state, the CPU will not generate a wait state.

This signal must be input in synchronization with external clock signals to satisfy the setup/hold time for normal operation.

$\overline{\text{POLL}}$ (Poll)

For small- and large-scale systems.

The CPU checks this input upon execution of the $\overline{\text{POLL}}$ instruction. If the input is low, then execution continues. If the input is high, the CPU will check the $\overline{\text{POLL}}$ input every five clock cycles until the input becomes low again.

The $\overline{\text{POLL}}$ and READY functions are used to synchronize CPU program execution with the operation of external devices.

$\overline{\text{RD}}$ (Read Strobe)

For small- and large-scale systems.

The CPU outputs this strobe signal during data read from an I/O device or memory. The IO/$\overline{\text{M}}$ signal is used to select between I/O and memory.

The three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

S/$\overline{\text{LG}}$ (Small/Large)

For small- and large-scale systems.

This signal determines the operation mode of the CPU. This signal is fixed at either a high or low level. When this signal is a high level, the CPU will operate in small-scale system mode, and when low, in the large-scale system mode.

Pins 24 to 31 and pin 34 function differently depending on the operating mode of the CPU. Separate nomenclature is adopted for these signals in the two operating modes.

Pin No.	Function	
	S/$\overline{\text{LG}}$-high	S/$\overline{\text{LG}}$-low
24	$\overline{\text{INTAK}}$	QS_1
25	ASTB	QS_0
26	$\overline{\text{BUFEN}}$	BS_0
27	BUFR/$\overline{\text{W}}$	BS_1
28	IO/$\overline{\text{M}}$	BS_2
29	$\overline{\text{WR}}$	$\overline{\text{BUSLOCK}}$
30	HLDAK	$\overline{\text{RQ}}/\overline{\text{AK}}_1$
31	HLDRQ	$\overline{\text{RQ}}/\overline{\text{AK}}_0$
34	LBS_0	Always high

$\overline{\text{INTAK}}$ (Interrupt Acknowledge)

For small-scale systems.

The CPU generates the $\overline{\text{INTAK}}$ signal low when it accepts an INT signal.

The interrupting device synchronizes with this signal and outputs the interrupt vector to the CPU via the data bus (AD_7-AD_0).

ASTB (Address Strobe)

For small-scale systems.

The CPU outputs this strobe signal to latch address information at an external latch.

ASTB is held at a low level during standby mode however, goes high at one time for a half clock cycle to latch L_{BS0} output.

$\overline{\text{BUFEN}}$ (Buffer Enable)

For small-scale systems

This is used as the output enable signal for an external bidirectional buffer. The CPU generates this signal during data transfer operations with external memory or I/O devices or during input of an interrupt vector.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

BUFR/$\overline{\text{W}}$ (Buffer Read/Write)

For small-scale systems.

The output of this signal determines the direction of data transfer with an external bidirectional buffer. A high output causes transmission from the CPU to the external device; a low signal causes data transfer from the external device to the CPU.

BUFR/$\overline{\text{W}}$ is a three-state output and becomes impedance during hold acknowledge.

IO/$\overline{\text{M}}$ (IO/Memory)

For small-scale systems.

The CPU generates this signal to specify either I/O access or memory access. A high-level output specifies I/O and a low-level signal specifies memory.

IO/M's output is three state and becomes high impedance during hold acknowledge.

$\overline{\text{WR}}$ (Write Strobe)

For small-scale systems.

The CPU generates this strobe signal during data write to an I/O device or memory. Selection of either I/O or memory is performed by the IO/M signal.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

HLDAK (Hold Acknowledge)

For small-scale systems.

The HLDAK signal is used to indicate that the CPU accepts the hold request signal (HLDRQ). When this signal is a high level, the address bus, address/data bus, and the control lines become high impedance.

HLDRQ (Hold Request)

For small-scale systems.

This input signal is used by external devices to request the CPU to release the address bus, address/data bus, and the control bus.

This signal must be input in synchronization with external clock signals to satisfy the setup/hold time for normal operation.

LBS_0 (Latched Bus Status 0)

For small-scale systems.

The CPU uses the signal along with the IO/$\overline{\text{M}}$ and BUFR/$\overline{\text{W}}$ signals to inform an external device what the current bus cycle is.

$IO/\overline{M}$	$BUFR/\overline{W}$	LBS_0	Bus Cycle
0	0	0	Program fetch
0	0	1	Memory read
0	1	0	Memory write
0	1	1	Passive state
1	0	0	Interrupt acknowledge
1	0	1	I/O read
1	1	0	I/O write
1	1	1	Halt

A_{19}/PS_3-A_{16}/PS_0 (Address Bus/Processor Status)

For small- and large-scale systems.

These pins are time multiplexed to operate as an address bus and as processor status signals.

When used as the address bus, these pins are the high 4 bits of the 20-bit memory address. During I/O access, all 4 bits output data 0.

The processor status signal are provided for both memory and I/O use. PS_3 is always 0 in the native mode and 1 in 8080 emulation mode. The interrupt enable flag (IE) is output to PS_2. Pins PS_1 and PS_0 indicate which memory segment is being accessed.

A_{17}/PS_1	A_{16}/PS_0	Segment
0	0	Data segment 1
0	1	Stack segment
1	0	Program segment
1	1	Data segment 0

The output of these pins is three state and becomes high impedance during hold acknowledge.

QS_1, QS_0 (Queue Status)

For large-scale systems.

The CPU uses these signals to allow external devices, such as the floating-point arithmetic processor chip, about the status of the internal CPU instruction queue.

QS_1	QS_0	Instruction Queue Status
0	0	NOP (Queue does not change)
0	1	First byte of instruction
1	0	Queue empty
1	1	Subsequent bytes of instruction

The insturction queue status indicated by these signals is the status when the execution unit (EXU) accesses the instruction queue. The data output from these pins is therefore valid only for one clock cycle immediately following queue access. These status signals are provided so that the floating-point processor chip can monitor the CPU's program execution status and synchronize its operation with the CPU when control is passed to it by the FPO (Floating Point Operation) instructions.

BS_2-BS_0 (Bus Status)

For large-scale systems.

The CPU uses these status signals to allow an external bus controller to monitor what the current bus cycle is.

The external bus controller decodes these signals and generates the control signals required to perform access of the memory or I/O device.

BS_2	BS_1	BS_0	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Program fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state

The output of these signals is three state and becomes high impedance during hold acknowledge.

These signals will be high from the rising edge of clock immediately after RESET signal is active to the next clock rise.

$\overline{\text{BUSLOCK}}$ (Bus Lock)

For large-scale systems.

The CPU uses this signal to secure the bus while executing the instruction immediately following the BUSLOCK prefix instruction, or during an interrupt acknowledge cycle. It is a status signal to the other bus masters in a multiprocessor system, inhibiting them from using the system bus during this time.

The output of this signal is three state and becomes high impedance during hold acknowledge. $\overline{\text{BUSLOCK}}$ is high during standby mode except if the HALT instruction has a BUSLOCK prefix.

$\overline{RQ}/\overline{AK_1}$, $\overline{RQ}/\overline{AK_0}$ (Hold Request/Acknowledge)

For large-scale systems.

These pins function as bus hold request inputs ($\overline{RQ}$) and as bus hold acknowledge outputs ($\overline{AK}$). $\overline{RQ}/\overline{AK_0}$ has a higher priority than $\overline{RQ}/\overline{AK_1}$.

These pins have three-state outputs with on-chip pull-up resistors which keep the pin at a high level when the output is high impedance.

Bus Hold Request Input (RQ) must be input in synchronization with external clock signals to satisfy the setup/hold time for normal operation.

V_{DD} (Power Supply)

For small- and large-scale systems.

This pin is used for the +5V power supply.

GND (Ground)

For small- and large-scale systems. This pin is used for ground.

IC (Internally Connected)

This pin is used for tests performed at the factor by SHARP. The LH70108 is used with this pin at ground potential.

■ Register Configuration

Program Counter (PC)

The program counter is a 16-bit binary counter that contains the segment offset address of the next instruction which the EXU is to execute.

The PC increments each time the microprogram fetches an instruction from the instruction queue. A new location value is loaded into the PC each time a branch, call, return, or break instruction is executed. At this time, the contents of the PC are the same as the Prefetch Pointer (PFP).

Prefetch Pointer (PFP)

The prefetch pointer (PFP) is a 16-bit binary counter which contains a segment offset which is used to calculate a program memory address that the bus control unit (BCU) uses to prefetch the next byte for the instruction queue. The contents of PFP are an offset from the PS (Program Segment) register.

The PFP is incremented each time the BCU prefetches an instruction from the program memory. A new location will be loaded into the PFP whenever a branch, call, return, or break instruction is executed. At that time the contents of the PFP will be the same as those of the PC (Program Counter).

Segment Registers (PC, SS, DS_0, and DS_1)

The memory addresses accessed by the LH70108 are divided into 64K-byte logical segments. The starting (base) address of each segment is specified by a 16-bit segment register, and the offset from this starting address is specified by the contents of another register or by the effective address.

These are the four types of segment registers used.

Segment Register	Default Offset
PS (Program Segment)	PFP
SS (Stack Segment)	PS, effective address
DS_0 (Data Segment 0)	IX, effective address
DS_1 (Data segment 1)	IY

General-Purpose Registers (AW, BW, CW, and DW)

There are four 16-bit general-purpose registers. Each one can be used as one 16-bit register or as two 8-bit registers by dividing them into their high and low bytes (AH, AL, BH, BL, CH, CL, DH, DL).

Each register is also used as a default register for processing specific instructions. The default assignments are:

AW: Word multiplication/division, word I/O, data conversion

AL: Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation

AH: Byte multiplication/division

BW: Translation

CW: Loop control branch, repeat prefix

CL: Shift instructions, rotation instructions, BCD operations

DW: Work multiplication/division, indirect addressing I/O

Pointers (SP, BP) and Index Registers (IX, IY)

These registers serve as base pointers or index registers when accessing the memory using based addressing, indexed addressing, or based indexed addressing.

These registers can also be used for data transfer and arithmetic and logical operations in the same manner as the general-purpose registers. They cannot be used as 8-bit registers.

Also, each of these registers acts as a default register for specific operations. The default assignments are:

SP: Stack operations

IX: Block transfer (source), BCD string operations

IY: Block transfer (destination), BCD string operations

Program Status Word (PSW)

The program status word consists of the following six status and four control flags.

Status Flags	Control Flags
• V (Overflow)	• MD (Mode)
• S (Sign)	• DIR (Direction)
• Z (Zero)	• IE (Interrupt Enable)

- AC (Auxiliary Carry) • BRK (Break)
- P (Parity)
- CY (Carry)

When the PSW is pushed on the stack, the word images of the various flags are as shown here.

PSW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD	1	1	1	V	DIR	IE	BRK	S	Z	0	AC	0	P	1	CY

The status flags are set and reset depending upon the result of each type of instruction executed.

Instructions are provided to set, reset, and complement the CY flag directly.

Other instructions set and reset the control flags and control the operation of the CPU.

Q_0-Q_3 (Prefetch Queue)

The LH70108 has 4 byte instruction queue (FIFO), and it can store up to 4 instruction byte prefetched by the BCU. The instruction codes stored in the queue are fetched and executed by the EXU. The queue is cleared and prefetched with branch, call, return, or break instruction has been executed and when external interrupt has been acknowledged. Normally, the LH70108 prefetches if the queue has one byte or more space. If the time required to prefetch the instruction code from the external memory is less than the mean execution time of instructions which are executed sequentially, then the actual instructions cycle will be shortened by this amount of time i. e. the instruction code to be next executed by the EXU can be available in the queue immediately after the completion of one instruction. As the result, processing speed is highly upgraded compared with the conventional CPU which fetch and execute instructions one by one. Queuing effect is lowered if there were many instructions which clears queue like the branch instruction or in the case of continuous instructions with too short instruction time.

DP (Data Pointer)

The data pointer is a 16-bit register indicates read/write addresses of variables. Effective address made in the effective address generator and the register contents including memory address offsets are transferred to the DP.

TEMP (Temporary Communication Register)

This is a 16-bit temporary register used by communications between external data bus and the EXU. The TEMP can be read or written by upper byte or lewer byte independently for byte access. Basically, the EXU completes write operation with transferring data to the TEMP and completes read operation with recognizing the data has been transferred to the TEMP from external data bus.

EAG (Effective Address Generator)

The Effective Address Generator (EAG) performs high-speed effective address calculation necessary for memory access. This completes all the calculations with 2 clocks for every addressing mode.

This fetches the instruction byte (2nd or 3rd byte) which has operand specifying field, if the instruction needs memory access. Then calculates effective address and transfers it to the DP (Data Pointer) and generates control signals relating to handling ALU and corresponding registers. In addition, if it is necessary, the EAG requests to the BCU for starting the bus cycle (memory read).

Instruction Decoder

The Instruction Decoder classifies 1st byte of an instruction code into some groups with specific function and holds them during micro-instruction execution.

Microaddress Register

The microaddress register specifies the address of a microinstruction ROM to be next executed. At starting of a microinstruction execution, the 1st byte of instruction bytes stored in the queue is fetched in this register and it specifies a start address of the corresponding microinstruction sequence.

Microinstruction ROM

The Microinstruction ROM has 1024 words by 29 bits of micro-instructions.

Microinstruction Sequencer

The Microinstruction Sequencer controls the microaddress register operation, microinstruction ROM output, and synchronizing the EXU with BCU.

ADM (Address Modifier)

Address Modifier performs the generation of physical address (adding segment register and PFP or DP) and increment of PFP (Prefetch Pointer).

TA/TB (Temporary Register/Shifter A, B)

The TA/TB are 16-bit temporary register/shifter used with execution of multiply/divided and shift/rotate (including BCD rotate) instructions. When executing multiply or divide instruction TA +TB operates as a 32-bit temporary register/shifter when executing shift/rotate instructions. Both the TA and TB can be read or written to and from the internal bus by upper byte or lower byte independently. The contents of the TA and TB are input to the ALU.

TC (Temporary Register C)

The TC is a 16-bit temporary register used with internal processing like the multiply or divide operation, etc. The TC content is output to the ALU.

ALU (Arithmetic & Logic Unit)

The Arithmetic and Logic Unit consists of a full adder and logical operation circuit and performs these operations:

1) Arithmetic operation (Add, Subtract, Multiply, Divide, increment, decrement, and complement)
2) Logical operation (test, AND, OR, XOR and bit test, set, clear, and complement)

LC (Loop Counter)

The Loop Counter (LC) is a 16-bit register which counts below items.

1) Loop number of the primitive block transfer and input/output instructions (MOVBK, OUTM, etc.) controlled with repeat prefix instructions (REP, REPC, etc.).
2) Shift number of the multi-bit shift/rotate instructions.

High-Speed Execution of Instructions

This section highlinghts the major architectural features that enhance the performance of the LH70108.

- Dual data bus in EXU
- Effective address generator
- 16/32-bit temporary registers/shifters (TA, TB)
- 16-bit loop counter
- PC and PFP

Dual Data Bus Method

To reduce the number of processing steps for instruction execution, the dual data bus method has been adopted for the LH70108 (figure 1). The two data buses (the main data bus and the subdata bus) are both 16 bits wide. For addition/subtraction and logical and comparison operations, processing time has been speeded up some 30% over single-bus systems.

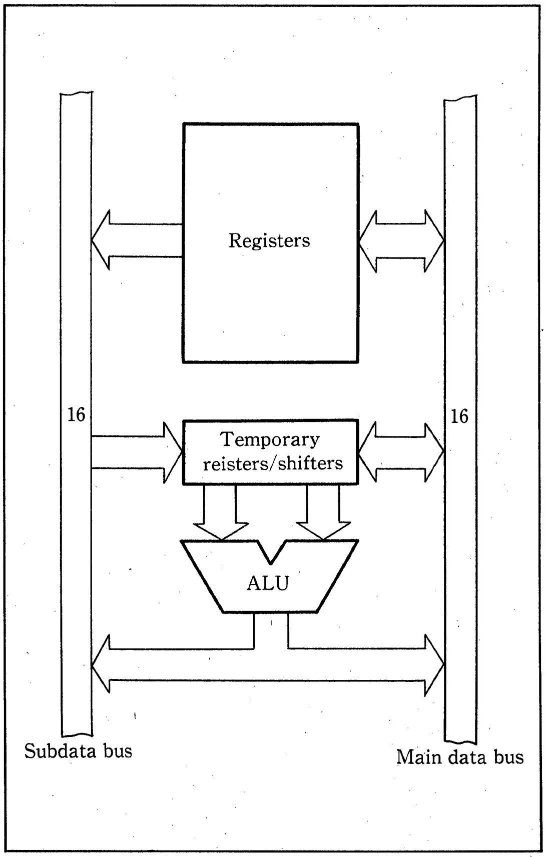


Fig. 1 Dual Data Buses

Example

ADD AW, BW; AW←AW+BW

Single Bus	**Dual Bus**
Step 1 TA←AW	TA←AW, TB←BW
Step 2 TB←BW	AW←TA+TB
Step 3 AW←TA+TB	

Effective Address Generator

This circuit (figure 2) performs high-speed processing to calculate effective addresses for accessing memory.

Calculation an effective address by the microprogramming method normally requires 5 to 12 clock cycles. This circuit requires only two clock cycles for address to be generated for any addressing mode. Thus, processing is several times faster.

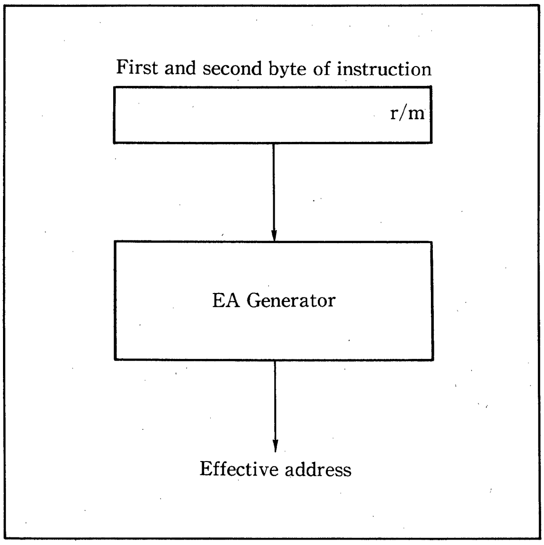


Fig. 2 Effective Address Generator

16/32-Bit Temporary Registers/Shifters (TA, TB)

These 16-bit temporary registers/shifters (TA, TB) are provided for multiplication/division and shift/rotation instructions.

These circuits have decreased the execution time of multiplication/division instructions. In fact, these instructions can be executed about four times faster than with the microprogramming method.

TA+TB: 32-bit temporary register/shifter for multiplication and divison instructions.

TB 16-bit temporary register/shifter for shift/rotation instructions.

Loop Counter (LC)

This counter is used to count the number of loops for a primitive block transfer instruction controlled by a repeat prefix instruction and the number of shifts that will be performed for a multiple bit shift/rotation instruction.

The processing performed for a multiple bit rotation of a register is shown below. The average speed is approximately doubled over the microprogram method.

Example

RORC AW, CL; CL=5

Microprogram method	**LC method**
8+(4×5)=28 clocks	7+5=12 clocks

Program Counter and Prefetch Pointer (PC and PFP)

The LH70108 microprocessor has a program counter, (PC) which addresses the program memory location of the instruction to be executed next, and a prefetch pointer (PFP), which addresses the program memory location to be accessed next. Both functions are provided in hardware. A time saving of several clocks is realized for branch, call, return, and break instruction execution, compared with microprocessors that have only one instruction pointer.

■ Unique Instructions

Variable Length Bit Field Operation Instructions

This category has two instructions: INS (insert Bit Field) and EXT (Extract Bit Field). These instructions are highly effective for computer graphics and high-level languages. They can, for example, be used for data structures such as packed arrays and record type data used in PASCAL.

(1) INS reg8, reg8/INS reg8, imm4

This instruction (figure 3) transfers low bits from the 16-bit AW register (the number of bits is specified by the second operand) to the memory location specified by the segment base (DS_1 register) pulse the byte offset (IY register). The starting bit position within this byte is specified as an offset by the lower 4 bits of the first operand.

After each complete data transfer, the IY register and the register specified by the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may specify the number of bits transferred (second operand). Because the maximum transferable bit length is 16 bits, only the lower 4 bits of the specified register (00H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

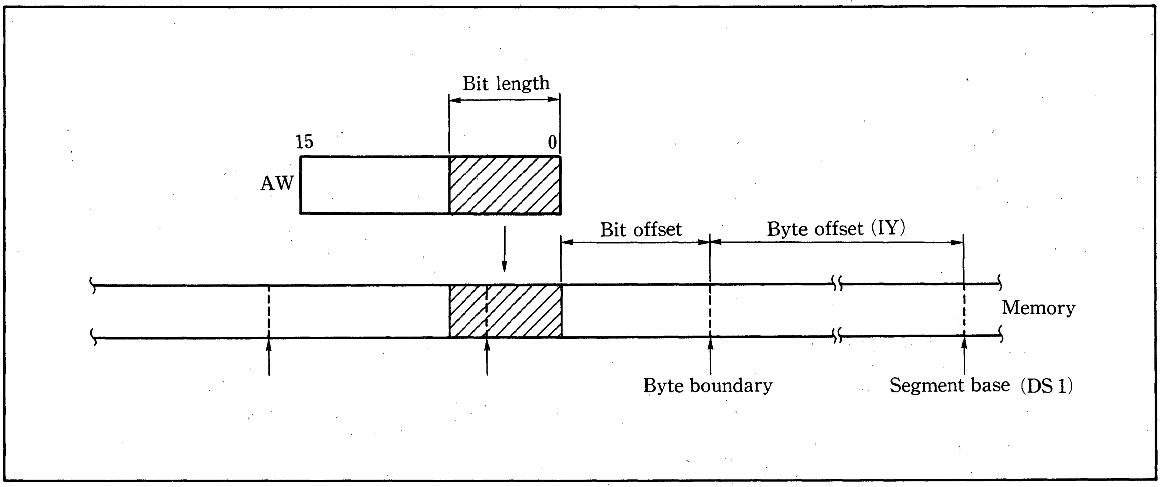


Fig. 3 Bit Field Insertion

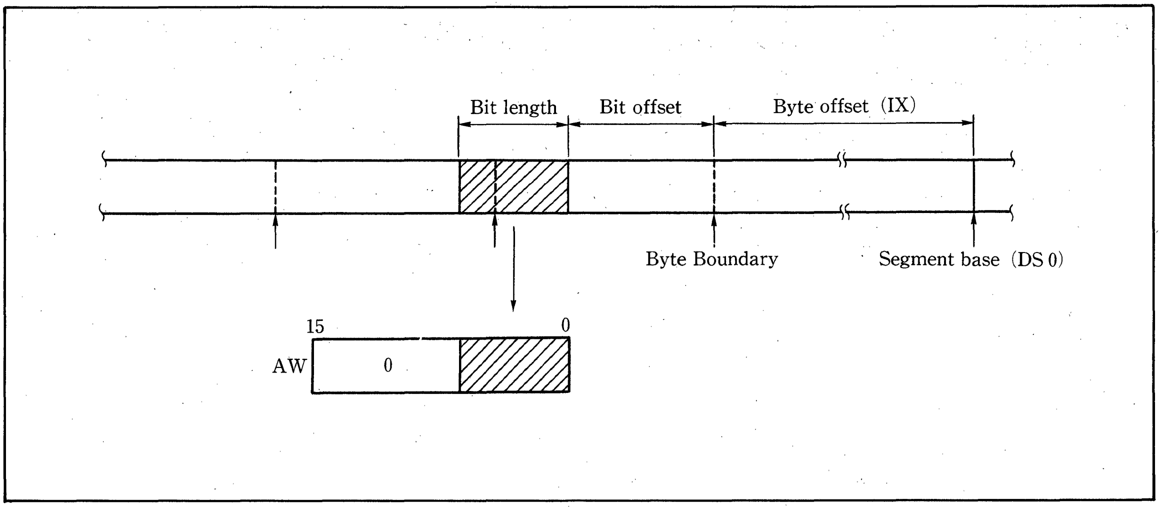


Fig. 4 Bit Field Extraction

(2) EXT reg8, reg8/EXT reg8, imm4

This instruction (figure 4) loads to the AW register the bit field data whose bit length is specified by the second operand of the instruction from the memory location that is specified by the DS_0 segment register (segment base), the IX index register (byte offset), and the lower 4 bits of the first operand (bit offset).

Packed BCD Operation Instructions

The instructions described here process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte-format operands (ROR4, ROL4).

After the transfer is complete, the IX register and the lower 4 bits of the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may be specified for the second operand. Because the maximum transferrable bit length is 16 bits, however, only the lower 4 bits of the specified register (0H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

(1) ADD4S

This instruction adds the packed BCD string addressed by the IX index register to the packed BCD string addressed by the IY index register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

BCD string (IY, CL)←BCD string (IY, CL)+BCD string (IX, CL)

(2) SUB4S

This instruction subtracts the packed BCD string addressed by the IX index registér from the packed BCD string addressed by the IY register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

BCD string (IY, CL)←BCD string (IY, CL)-BCD String (IX, CL)

(3) CMP4S

This instruction performs the same operation as SUB4S except that the result is not stored and only the overflow (V), carry flags (CY) and zero flag (Z) are affected.

BCD string (IY, CL)-BCD string (IX, CL)

(4) ROL4

This instruction (figure 5) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4 bits of the AL register (AL_L) to rotate that data one BCD digit to the left.

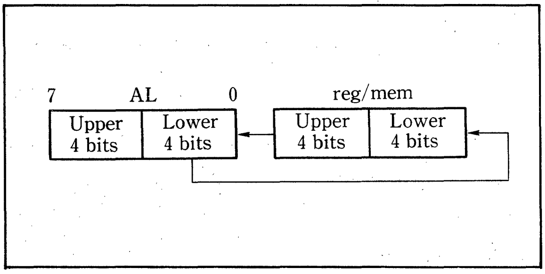


Fig. 5 BCD Rotate Left (ROL4)

(5) ROR4

This instruction (figure 6) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4 bits of the AL register (AL_L) to rotate the data one BCD digit to the right

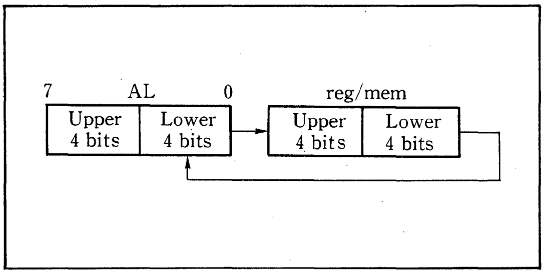


Fig. 6 BCD Rotate Right (ROR4)

Stack Operation Instruction

(1) PREPARE imm16, imm8

This instruction is used to generate "stack frames" required by the block structures of high-level languages such as Pascal and Ada. The stack frame includes a local variable area as well as pointers. These frame pointers point to the frame containing the variables that can be referenced from the current procedure.

The program example based upon Pascal language is shown below.

```
program EXAMPLE;
  procedure P;
    var a, b, c;
    procedure Q;
      var d, e;
      procedure R;
        var f, g;
        begin
          d:=a+f+g;
        end:
      begin
        R;
        b:=d+e;
      end;
    begin
      a:=b+c;
      Q;
    end;
(*main program*)
  begin
    P;
  end.
```

Note: The variables are defined as the words.

This program is an example of a procedure block with a triple nesting.

Procedure	Variables
P	a, b, c
Q	d, e
R	f, g

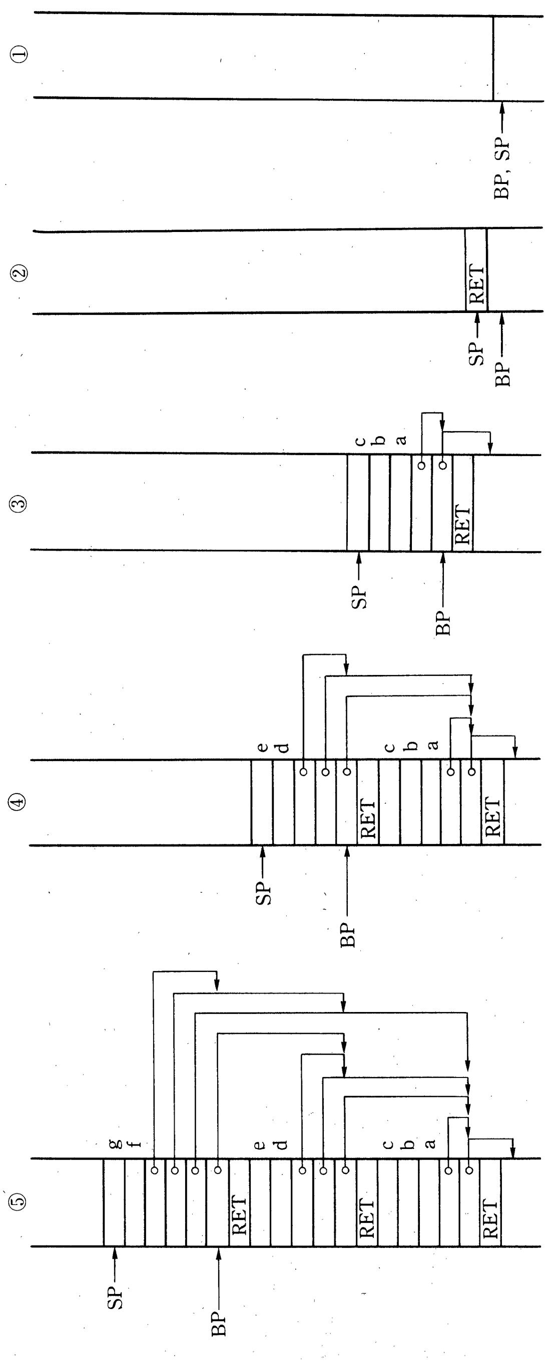
Accordingly, the global variables of a, b and c are referenced from the procedure Q, and a, b, c, d and e from the procedure R.

This instruction copies frame-pointers to reserve the local variable area and to enable global variable references. The first operand (16-bit immediate data) specifies (in bytes) the size of the local variable area. The second operation (8-bit immediate data) specifies the depth (or lexical level) of the procedure block. The frame base address generated by this instruction is set in the BP base pointer.

To compile the EXAMPLE program follows the assembler program shown next. (The DISPOSE instruction in the assembler program is used to return the stack pointer SP and the base pointer BP to the state just before the PREPARE instruction is executed. See DISPOSE section mentioned later.)

```
START: MOV      SP, SPTOP
       MOV      BP, SP               ;①
       CALL     P                    ;②
       BR       SYSTEM
P:     PREPARE  6, 1                 ;③
       MOV      AW, 〔BP〕〔B+BLEVEL*2〕
       ADD      AW, 〔BP〕〔C+CLEVEL*2〕
       MOV      〔BP〕〔A+ALEVEL*2〕, AW
       CALL     Q
       DISPOSE
       RET
Q:     PREPARE  4, 2                 ;④
       CALL     R
       MOV      AW, 〔BP〕〔D+DLEVEL*2〕
       ADD      AW, 〔BP〕〔E+ELEVEL*2〕
       MOV      IY, 〔BP〕〔BLEVEL*2〕
       MOV      SS: 〔IY〕〔B+BLEVEL*2〕, AW
       DISPOSE
       RET
R:     PREPARE  4, 3                 ;⑤
       MOV      AW, 〔BP〕〔F+FLEVEL*2〕
       ADD      AW, 〔BP〕〔G+GLEVEL*2〕
       MOV      IY, 〔BP〕〔ALEVEL*2〕
       ADD      AW, SS: 〔IY〕〔A+ALEVEL*2〕
       MOV      IY, 〔BP〕〔DLEVEL*2〕
       MOV      SS: 〔IY〕〔D+DLEVEL*2〕, AW
       DISPOSE
       RET
```

```
;A=-2     ALEVEL=-1
;B=-4     BLEVEL=-1
;C=-6     CLEVEL=-1
;D=-2     DLEVEL=-2
;E=-4     ELEVEL=-2
;F=-2     FLEVEL=-3
;G=-4     GLEVEL=-3
```



The process of the generation of the stack frame according to the program is shown next. The numbers are referred to that in the program.

First the old BP value is saved to the stack. This is done so that BP of the calling procedure can be restored when the called procedure terminates. The frame pointer (BP value saved to the stack) that indicates the range of variables that can be referenced by the called procedure is placed on the stack. This range is always a value one less than the lexical level of the procedure.

If the lexical level of a procedure is greater than 1, the pointers of that procedure will also be saved on the stack. This is so that the frame pointer of the calling procedure can also be copied when frame pointer copy is performed within the called procedure.

Next the new frame pointer value is set in BP and the area for local variables used by the procedure is reserved in the stack. In other words, SP is decremented only for the amount of stack memory required by the local variables.

```
    display=2nd operand
    dynamics=1st operand
  SP=SP-2;
  (SP)=BP;
  temp=SP;
  if display>0 then begin
    repeat display-1 times
      begin
        SP=SP-2;
        BP=BP-2;
        (SP)=(BP);
      end;
    SP=SP-2;
    (SP)=temp;
    end;
BP=temp/
SP=SP-dynamics
```

Data Access

(1) Local variable access

The local variables are assigned in the frame of the procedure. The effective address EA. L of the local variables is defined by the formula:

EA. L=SS: (BP+offset)

The offset value is defined as the sum of the frame size (referenced frame base) and the variable from the base of the local variable area.

(2) Global variable access

The global variable is located at the address added by the offset of variables which are referenced to the accessed value of the base pointer of the old one saved on the stack frame.

The effective address EA. G is defined as below.

EA. G=SS: ((SS: (BP+offset 1))+offset 2)

The offset 1 is defined by the offset value from the frame base (BP) to the address stored by the base address of the frame including the global variables.

The offset 2 is defined by the offset value from the frame base including variables to be referenced to the variables.

DISPOSE

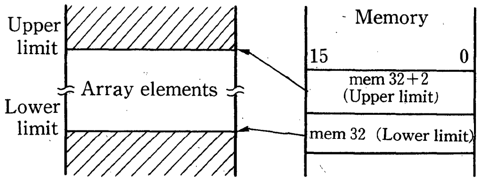
This instruction releases the last stack frame generated by the PREPARE instruction. It returns the stack and base pointers to the values they had before the PREPARE instruction was used to call a procedure.

```
SP=BP;
BP=(SP);
SP=SP+2
```

Check Array Boundary Instruction

This instruction is used to verify that index values pointing to the elements of an array data structure are within the defined range. The lower limit of the array should be in memory location mem32, the upper limit in mem32+2. If the index value in reg16 is not between these limits when CHKIND is executed, a BRK 5 will occur. This causes a jump to the location in interrupt vector 5.

```
CHKIND reg16, mem32
When (mem32)>reg16 or (mem32+2)<reg16
  TA←(015H, 014H)                     )
  TC←(017H, 016H)                     |
  SP←SP-2, (SP+1, SP)←PSW             |
  IE←0, BREK←0                        |
  SP←SP-2, (SP+1, SP)←PS              } =BRK 5
  PS←TC                               |
  SP←SP-2, (SP+1, SP)←PC              |
  PC←TA                               )
```



Mode Operating Instructions

The LH70108 has two operating modes (figure 7). One is the native mode, and the other is the emulation mode in which the instruction set of the 8080A is emulated. A mode flag (MD) is provided to select between these two modes. Native mode is selected when MD is 1 and emulation mode when MD is 0. MD is set and reset, directly and indirectly, by executing the mode manipulation instructions.

Two instructions are provided to switch operation from the native mode to the emulation mode and back: BRKEM (Break for Emulation), and RETEM (Return from Emulation).

Two instructions are used to switch from the emulation mode to the native mode and back: CALLN (Call Native Routine), and RETI (Return from Interrupt).

The system will return from the 8080 emulation mode to the native mode when the RESET signal is present, or when an external interrupt (NMI or INT) is present.

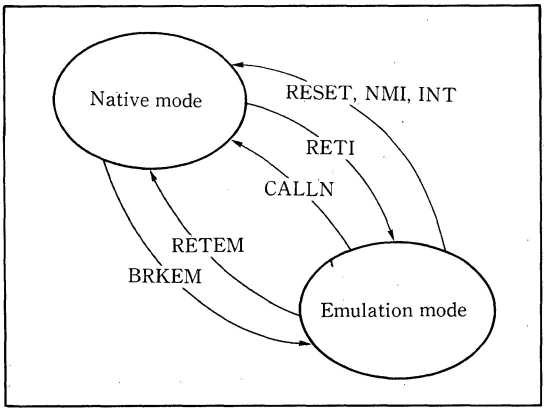


Fig. 7 V20 Modes

(1) BRKEN imm8

This is the basic instruction used to start the 8080 emulation mode. This instruction operates exactly the same as the BRK instruction, except that BRKEM resets the mode flag (MD) to 0. PSW, PS, and PC are saved to the stack. MD is then reset and the interrupt vector specified by the operand imm8 of this command is loaded into PS and PC.

The instruction codes of the interrupt processing routine jumped to are then fetched. Then the CPU executes these codes as 8080A instructions.

In 8080 emulation mode, registers and flags of the 8080A are performed by the following registers and flags of the LH70108.

In the native mode, SP is used for the stack pointer. In the 8080 emulation mode this function is performed by BP.

This use of independent stack pointers allows independent stack areas to be secured for each mode

	8080A	LH70108
Registers:	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
Flags:	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

and keeps the stack of one of the modes from being destroyed by an erroneous stack operation in the other mode.

The SP, IX, IY and AH registers and the four segment registers (PS, SS, DS_0, and DS_1) used in the native mode are not affected by operations in 8080 emulation mode.

In the 8080 emulation mode, the segment register for instructions is determined by the PS register (set automatically by the interrupt vector) and the segment register for data is the DS_0 register (set by the programmer immediately before the 8080 emulation mode is entered).

It is prohibited to nest BRKEM instructions.

(2) RETEM (no operand)

When RETEM is executed in 8080 emulation mode (interpreted by the CPU as a 8080A instruction), the CPU restores PS, PC, and PSW (as it would when returning from an interrupt processing routine), and returns to the native mode. At the same time, the contents of the mode flag (MD) which was saved to the stack by the BRKEM instruction, is restored to MD=1. The CPU is set to the native mode.

(3) CALLN imm8

This instruction makes it possible to call the native mode subroutines from the 8080 emulation mode. To return from subroutine to the emulation mode, the RETI instruction is used.

The processing performed when this instruction is executed in the 8080 emulatiốn mode (it is interpreted by the CPU as 8080A instruction), is similar to that performed when a BRK instruction is executed in the native mode. The imm8 operand specifies an interrupt vector type. The contents of PS, PC, and PSW are pushed on the stack and an MD flag value of 0 is saved. The mode flag is set to 1 and the interrupt vector specified by the operand is loaded into PS and PC.

(4) RETI (no operand)

This is a general-purpose instruction used to return from interrupt routines entered by the BRK instruction or by an external interrupt in the native mode. When this instruction is executed at the end of a subroutine entered by the execution of the CALLN instruction, the operation that restores PS, PC, and PSW is exactly the same as the native mode execution. When PSW is restored, however, the 8080 emulation mode value of the mode flag (MD) is restored, the CPU is set in emulation mode, and all subsequent instructions are interpreted and executed as 8080A instructions.

RETI is also used to return from an initerrupt procedure initiated by an NMI or INT interrupt in the emulation mode.

■ Floating Point Operation Chip Instructions

FPO1 fp-op, mem/FPO2 fp-op, mem

These instructions are used for the external floating point processor. The floating point operation is passed to the floating point processor when the CPU fetches one of these instructions. From this point the CPU performs only the necessary auxiliary processing (effective address calculation, generation of physical addresses, and start-up of the memory read cycle).

The floating point processor always monitors the instructions fetched by the CPU. When it interprets one as an instruction to itself, it performs the appropriate processing. At this time, the floating point processor chip uses either the address alone or both the address and read data of the memory read cycle executed by the CPU. This difference in the data used depends on which of these instructions is executed.

Note: During the memory read cycle initiated by the CPU for FPO1 or FPO2 execution, the CPU does not accept any read data on the data bus from memory. Although the CPU generates the memory address, the data is used by the floating point processor.

■ Interrupt Operation

The interrupts used in the LH70108 can be divided into two types: interrupts generated by external interrupt requests and interrupts generated by software processing. These are the classifications.

(1) External Interrupts

(a) NMI input (nonmaskable)
(b) INT input (maskable)

(2) Software Processing

As the result of instruction execution

- ●When a divide error occurs during execution of the DIV or DIVU instruction
- ●When a memory-boundary-over error is detected by the CHKIND instruction

Conditional break instruction

- ●When V=1 during execution of the BRKV instruction

Unconditional break instructions

- ●1-byte break instruction: BRK3
- ●2-byte break instruction: BRK imm8

Flag processing

- ●When stack operations are used to set the BRK flag

8080 Emmulation mode instructions

- ●BRKEM imm8
- ●CALLN imm8

Interrupt Vectors

Starting addresses for interrupt processing routines are either determined automatically by a single location of the interrupt vector table or selected each time interrupt processing is entered.

The interrupt vector table is shown in figure 8. The table uses 1K bytes of memory addresses 000H to 3FFH and can store starting address data for a maximum of 256 vectors (4 bytes per vector).

The corresponding interrupt sources for vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved. These vectors consequently cannot be used for general applications

The BRKEM instruction and CALLN instruction (in the emulation mode) and the INT input are available for general applications for vectors 32 to 255.

A single interrupt vector is made up of 4 bytes (figure 9). The 2 bytes in the low addresses of memory are loaded into PC as the offset, and the high 2 bytes are loaded into PS as the base address. The bytes are combined in reverse order. The lower-order bytes in the vector become the most significant bytes in the PC and PS, and the higher-order bytes become the least significant bytes.

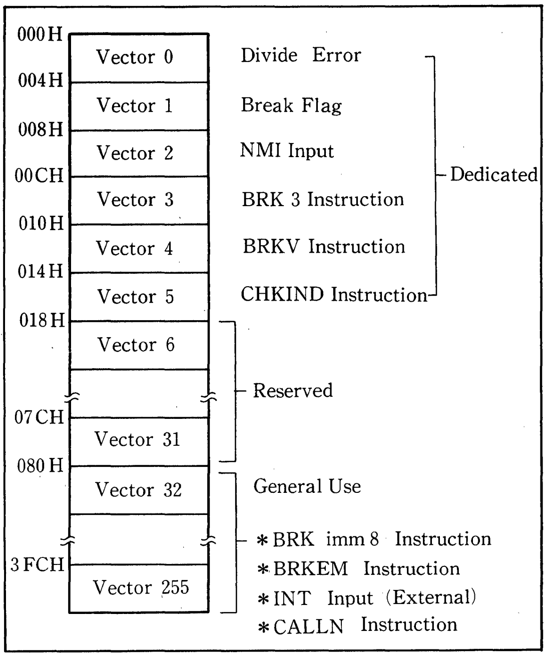


Fig. 8 Interrupt Vector Table

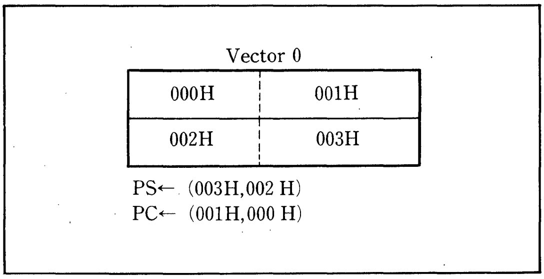


Fig. 9 Interrupt Vector 0

Based on this format, the contents of each vector should be initialized at the beginning of the program.

The basic steps to jump to an interrupt processing routine are now shown.

TA←vector low bytes (offset)
TC←vector high bytes (segment base)
SP←SP-2, (SP+1, SP)←PSW
IE←0, BRK←0, MD←0
SP←SP-2, (SP+1, SP)←PS
PS←TC
SP←SP-2, (SP+1, SP)←PC
PC←TA

Standby Function

The LH70108 has a standby mode to reduce power consumption during program wait states. This mode is set by the HALT instruction in both the native and the emulation mode.

In the standby mode, the internal clock is supplied only to those circuits related to functions required to release this mode and bus hold control functions. As a result, power consumption can be reduced to 1/10 the level of normal operation in either native or emulation mode.

The standby mode is released by inputting a RESET signal or an external interrupt (NMI, INT).

The bus hold function is effective during standby mode. The CPU returns to standby mode when the bus hold request is removed.

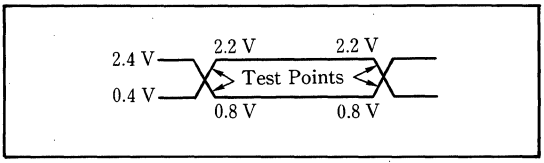
During standby mode, all control outputs are disabled and the address/data bus will be at either high or low levels.

I/O Address Reservation

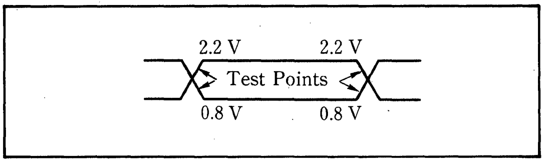
Reserve upper 256 bytes of I/O address (FF00H-FFFH) in case it may be used in future.

Timing Diagram

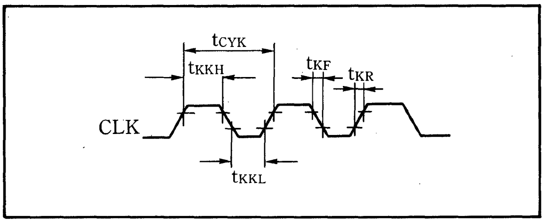
(1) AC test input waveform (Except CLK)



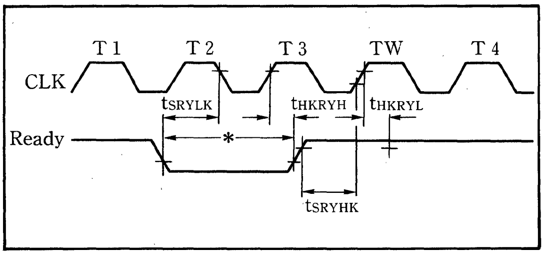
(2) AC output test points



(3) Clock timing

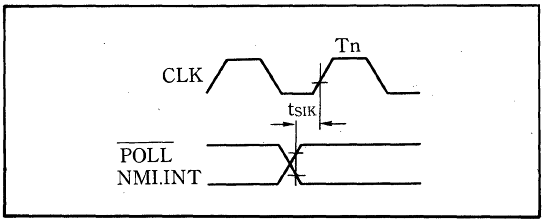


(4) Wait (Ready) timing

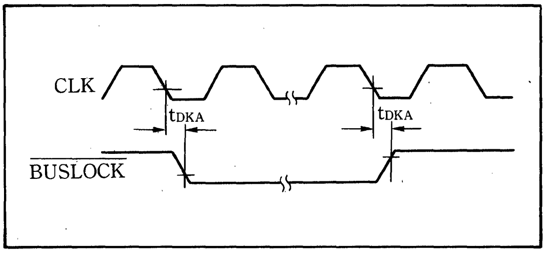


*Read signal must be held at LOW or HIGH during this period.

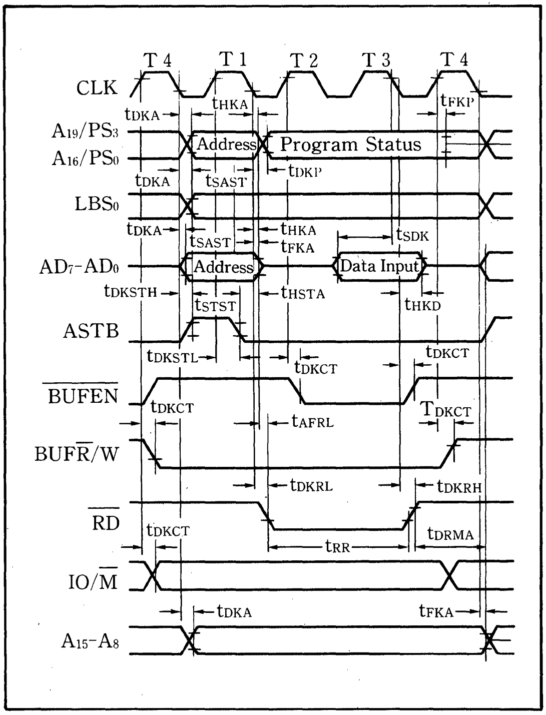
(5) POLL, NMI, INT input timing



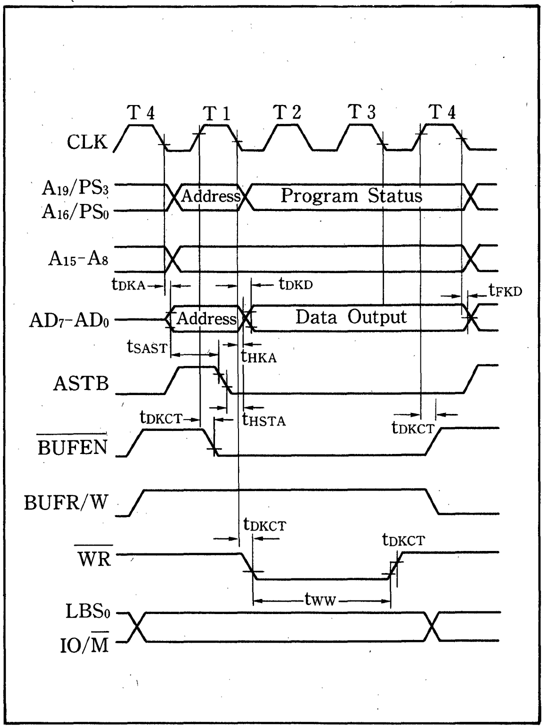
(6) BUSLOCK output timing



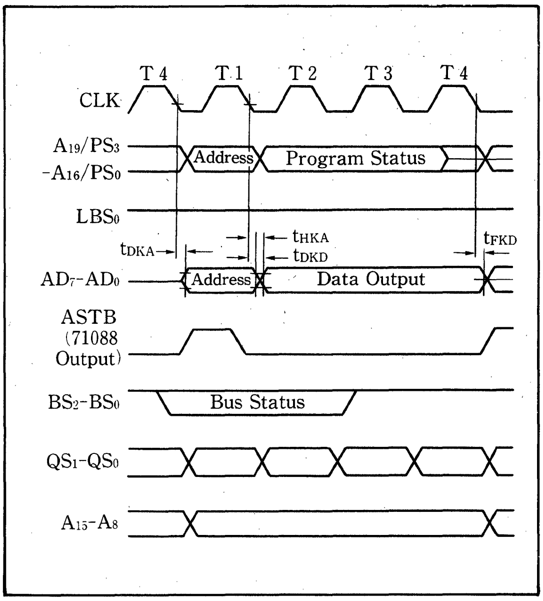
(7) Read timing (small scale)



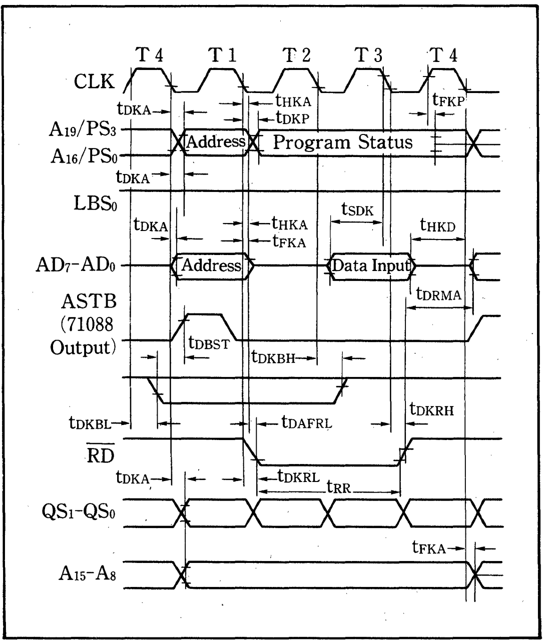
(8) Write timing (small scale)



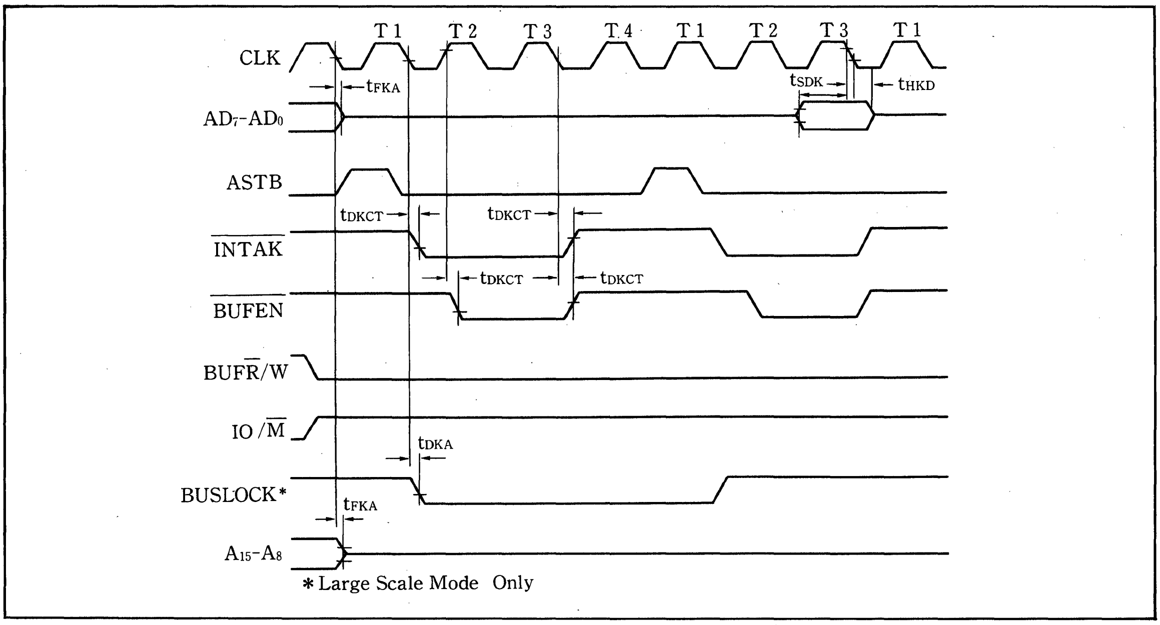
(10) Write timing (large scale)



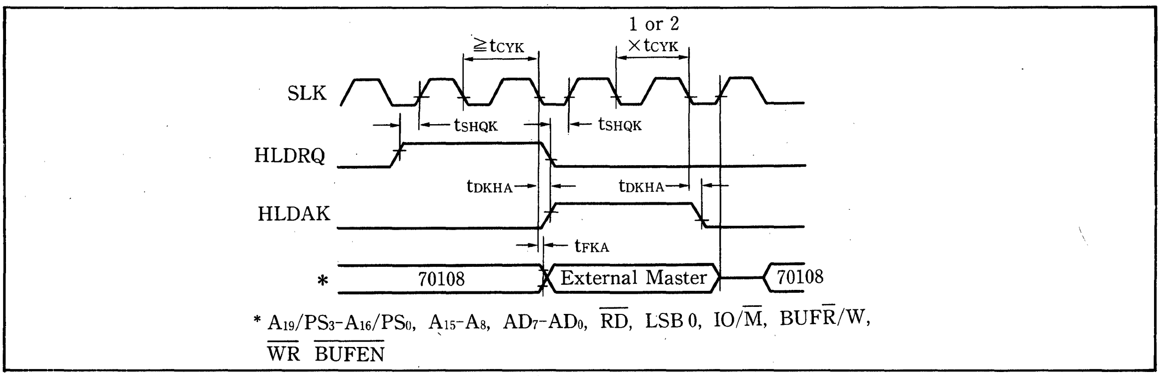
(9) Read timing (large scale)



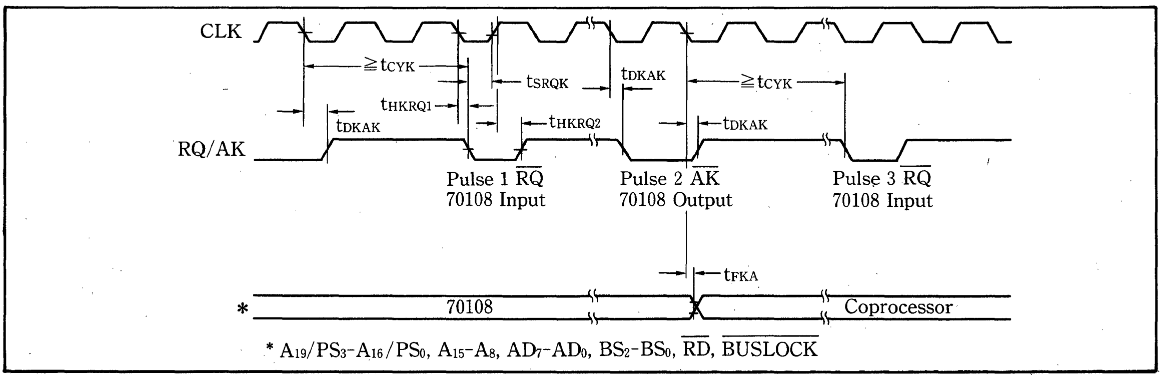
(11) Interrupt acknowledge timing



(12) Hold request/acknowledge timing (small scale)



(13) Bus request/acknowledge timing (large scale)



■ Instruction Set

The following tables briefly describe the LH70108's instruction set.

□ Operation and Operand Type-defines abbreviations used in the Instruction Set table.

□ Flag Operations-defines the symbols used to describe flag operations.

□ Memory Addressing-shows how mem and mod-combinations specify memory addressing modes.

□ Selection of 8-and 16-Bit Registers-shows how reg and W select a register when mod=111.

□ Selection of Segment Registers-shows how sreg selects a segment register.

□ Instruction Set-shows the instruction mnemonics, their sffect, their operation codes the number of bytes in the instruction, the number of clocks required for execution, and the effect on the LH70108 flags.

Table 1 Operand Types

Identifier	Description
reg	8- or 16-bit general-purpose register
reg8	8-bit general-pourpose register
reg16	16-bit general-purpose register
dmem	8- or 16-bit direct memory location
mem	8- or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory location
mem32	32-bit memory location
imm	Constant (0 to FFFFH)
imm16	Constant (0 to FFFFH)
imm8	Constant (0 to FFH)
imm4	Constant (0 to FH)
imm3	Constant (0 to 7)
acc	AW or AL register
sreg	Segment register
src-table	Name of 256-byte translation table
src-block	Name of block addressed by the IX register
dst-block	Name of block addressed by the IY register
near-proc	Procedure within the current program segment
far-proc	Procedure located in another program segment
near-label	Label in the current program segment
short-label	Label between −128 and +127 bytes from the end of instruction
far-label	Label in another program segment
memptr16	Word containing the offset of the memory location whin the current program segment to which control is to be transferred
memptr32	Double word containing the offset and segment base address of the memory location to which control is to be transferred
regptr16	16-bit register containing the offset of the memory location within the program segment to which control is to be transferred
pop-value	Number of bytes of the stack to be discarded (0 to 64K bytes, usually even addresses)
fp-op	Immediate data to identify the instruction code of the external floating point operation

Table 2 Operation Code Types

Identifier	Description
R	Register set
W	Word/byte field (0 to 1)
reg	Register field (000 to 111)
mem	Memory field (000 to 111)
mod	Mode field (00 to 10)
S : W	When S : W=01 or 11, data=16 bits. At all other times, data=8 bits.
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip

Table 3 Operational Description

Identifier	Description
AW	Accumulator (16 bits)
AH	Accumulator (high byte)
AL	Accumulator (low byte)
BW	BW register (16 bits)
CW	CW register (16 bits)
CL	CW register (low byte)
DW	DW register (16 bits)
BP	Base pointer (16 bits)
SP	Stack pointer (16 bits)
PC	Program counter (16 bits)
PSW	Program status word (16 bits)
IX	Index register (source)(16 bits)
IY	Index register (destination)(16 bits)
PS	Program segment register (16 bit)
SS	Stack segment register (16 bits)
DS_0	Data segment 0 register (16 bits)
DS_1	Data segment 1 register (16 bits)
AC	Auxiliary carry flag
CY	Carry flag
P	Parity flag
S	Sign flag
Z	Zero flag
DIR	Direction flag
IE	Interrupt enable flag
V	Overflow flag
BRK	Break flag
MD	Mode flag
(…)	Values in parentheses are memory contents
disp	Displacement (8 or 16 bits)
ext-disp8	16-bit displacement (sign-extension byte+8-bit displacement)
temp	Temporary register (8/16/32 bit) TA: Temporary resister A (16 bits) TB: Temporary resister B (16 bits) TC: Temporary resister C (16 bits)

Identifier	Description
tmpcy	Temporary carry flag (1 bit)
seg	Immediate segment data (16 bits)
offset	Immediate offset data (16 bit)
←	Transfer direction
+	Addition
−	Subtraction
×	Multiplication
÷	Division
%	Modulo
AND A	Logical product
OR V	Logical sum
XOR ∀	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value

Table 4 Flag Operations

Identifier	Description
(blank)	No change
0	Cleared to 0
1	Set to 1
X	Set or cleared according to the result
U	Undefined
R	Value saved earlier is restored

Table 5 Memory Addressing

	mod		
men	00	01	10
000	BW+IX	BW+IX+disp8	BW+IX+disp16
001	BW+IY	BW+IY+disp8	BW+IY+disp16
010	BP+IX	BP+IX+disp8	BP+IX+disp16
011	BP+IY	BP+IY+disp8	BP+IY+disp16
100	IX	IX+disp8	IX+disp16
101	IY	IY+disp8	IY+disp16
110	Direct address	BP+disp8	BP+disp16
111	BW	BW+disp8	BW+disp16

Table 6 Selection of 8- and 16-Bit Registers (mod 11)

reg	W=0	W=1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

Table 7 Selection of Segment Registers

sreg	
00	DS_1
01	PS
10	SS
11	DS_0

The table on the following pages shows the instruction set.

At "No. of Clocks," for instructions referencing memory operands, the left side of the slash (/) is the number of clocks for byte opereands and the right side is for word operands. For conditional control transfer instructions, the left side of the slash (/) is the number of clocks if a control transfer takes place. The right side is the number of clocks when no control transfer or branch occurs. Some instructions show a range of clock times, separated by a hyphen. The execution time of these instructions varies from the minimum value to the maximum, depending on the operands involved.

"No. of Clocks"includes these times:

- Decoding
- Effective address generation
- Operand fetch
- Execution

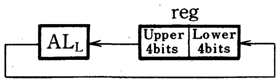
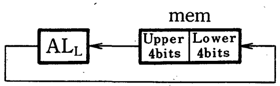
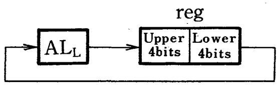
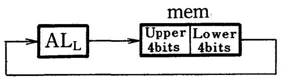
It assumes that the instruction bytes have been prefetched.

Mnemonic	Operand	Operation	Operation Code																No. of	No. of	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	AC	CY	V	P	S	Z
Data Transfer Instructions																										
MOV	reg, reg	reg←reg	1	0	0	0	1	0	1	W	1	1		reg			reg		2	2						
	mem, reg	(mem)←reg	1	0	0	0	1	0	0	W	mod			reg			mem		9/13	2-4						
	reg, mem	reg←(mem)	1	0	0	0	1	0	1	W	mod			reg			mem		11/15	2-4						
	mem, imm	(mem)←imm	1	1	0	0	0	1	1	W	mod		0	0	0		mem		11/15	3-6						
	reg, imm	reg←imm	1	0	1	1	W		reg										4	2-3						
	acc, dmem	When W= 0 AL←(dmem) When W= 1 AH←(dmem+1), AL←(dmem)	1	0	1	0	0	0	0	W									10/14	3						
	dmen, acc	When W=0 (dmem)←AL When W=1 (dmem+1)←AH,(dmem)←AL	1	0	1	0	0	0	1	W									9/13	3						
	sreg, reg16	sreg←reg16 sreg: SS, DS0, DS1	1	0	0	0	1	1	1	0	1	1	0		sreg		reg		2	2						
	sreg,mem16	sreg←(mem16) sreg: SS, DS0, DS1	1	0	0	0	1	1	1	0	mod		0		sreg		mem		15	2-4						
	reg16, sreg	reg16←sreg	1	0	0	0	1	1	0	0	1	1	0		sreg		reg		2	2						
	mem16, sreg	(mem16)←sreg	1	0	0	0	1	1	0	0	mod		0		sreg		mem		14	2-4						
	DS0, reg16, mem32	reg16←(mem32) DS0←(mem32+2)	1	1	0	0	0	1	0	1	mod			reg			mem		26	2-4						
	DS1, reg16, mem32	reg16←(mem32) DS1←(mem32+2)	1	1	0	0	0	1	0	0	mod			reg			mem		26	2-4						
	AH, PSW	AH←S, Z, x, AC, x, P, x, CY	1	0	0	1	1	1	1	1									2	1	x	x		x	x	x
	PSW, AH	S, Z, x, AC, x, P, x, CY←AH	1	0	0	1	1	1	1	0									3	1	x	x		x	x	x
LDEA	reg16, mem16	reg16←mem16	1	0	0	0	1	1	0	1	mod			reg			mem		4	2-4						
TRANS	src-table	AL←(BW+AL)	1	1	0	1	0	1	1	1									9	1						
XCH	reg, reg	reg↔reg	1	0	0	0	0	1	1	W	1	1		reg			reg		3	2						
	mem, reg or reg, mem	(mem)↔reg	1	0	0	0	0	1	1	W	mod			reg			mem		16/24	2-4						
	AW, reg16 or reg16,AW	AW↔reg16	1	0	0	1	0		reg										3	1						
Repeat Prefixes																										
REPC		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed. When CY≠1, exit the loop.	0	1	1	0	0	1	0	1									2	1						
REPNC		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed.When CY≠0, exit the loop.	0	1	1	0	0	1	0	0									2	1						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Repeat Prefixes (cont)																										
REP REPE REPZ		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, If is processed. If the primitive block transfer instruction is CMPBK or CMPM and Z≠1, exit the loop.	1	1	1	1	0	0	1	1									2	1						
REPNE REPNZ		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed.If the primitive block transfer instruction is CMPBK or CMPM and Z≠0, exit the loop.	1	1	1	1	0	0	1	0									2	1						
Primitive Block Transfer Instructions																										
MOVBK	dst-block, src-block	When W=0 (IY)←(IX) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX−1, IY←IY−1 When W=1 (IY+1,IY)←(IX+1,IX) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX−2, IY−2	1	0	1	0	0	1	0	W									11+8n 11+16n	1						
CMPBK	src-block, dst-block	When W=0 (IX)−(IY) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX−1, IY←IY−1 When W=1 (IX+1, IX)−(IY+1, IY) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX−2, IY←IY−2	1	0	1	0	0	1	1	W									7+14n 7+22n	1	x	x	x	x	x	x
CMPM	dst-block	When W=0 AL−(IY) DIR=0: IY←IY+1; DIR=1: IY←IY−1 When W=1 AW−(IY+1, IY) DIR=0: IY←IY+2; DIR=1: IY←IY−2	1	0	1	0	1	1	1	W									7+10n 7+14n	1	x	x	x	x	x	x
LDM	src-block	When W=0 AL←(IX) DIR=0: IX←IX+1; DIR=1: IX←IX−1 When W=1 AW←(IX+1, IX) DIR=0: IX←IX+2; DIR=1: IX←IX−2	1	0	1	0	1	1	0	W									7+9n 7+13n	1						
STM	dst-block	When W=0 (IY)←AL DIR=0: IY←IY+1; DIR=1: IY←IY−1 When W=1 (IY+1, IY)←AW DIR=0: IY←IY+2; DIR=1: IY←IY−2	1	0	1	0	1	0	1	W									7+4n 7+8n	1						

n: number of transfers

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags					
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC	CY	V	P	S	Z
Bit Field Instructions												
INS	reg8, reg8	16-Bit field←AW	0 0 0 0 1 1 1 1 1 1 reg reg	0 0 1 1 0 0 0 1	35-133	3						
	reg8, imm4	16-Bit field←AW	0 0 0 0 1 1 1 1 1 1 0 0 0 reg	0 0 1 1 1 0 0 1	35-133	4						
Bit Field Transfer Instructions (cont)												
EXT	reg8, reg8	AW←16-Bit field	0 0 0 0 1 1 1 1 1 1 reg reg	0 0 1 1 0 0 1 1	34-59	3						
	reg8, imm4	AW←16-Bit filed	0 0 0 0 1 1 1 1 1 1 0 0 0 reg	0 0 1 1 1 0 1 1	34-59	4						
I/O Instructions												
IN	acc, imm8	When W=0 AL←(imm8) When W=1 AH←(imm8+1), AL←(imm8)	1 1 1 0 0 1 0 W		9/13	2						
	acc, DW	When W=0 AL←(DW) When W=1 AH←(DW+1), AL←(DW)	1 1 1 0 1 1 0 W		8/12	1						
OUT	imm8, acc	When W=0 (imm8)←AL When W=1 (imm8+1)←AH, (imm8)←AL	1 1 1 0 0 1 1 W		8/12	2						
	DW, acc	When W=0 (DW)←AL When W=1 (DW+1)←AH, (DW)←AL	1 1 1 0 1 1 1 W		8/12	1						
Primitive I/O Instrctions												
INM	dst-block, DW	When W=0 (IY)←(DW) DIR=0: IY←IY+1; DIR=1: IY←IY−1 When W=1 (IY+1, IY)←(DW+1, DW) DIR=0: IY←IY+2; DIR=1: IY←IY−2	0 1 1 0 1 1 0 W		9+8n 9+16n	1						
OUTM	DW,src-block	When W=0 (DW)←(IX) DIR=0: IX←IX+1; DIR=1: IX←IX−1 When W=1 (DW+1, DW)←(IX+1, IX) DIR=0: IX←IX+2; DIR=1: IX←IX−2	0 1 1 0 1 1 1 W	n: number of transfers	9+8n 9+16n	1						
Addition/Subtraction Instructions												
ADD	reg, reg	reg←reg+reg	0 0 0 0 0 0 1 W	1 1 reg reg	2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)+reg	0 0 0 0 0 0 0 W	mod reg mem	16/24	2-4	x	x	x	x	x	x
	reg,mem	reg←reg+(mem)	0 0 0 0 0 0 1 W	mod reg mem	11/15	2-4	x	x	x	x	x	x
	regm imm	reg←reg+imm	1 0 0 0 0 0 S W	1 1 0 0 0 reg	4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)+imm	1 0 0 0 0 0 S W	mod reg mem	18/26	3-6	x	x	x	x	x	x
	acc,imm	When W=0 AL←AL+imm When W=1 AW←AW+imm	0 0 0 0 0 1 0 W		4	2-3	x	x	x	x	x	x
ADDC	reg, reg	reg←reg+reg+CY	0 0 0 1 0 0 1 W	1 1 reg reg	2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)+reg+CY	0 0 0 1 0 0 0 W	mod reg mem	16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg+(mem)+CY	0 0 0 1 0 0 1 W	mod reg mem	11/15	2-4	x	x	x	x	x	x
	reg, imm	reg←reg+imm+CY	1 0 0 0 0 0 S W	1 1 0 1 0 reg	4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)+imm+CY	1 0 0 0 0 0 S W	mod 0 1 0 mem	18/26	3-6	x	x	x	x	x	x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Addition/Subtraction Instructions (cont)																										
ADDC	acc, imm	When W=0 AL←AL+imm+CY When W=1 AW←AW+imm+CY	0	0	0	1	0	1	0	W									4	2-3	x	x	x	x	x	x
SUB	reg, reg	reg←reg−reg	0	0	1	0	1	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)−reg	0	0	1	0	1	0	0	W	mod			reg			mem		16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg−(mem)	0	0	1	0	1	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	reg, imm	reg←reg−imm	1	0	0	0	0	0	S	W	1	1	1	0	1		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)−imm	1	0	0	0	0	0	S	W	mod		1	0	1		mem		18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL−imm When W=1 AW←AW−imm	0	0	1	0	1	1	0	W									4	2-3	x	x	x	x	x	x
SUBC	reg, reg	reg←reg−reg−CY	0	0	0	1	1	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)−reg−CY	0	0	0	1	1	0	0	W	mod			reg			mem		16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg−(mem)−CY	0	0	0	1	1	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	
	reg, imm	reg←reg−imm−CY	1	0	0	0	0	0	S	W	1	1	0	1	1		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)−imm−CY	1	0	0	0	0	0	S	W	mod		0	1	1		mem		18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL−imm−CY When W=1 AW←AW−imm−CY	0	0	0	1	1	1	0	W									4	2-3	x	x	x	x	x	x
BCD Operation Instructions																										
ADD4S		dst BCD string←dst BCD string +src BCD string	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	7+19n	2	u	x	u	u	u	x
SUB4S		dst BCD string←dst BCD string −src BCD string	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	7+19n	2	u	x	u	u	u	x
CMP4S		dst BCD string−scrBCD string	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	7+19n	2	u	x	u	u	u	x
			n: number of BCD digits divided by 2																							
ROL4	reg8		0 1	0 1	0 0	0 0	1 0	1	1 reg	1	0	0	1	0	1	0	0	0	13	3						
	mem8		0 mod	0	0 0	0 0	1	1 mem	1	1	0	0	1	0	1	0	0	0	28	3-5						
ROR4	reg8		0 1	0 1	0 0	0 0	1 0	1	1 reg	1	0	0	1	0	1	0	1	0	17	3						
	mem8		0 mod	0	0 0	0 0	1 0	1	1 mem	1	0	0	1	0	1	0	1	0	32	3-5						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Increment/Decrement Instructions (cont)																										
INC	reg8	reg8←reg8+1	1	1	1	1	1	1	1	0	1	1	0	0	0	reg			2	2	x		x	x	x	x
	mem	(mem)←(mem)+1	1	1	1	1	1	1	1	W	mod		0	0	0	mem			16/24	2-4	x		x	x	x	x
	reg16	reg16←reg16+1	0	1	0	0	0	reg											2	1	x		x	x	x	x
DEC	reg8	reg8←reg8−1	1	1	1	1	1	1	1	0	1	1	0	0	1	reg			2	2	x		x	x	x	x
	mem	(mem)←(mem)−1	1	1	1	1	1	1	1	W	mod		0	0	1	mem			16/24	2-4	x		x	x	x	x
	reg16	reg16←reg16−1	0	1	0	0	1	reg											2	1	x		x	x	x	x
Multiplication Instructions																										
MULU	reg8	AW←AL×reg8 AH=0: CY←0, V←0 AH≠0: CY←1, V←1	1	1	1	1	0	1	1	0	1	1	1	0	0	reg			21-22	2	u	x	x	u	u	u
	mem8	AW←AL×(mem8) AH=0: CY←0, V←0 AH≠0: CY←1, V←1	1	1	1	1	0	1	1	0	mod		1	0	0	mem			27-28	2-4	u	x	x	u	u	u
	reg16	DW, AW←AW×reg16 DW=0: CY←0, V←0 DW≠0: CY←1,V←1	1	1	1	1	0	1	1	1	1	1	1	0	0	reg			29-30	2	u	x	x	u	u	u
	mem16	DW, AW←AW×(mem16) DW=0: CY←0, V←0 DW≠0: CY←1,V←1	1	1	1	1	0	1	1	1	mod		1	0	0	mem			39-40	2-4	u	x	x	u	u	u
MUL	reg8	AW←AL×reg8 AH=AL sign expansion: CY←0, V←0 AH≠AL sign expansion: CY←1, V←1	1	1	1	1	0	1	1	0	1	1	1	0	1	reg			33-39	2	u	x	x	u	u	u
	mem8	AW←AL×(mem)8 AH=AL sign expansion: CY←0, V←0 AH≠AL sign expansion: CY←1, V←1	1	1	1	1	0	1	1	0	mod		1	0	1	mem			39-45	2-4	u	x	x	u	u	u
	reg16	DW, AW←AW×reg8 DW=AW sign expansion: CY←0, V←0 DW≠AW sign expansion: CY←1, V←1	1	1	1	1	0	1	1	1	1	1	1	0	1	reg			41-47	2	u	x	x	u	u	u
	mem16	DW, AW←AW×(mem)8 DW=AW sign expansion: CY←0, V←0 DW≠AW sign expansion: CY←1, V←1	1	1	1	1	0	1	1	1	mod		1	0	1	mem			51-57	2-4	u	x	x	u	u	u
	reg16, (reg16,) imm8	reg16←reg16×imm8 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	1	1	1	1	reg			reg			28-34	3	u	x	x	u	u	u
	reg16, mem16, imm8	reg16←(mem16)×imm8 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	1	1	mod		reg			mem			38-44	3-5	u	x	x	u	u	u
	reg16, (reg16,) imm16	reg16←reg16×imm16 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	0	1	1	1	reg			reg			36-42	4	u	x	x	u	u	u
	reg16, mem16, imm16	reg16←(mem16)×imm16 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	0	1	mod		reg			mem			46-52	4-6	u	x	x	u	u	u

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Unsigned Division Instructions																										
DIVU	reg8	temp←AW When temp÷reg8≦FFH AH←temp%reg8, AL←temp÷reg8 When temp÷reg8>FFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	1	1	1	1	0		reg		19	2	u	u	u	u	u	u
	mem8	temp←AW When temp÷(mem8)≦FFH AH←temp%(mem8), AL←temp÷(mem8) When temp÷(mem8)>FFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, 1E←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	mod		1	1	0		mem		24	2-4	u	u	u	u	u	u
	reg16	temp←DW, AW When temp÷reg16≦FFFFH DW←temp%reg16, AW←temp÷reg16 When temp÷reg16>FFFFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	1	1	1	1	0		reg		25	2	u	u	u	u	u	u
	mem16	temp←DW, AW When temp÷(mem16)≦FFFFH DW←temp%(mem16),AW←temp÷(mem16) When temp÷(mem16)>FFFFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1,SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC,PC←TA	1	1	1	1	0	1	1	1	mod		1	1	0		mem		34	2-4	u	u	u	u	u	u
Signed Division Instructions																										
DIV	reg8	temp←AW When temp÷reg8>0, temp÷reg8≦7FH or temp÷reg8<0, temp÷reg8>0−7FH−1, AH←temp%reg8, AL←temp÷reg8 When temp÷reg8>0, temp÷reg8>7FH or temp÷reg8<0, temp÷temp8≦0−7FH−1, TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS,PS←TC SP←SP−2,(SP+1, SP)←PC,PC←TA	1	1	1	1	0	1	1	0	1	1	1	1	1		reg		29-34	2	u	u	u	u	u	u

Mnemonic	Operand	Operation	Operation Code																No. of	No. of	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	AC	CY	V	P	S	Z
Signed Division Instructions (cont)																										
DIV	mem8	temp←AW When temp÷(mem8)>0, temp÷(mem8)≦7FH or temp÷(mem8)<0, temp÷(mem8)>0−7FH−1 AH←temp%(mem8), AL←temp÷(mem8) When temp÷(mem8)>0, temp÷(mem8)>7FH or temp÷(mem8)<0, temp÷(mem8)≦0−7FH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	mod		1	1	1	mem			34-39	2-4	u	u	u	u	u	u
	reg16	temp←DW, AW When temp÷reg16>0, temp÷reg16≦7FFFH or temp÷reg16<0, temp÷reg16>0−7FFFH−1 DW←temp%reg16, AW←temp÷reg16 When temp÷reg16>0, temp÷reg16>7FFFH temp÷reg16<0, temp÷reg16≦0−7FFFH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	1	1	1	1	1	reg			38-43	2	u	u	u	u	u	u
	mem16	temp←DW, AW When temp÷(mem16)>0, temp÷(mem16)≦7FFFH or temp÷(mem16)<0, temp÷(mem16)>0−7FFFH−1 DW←temp%(mem16), AW←temp÷(mem16) When temp÷(mem16)>0, temp÷(mem16)>7FFFH or temp÷(mem16)<0, temp÷(mem16)≦0−7FFFH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2,(SP+1, SP)←PSW,IE←0, BRK←0 SP←SP−2,(SP+1, SP)←PS, PS←TC SP←SP−2,(SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	mod		1	1	1	mem			47-52	2-4	u	u	u	u	u	u
BCD Adjust Instructions																										
ADJBA		When (AL AND 0FH)>9 or AC=1, AL←AL+6, AH←AH+1, AC←1, CY←AC, AL←AL AND 0FH	0	0	1	1	0	1	1	1									7	1	x	x	u	u	u	u
ADJ4A		When (AL AND 0FH)>9 or AC=1, AL←AL+6, AC←1, When AL>9FH, or CY=1 AL←AL+60H, CY←1	0	0	1	0	0	1	1	1									3	1	x	x	u	x	x	x
ADJSB		When (AL AND 0FH)>9 or AC=1, AL←AL−6, AH←AH−1, AC←1, CY←AC, AL←AL AND 0FH	0	0	1	1	1	1	1	1									7	1	x	x	u	u	u	u
ADJ4S		When (AL AND 0FH)>9 or AC=1, AL←AL−6, AC←1 When AL>9FH or CY=1 AL←AL−60H, CY←1	0	0	1	0	1	1	1	1									3	1	x	x	u	x	x	x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Data Conversion Instructions																										
CVTBD		AH←AL=0AH, AL←AL%0AH	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	15	2	u	u	u	x	x	x
CVTDB		AH←0, AL←AH×0AH+AL	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	7	2	u	u	u	x	x	x
CVTBW		When AL<80H, AH←0, all other times AH←FFH	1	0	0	1	1	0	0	0									2	1						
CVTWL		When AL<8000H, DW←0, all other times DW←FFFFH	1	0	0	1	1	0	0	1									4-5	1						
Comparison Instructions																										
CMP	reg, reg	reg-reg	0	0	1	1	1	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)-reg	0	0	1	1	1	0	0	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	reg, mem	reg−(mem)	0	0	1	1	1	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	reg, imm	reg−imm	1	0	0	0	0	0	S	W	1	1	1	1	1		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)−imm	1	0	0	0	0	0	S	W	mod		1	1	1		mem		13/17	3-6	x	x	x	x	x	x
	acc, imm	When W=0, AL−imm When W=1, AW−imm	0	0	1	1	1	1	0	W									4	2-3	x	x	x	x	x	x
Complement Instructions																										
NOT	reg	reg←$\overline{\text{reg}}$	1	1	1	1	0	1	1	W	1	1	0	1	0		reg		2	2						
	mem	(mem)←$\overline{\text{(mem)}}$	1	1	1	1	0	1	1	W	mod		0	1	0		mem		16/24	2-4						
NEG	reg	reg←$\overline{\text{reg}}$+1	1	1	1	1	0	1	1	W	1	1	0	1	1		reg		2	2	x	x	x	x	x	x
	mem	(mem)←$\overline{\text{(mem)}}$+1	1	1	1	1	0	1	1	W	mod		0	1	1		mem		16/24	2-4	x	x	x	x	x	x
Logical Operation Instructions																										
TEST	reg, reg	reg AND reg	1	0	0	0	0	1	0	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg or reg, mem	(mem) AND reg	1	0	0	0	0	1	0	W	mod			reg			mem		10/14	2-4	u	0	0	x	x	x
	reg, imm	reg AND imm	1	1	1	1	0	1	1	W	1	1	0	0	0		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem) AND imm	1	1	1	1	0	1	1	W	mod		0	0	0		mem		11/15	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL AND imm8 When W=1, AW AND imm8	1	0	1	0	1	0	0	W									4	2-3	u	0	0	x	x	x
AND	reg, reg	reg←reg AND reg	0	0	1	0	0	0	1	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) AND reg	0	0	1	0	0	0	0	W	mod			reg			mem		16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg AND (mem)	0	0	1	0	0	0	1	W	mod			reg			mem		11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg AND imm	1	0	0	0	0	0	0	W	1	1	1	0	0		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) AND imm	1	0	0	0	0	0	0	W	mod		1	0	0		mem		18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL←AL AND imm8 When W=1, AW←AW AND imm16	0	0	1	0	0	1	0	W									4	2-3	u	0	0	x	x	x
OR	reg, reg	reg←reg OR reg	0	0	0	0	1	0	1	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) OR reg	0	0	0	0	1	0	0	W	mod			reg			mem		16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg OR (mem)	0	0	0	0	1	0	1	W	mod			reg			mem		11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg OR imm	1	0	0	0	0	0	0	W	1	1	0	0	1		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) OR imm	1	0	0	0	0	0	0	W	mod		0	0	1		mem		18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL←AL OR imm8 When W=1, AW←AW or imm16	0	0	0	0	1	1	1	W									4	2-3	u	0	0	x	x	x

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags					
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC	CY	V	P	S	Z
Logical Operation Instructions (cont)												
XOR	reg, reg	reg←reg XOR reg	0 0 1 1 0 0 1 W	1 1 reg reg	2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) XOR reg	0 0 1 1 0 0 0 W	mod reg mem	16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg XOR (mem)	0 0 1 1 0 0 1 W	mod reg mem	11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg XOR imm	1 0 0 0 0 0 0 W	1 1 1 1 0 reg	4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) XOR imm	1 0 0 0 0 0 0 W	mod 1 1 0 mem	18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL←AL XOR imm8 When W=1, AW←AW XOR imm16	0 0 1 1 0 1 0 W		4	2-3	u	0	0	x	x	x
Bit Operation Instructions												
			2nd byte★	3rd byte★								
TEST1	reg8, CL	reg8 bit no. CL=0: Z←1 reg8 bit no. CL=1: Z←0	0 0 0 1 0 0 0 0	1 1 0 0 0 reg	3	3	u	0	0	u	u	x
	mem8, CL	(mem8) bit no. CL=0: Z←1 (mem8) bit no. CL=1: Z←0	0 0 0 1 0 0 0 1	mod 0 0 0 mem	8	3-5	u	0	0	u	u	x
	reg16, CL	reg16 bit no. CL=0: Z←1 reg16 bit no. CL=1: Z←0	0 0 0 1 0 0 0 1	1 1 0 0 0 reg	3	3	u	0	0	u	u	x
	mem16, CL	(mem16) bit no. CL=0: Z←1 (mem16) bit no. CL=1: Z←0	0 0 0 1 0 0 0 0	mod 0 0 0 mem	12	3-5	u	0	0	u	u	x
	reg8, imm3	reg8 bit no. imm3=0: Z←1 reg8 bit no. imm3=1: Z←0	0 0 0 1 1 0 0 0	1 1 0 0 0 reg	4	4	u	0	0	u	u	x
	mem8, imm3	(mem8) bit no. imm3=0: Z←1 (mem8) bit no. imm3=1: Z←0	0 0 0 1 1 0 0 0	mod 0 0 0 mem	9	4-6	u	0	0	u	u	x
	reg16, imm4	reg16 bit no. imm4=0: Z←1 reg16 bit no. imm4=1: Z←0	0 0 0 1 1 0 0 1	1 1 0 0 0 reg	4	4	u	0	0	u	u	x
	mem16, imm4	(mem16) bit no. imm4=0: Z←1 (mem16) bit no. imm4=1: Z←0	0 0 0 1 1 0 0 1	mod 0 0 0 mem	13	4-6	u	0	0	u	u	x
			2nd byte★	3rd byte★								
			★Note: First byte=0FH									
			2nd byte★	3rd byte★								
NOT1	reg8, CL	reg8 bit no. CL←$\overline{\text{reg8 bit no. CL}}$	0 0 0 1 0 1 1 0	1 1 0 0 0 reg	4	3						
	mem8, CL	(mem8) bit no. CL←$\overline{\text{(mem8) bit no. CL}}$	0 0 0 1 0 1 1 0	mod 0 0 0 mem	13	3-5						
	reg16, CL	reg16 bit no. CL←$\overline{\text{reg16 bit no. CL}}$	0 0 0 1 0 1 1 1	1 1 0 0 0 reg	4	3						
	mem16, CL	(mem16) bit no. LC←$\overline{\text{(mem16) bit no. CL}}$	0 0 0 1 0 1 1 1	mod 0 0 0 mem	21	3-5						
	reg8, imm3	reg8 bit no. imm3←$\overline{\text{reg8 bit no. imm3}}$	0 0 0 1 1 1 1 0	1 1 0 0 0 reg	5	4						
	mem8, imm3	(mem8) bit no. imm3←$\overline{\text{(mem8) bit no. imm3}}$	0 0 0 1 1 1 1 0	mod 0 0 0 mem	14	4-6						
	reg16, imm4	reg16 bit no. imm4←$\overline{\text{(reg16) bit no. imm4}}$	0 0 0 1 1 1 1 1	1 1 0 0 0 reg	5	4						
	mem16, imm4	(mem16) bit no. imm4←$\overline{\text{(mem16) bit no. imm4}}$	0 0 0 1 1 1 1 1	mod 0 0 0 mem	22	4-6						
			2nd byte★	3rd byte★								
			★Note: First byte=0FH									
	CY	CY←$\overline{\text{CY}}$	1 1 1 1 0 1 0 1		2	1		x				

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Bit Operation Instructions (cont)																										
			2nd byte★								3rd byte★															
CLR1	reg8, CL	reg8 bit no. CL←0	0	0	0	1	0	0	1	0	1	1	0	0	0	reg			5	3						
	mem8, CL	(mem8) bit no. CL←0	0	0	0	1	0	0	1	0	mod		0	0	0	mem			14	3-5						
	reg16, CL	reg16 bit no. CL←0	0	0	0	1	0	0	1	1	1	1	0	0	0	reg			5	3						
	mem16, CL	(mem16) bit no. CL←0	0	0	0	1	0	0	1	1	mod		0	0	0	mem			22	3-5						
	reg8, imm3	reg8 bit no. imm3←0	0	0	0	1	1	0	1	0	1	1	0	0	0	reg			6	4						
	mem8, imm3	(mem8) bit no. imm3←0	0	0	0	1	1	0	1	0	mod		0	0	0	mem			15	4-6						
	reg16, imm4	reg16 bit no. imm4←0	0	0	0	1	1	0	1	1	1	1	0	0	0	reg			6	4						
	mem16, imm4	(mem16) bit no. imm4←0	0	0	0	1	1	0	1	1	mod		0	0	0	mem			23	4-6						
			2nd byte★ ★Note: First byte=0FH								3rd byte★															
	CY	CY←0	1	1	1	1	1	0	0	0									2	1		0				
	DIR	DIR←0	1	1	1	1	1	1	0	0									2	1						
SET1	reg8, CL	reg8 bit no. CL←1	0	0	0	1	0	1	0	0	1	1	0	0	0	reg			4	3						
	mem8, CL	(mem8) bit no. CL←1	0	0	0	1	0	1	0	0	mod		0	0	0	mem			13	3-5						
	reg16, CL	reg16 bit no. CL←1	0	0	0	1	0	1	0	1	1	1	0	0	0	reg			4	3						
	mem16, CL	(mem16) bit no. CL←1	0	0	0	1	0	1	0	1	mod		0	0	0	mem			21	3-5						
	reg8, imm3	reg8 bit no. imm3←1	0	0	0	1	1	1	0	0	1	1	0	0	0	reg			5	4						
	mem8, imm3	(mem8) bit no. imm3←1	0	0	0	1	1	1	0	0	mod		0	0	0	mem			14	4-6						
	reg16, imm4	reg16 bit no. imm4←1	0	0	0	1	1	1	0	1	1	1	0	0	0	reg			5	4						
	mem16, imm4	(mem16) bit no. imm4←1	0	0	0	1	1	1	0	1	mod		0	0	0	mem			22	4-6						
		★Note: First byte=0FH	2nd byte★								3rd byte★															
	CY	CY←1	1	1	1	1	1	0	0	1									2	1		1				
	DIR	DIR←1	1	1	1	1	1	1	0	1									2	1						
Shift Instructions																										
SHL	reg, 1	CY←MSB or reg, reg←×2 When MSB of reg≠CY, V←1 When MSB of reg=CY, V←0	1	1	0	1	0	0	0	W	1	1	1	0	0	reg			2	2	u	x	x	x	x	x
	mem, 1	CY←MSB or (mem), (mem)←(mem)×2 When MSB or (mem)≠CY, V←1 When MSB of (mem)=CY, V←0	1	1	0	1	0	0	0	W	mod		1	0	0	mem			16/24	2-4	u	x	x	x	x	x
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB or reg, reg←reg×2, temp←temp−1	1	1	0	1	0	0	1	W	1	1	1	0	0	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2, temp←temp−1	1	1	0	1	0	0	1	W	mod		1	0	0	mem			19/27+n	2-4	u	x	u	x	x	x
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB or reg, reg←reg×2, temp←temp−1	1	1	0	0	0	0	0	W	1	1	1	0	0	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB or (mem), (mem)←(mem)×2, temp←temp−1	1	1	0	0	0	0	0	W	mod		1	0	0	mem			19/27+n	3-5	u	x	u	x	x	x
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Shift Instructions (cont)																										
SHR	reg, 1	CY←LSB of reg, reg←reg÷2 When MSB of reg≠bit following MSB of reg: V←1 When MSB of reg=bit following MSB of reg: V←0	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			2	2	u	x	x	x	x	x
	mem, 1	CY←LSB of (mem), (mem)←(mem)÷2 When MSB of (mem)≠bit following MSB of (mem): V←1 When MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		1	0	1	mem			16/24	2-4	u	x	x	x	x	x
	reg, CL	temp←CL, while time≠0, repeat this operation, CY←LSB or reg, reg←reg÷2, temp←temp−1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0 repeat this operation, CY←LSB or (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	1	0	0	1	W	mod		1	0	1	mem			19/27+n	2-4	u	x	u	x	x	x
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB or reg, reg←reg÷2, temp←temp−1	1	1	0	0	0	0	0	W	1	1	1	0	1	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0 repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	0	0	0	0	W	mod		1	0	1	mem			19/27+n	3-5	u	x	u	x	x	x
			n: number of shifts																							
SHRA	reg, 1	CY←LSB of reg, reg←reg÷2,V←0 MSB of operand does not change	1	1	0	1	0	0	0	W	1	1	1	1	1	reg			2	2	u	x	0	x	x	x
	mem, 1	CY←LSB of (mem), (mem)←(mem)÷2, V←0, MSB of operand does not change	1	1	0	1	0	0	0	W	mod		1	1	1	mem			16/24	2-4	u	x	0	x	x	x
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, temp←temp−1 MSB of operand does not change	1	1	0	1	0	0	1	W	1	1	1	1	1	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1 MSB of operand does not change	1	1	0	1	0	0	1	W	mod		1	1	1	mem			19/27+n	2-4	u	x	u	x	x	x
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, temp←temp−1 MSB of operand does not change	1	1	0	0	0	0	0	W	1	1	1	1	1	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1 MSB of operand does not change	1	1	0	0	0	0	0	W	mod		1	1	1	mem			19/27+n	3-5	u	x	u	x	x	x
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions																										
ROL	reg, 1	CY←MSB of reg, reg←reg×2+CY MSB of reg≠CY: V←1 MSB of reg=CY: V←0	1	1	0	1	0	0	0	W	1	1	0	0	0	reg			2	2		x	x			
	mem, 1	CY←MSB of (mem), (mem)←(mem)×2+CY MSB of (mem)≠CY: V←1 MSB of (mem)=CY: V←0	1	1	0	1	0	0	0	W	mod		0	0	0	mem			16/24	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of reg, reg←reg×2+CY temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	0	0	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2+CY temp←temp−1	1	1	0	1	0	0	1	W	mod		0	0	0	reg			19/27+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB of reg, reg←reg×2+CY temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	0	0	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2+CY temp←temp−1	1	1	0	0	0	0	0	W	mod		0	0	0	mem			19/27+n	3-5		x	u			
			n: number of shifts																							
ROR	reg, 1	CY←LSB of reg, reg←reg÷2 MSB of reg←CY MSB of reg≠bit following MSB of reg: V←1 MSB of reg=bit following MSB of reg: V←0	1	1	0	1	0	0	0	W	1	1	0	0	1	reg			2	2		x	x			
	mem, 1	CY←LSB of (mem),(mem)←(mem)÷2 MSB of (mem)←CY MSB of (mem)≠bit following MSB of (mem): V←1 MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		0	0	1	mem			16/24	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, MSB of leg←CY temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	0	1	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, MSB of (mem)←CY temp←temp−1	1	1	0	1	0	0	1	W	mod		0	0	1	mem			19/27+n	2-4		x	u			
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions (cont)																										
ROR	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, MSB of reg←CY temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	0	1		reg		7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	0	0	0	0	W	mod		0	0	1		mem		19/27+n	3-5		x	u			
			n: number of shifts																							
Rotation Instructions																										
ROLC	reg, 1	tmpcy←CY, CY←MSB of reg reg←reg×2+tmpcy MSB of reg=CY: V←0 MSB of reg≠CY: V←1	1	1	0	1	0	0	0	W	1	1	0	1	0		reg		2	2		x	x			
	mem, 1	tmpcy←CY, CY←MSB of (mem) (mem)←(mem)×2+tmpcy MSB of (mem)=CY: V←0 MSB of (mem)≠CY: V←1	1	1	0	1	0	0	0	W	mod		0	1	0		mem		16/24	2-4		x	x			
	reg, CL	temp←CL, while temp≠0 repeat this operation, tmpcy←CY, CY←MSB of reg, reg←reg×2+tmpcy temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	1	0		reg		7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0 repeat this operation, tmpcy←CY, CY←MSB of (mem), (mem)←(mem)×2+tmpcy temp←temp−1	1	1	0	1	0	0	1	W	mod		0	1	0		mem		19/27+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0 repeat this operation, tmpcy←CY, CY←MSB of reg, reg←reg×2+tmpcy temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	1	0		reg		7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0 repeat this operation, tmpcy←CY, CY←MSB of (mem) (mem)←(mem)×2+tmpcy temp←temp−1	1	1	0	0	0	0	0	W	mod		0	1	0		mem		19/27+n	3-5		x	u			
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions (cont)																										
RORC	reg, 1	tmpcy←CY, CY←LSB of reg reg←reg÷2, MSB of reg←tmpcy MSB of reg≠bit following MSB of reg: V←1 MSB of reg=bit following MSB of reg: V←1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			2	2		x	x			
	mem, 1	tmpcy←CY, CY←LSB of (mem) (mem)←(mem)÷2, MSB of (mem)←tmpcy MSB of (mem)≠bit following MSB of (mem): V←1 MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		0	1	1	mem			16/24	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operatios, tmpcy←CY, CY←LSB of reg, reg←reg÷2, MSB of reg←tmpcy, temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	1	1	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, tmpcy←CY, CY←LSB of (mem), (mem)←(mem)÷2, MSB of (mem)←tmpcy, temp←temp−1	1	1	0	1	0	0	1	W	mod		0	1	1	mem			19/27+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0 repeat this operation, tmpcy←CY, CY←LSB of reg, reg←reg÷2, MSB of reg←tmpcy, temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	1	1	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, tmpcy←CY, CY←LSB of (mem), (mem)←(mem)÷2 MSB of (mem)←tmpcy, temp←temp−1	1	1	0	0	0	0	0	W	mod n: number of shifts		0	1	1	mem			19/27+n	3-5		x	u			
Subroutine Control Instructions																										
CALL	near-proc	(SP−1, SP−2)←PC, SP←SP−2 PC←PC+disp	1	1	1	0	1	0	0	0									20	3						
	regptr16	(SP−1, SP−2)←PC, SP←SP−2 PC←regptr16	1	1	1	1	1	1	1	1	1	1	0	1	0	reg			18	2						
	memptr16	(SP−1, SP−2)←PC, SP←SP−2 PC←(memptr16)	1	1	1	1	1	1	1	1	mod		0	1	0	mem			31	2-4						
	far-proc	(SP−1, SP−2)←PS, (SP−3, SP−4)←PC SP←SP−4, PS←seg, PC←offset	1	0	0	1	1	0	1	0									29	5						
	memptr32	(SP−1, SP−2)←PS, (SP−3, SP−4)←PC SP←SP−4, PS←(memptr32+2) PC←(memptr32)	1	1	1	1	1	1	1	1	mod		0	1	1	mem			47	2-4						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Subroutine Control Instructions (cont)																										
RET		PC←(SP+1, SP), SP←SP+2	1	1	0	0	0	0	1	1									19	1						
	pop-value	PC←(SP+1, SP), SP←SP+2, SP←SP+pop-value	1	1	0	0	0	0	1	0									24	3						
		PC←(SP+1, SP), PS←(SP+3, SP+2) SP←SP+4	1	1	0	0	1	0	1	1									29	1						
	pop-value	PC←(SP+1, SP), PS←(SP+3, SP+2) SP←SP+4, SP←SP+pop-value	1	1	0	0	1	0	1	0									32	3						
Stack Manipulation Instructions																										
PUSH	mem16	SP←SP−2 (SP+1, SP)←(mem16)	1	1	1	1	1	1	1	1	mod		1	1	0	mem			26	2-4						
	reg16	SP←SP−2 (SP+1, SP)←reg16	0	1	0	1	0	reg											12	1						
	sreg	SP←SP−2 (SP+1, SP)←sreg	0	0	0	sreg		1	1	0									12	1						
	PSW	SP←SP−2 (SP+1, SP)←PSW	1	0	0	1	1	1	0	0									12	1						
	R	Push registers on the stack	0	1	1	0	0	0	0	0									67	1						
	imm	(SP−1, SP−2)←imm SP←SP−2, when S=1, sign extension	0	1	1	0	1	0	S	0									11/12	2-3						
POP	mem16	(mem16)←(SP+1, SP), SP←SP+2	1	0	0	0	1	1	1	1	mod		0	0	0	mem			25	2-4						
	reg16	reg 16←(SP+1, SP), SP←SP+2	0	1	0	1	1	reg											12	1						
	sreg	sreg←(SP+1, SP) sreg: SS, DS0 DS1 SP←SP+2	0	0	0	sreg		1	1	1									12	1						
	PSW	PSW←(SP+1, SP), SP←SP+2	1	0	0	1	1	1	0	1									12	1	R	R	R	R	R	R
	R	Pop registers from the stack	0	1	1	0	0	0	0	1									75	1						
PREPARE	imm16,imm8	Prepare new stack frame	1	1	0	0	1	0	0	0									*	4						
			*: imm8<0: 16 imm8<1: 23+16 (imm8−1)																							
DISPOSE		Dispose of stack frame	1	1	0	0	1	0	0	1									10	1						
Branch Instructions																										
BR	near-label	PC←PC+disp	1	1	1	0	1	0	0	1									13	3						
	short-label	PC←PC+ext-disp8	1	1	1	0	1	0	1	1									12	2						
	regptr16	PC←regptr16	1	1	1	1	1	1	1	1	1	1	1	0	0	reg			11	2						
	memptr16	PC←(memptr16)	1	1	1	1	1	1	1	1	mod		1	0	0	mem			24	2-4						
	far-label	PS←seg, PC←offset	1	1	1	0	1	0	1	0									15	5						
	memptr32	PS←(memptr32+2), PC←(memptr32)	1	1	1	1	1	1	1	1	mod		1	0	1	mem			35	2-4						
Conditional Branch Instructions																										
BV	short-label	if V=1, PC←PC+ext-disp8	0	1	1	1	0	0	0	0									14/4	2						
BNV	short-label	if V=0, PC←PC+ext-disp8	0	1	1	1	0	0	0	1									14/4	2						
BC, BL	short-label	if CY=1, PC←PC+ext-disp8	0	1	1	1	0	0	1	0									14/4	2						
BNC,BNL	short-label	if CY=0, PC←PC+ext-disp8	0	1	1	1	0	0	1	1									14/4	2						
BE,BZ	short-label	if Z=1, PC←PC+ext-disp8	0	1	1	1	0	1	0	0									14/4	2						
BNE,BNZ	short-label	if Z=0, PC←PC+ext-disp8	0	1	1	1	0	1	0	1									14/4	2						
BNH	short-label	if CY OR Z=1, PC←PC+ext-disp8	0	1	1	1	0	1	1	0									14/4	2						
BH	short-label	if CY OR Z=0, PC←PC+disp8	0	1	1	1	0	1	1	1									14/4	2						
BN	short-label	if S=1, PC←PC+ext-disp8	0	1	1	1	1	0	0	0									14/4	2						
BP	short-label	if S=0, PC←PC+ext-disp8	0	1	1	1	1	0	0	1									14/4	2						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Conditional Branch Instructions (cont)																										
BPE	short-label	if P=1, PC←PC+ext-disp8	0	1	1	1	1	0	1	0									14/4	2						
BPO	short-label	if P=0, PC←PC+ext-disp8	0	1	1	1	1	0	1	1									14/4	2						
BLT	short-label	if S XOR V=1, PC←PC+ext-disp8	0	1	1	1	1	1	0	0									14/4	2						
BGE	short-label	if S XOR V=0, PC←PC+ext-disp8	0	1	1	1	1	1	0	1									14/4	2						
BLE	short-label	if (S XOR V) OR Z=1, PC←PC+ext-disp8	0	1	1	1	1	1	1	0									14/4	2						
BGT	short-label	if (S XOR V) OR Z=0, PC←PC+ext-disp8	0	1	1	1	1	1	1	1									14/4	2						
DBNZNE	short-label	CW←CW−1 if Z=0 and CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	0	0									14/5	2						
DBNZE	short-label	CW←CW−1 if Z=1 and CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	0	1									14/5	2						
DBNZ	short-label	CW←CW−1 if CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	1	0									13/5	2						
BCWZ	short-label	if CW=0, PC←PC+ext-disp8	1	1	1	0	0	0	1	1									13/5	2						
Interrupt Instructions																										
BRK	3	TA←(00DH, 00CH), TA←(00FH, 00EH) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	0	0									50	1						
	imm8 (≠3)	TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=Imm8 SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	0	1									50	2						
BRKV		When V=1 TA←(011H, 010H), TC←(013H, 012H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP− 2 , (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	1	0									52/3	1						
RETI		PC←(SP+1, SP), PS←(SP+3, SP+2), PSW←(SP+5, SP+4), SP←SP+6	1	1	0	0	1	1	1	1									39	1	R	R	R	R	R	R
CHKIND	reg16, mem32	When (mem32)>reg 16 or (mem32+2)<reg16 TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=imm8 SP←SP−2, (SP+1, SP)←PSW, MD←0 MD Bit Write Enable SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	0	1	1	0	0	0	1	0	mod			reg			mem		73-76/ 26	2-4						
BRKEM	imm8	TA←(015H, 014H), TC←(017H, 016H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	50	3						

Mnemonic	Operand	Operation	Operation Code																No. of	No. of	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	Clocks	Bytes	AC	CY	V	P	S	Z
CPU Control Instructions																										
HALT		CPU Halt	1	1	1	1	0	1	0	0									2	1						
BUSLOCK		Bus Lock Prefix	1	1	1	1	0	0	0	0									2	1						
FPO1	fp-op	No Operation	1	1	0	1	1	X	X	X	1	1	Y	Y	Y	Z	Z	Z	2	2						
	fp-op, mem	data bus←(mem)	1	1	0	1	1	X	X	X	mod		Y	Y	Y	mem			15	2-4						
FPO2	fp-op	No Operation	0	1	1	0	0	1	1	X	1	1	Y	Y	Y	Z	Z	Z	2	2						
	fp-op, mem	data bus←(mem)	0	1	1	0	0	1	1	X	mod		Y	Y	Y	mem			15	2-4						
POLL		Poll and wait n: number of times POLL pin is sampled	1	0	0	1	1	0	1	1									2+5n	1						
NOP		No Operation	1	0	0	1	0	0	0	0									3	1						
DI		IE←0	1	1	1	1	1	0	1	0									2	1						
EI		IE←1	1	1	1	1	1	0	1	1									2	1						
8080 Mode Instructions																										
RETEM		PC←(SP+1, SP), PS←(SP+3, SP+2), PSW←(SP+5, SP+4), SP←SP+6, MD Bit Write Disable	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	39	2	R	R	R	R	R	R
CALLN	imm8	TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=imm8 SP←SP−2, (SP+1, SP)←PSW, MD←1 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	58	3						

LH70116 (V30) High-Performance 16-Bit Microprocessor

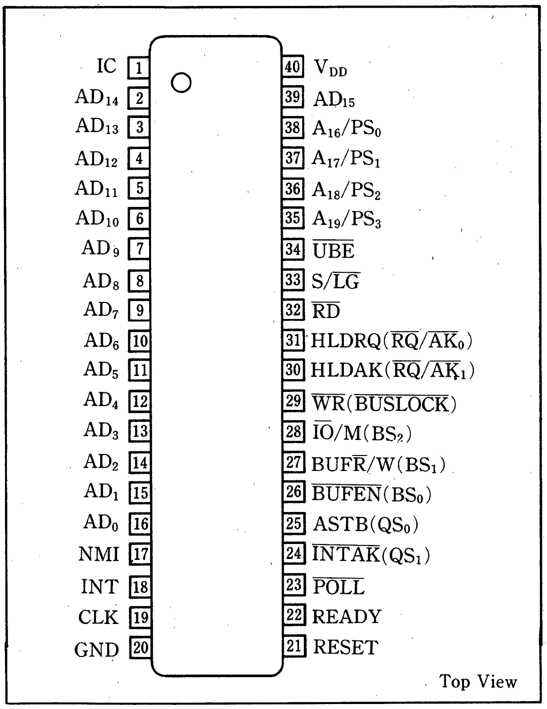
Description

The LH70116 (V30) is a CMOS 16-bit microprocessor with internal 16-bit architecture and a 16-bit external data bus. The LH70116 additionally has a powerful instruction set including bit processing, packed BCD operations, and high-speed multiplication/division operations. The LH70116 can also execute the entire 8080 instruction set and comes with a standby mode that significantly reduces power consumption. It is software-compatible with the LH70108 16-bit microprocessor.

Features

1. Minimum instruction execution time: 250ns (at 8MHz)
2. Maximum addressable memory: 1 Mbyte
3. Abundant memory addressing modes
4. 14×16-bit register set
5. 101 instructions
6. Bit, byte, word, and block operations
7. Bit field operation instructions
8. Packed BCD instructions
9. Multiplication/division instruction execution time: 2.4 μs to 7.1 μs (at 8MHz), 3.8 μs to 11.4 μs (at 5MHz)
10. High-speed block transfer instructions: 2 Mbyte /s (at 8MHz)
11. High-speed calculation of effective addresses: 2 clock cycles in any addressing mode
12. Maskable (INT) and nonmaskable (NMI) interrupt inputs
13. IEEE-796 bus compatible interface
14. 8080 emulation mode
15. CMOS technology
16. Low-power consumption
17. Low-power standby mode
18. Single power supply
19. 5MHz or 8MHz clock
20. 40-pin DIP (DIP40-P-600)

Pin Connections

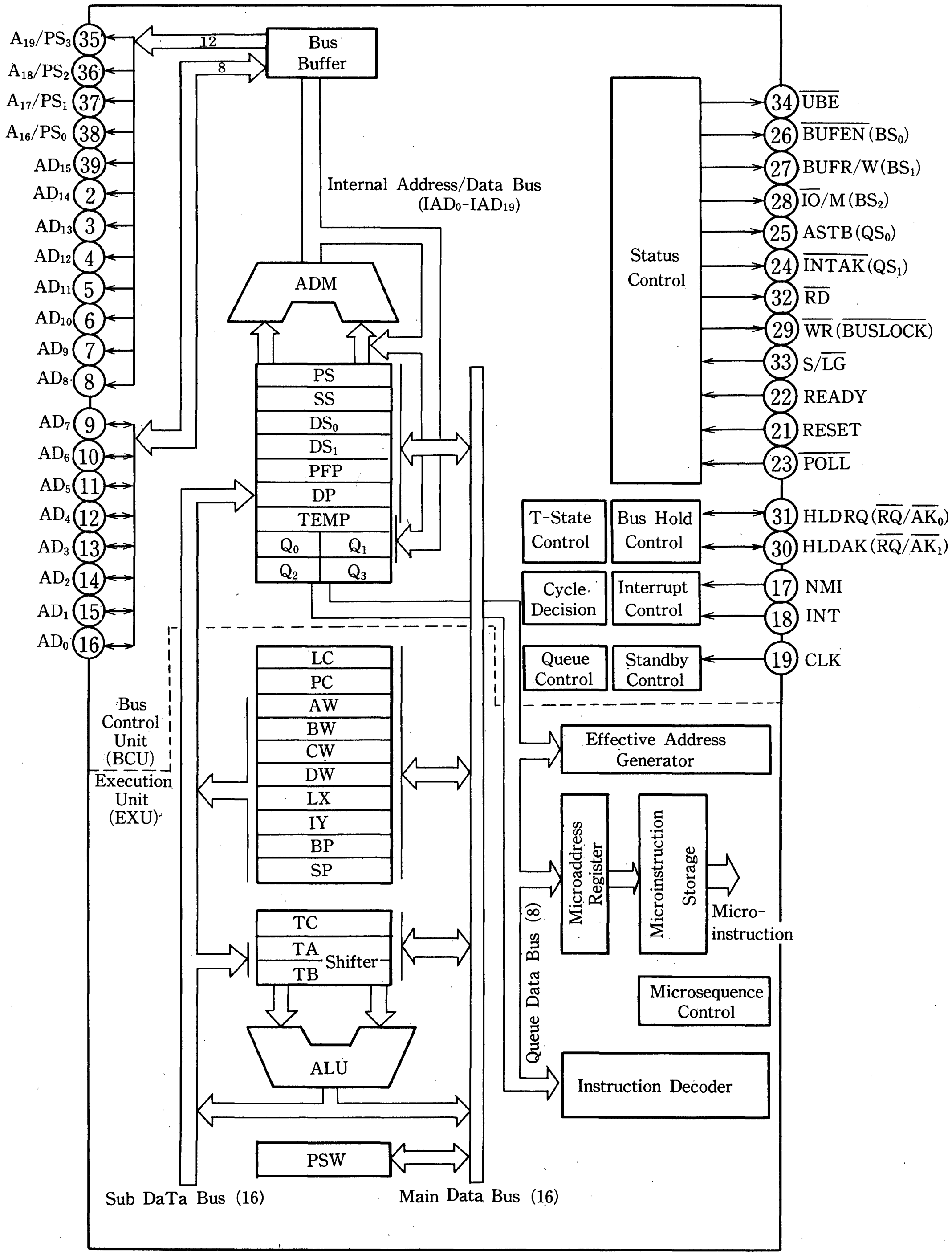


Ordering Information

LH70116-X
└Frequency
5: 5MHz
8: 8MHz
└Model No.

*V30 is a trademark of NEC corporation.

Block Diagram



■ Pin Identification

No.	Symbol	Direction	Function
1	IC*		Internally connected
2-8	AD_{14}-AD_8	Out	Address bus , middle bits
9-16	AD_7-AD_0	In/Out	Address/data bus
17	NMI	In	Nonmaskable interrupt input
18	INT	In	Maskable interrupt input
19	CLK	In	Clock input
20	GND		Ground potential
21	RESET	In	Reset input
22	READY	In	Ready input
23	$\overline{\text{POLL}}$	In	Poll input
24	$\overline{\text{INTAK}}$ (QS_1)	Out	Interrupt acknowledge output (queue status bit 1 output)
25	ASTB (QS_0)	Out	Address strobe output (queue status bit 0 output)
26	$\overline{\text{BUFEN}}$ (BS_0)	Out	Buffer enable output (bus status bit 0 output)
27	BUF$\overline{\text{R}}$/W (BS_1)	Out	Buffer read/write output (bus status bit 1 output)
28	$\overline{\text{IO}}$/M (BS_2)	Out	Access is I/O or memory (bus status bit 2 output)
29	WR ($\overline{\text{BUSLOCK}}$)	Out	Write strobe output. (bus lock output)
30	HLDAK ($\overline{\text{RQ}}/\overline{\text{AK}}_1$)	Out (In/Out)	Hold acknowledge output, (bus hold request input/ acknowledge output 1)
31	HLDRQ ($\overline{\text{RQ}}/\overline{\text{AK}}_0$)	In (In/Out)	Hold request input (bus hold request input/acknowledge output 0)
32	$\overline{\text{RD}}$	Out	Read strobe output
33	S/$\overline{\text{LG}}$	In	Small-scale/large-scale system input
34	$\overline{\text{UBE}}$	Out	Latched bus status output 0 (always high in large-scale systems)
35-38	A_{19}/PS_3- A_{16}/PS_0	Out	Address bus, high bits or processor status output
39	AD_{15}	Out	Address bus, bit 15
40	V_{DD}		Power supply

Notes: *IC should be connected to ground.
Where pins have different functions in small-and large-systems, the large-scale system pin symbol and function are in parentheses.
Unused input pins should be tied to ground or V_{DD} to minimize power dissipation and prevent the flow of potentially harmful currents.

■ Absolute Maximum Ratings

(Ta=+25°C)

Parameter	Symbol	Ratings	Units
Supply voltage	V_{DD}	−0.5 to +7.0	V
Power dissipation	PD_{MAX}	0.5	W
Input voltage	V_I	−0.5 to V_{DD}+0.3	V
CLK input voltage	V_K	−0.5 to V_{DD}+1.0	V
Output voltage	V_O	−0.5 to V_{DD}+0.3	V
Operating temperature	Topr	−40 to +85	°C
Storage temperature	Tstg	−65 to +150	°C

■ Capacitance

(Ta=+25°C, V_{DD}=0V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Input capacitance	C_I	fc=1MHz Unmeasured pins returned to 0V		15	pF
I/O capacitance	C_{IO}			15	pF

■ DC Characteristics

(LH70116-5, Ta=−40℃ to +85℃, V_{DD}=+5V±10%)
(LH70116-8, Ta=−10℃ to +70℃, V_{DD}=+5V±5%)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input HIGH voltage	V_{IH}			2.2		V_{DD}+0.3	V
Input LOW voltage	V_{IL}			−0.5		0.8	V
CLK input HIGH voltage	V_{KH}			3.9		V_{DD}+1.0	V
CLK input LOW voltage	V_{KL}			−0.5		0.6	V
Output HIGH voltage	V_{OH}	I_{OH}=−400μA		0.7V_{DD}			V
Output LOW voltage	V_{OL}	I_{OL}=2.5mA				0.4	V
Input leakage HIGH current	I_{LIH}	V_I=V_{DD}				10	μA
Input leakage LOW current	I_{LIL}	V_I=0V				−10	μA
Output leakage HIGH current	I_{LOH}	V_O=V_{DD}				10	μA
Output leakage LOW current	I_{LOL}	V_O=0V				−10	μA
Supply current	I_{DD}	Normal operation	70108-5		30	60	mA
		Standby mode	5MHz		5	10	mA
		Normal operation	70108-8		45	80	mA
		Standby mode	8MHz		6	12	mA

6

■ AC Characteristics

(LH70116-5, Ta=−40℃ to +85℃, V_{DD}=+5V±10%)
(LH70116-8, Ta=−10℃ to +70℃, V_{DD}=+5V±5%)

Parameter	Symbol	Conditions	LH70116-5		LH70116-8		Unit
			MIN.	MAX.	MIN.	MAX.	
Small/Larg Scale							
Clock cycle	t_{CYK}		200	500	125	500	ns
Clock pulse HIGH width	t_{KKH}	V_{KH}=3.0V	69		44		ns
Clock pulse LOW width	t_{KKL}	V_{KL}=1.5V	90		60		ns
Clock rise time	t_{KR}	1.5V to 3.0V		10		10	ns
Clock fall time	t_{KF}	3.0V to 1.5V		10		10	ns
READY inactive setup to CLK ↓	t_{SRYLK}		−8		−8		ns
READY incative hold after CLK ↑	t_{HKRYH}		30		20		ns
READY active setup to CLK ↑	t_{SRYHK}		$t_{KKL}-8$		$t_{KKL}-8$		ns
READY active hold after CLK ↑	t_{HKRYL}		30		20		ns
Data setup time to CLK ↓	t_{SDK}		30		20		ns
Data hold time after CLK ↓	t_{HKD}		10		10		ns
NMI, INT, $\overline{\text{POLL}}$ setup time to CLK ↑	t_{SIK}		30		15		ns
Input rise time (except CLK)	t_{IR}	0.8V to 2.2V		20		20	ns
Input fall time (except CLK)	t_{IF}	2.2V to 0.8V		12		12	ns
Output rise time	t_{OR}	0.8V to 2.2V		20		20	ns
Output fall time	t_{OF}	2.2V to 0.8V		12		12	ns
Small Scale							
Address delay time from CLK ↓	t_{DKA}	C_L=100pF	10	90	10	60	ns
Address hold time CLK ↓	t_{HKA}		10		10		ns
PS delay time from CLK ↓	t_{DKP}		10	90	10	60	ns
PS float delay time from CLK ↑	t_{FKP}		10	80	10	60	ns
Address setup time to ASTB ↓	t_{SAST}		$t_{KKL}-60$		$t_{KKL}-30$		ns
Address float delay time from CLK ↓	t_{FKA}		t_{HKA}	80	t_{HKA}	60	ns
ASTB ↑ delay time from CLK ↓	t_{DKSTH}			80		50	ns
ASTB ↓ delay time from CLK ↑	t_{DKSTL}			85		55	ns
ASTB HIGH width	t_{STST}		$t_{KKL}-20$		$t_{KKL}-10$		ns
Address hold time from ASTB ↓	t_{HSTA}		$t_{KKH}-10$		$t_{KKH}-10$		ns
Control delay time from CLK	t_{DKCT}	C_L=100pF	10	110	10	65	ns
Address float to $\overline{\text{RD}}$ ↓	t_{AFRL}		0		0		ns
$\overline{\text{RD}}$ ↓ delay time from CLK ↓	t_{DKRL}		10	165	10	80	ns
$\overline{\text{RD}}$ ↑ delay time from CLK ↓	t_{DKRH}		10	150	10	80	ns
Address delay time from $\overline{\text{RD}}$ ↑	t_{DRHA}		$t_{CYK}-45$		$t_{CYK}-40$		ns
$\overline{\text{RD}}$ LOW width	t_{RR}		$2t_{CYK}-75$		$2t_{CYK}-50$		ns
Data output delay time from CLK ↓	t_{DKD}		10	90	10	60	ns
Data float delay time from CLK ↓	t_{FKD}		10	80	10	60	ns
$\overline{\text{WR}}$ LOW width	t_{WW}		$2t_{CYK}-60$		$2t_{CYK}-40$		ns
HLDRQ setup time to CLK ↑	t_{SHQK}		35		20		ns
HLDAK delay time from CLK ↓	t_{DKHA}		10	160	10	100	ns

■ AC Characteristics (Cont)

(LH70116-5, Ta = −40℃ to +85℃, V_{DD} = +5V ± 10%)
(LH70116-8, Ta = −10℃ to +70℃, V_{DD} = +5V ± 5%)

Parameter	Symbol	Conditions	LH70116-5		LH70116-8		Unit
			MIN.	MAX.	MIN.	MAX.	
Large Scale							
Address delay time from CLK	t_{DKA}	C_L = 100pF	10	90	10	60	ns
Address hold time from CLK	t_{HKA}		10		10		ns
PS delay time from CLK ↓	t_{DKP}		10	90	10	60	ns
PS float delay time from CLK ↑	t_{FKP}		10	80	10	60	ns
Address float delay time from CLK ↓	t_{FKA}		t_{HKA}	80	t_{HKA}	60	ns
Address delay time from $\overline{RD}$ ↑	t_{DRHA}		$t_{CYK}-45$		$t_{CYK}-40$		ns
ASTB delay time from BS ↓	t_{DBST}			15		15	ns
BS ↓ delay time from CLK ↑	t_{DKBL}		10	110	10	60	ns
BS ↑ delay time from CLK ↓	t_{DKBH}		10	130	10	65	ns
$\overline{RD}$ ↓ delay time from address float	t_{DAFRL}		0		0		ns
$\overline{RD}$ ↓ delay time from CLK ↓	t_{DKRL}		10	165	10	80	ns
$\overline{RD}$ ↑ delay time from CLK ↓	t_{DKRH}		10	150	10	80	ns
$\overline{RD}$ LOW width	t_{RR}		$2t_{CYK}-75$		$2t_{CYK}-50$		ns
Data output delay time from CLK ↓	t_{DKD}		10	90	10	60	ns
Data float delay time from CLK ↑	t_{FKD}		10	80	10	60	ns
$\overline{AK}$ delay time from CLK ↓	t_{DKAK}			70		50	ns
$\overline{RQ}$ setup time to CLK ↑	t_{SRQK}		20		10		ns
$\overline{RQ}$ hold time after CLK ↓	t_{HKRQ1}		0		0		ns
$\overline{RQ}$ hold time after CLK ↑	t_{HKRQ2}		40		30		ns

Pin Functions

Some pins of the LH70116 have different functions according to whether the microprocessor is used in a small- or large-scale system. Other pins function the same way in either type of system.

AD_{15}-AD_0 (Address/Data Bus)

For small and large scale systems

The AD_{15}-AD_0 is a time-multiplexed Address/Data bus. This performs output of lower 16 bits of 20 bits address information and input/output of byte or word data. The LH70116 locates memory and I/O operand to a byte-data bank to be accessed with even address ($AD_0=0$) and a byte-data bank to be accessed with odd address ($AD_0=1$). The LSB (AD_0) has no meaning as a word data address but used for selecting the odd or even address bank. The $\overline{\text{UBE}}$ (Upper Byte Enable) signal is provided to access byte/word data besides AD_0. This is used as the combination of the following table.

Operand	$\overline{\text{UBE}}$	AD_0	No. of bus cycle
Word of Even Address	0	0	1
Word of Odd Address	0 1	1★ 0★★	2
Byte of Even Address	1	0	1
Byte of Odd Address	0	1	1

★First time, ★★Second time

A word operand in odd address is performed via two continuous access of odd-byte bank and even--byte bank. In this case, first $AD_0=1$ showing odd bank is output, and second $AD_0=1$ showing continuous even bank is output automatically. These outputs are held to high or low level in the standby mode. These are 3-state output and becomes high impedance in the hold acknowledge and interrupt acknowledge states.

NMI (Nonmaskable Interrupt)

For small- and large-scale systems.

This pin is used to input nonmaskable interrupt requests. NMI cannot be masked by software. This input is positive edge triggered and must be held high for five clocks to guarantee recognition. Actual interrupt processing begins, however, after completion of the instruction in progress.

The contents of interrupt vector 2 determine the starting address for the interrupt-servicing routine. Note that a hold request will be accepted even during NMI acknowledge.

This interrupt will cause the LH70116 to exit the standby mode.

INT (Maskable Interrupt)

For small- and large-scale systems.

This pin is an interrupt request that can be masked by software.

INT is active high level and is sensed during the last clock of the instruction. The interrupt will be accepted if the interrupt enable flag IE is set. The CPU outputs the $\overline{\text{INTAK}}$ signal to inform external devices that the interrupt request has been granted. INT must be asserted until the interrupt acknowledge is returned.

If NMI and INT interrupts occur at the same time, NMI has higher priority than INT and INT cannot be accepted. A hold request will be accepted during INT acknowledge.

This interrupt causes the LH70116 to exit the standby mode.

CLK (Clock)

For small- and large-scale systems.

This pin is used for external clock input.

RESET (Reset)

For small- and large-scale systems.

This pin is used for the CPU reset signal. It is an active high level. Input of this signal has priority over all other operations. After the reset signal input returns to a low level, the CPU begins execution of the program starting at address FFFF0H.

In addition to causing normal CPU start, RESET input will cause the LH70116 to exit the standby mode.

READY (Ready)

For small- and large-scale systems.

When the memory or I/O device being accessed cannot complete data read or write within the CPU basic access time, it can generate a CPU wait state (Tw) by setting this signal to inactive (low level) and requesting a read/write cycle delay.

If the READY signal is active (high level) during either the T3 or Tw state, the CPU will not generate a wait state.

This signal must be input in synchronization with external clock signals to satisfy the setup/hold time for normal operation.

$\overline{\text{POLL}}$ (Poll)

For small- and large-scale systems.

The CPU checks this input upon execution of the $\overline{\text{POLL}}$ instruction. If the input is low, then execution continues. If the input is high, the CPU will check the $\overline{\text{POLL}}$ input every five clock cycles until the input becomes low again.

The $\overline{\text{POLL}}$ and READY functions are used to synchronize CPU program execution with the operation of external devices.

$\overline{\text{RD}}$ (Read Strobe)

For small- and large-scale systems.

The CPU outputs this strobe signal during data read from an I/O device or memory. The $\overline{\text{IO}}$/M signal is used to select between I/O and memory.

The three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

S/LG (Small/Large)

For small- and large-scale systems.

This signal determines the operation mode of the CPU. This signal is fixed at either a high or low level. When this signal is a high level, the CPU will operate in small-scale system mode, and when low, in the large-scale system mode.Pins 24 to 31 and 34 function differently depending on the operating mode of the CPU. Separate nomenclature is adopted for these signals in the two operating modes.

Pin No.	Function	
	S/$\overline{\text{LG}}$-high	S/$\overline{\text{LG}}$-low
24	$\overline{\text{INTAK}}$	QS_1
25	ASTB	QS_0
26	$\overline{\text{BUFEN}}$	BS_0
27	BUF$\overline{\text{R}}$/W	BS_1
28	$\overline{\text{IO}}$/M	BS_2
29	$\overline{\text{WR}}$	$\overline{\text{BUSLOCK}}$
30	HLDAK	$\overline{RQ/AK_1}$
31	HLDRQ	$\overline{RQ/AK_0}$

INTAK (Interrupt Acknowledge)

For small-scale systems.

The CPU generates the $\overline{\text{INTAK}}$ signal low when it accepts an INT signal.

The interrupting device synchronizes with this signal and outputs the interrupt vector to the CPU via the data bus (AD_7-AD_0).

ASTB (Address Strobe)

For small-scale systems.

The CPU outputs this strobe signal to latch address information at an external latch.

ASTB is held at a low level after held at a high level for a half clock cycle during standby mode.

$\overline{\text{BUFEN}}$ (Buffer Enable)

For small-scale systems

This is used as the output enable signal for an external bidirectional buffer. The CPU genarates this signal during data transfer operations with external memory or I/O devices or during input of an interrupt vector.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

BUF$\overline{\text{R}}$/W (Buffer Read/Write)

For small-scale systems.

The output of this signal determines the direction of data transfer with an external bidirectional buffer. A high output causes transmission from the CPU to the external device; a low signal causes data transfer from the external device to the CPU.

BUFR/W is a three-state output and becomes high impedance during hold acknowledge.

$\overline{\text{IO}}$/M (IO/Memory)

For small-scale systems.

The CPU generates this signal to specify either I/O access or memory access. A high-level output specifies memory and a low-level signal specifies I/O.

IO/M's output is three state and becomes high impedance during hold acknowledge.

$\overline{\text{WR}}$ (Write strobe)

For small-scale systems.

The CPU generates this strobe signal during data write to an I/O device or memory. Selection of either I/O or memory is performed by the $\overline{\text{IO}}$/M signal.

This three-state output is held high during standby mode and enters the high-impedance state during hold acknowledge.

HLDAK (Hold Acknowledge)

For small-scale systems.

The HLDAK signal is used to indicate that the CPU accepts the hold request signal (HLDRQ). When this signal is a high level, the address bus, address/data bus, and the control lines become high impedance.

HLDRQ (Hold Request)

For small-scale systems.

This input signal is used by external devices to request the CPU to release the address bus, address/data bus, and the control bus.

This signal must be input in synchronization with external clock signals to satisfy the setup/hold time for normal operation.

UBE (Upper Byte Enable)

For small- and large-scale systems.

This output indicates the use of the upper 8 bits (AD_{15}-AD_8) of the Address/Data bus during T2-T4 of bus cycle. This signal is active low and output during T1-T4 of the bus cycle. Bus cycles in which the UBE is active are shown in the following table.

Operand	UBE	AD_0	No. of bus cycle
Word of Even Address	0	0	1
Word of Odd Address	0 1	1 ★ 0 ★★	2
Byte of Even Address	1	0	1
Byte of Odd Address	0	1	1

★First time, ★★Second time

The UBE signal goes low level continuously during interrupt acknowledge state (because of necessity of word access of even address for vector read).

This signal is held to high level in the standby mode. The UBE is 3-state output and becomes high impedance during hold acknowledge.

A_{19}/PS_3-A_{16}/PS_0 (Address Bus/Processor Status)

For small-and large-scale systems.

These pins are time multiplexed to operate as an address bus and as proccessor status signals.

When used as the address bus, these pins are the high 4 bits of the 20-bit memory address. During I/O access, all 4 bits output data 0.

The proccessor status signals are provided for both memory and I/O use. PS_3 is always 0 in the native mode and 1 in 8080 emulation mode. The interrupt enable flag (IE) is output to PS_2. Pins PS_1 and PS_0 indicate which memory segment is being accessed.

A_{17}/PS_1	A_{16}/PS_0	Segment
0	0	Data segment 1
0	1	Stack segment
1	0	Program segment
1	1	Data segment 0

The output of these pins is three state and becomes high impedance during hold acknowledge.

QS_1, QS_0 (Queue Status)

For large-scale systems.

The CPU uses these signals to allow external devices, such as the floating-point arithmetic processor chip, about the status of the internal CPU instruction queue.

QS_1	QS_0	Instruction Queue Status
0	0	NOP (Queue does not change)
0	1	First byte of instruction
1	0	Queue empty
1	1	Subsequent bytes of instruction

The instruction queue status indicated by these signals is the status when the execution unit (EXU) accesses the instruction queue. The data output from these pins is therefore valid only for one clock cycle immediately following queue access. These status signals are provided so that the floating-point processor chip can monitor the CPU's program execution status and synchronize its operation with the CPU when control is passed to it by the FPO (Floating Point Operation) instructions.

BS_2-BS_0 (Bus Status)

For large-scale systems.

The CPU uses these status signals to allow an external bus controller to monitor what the current bus cycle is.

The external bus controller decodes these signals and generates the control signals required to perform access of the memory or I/O device.

BS_2	BS_1	BS_0	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Program fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state

The output of these signals is three state and becomes high impedance during hold acknowledge.

These signals will be high from the rising edge of clock immediately after RESET signal is active to the next clock rise.

BUSLOCK (Bus Lock)

For large-scale systems.

The CPU uses this signal to secure the bus while executing the instruction immediately following the BUSLOCK prefix instruction, or during an interrupt acknowledge cycle. It is a status signal to the other bus masters in a multiprocessor system, inhibiting them from using the system bus during this time.

The output of this signal is three state and becomes high impedance during hold acknowledge.

$\overline{\text{BUSLOCK}}$ is high during standby mode except if the HALT instruction has a BUSLOCK prefix.

$\overline{\text{RQ}}/\overline{\text{AK}}_1$, $\overline{\text{RQ}}/\overline{\text{AK}}_0$ (Hold Request/Acknowledge)

For large-scale systems.

These pins function as bus hold request inputs ($\overline{\text{RQ}}$) and as bus hold acknowledge outputs ($\overline{\text{AK}}$). $\overline{\text{RQ}}/\overline{\text{AK}}_0$ has a higher priority than $\overline{\text{RQ}}/\overline{\text{AK}}_1$.

These pins have three-state outputs with on-chip pull-up resistors which keep the pin at a high level when the output is high impedance.

Bus Hold Request Input (RQ) must be input in synchronization with external clock signal to satisfy the setup/hold time for normal operation.

V_{DD} (Power Supply)

For small- and large-scale systems.

This pin is used for the +5V power supply.

GND (Ground)

For small- and large-scale systems. This pin is used for ground.

IC (Internally Connected)

The LH70116 is used with this pin at grouwd potential.

■ Register Configuration

Program Counter (PC)

The program counter is a 16-bit binary counter that contains the segment offset address of the next instruction which the EXU is to execute.

The PC increments each time the microprogram fetches an instruction from the instruction queue. A new location value is loaded into the PC each time a branch, call, return, or break instruction is executed. At this time, the contents of the PC are the same as the Prefetch Pointer (PEP).

Prefetch Pointer (PFP)

The prefetch pointer (PFP) is a 16-bit binary counter which contains a segment offset which is used to calculate a program memory address that the bus control unit (BCU) uses to prefetch the next byte for the instruction queue. The contents of PFP are an offset from the PS (Program Segment) register.

The PFP is incremented each time the BCU prefetches an instruction from the program memory. A new location will be loaded into the PFP whenever a branch, call, return, or break instruction is executed. At that time the contents of the PFP will be the same as those of the PC (Program Counter).

Segmet Registers (PC, SS, DS_0, and DS_1)

The memory addresses accessed by the LH70116 are divided into 64K-byte logical segments. The starting (base) address of each segment is specified by a 16-bit segment register, and the offset from this starting address is specified by the contents of another register or by the effective address.

These are the four types of segment registers used.

Segment Register	Default Offset
PS (Program Segment)	PFP
SS (Stack Segment)	PS, effective address
DS_0 (Data Segment 0)	IX, effective address
DS_1 (Data Segment 1)	IY

General-Purpose Registers (AW, BW, CW, and DW)

There are four 16-bit general-purpose registers. Each one can be used as one 16-bit register or as two 8-bit registers by dividing them into their high and low bytes (AH, AL, BH, BL, CH, CL, DH, DL).

Each register is also used as a default register for processing specific instructions. The default assignments are:

AW: Word multiplication/division, word I/O, data conversion
AL: Byte multiplication/division, byte I/O, BCD rotation, data conversion, translation
AH: Byte multiplication/division
BW: Translation
CW: Loop control branch, repeat prefix
CL: Shift instructions, rotation instructions, BCD operations
DW: Work multiplication/division, indirect addressing I/O

Pointers (SP, BP) and Index Registers (IX, IY)

These registers serve as base pointers or index registers when accessing the memory using based addressing, indexed addressing, or based indexed addressing.

These registers can also be used for data transfer and arithmetic and logical operations in the same manner as the general-purpose registers. They cannot be used as 8-bit registers.

Also, each of these registers acts as a default register for specific operations. The default assignments are:

SP: Stack operations
IX: Block transfer (source), BCD string operations

IY: Block transfer (destination), BCD string operations

Program Status Word (PSW)

The program status word consists of the following six status and four control flags.

Status Flags	Control Flags
• V (Overflow)	• MD (Mode)
• S (Sign)	• DIR (Direction)
• Z (Zero)	• IE (Interrupt Enable)
• AC (Auxiliary Carry)	• BRK (Break)
• P (Parity)	
• CY (Carry)	

When the PSW is pushed on the stack, the word images of the various flags are as shown here.

PSW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M D	1	1	1	V	D I R	I E	B R K	S	Z	0	A C	0	P	I	C Y

The status flags are set and reset depending upon the result of each type of instruction executed.

Instructions are provided to set, reset, and complement the CY flag directly.

Other instructions set and reset the control flags and control the operation of the CPU.

Q_0-Q_5 (Prefetch Queue)

The LH70116 has 6 byte instruction queue (FIFO), and it can store up to 6 instruction byte prefetched by the BCU. The instruction codes stored in the queue are fetched and executed by the EXU. The queue is cleared and prefetched with branch, call, return, or break instruction has been executed and when external interrupt has been acknowledged. Normally, the LH70108 prefetches if the queue has one word (two bytes) or more space. If the time required to prefetch the instruction code from the external memory is less than the mean execution time of instructions which are executed sequentially, then the actual instructions cycle will be shortened by this amount of time i. e. the instruction code to be next executed by the EXU can be available in the queue immediately after the completion of one instruction. As the result, processing speed is highly upgraded compared with the conventional CPU which fetch and execute instructions one by one. Queuing effect is lowered if there were many instructions which clears queue like the branch instruction or in the case of continuous instructions with too short instruction time.

DP (Data Pointer)

The data pointer is a 16-bit register indicates read/write addresses of variables. Effective address made in the effective address generator and the register contents including memory address offsets are transferred to the DP.

TEMP (Temporary Communication Register)

This is a 16-bit temporary register used by communications between external data bus and the EXU. The TEMP can be read or written by upper byte or lewer byte independently for byte access. Basically, the EXU completes write operation with transferring data to the TEMP and completes read operation with recognizing the data has been transferred to the TEMP from external data bus.

EAG (Effective Address Generator)

The Effective Address Generator (EAG) performs high-speed effective address calculation necessary for memory access. This completes all the calculations with 2 clocks for every addressing mode.

This fetches the instruction byte (2nd or 3rd byte) which has operand specifying field, if the instruction needs memory access. Then calculates effective address and transfers it to the DP (Data Pointer) and generates control signals relating to handling ALU and corresponding registers. In addition, if it is necessary, the EAG requests to the BCU for starting the bus cycle (memory read).

Instruction Decoder

The Instruction Decoder classifies lst byte of an instruction code into some groups with specific function and holds them during micro-instruction execution.

Microaddress Register

The microaddress register specifies the address of a microinstruction ROM to be next executed. At starting of a microinstruction execution, the lst byte of instruction bytes stored in the queue is fetched in this register and it specifies a start address of the corresponding microinstruction sequence.

Microinstruction ROM

The Microinstruction ROM has 1024 words by 29 bits of microinstructions.

Microinstruction Sequencer

The Microinstruction Sequencer controls the microaddress register operation, microinstruction

ROM output, and synchronizing the EXU with BCU.

ADM (Address Modifier)

Address Modifier performs the generation of physical address (adding segment register and PFP or DP) and increment of PFP (Prefetch Pointer).

TA/TB (Temporary Register/Shifter A, B)

The TA/TB are 16-bit temporary register/shifter used with execution of multiply/divided and shift/rotate (including BCD rotate) instructions. When executing multiply or divide instruction TA +TB operates as a 32-bit temporary register/shifter when executing shift/rotate instructions. Both the TA and TB can be read or written to and from the internal bus by upper byte or lower byte independently. The contents of the TA and TB are input to the ALU.

TC (Temporary Register C)

The TC is a 16-bit temporary register used with internal processing like the multiply or divide operation, etc. The TC content is output to the ALU.

ALU (Arithmetic & Logic Unit)

The Arithmetic and Logic Unit consists of a full adder and logical operation circuit and performs these operations:

1) Arithmetic operation (Add, Subtract, Multiply, Divide, increment, decrement, and complement)
2) Logical operation (test, AND, OR, XOR and bit test, set, clear, and complement)

LC (Loop Counter)

The Loop Counter (LC) is a 16-bit register which counts below items.

1) Loop number of the primitive block transfer and input/output instructions (MOVBK, OUTM, etc.) controlled with repeat prefix instructions (REP, REPC, etc.).
2) Shift number of the multi-bit shift/rotate instructions.

■ High-Speed Execution of Instructions

This section highlights the major architectural features that enhance the performance of the LH70116.

- Dual data bus in EXU
- Effective address generator
- 16/32-bit temporary registers/shifters (TA, TB)
- 16-bit loop counter
- PC and PFP

Dual Data Bus Method

To reduce the number of processing steps for instruction execution, the dual data bus method has been adopted for the LH70116 (figure 1). The two data buses (the main data bus and the subdata bus) are both 16 bits wide. For addition/subtraction and logical and comparison operations, processing time has been speeded up some 30% over single-bus systems.

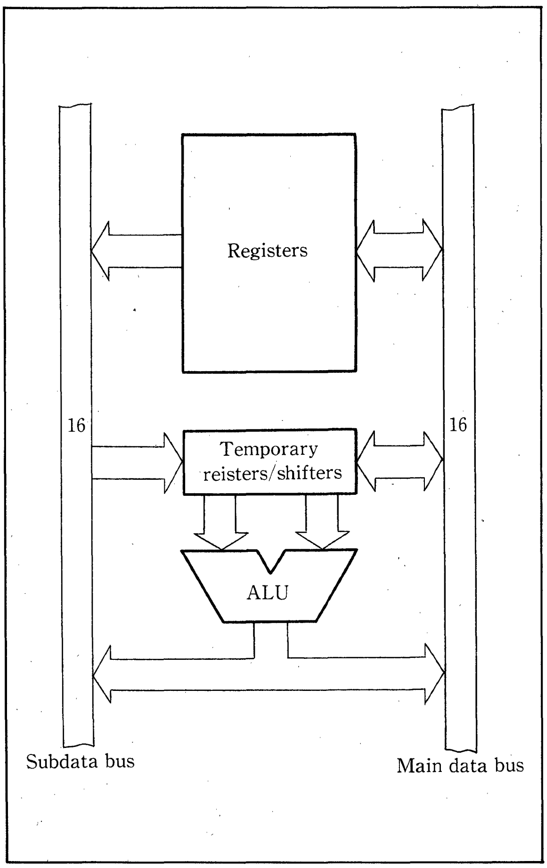


Fig. 1 Dual Data Buses

Example

ADD AW, BW; AW←AW+BW

Single Bus	Dual Bus
Step 1 TA←AW	TA←AW, TB←BW
Step 2 TB←BW	AW←TA+TB
Step 3 AW←TA+TB	

Effective Address Generator

This circuit (figure 2) performs high-speed processing to calculate effective addresses for accessing memory.

Calculation an effective address by the microprogramming method normally requires 5 to 12 clock cycles. This circuit requires only two clock cycles for address to be generated for any addressing mode. Thus, processing is several times faster.

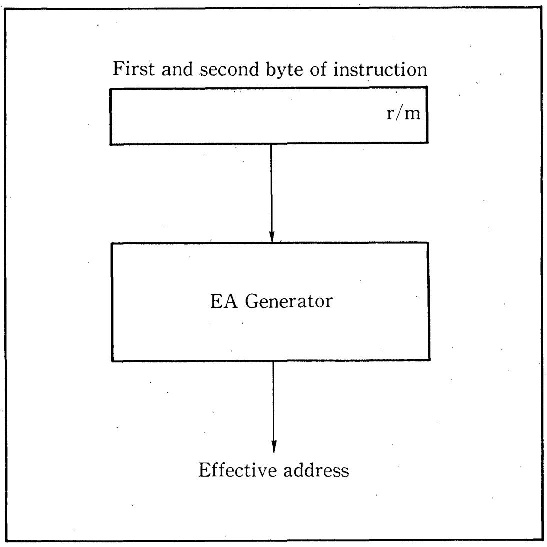


Fig. 2 Effective Address Generator

16/32-Bit Temporary Registers/Shifters (TA, TB)

These 16-bit temporary registers/shifters (TA, TB) are provided for multiplication/division and shift/rotation instructions.

These circuits have decreased the execution time of multiplication/division instructions. In fact, these instructions can be executed about four times faster than with the microprogramming method.

TA+TB: 32-bit temporary register/shifter for multiplication and division instructions.

TB 16-bit temporary register/shifter for shift/rotation instructions.

Loop Counter (LC)

This counter is used to count the number of loops for a primitive block transfer instruction controlled by a repeat prefix instruction and the number of shifts that will be performed for a multiple bit shift/rotation instruction.

The processing performed for a multiple bit rotation of a register is shown below. The average speed is approximately doubled over the microprogram method.

Example

RORC AW, CL; CL=5

Microprogram method	LC method
8+(4×5)=28 clocks	7+5=12 clocks

Program Counter and Prefetch Pointer (PC and PFP)

The LH70116 microprocessor has a program counter (PC), which addresses the program memory location of the instruction to be executed next, and a prefetch pointer (PFP), which addresses the program memory location to be accessed next. Both functions are provided in hardware. A time saving of several clocks is realized for branch, call, return, and break instruction execution, compared with microprocessors that have only one instruction pointer.

■ Unique Instructions

Variable Length Bit Field Operation Instructions

This category has two instructions: INS (insert Bit Field) and EXT (Extract Bit Field). These instructions are highly effective for computer graphics and high-level languages. They can, for example, be used for data structures such as packed arrays and record type data used in PASCAL.

(1) INS reg8, reg8/INS reg8, imm4

This instruction (figure 3) transfers low bits from the 16-bit AW register (the number of bits is specified by the second operand) to the memory location specified by the segment base (DS_1 register) pulse the byte offset (IY register). The starting bit position within this byte is specified as an offset by the lower 4 bits of the first operand.

After each complete data transfer, the IY register and the register specified by the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may specify the number of bits transferred (second operand). Because the maximum transferable bit length is 16 bits, only the lower 4 bits of the specified register (00H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

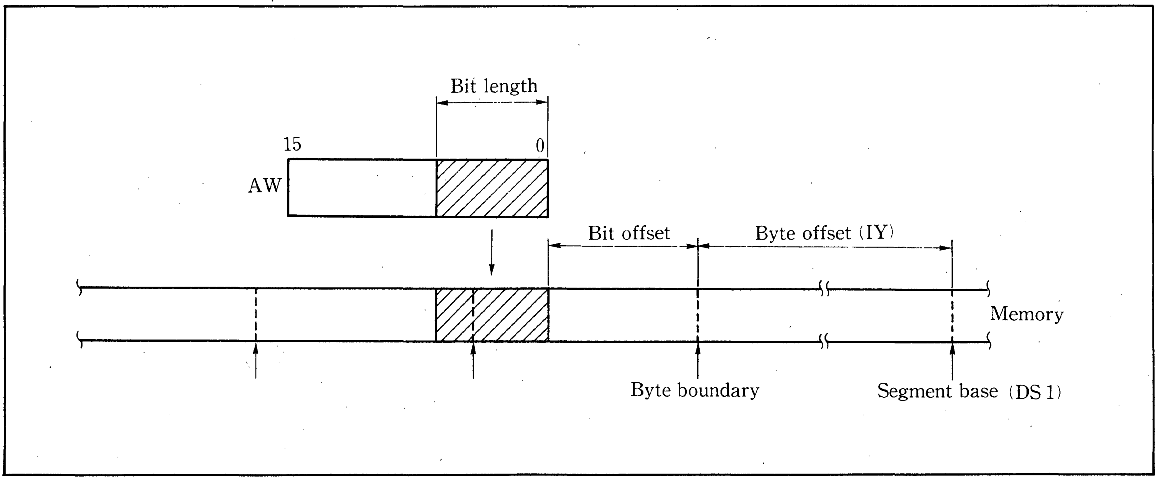


Fig. 3 Bit Field Insertion

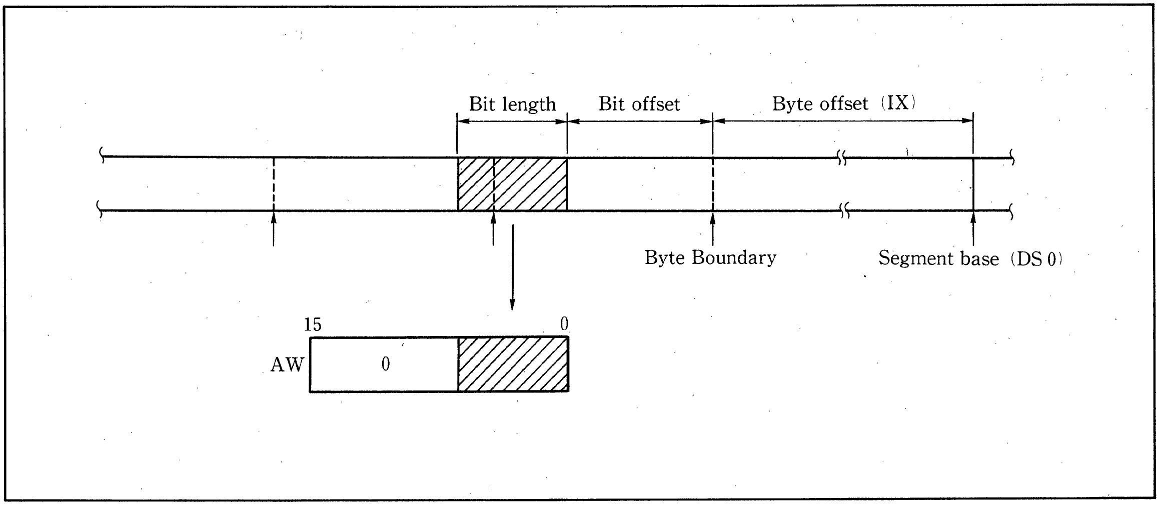


Fig. 4 Bit Field Extraction

(2) EXT reg8, reg8/EXT reg8, imm4

This instruction (figure 4) loads to the AW register the bit field data whose bit length is specified by the second operand of the instruction from the memory location that is specified by the DS_0 segment register (segment base), the IX index register (byte offset), and the lower 4 bits of the first operand (bit offset).

Packed BCD Operation Instructions

The instructions described here process packed BCD data either as strings (ADD4S, SUB4S, CMP4S) or byte-format operands (ROR4, ROL4). After the transfer is complete, the IX register and the lower 4 bits of the first operand are automatically updated to point to the next bit field.

Either immediate data or a register may be specified for the second operand. Because the maximum transferrable bit length is 16 bits, however, only the lower 4 bits of the specified register (0H to 0FH) will be valid.

Bit field data may overlap the byte boundary of memory.

(1) ADD4S

This instruction adds the packed BCD string addressed by the IX index register to the packed BCD string addressed by the IY index register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

BCD string (IY, CL)←BCD string (IY, CL)+BCD string (IX, CL)

(2) SUB4S

This instruction subtracts the packed BCD string addressed by the IX index register from the packed BCD string addressed by the IY register, and stores the result in the string addressed by the IY register. The length of the string (number of BCD digits) is specified by the CL register, and the result of the operation will affect the overflow flag (V), the carry flag (CY), and zero flag (Z).

BCD string (IY, CL)←BCD string (IY, CL)−BCD String (IX, CL)

(3) CMP4S

This instruction performs the same operation as SUB4S except that the result is not stored and only the overflow (V), carry flags (CY) and zero flag (Z) are affected.

BCD string (IY, CL)−BCD string (IX, CL)

(4) ROL4

This instruction (Fig. 5) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4-bits of the AL register (AL_L) to rotate that data one BCD digit to the left.

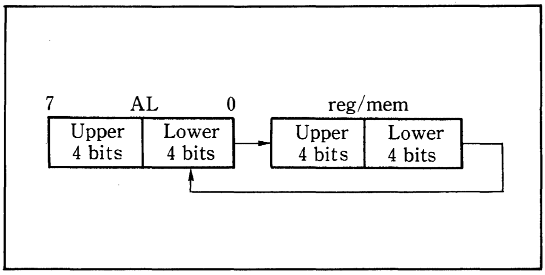


Fig. 5 BCD Rotate Left (ROL4)

(5) ROR4

This instruction (Fig. 6) treats the byte data of the register or memory directly specified by the instruction byte as BCD data and uses the lower 4-bits of the AL register (AL_L) to rotate the data one BCD digit to the right

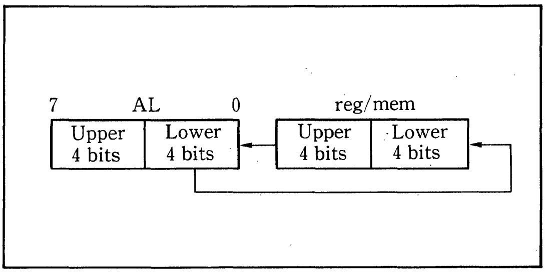


Fig. 6 BCD Rotate Right (ROR4)

Stack Operation Instruction

(1) PREPARE imm16, imm8

This instruction is used to generate "stack frames" required by the block structures of high-level languages such as Pascal and Ada. The stack frame includes a local variable area as well as pointers. These frame pointers point to the frame containing the variables that can be referenced from the current procedure.

The program example based upon Pascal language is shown below.

```
program EXAMPLE ;
  procedure P ;
    var a, b, c ;
    procedure Q ;
      var d, e ;
      procedure R ;
        var f, g ;
        begin
          d : =a+f+g ;
        end ;
      begin
        R ;
        b : =d+e ;
      end ;
    begin
      a ; =b+c ;
      Q :
    end ;
(*main program*)
  begin
    P ;
  end.
```

Note: The variables are defined as the words.

This program is an example of a procedure block with a triple nesting.

Procedure	Variables
P	a, b, c
Q	d, e
R	f, g

Accordingly, the global variables of a, b and c are referenced from the procedure Q, and a, b, c, d and e from the procedure R.

This instruction copies frame-pointers to reserve the local variable area and to enable global variable references. The first operand (16-bit immediate data) specifies (in bytes) the size of the local variable area. The second operation (8-bit immediate data) specifies the depth (or lexical level) of the procedure block. The frame base address generated by this instruction is set in the BP base pointer.

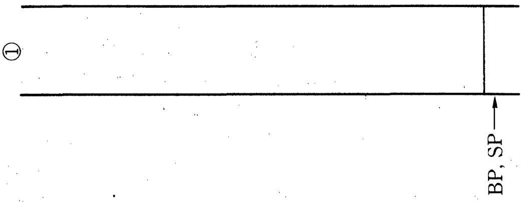
To compile the EXAMPLE program follows the assembler program shown next. (The DISPOSE instruction in the assembler program is used to return the stack pointer SP and the base pointer BP to the state just before the PREPARE instruction is executed. See DISPOSE section mentioned later.)

```
START : MOV       SP, SPTOP
        MOV       BP, SP              ;①
        CALL      P                   ;②
        BR        SYSTEM
P :     PREPARE   6, 1                ;③
        MOV       AW, [BP][B+BLEVEL×2]
        ADD       AW, [BP][C+CLEVEL×2]
        MOV       [BP][A+ALEVEL×2], AW
        CALL      Q
        DISPOSE
        RET
```

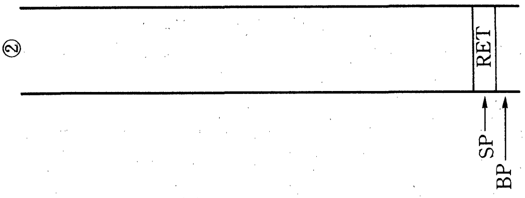
```
Q :    PREPARE   4, 2                 ;④
       CALL      R
       MOV       AW, [BP][D+DLEVEL×2]
       ADD       AW, [BP][E+ELEVEL×2]
       MOV       IY, [BP][BLEVEL×2]
       MOV       SS : [IY][B+BLEVEL×2], AW
       DISPOSE
       RET
R :    PREPARE   4, 3                 ;⑤
       MOV       AW, [BP][F+FLEVEL×2]
       ADD       AW, [BP][G+GLEVEL×2]
       MOV       IY, [BP][ALEVEL×2]
       ADD       AW, SS : [IY][A+ALEVEL×2]
       MOV       IY, [BP][DLEVEL×2]
       MOV       SS : [IY][D+DLEVEL×2], AW
       DISPOSE
       RET
  ;A=-2          ALEVEL=-1
  ;B=-4          BLEVEL=-1
  ;C=-6          CLEVEL=-1
  ;D=-2          DLEVEL=-2
  ;E=-4          ELEVEL=-2
  ;F=-2          FLEVEL=-3
  ;G=-4          GLEVEL=-3
```

The process of the generation of the stack frame according to the program is shown next. The numbers are referred to that in the program.

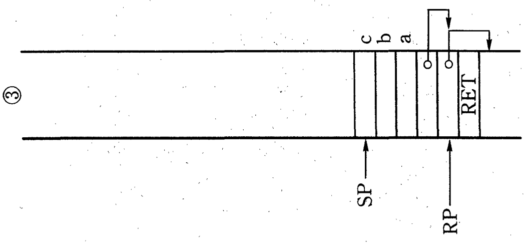
①



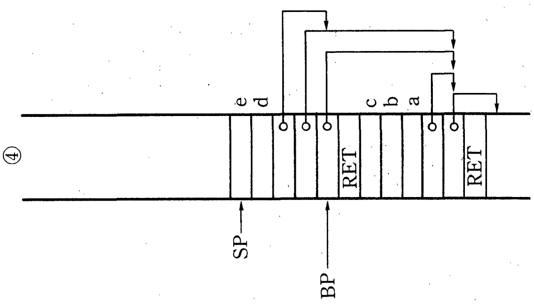
②



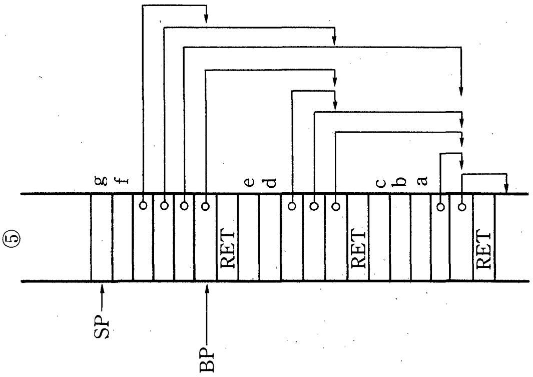
③



④



⑤



First the old BP value is saved to the stack. This is done so that BP of the calling procedure can be restored when the called procedure terminates. The frame pointer (BP value saved to the stack) that indicates the range of variables that can be referenced by the called procedure is placed on the stack. This range is always a value one less than the stack. This range is always a value one less than the lexical level of the procedure.

If the lexical level of a procedure is greater than 1, the pointers of that procedure will also be saved on the stack. This is so that the frame pointer of the calling procedure can also be copied when frame pointer copy is performed within the called procedure.

Next the new frame pointer value is set in BP and the area for local variables used by the procedure is reserved in the stack. In other words, SP is decremented only for the amount of stack memory required by the local variables.

```
  display=2nd operand
  dynamics=1st operand
SP=SP-2 ;
(SP)=BP ;
temp=SP ;
if display>0 then begin
  repeat display-1 times
```

```
      begin
        SP=SP−2 ;
        SP=BP−2 ;
        (SP)=(BP) ;
      end ;
    SP=SP−2 ;
    (SP)=temp ;
    end ;
  BP=temp ;
  SP=SP−dynamics
```

Mode Operating Instructions

The LH70116 has two operating modes (Fig. 7). One is the native mode, the other is the emulation mode in which the instruction set of the 8080A is emulated. A mode flag (MD) is provided to select between these two modes. Native mode is selected when MD is 1 and emulation mode when MD is 0. MD is set and reset, directly and indirectly, by executing the mode manipulation instructions.

Two instructions are provided to switch operation from the native mode to the emulation mode and back: BRKEM (Break for Emulation), and RETEM (Return from Emulation).

Two instructions are used to switch from the emulation mode to the native mode and back: CALLN (Call Native Routine), and RETI (Return from Interrupt).

The system will return from the 8080 emulation mode to the native mode when the RESET signal is present, or when an external interrupt (NMI or INT) is present.

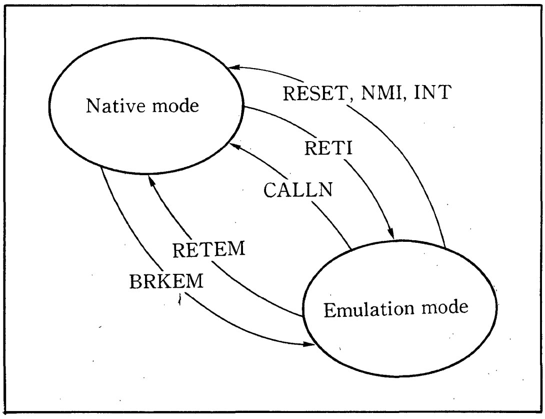


Fig. 7 V30 Modes

Data Access

(1) Local variable access

The local variables are assigned in the frame of the procedure. The effective address EA. L of the local variables is defined by the formula :

EA. L=SS : (BP+offset)

The offset value is defined as the sum of the frame size (referenced frame base) and the variable from the base of the local variable area.

(2) Global variable access

The global variable is located at the address added by the offset of variables which are referenced to the accessed value of the base pointer of the old one saved on the stack frame.

The effective address EA. G is defined as below.

EA. G=SS : ((SS : (BP+offset 1))+offset 2)

The offset 1 is defined by the offset value from the frame base (BP) to the address stored by the base address of the frame including the global variables.

The offset 2 is defined by the offset value from the frame base including variables to be referenced to the variables.

DISPOSE

This instruction releases the last stack frame generated by the PREPARE instruction. It returns the stack and base pointers to the values they had before the PREPARE instruction was used to call a procedure.

```
SP=BP ;
BP=(SP) ;
SP=SP+2
```

Check Array Boundary Instruction

This instruction is used to verify that index values pointing to the elements of an array data structure are within the defined range. The lower limit of the array should be in memory location mem32, the upper limit in mem32+2. If the index value in reg16 is not between these limits when CHKIND is executed, a BRK 5 will occur. This causes a jump to the location in interrupt vector 5.

```
      CHKIND reg16, mem32
When (mem32)>reg16 or (mem32+2)<reg16
      TA←(015H, 014H)                 |
      TC←(017H, 016H)                 |
      SP←SP−2, (SP+1, SP)←PSW         |
      IE←0, BREK←0                    |
      SP←SP−2, (SP+1, SP)←PS          } =BRK 5
      PS←TC                           |
      SP←SP−2, (SP+1, SP)←PC          |
      PC←TA                           |
```

(1) BRKEN imm8

This is the basic instruction used to start the 8080 emulation mode. This instruction operates exactly the same as the BRK instruction, except that BRKEM resets the mode flag (MD) to 0. PSW, PS, and PC are saved to the stack. MD is then reset and the interrupt vector specified by the operand imm8 of this command is loaded into PS and PC.

The instruction codes of the interrupt processing routine jumped to are then fetched. Then the CPU executes these codes as 8080A instructions.

In 8080 emulation mode, registers and flags of the 8080A are performed by the following registers and flags of the LH70116

	8080A	LH70116
Registers:	A	AL
	B	CH
	C	CL
	D	DH
	E	DL
	H	BH
	L	BL
	SP	BP
	PC	PC
Flags:	C	CY
	Z	Z
	S	S
	P	P
	AC	AC

In the native mode, SP is used for the stack pointer. In the 8080 emulation mode this function is performed by BP.

The use of independent stack pointers allows independent stack areas to be secured for each mode and keeps the stack of one of the modes from being destroyed by an erroneous stack operation in the other mode.

The SP, IX, IY and AH registers and the four segment registers (PS, SS, DS_0, and DS_1) used in the native mode are not affected by operations in 8080 emulation mode.

In the 8080 emulation mode, the segment register for instructions is determined by the PS register (set automatically by the interrupt vector) and the segment register for data is the DS_0 register (set by the programmer immediately before the 8080 emulation mode is entered).

It is prohibited to nest BRKEM instructions.

(2) RETEM (no operand)

When RETEM is executed in 8080 emulation mode (interpreted by the CPU as an 8080A instruction), the CPU restores PS, PC, and PSW (as it would when returning from an interrupt processing routine), and returns to the native mode. At the same time, the contents of the mode flag (MD) which was saved to the stack by the BRKEM instruction, is restored to MD=1. The CPU is set to the native mode.

(3) CALLN imm8

This instruction makes it possible to call the native mode subroutines from the 8080 emulation mode. To return from subroutine to the emulation mode, the RETI instruction is used.

The processing performed when this instruction is executed in the 8080 emulation mode (it is interpreted by the CPU as 8080A instruction), is similar to that performed when a BRK instruction is executed in the native mode. The imm8 operand specifies an interrupt vector type. The contents of PS, PC, and PSW are pushed on the stack and an MD flag value of 0 is saved. The mode flag is set to 1 and the interrupt vector specified by the operand is loaded into PS and PC.

(4) RETI (no operand)

This is a general-purpose instruction used to return from interrupt routines entered by the BRK instruction or by an external interrupt in the native mode. When this instruction is executed at the end of a subroutine entered by the execution of the CALLN instruction, the operation that restores PS, PC, and PSW is exactly the same as the the native mode execution. When PSW is restored, however, the 8080 emulation mode value of the mode flag (MD) is restored, the CPU is set in emulation mode, and all subsequent instructions are interpreted and executed as 8080A instructions.

RETI is also used to return from an interrupt procedure initiated by an NMI or INT interrupt in the emulation mode.

■ Floating Point Operation Chip Instructions FPO1 fp-op, mem/FPO2 fp-op, mem

These instructions are used for the external floating point processor. The floating point operation is passed to the floating point processor when the CPU fetches one of these instructions. From this point the CPU performs only the necessary auxiliary processing (effective address calculation, generation of physical addresses, and start-up of the memory read cycle).

The floating point processor always monitors the instructions fetched by the CPU. When it interprets one as an instruction to itself, it performs the appropriate processing. At this time, the floating point processor chip uses either the address alone or both the address and read data of the memory read cycle executed by the CPU. This difference in the data used depends on which of these instructions is executed.

Note: During the memory read cycle initiated by the CPU for FPO1 or FPO2 execution, the CPU does not accept any read data on the data bus from memory. Although the CPU generates the memory address, the data is used by the floating point processor.

■ Interrupt Operation

The interrupts used in the LH70116 can be divided into two types: interrupts generated by external interrupt requests and interrupts generated by software processing. These are the classifications.

(1) External Interrupts

(a) NMI input (nonmaskable)

(b) INT input (maskable)

(2) Software Processing

As the result of instruction execution

- When a divide error occurs during execution of the DIV or DIVU instruction
- When a memory-boundary-over error is detected by the CHKIND instruction

Conditional break instruction

- When V=1 during execution of the BRKV instruction

Unconditional break instructions

- 1-byte break instruction: BRK3
- 2-byte break instruction: BRK imm8

Flag processing

- When stack operaitons are used to set the BRK flag

8080 Emulation mode instructions

- BRKEM imm8
- CALLN imm8

Interrupt Vectors

Starting addresses for interrupt processing routines are either determined automatically by a single location of the interrupt vector table or selected each time interrupt processing is entered.

The interrupt vector table is shown in figure 8. The table uses 1K bytes of memory addresses 000H to 3FFH and can store starting address data for a maximum of 256 vectors (4 bytes per vector).

The correponding interrupt sources for vectors 0 to 5 are predetermined and vectors 6 to 31 are reserved These vectors consequently cannot be used for general applications

The BRKEM instruction and CALLN instruction (in the emulation mode) and the INT input are available for general applications for vectors 32 to 255.

A single interrupt vector is made up of 4 bytes (Fig. 9). The 2 bytes in the low addresses of memory are loaded into PC as the offset, and the high 2 bytes are loaded into PS as the base address. The bytes are combined in reverse order. The lower-order bytes in the vector become the most significant bytes in the PC and PS, and the higher-order bytes become the least significant bytes.

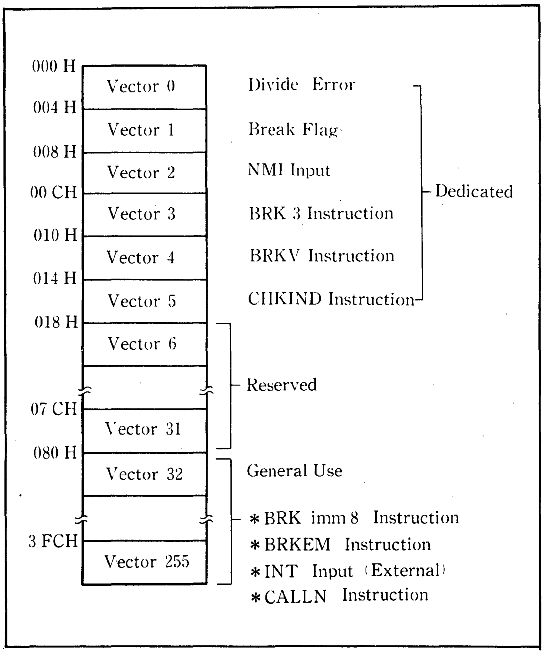


Fig. 8 Interrupt Vector Table

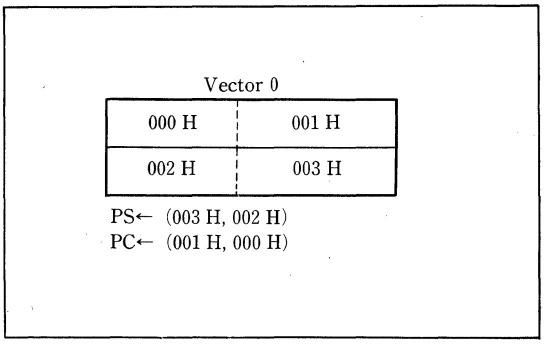


Fig. 9 Interrupt Vector 0

Based on this format, the contents of each vector should be initialized at the beginning of the program.

The basic steps to jump to an interrupt processing routine are now shown.

TA←vector low bytes (offset)
TC←vector high bytes (segment base)
SP←SP−2, (SP+1, SP)←PSW
IE←0, BRK←0, MD←0
SP←SP−2, (SP+1, SP)←PS
PS←TC
SP←SP−2, (SP+1, SP)←PC
PC←TA

Standby Function

The LH70116 has a standby mode to reduce power consumption during program wait states. This mode is set by the HALT instruction in both the native and the emulation mode.

In the standby mode, the internal clock is supplied only to those circuits related to functions required to release this mode and bus hold control functions. As a result, power consumption can be reduced to 1/10 the level of normal operation in either native or emulation mode.

The standby mode is released by inputting a RESET signal or an external interrupt (NMI, INT).

The bus hold function is effective during standby mode. The CPU returns to standby mode when the bus hold request is removed.

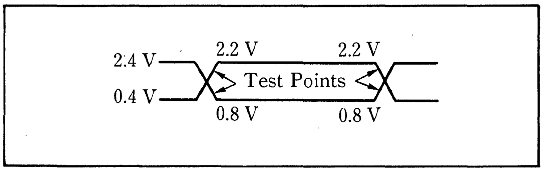
During standby mode, all control outputs are disabled and the address/data bus will be at either high or low levels.

I/O Address Reservation

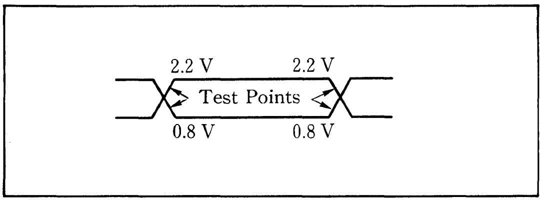
Reserve upper 256 bytes of I/O address (FF00H-FFFFH) in case it may be used in future.

Timing Diagram

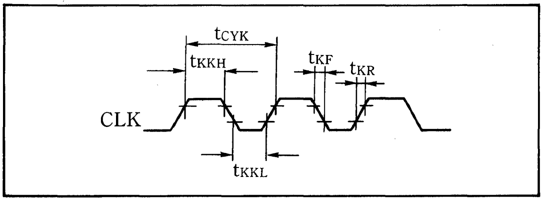
(1) AC Test Input Waveform (Except CLK)



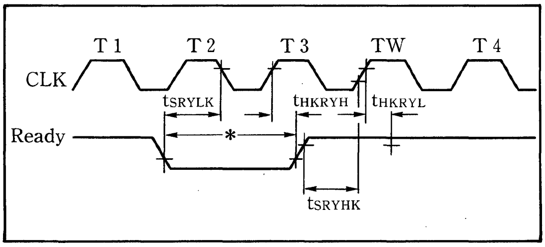
(2) AC Output Test Points



(3) Clock Timing

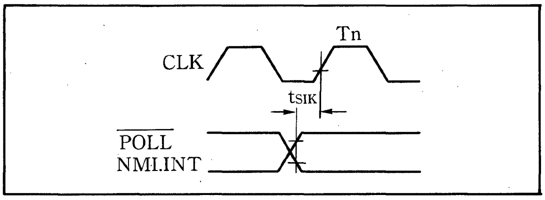


(4) Wait (Ready) Timing

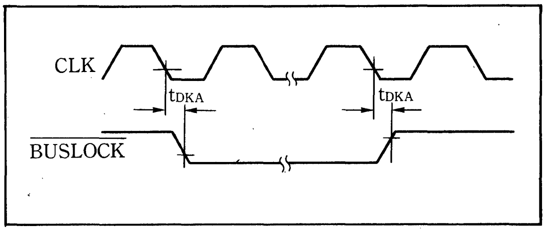


*Read signal must be held at LOW or HIGH during this period.

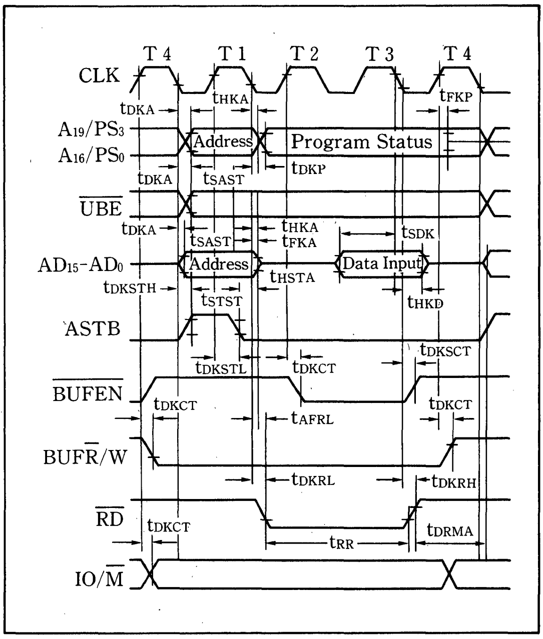
(5) POLL, NMI, INT Input Timing



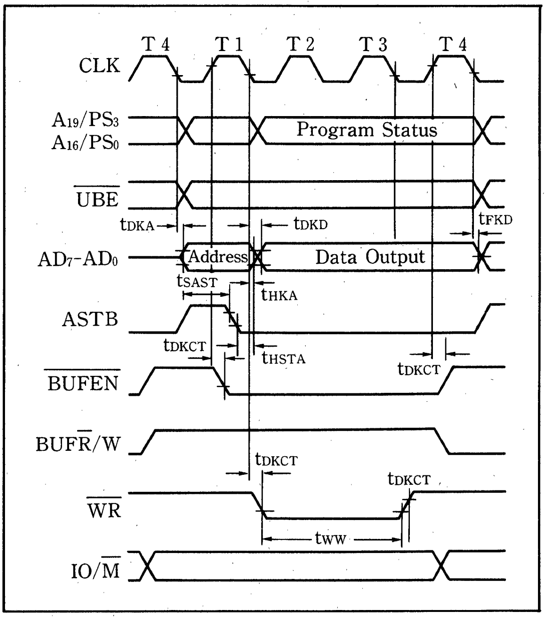
(6) BUSLOCK Output Timing



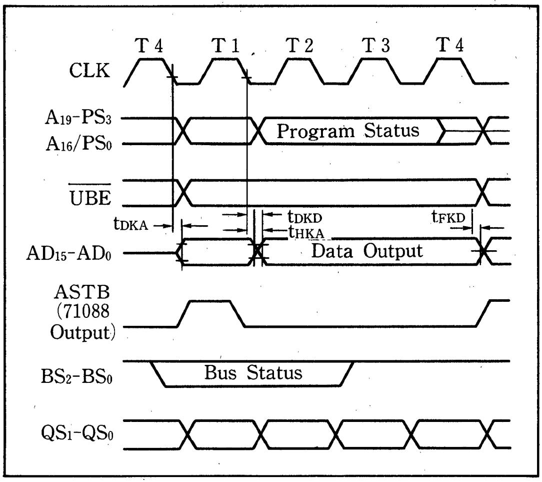
(7) Read Timing (Small Scale)



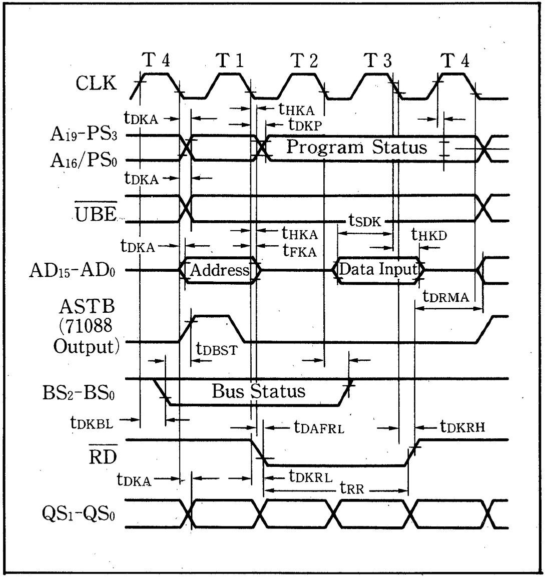
(8) Write Timing (Small Scale)



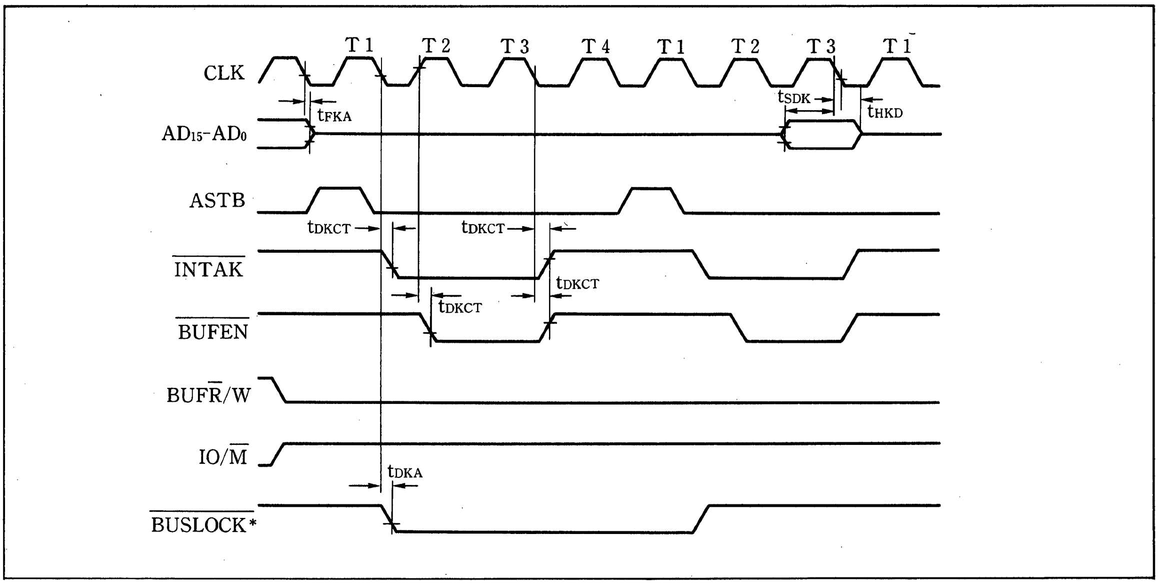
(10) Write Timing (Large Scale)



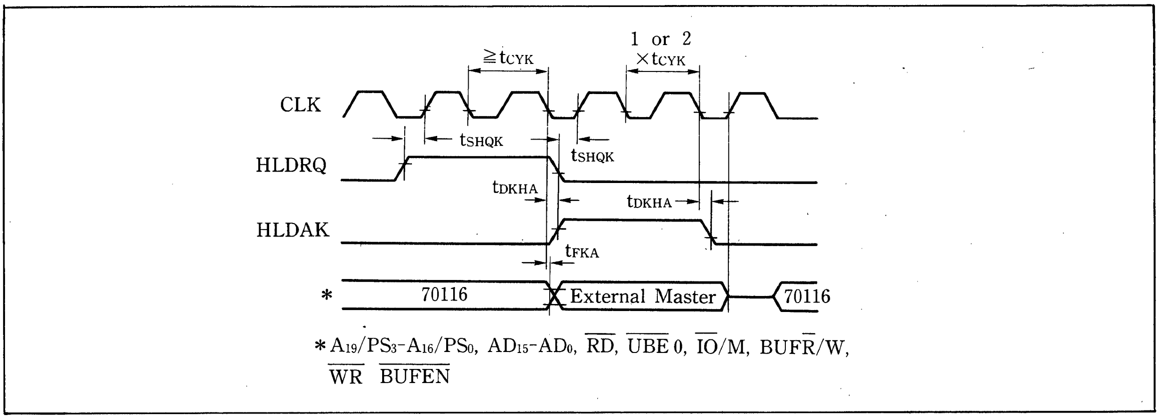
(9) Read Timing (Large Scale)



(11) Interrupt Acknowledge Timing

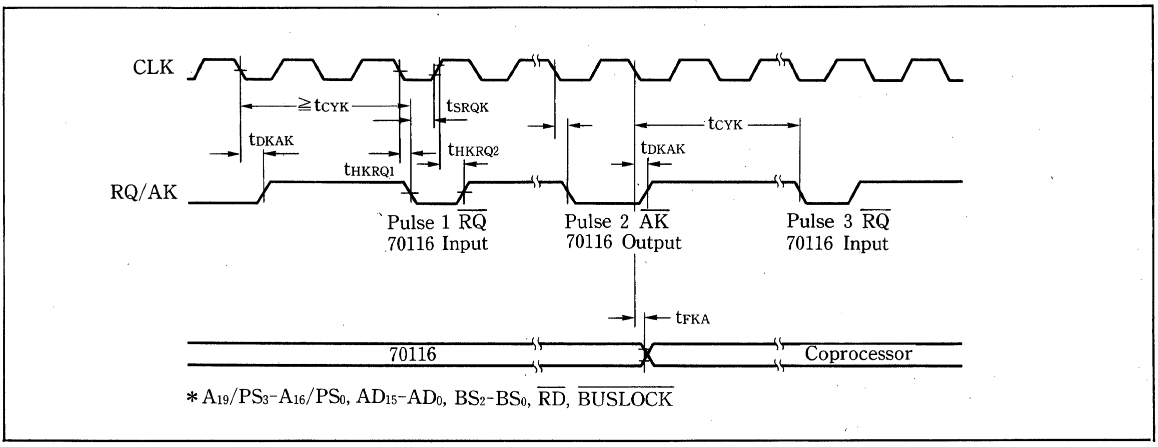


(12) Hold Request/Acknowledge Timing (Small Scale)



* A19/PS3-A16/PS0, AD15-AD0, RD, UBE 0, IO/M, BUFR/W, WR BUFEN

(13) Bus Request/Acknowledge Timing (Large Scale)



* A19/PS3-A16/PS0, AD15-AD0, BS2-BS0, RD, BUSLOCK

Table 1 Operand Types

Identifier	Description
reg	8-or 16-bit general-purpose register
reg8	8-bit general-pourpose register
reg16	16-bit general-purpose register
dmem	8-or 16-bit direct memory location
mem	8-or 16-bit memory location
mem8	8-bit memory location
mem16	16-bit memory locotion
mem32	32-bit memory location
imm	Constant (0 to FFFFH)
imm16	Constant (0 to FFFFH)
imm8	Constant (0 to FFH)
imm4	Constant (0 to FH)
imm3	Constant (0 to 7)
acc	AW or AL register
sreg	Segment register
src-table	Name of 256-byte translation table
src-block	Name of block addressed by the IX register
dst-block	Name of block addressed by the IY register
near-proc	Procedure within the current program segment
far-proc	Procedure located in another program segment
near-label	Label in the current program segment
short-label	Label between −128 and +127 bytes from the end of instruction
far-label	Label in another program segment
memptr16	Word containing the offset of the memory location within the current program segment to which control is to be transferred
memptr32	Double word containing the offset and segment base address of the memory location to which control is to be transferred
regptr16	16-bit register containing the offset of the memory location within the program segment to which control is to be transferred
pop-value	Number of bytes of the stack to be discarded (0 to 64K bytes, usually even addresses)
fp-op	Immediate data to identify the instruction code of the external floating point operation
R	Register set

Table 2 Operation Code Types

Identifier	Description
W	Word/byte field (0 to 1)
reg	Register field (000 to 111)
mem	Memory field (000 to 111)
mod	Mode field (00 to 10)
S: W	When S: W=01 or 11, data=16 bits. At all other times, data=8 bits.
X, XXX, YYY, ZZZ	Data to identify the instruction code of the external floating point arithmetic chip

Table 3 Operational Description Types

Identifier	Description
AW	Accumulator (16 bits)
AH	Accumulator (high byte)
AL	Accumulator (low byte)
BW	BW register (16 bits)
CW	CW register (16 bits)
CL	CW register (low byte)
DW	DW register (16 bits)
SP	Stack pointer (16 bits)
PC	Program counter (16 bits)
PSW	Program status word (16 bits)
IX	Index register (source 16 bits)
IY	Index register (destination 16 bits)
PS	Program segment register (16 bits)
SS	Stack segment register (16 bits)
DS_0	Data segment 0 register (16 bits)
DS_1	Data segment 1 register (16 bits)
AC	Auxiliary carry flag
CY	Carry flag
P	Parity flag
S	Sign flag
Z	Zero flag
DIR	Direction flag
IE	Interrupt enable flag
V	Overflow flag
BRK	Break flag
MD	Mode flag
(...)	Values in parentheses are memory contents
disp	Displacement (8 or 16 bits)
ext-disp8	16-bit displacement (sign-extension byte +8-bit displacement)
temp	Temporary register (8/16/32 bits) TA: Temporary register A (16 bits) TB: Temporary register B (16 bits) TC: Temporary register C (16 bits)

Operational Description Types (cont)

Identifier	Description
tmpcy	Temporary carry flag (1 bit)
seg	Immediate segment data (16 bits)
offset	Immediate offset data (16 bits)
←	Transfer direction
+	Addition
−	Subtraction
×	Multiplication
÷	Division
%	Modulo
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
XXH	Two-digit hexadecimal value
XXXXH	Four-digit hexadecimal value

Table 4 Operations

Identifier	Description
(blank)	No change
0	Cleared to 0
1	Set to 1
X	Set or cleared according to the result
U	Undefind
R	Value seved earlier is restored

Table 5 Memory Address

mem	mod		
	00	01	10
000	BW+IX	BW+IX+disp8	BW+IX+disp16
001	BW+IY	BW+IY+disp8	BW+IY+disp16
010	BP+IX	BP+IX+disp8	BP+IX+disp16
011	BP+IY	BP+IY+disp8	BP+IY+disp16
100	IX	IX+disp8	IX+disp16
101	IY	IY+disp8	IY+disp16
110	Direct address	BP+disp8	BP+disp16
111	BW	BW+disp8	BW+disp16

Table 6 Selection of 8-and 16-Bit Registers (mod 11)

reg	W=0	W=1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

Table 7 Selection of Segment Registers

sreg	
00	DS_1
01	PS
10	SS
11	DS_0

The table on the following pages shows the instruction set.

At "No. of Clocks," for instructions referencing memory operands, the left side of the slash (/) is the number of clocks for byte operands and the right side is for word operands. For conditional control transfer instructions, the left side of the slash (/) is the number of clocks if a control transfer takes place. The right side is the number of clocks when no control transfer or branch occurs. Some instructions show a range of clock times, separated by a hyphen. The execution time of these instructions varies from the minimum value to the maximum, depending on the operands involved.

Note: Add four clocks these times for each word transfer made to an odd address.

"No. of Clocks" includes these times:

- Decoding
- Effective address generation
- Operand fetch
- Execution

It assumes that the instruction bytes have been prefetched.

Table 8 Primitive Block Transfer Instructions

(n: number of transfers)

Mnemonic	Byte clocks (W=0)	Word clock (W=1)		
		Odd, Odd address	Odd, Even address	Even, Even address
MOVBK	11+8n (11)	11+16n (19)	11+12n (15)	11+8n (11)
CMPBK	7+14n (13)	7+22n (21)	7+18n (17)	7+14n (13)

Note : Values in parentheses apply to the case of single processing.

Table 9 Primitive I/O Instructions

(n: number of transfers)

Mnemonic	Byte clocks (W=0)	Word clock (W=1)	
		Odd address	Even address
CMPM	7+10n (7)	7+14n (11)	7+10n (7)
LDM	7+9n (7)	7+13n (11)	7+9n (7)
STM	7+4n (7)	7+8n (11)	7+4n (7)
INM	9+8n (10)	9+16n (18)	9+8n (10)
OUTM	9+8n (10)	9+16 (18)	9+8 (10)

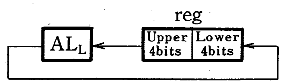
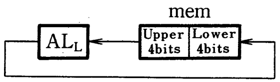
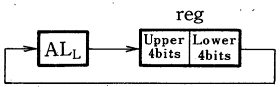
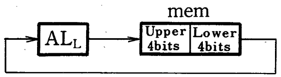
Note : Values in parentheses apply to the case of single proccesing.

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Data Transfer Instructions																										
MOV	reg, reg	reg←reg	1	0	0	0	1	0	1	W	1	1		reg			reg		2	2						
	mem, reg	(mem)←reg	1	0	0	0	1	0	0	W	mod			reg			mem		9/13	2-4						
	reg, mem	reg←(mem)	1	0	0	0	1	0	1	W	mod			reg			mem		11/15	2-4						
	mem, imm	(mem)←imm	1	1	0	0	0	1	1	W	mod		0	0	0		mem		11/15	3-6						
	reg, imm	reg←imm	1	0	1	1	W		reg										4	2-3						
	acc, dmem	When W=0 AL←(dmem) When W=1 AH←(dmem+1), AL←(dmem)	1	0	1	0	0	0	0	W									10/14	3						
	dmem, acc	When W=0 (drem)←AL When W=1 (dmem+1)←AH, (dmem)←AL	1	0	1	0	0	0	1	W									9/13	3						
	sreg, reg16	sreg←reg16 sreg: SS, DS0, DS1	1	0	0	0	1	1	1	0	1	1	0		sreg		reg		2	2						
	sreg, mem16	sreg←(mem16) sreg: SS, DS0, DS1	1	0	0	0	1	1	1	0	mod		0		sreg		mem		11/15	2-4						
	reg16, sreg	reg16←sreg	1	0	0	0	1	1	0	0	1	1	0		sreg		reg		2	2						
	mem16, sreg	(mem16)←sreg	1	0	0	0	1	1	0	0	mod		0		sreg		mem		10/14	2-4						
	DS0, reg16, mem32	reg16←(mem32) DS0←(mem32+2)	1	1	0	0	0	1	0	1	mod			reg			mem		18/26	2-4						
	DS1, reg16, mem32	reg16←(mem32) DS1←(mem32+2)	1	1	0	0	0	1	0	0	mod			reg			mem		18/26	2-4						
	AH, PSW	AH←S, Z, x, AC, x, P, x, CY	1	0	0	1	1	1	1	1									2	1	x	x		x	x	x
	PSW, AH	S, Z, x, AC, x, P, x, CY←AH	1	0	0	1	1	1	1	0									3	1	x	x		x	x	x
LDEA	reg16,mem16	reg16←mem16	1	0	0	0	1	1	0	1	mod			reg			mem		4	2-4						
TRANS	src-table	AL←(BW+AL)	1	1	0	1	0	1	1	1									9	1						
XCH	reg, reg	reg←→reg	1	0	0	0	0	1	1	W	1	1		reg			reg		3	2						
	mem, reg or reg, mem	(mem)←→reg	1	0	0	0	0	1	1	W	mod			reg			mem		16/24	2-4						
	AW, reg16 or reg16, AW	AW←→reg16	1	0	0	1	0		reg										3	1						
Repeat Prefixes																										
REPC		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed. When CY≠1, exit the loop.	0	1	1	0	0	1	0	1									2	1						
REPNC		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed. When CY≠0, exit the loop.	0	1	1	0	0	1	0	0									2	1						
REP REPE REPZ		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed. If the primitive block transfer instruction is CMPBK or CMPM and Z≠1, exit the loop.	1	1	1	1	0	0	1	1									2	1						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Repeat Prefixes (cont)																										
REPNE REPNZ		While CW≠0, the next byte of the primitive block transfer instruction is executed and CW is decremented (−1). If there is a waiting interrupt, it is processed. If the primitive block transfer instruction is CMPBK or CMPM and Z≠0, exit the loop.	1	1	1	1	0	0	1	0									2	1						
Primitive Block Transfer Instructions																										
MOVBK	dst-block, src-block	When W=0 (IY)←(IX) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX−1, IY←IY−1 When W=1 (IY+1, IY)←(IX+1, IX) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX−2, IY←IY−2	1	0	1	0	0	1	0	W									See Table 8	1						
CMPBK	src-block, dst-block	When W=0 (IX)−(IY) DIR=0: IX←IX+1, IY←IY+1 DIR=1: IX←IX−1, IY←IY−1 When W=1 (IX+1, IX)←(IY+1, IY) DIR=0: IX←IX+2, IY←IY+2 DIR=1: IX←IX−2, IY←IY−2	1	0	1	0	0	1	1	W									7+13n 7+21n	1	x	x	x	x	x	x
CMPM	dst-block	When W=0 AL−(IY) DIR=0: IY←IY+1; DIR=1:IY←IY−1 When W=1 AW−(IY+1, IY) DIR=0: IY←IY+2; DIR=1:IY←IY−2	1	0	1	0	1	1	1	W									See Table 7-9	1	x	x	x	x	x	x
LDM	src-block	When W=0 AL−(IX) DIR=0: IX←IX+1; DIR=1:IX←IX−1 When W=1 AW←(IX+1, IX) DIR=0: IX←IX+2; DIR=1:IX←IX−2	1	0	1	0	1	1	0	W									See Table 7-9	1						
STM	dst-block	When W=0 (IY)←AL DIR=0: IY←IY+1; DIR=1:IY←IY−1 When W=1(IY+1, IY)←AW DIR=0: IY←IY+2; DIR=1:IY←IY−2	1	0	1	0	1	0	1	W									See Table 7-9	1						
Bit Field Transfer Instructions																										
INS	reg8, reg8	16-Bit field←AW	0 1	0 1	0	0 reg	1	1	1 reg	1	0	0	1	1	0	0	0	1	31-117 35-133	3						
	reg8, imm4	16-Bit field←AW	0 1	0 1	0 0	0 0	1 0	1	1 reg	1	0	0	1	1	1	0	0	1	31-117 35-133	4						
EXT	reg8, reg8	AW←16-Bit field	0 1	0 1	0	0 reg	1	1	1 reg	1	0	0	1	1	0	0	1	1	26-55 /34-59	3						
	reg8, imm4	AW←16-Bit field	0 1	0 1	0 0	0 0	1 0	1	1 reg	1	0	0	1	1	1	0	1	1	26-55 /34-59	4						

n: number of transfers

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
I/O Instructions																										
IN	acc, imm8	When W=0 AL←(imm8) When W=1 AH←(imm8+1), AL←(imm8)	1	1	1	0	0	1	0	W									9/13	2						
	acc, DW	When W=0 AL←(DW) When W=1 AH←(DW+1), AL←(DW)	1	1	1	0	1	1	0	W									8/12	1						
OUT	imm8, acc	When W=0 (imm8)←AL When W=1 (imm8+1)←AH, (imm8)←AL	1	1	1	0	0	1	1	W									8/12	2						
	DW, acc	When W=0 (DW)←AL When W=1 (DW+1)←AH, (DW)←AL	1	1	1	0	1	1	1	W									8/12	1						
Primitive I/O Instructions																										
INM	dst-block, DW	When W=0 (IY)←(DW) DIR=0: IY←IY+1; DIR=1:IY←IY−1 When W=1 (IY+1, IY)←(DW+1, DW) DIR=0: IY←IY+2; DIR=1:IY←IR−2	0	1	1	0	1	1	0	W									See Table 7-9	1						
OUTM	DW,src-block	When W=0 (DW)←(IX) DIR=0: IX←IX+1; DIR=1:IX←IX−1 When W=1 (DW+1, DW)←(IX+1, IX) DIR=0: IX←IX+2; DIR=1:IX←IX−2	0	1	1	0	1	1	1	W				n: number of transfers					See Table 7-9	1						
Addition/Subtraction Instructions																										
ADD	reg, reg	reg←reg+reg	0	0	0	0	0	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)+reg	0	0	0	0	0	0	0	W	mod			reg			mem		16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg+(mem)	0	0	0	0	0	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	regm, imm	reg←reg+imm	1	0	0	0	0	0	S	W	1	1	0	0	0		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)+imm	1	0	0	0	0	0	S	W	mod			reg			mem		18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL+imm When W=1 AW←AW+imm	0	0	0	0	0	1	0	W									4	2-3	x	x	x	x	x	x
ADDC	reg, reg	reg←reg+reg+CY	0	0	0	1	0	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)+reg+CY	0	0	0	1	0	0	0	W	mod			reg			mem		16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg+(mem)+CY	0	0	0	1	0	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	reg, imm	reg←reg+imm+CY	1	0	0	0	0	0	S	W	1	1	0	1	0		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)+imm+CY	1	0	0	0	0	0	S	W	mod		0	1	0		mem		18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL+imm+CY When W=1 AW←AW+imm+CY	0	0	0	1	0	1	0	W									4	2-3	x	x	x	x	x	x
SUB	reg, reg	reg←reg−reg	0	0	1	0	1	0	1	W	1	1		reg			reg		2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)−reg	0	0	1	0	1	0	0	W	mod			reg			mem		16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg−(mem)	0	0	1	0	1	0	1	W	mod			reg			mem		11/15	2-4	x	x	x	x	x	x
	reg, imm	reg←reg−imm	1	0	0	0	0	0	S	W	1	1	1	0	1		reg		4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)−imm	1	0	0	0	0	0	S	W	mod		1	0	1		mem		18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL−imm When W=1 AW←AW−imm	0	0	1	0	1	1	0	W									4	2-3	x	x	x	x	x	x

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags					
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC	CY	V	P	S	Z
Addition/Subtraction Instructions (cont)												
SUBC	reg, reg	reg←reg−reg−CY	0 0 0 1 1 0 1 W	1 1 reg reg	2	2	x	x	x	x	x	x
	mem, reg	(mem)←(mem)−reg−CY	0 0 0 1 1 0 0 W	mod reg mem	16/24	2-4	x	x	x	x	x	x
	reg, mem	reg←reg−(mem)−CY	0 0 0 1 1 0 1 W	mod reg mem	11/15	2-4	x	x	x	x	x	
	reg, imm	reg←reg−imm−CY	1 0 0 0 0 0 S W	1 1 0 1 1 reg	4	3-4	x	x	x	x	x	x
	mem, imm	(mem)←(mem)−imm−CY	1 0 0 0 0 0 S W	mod 0 1 1 mem	18/26	3-6	x	x	x	x	x	x
	acc, imm	When W=0 AL←AL−imm−CY When W=1 AW←AW−imm−CY	0 0 0 1 1 1 0 W		4	2-3	x	x	x	x	x	x
BCD Operation Instructions												
ADD4S		dst BCD string←dst BCD string +src BCD string	0 0 0 0 1 1 1 1	0 0 1 0 0 0 0 0	7+19n	2	u	x	u	u	u	x
SUB4S		dst BCD string←dst BCD string −src BCD string	0 0 0 0 1 1 1 1	0 0 1 0 0 0 1 0	7+19n	2	u	x	u	u	u	x
CMP4S		dst BCD string−scr BCD string	0 0 0 0 1 1 1 1 n: number of BCD digits divided by 2	0 0 1 0 0 1 1 0	7+19n	2	u	x	u	u	u	x
ROL4	reg8		0 0 0 0 1 1 1 1 1 1 0 0 0 reg	0 0 1 0 1 0 0 0	13	3						
	mem8		0 0 0 0 1 1 1 1 mod 0 0 mem	0 0 1 0 1 0 0 0	28	3-5						
ROR4	reg8		0 0 0 0 1 1 1 1 1 1 0 0 0 reg	0 0 1 0 1 0 1 0	17	3						
	mem8		0 0 0 0 1 1 1 1 mod 0 0 0 mem	0 0 1 0 1 0 1 0	32	3-5						
Increment/Decrement Instructions												
INC	reg8	reg8←reg8+1	1 1 1 1 1 1 1 0	1 1 0 0 0 reg	2	2	x		x	x	x	x
	mem	(mem)←(mem)+1	1 1 1 1 1 1 1 W	mod 0 0 0 mem	16/24	2-4	x		x	x	x	x
	reg16	reg16←reg16+1	0 1 0 0 0 reg		2	1	x		x	x	x	x
DEC	reg8	reg8←reg8−1	1 1 1 1 1 1 1 0	1 1 0 0 1 reg	2	2	x		x	x	x	x
	mem	(mem)←(mem)−1	1 1 1 1 1 1 1 W	mod 0 0 1 mem	16/24	2-4	x		x	x	x	x
	reg16	reg16←reg16−1	0 1 0 0 1 reg		2	1	x		x	x	x	x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Multiplication Instructions																										
MULU	reg8	AW←AL×reg8 AH=0: CY←0, V←0 AH≠0: CY←1, V←1	1	1	1	1	0	1	1	0	1	1	1	0	0	reg			21-22	2	u	x	x	u	u	u
	mem8	AW←AL×(mem8) AH=0: CY←0, V←0 AH≠0: CY←1, V←1	1	1	1	1	0	1	1	0	mod		1	0	0	mem			27-28	2-4	u	x	x	u	u	u
	reg16	DW, AW←AW×reg16 DW=0: CY←0, V←0 DW≠0: CY←1, V←1	1	1	1	1	0	1	1	1	1	1	1	0	0	reg			29-30	2	u	x	x	u	u	u
	mem16	DW, AW←AW×(mem16) DW=0: CY←0, V←0 DW≠0: CY←1, V←1	1	1	1	1	0	1	1	1	mod		1	0	0	mem			35-36 /39-40	2-4	u	x	x	u	u	u
MUL	reg8	AW←AL×reg8 AH=AL sign expansion: CY←0, V←0 AH≠AL sign expansion: CY←1, V←1	1	1	1	1	0	1	1	0	1	1	1	0	1	reg			33-39	2	u	x	x	u	u	u
	mem8	AW←AL×(mem)8 AH=AL sign expansion: CY←0, V←0 AH≠AL sign expansion: CY←1, V←1	1	1	1	1	0	1	1	0	mod		1	0	1	mem			39-45	2-4	u	x	x	u	u	u
	reg16	DW, AW←AL×(reg)8 DW=AW sign expansion: CY←0, V←0 DW≠AW sign expansion: CY←1, V←1	1	1	1	1	0	1	1	1	1	1	1	0	1	reg			41-47	2	u	x	x	u	u	u
	mem16	DW, AW←AL×(mem)8 DW=AW sign expansion: CY←0, V←0 DW≠AW sign expansion: CY←1, V←1	1	1	1	1	0	1	1	1	mod		1	0	1	mem			47-53 /51-57	2-4	u	x	x	u	u	u
	reg16, (reg16,) imm8	reg16←reg16×imm8 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	1	1	1	1	reg			reg			28-34	3	u	x	x	u	u	u
	reg16, mem16, imm8	reg16←(mem16)×imm8 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	1	1	mod		reg			mem			34-40 /38-44	3-5	u	x	x	u	u	u
	reg16, (reg16,) imm16	reg16←reg16×imm16 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	0	1	1	1	reg			reg			36-42	4	u	x	x	u	u	u
	reg16, mem16, imm16	reg16←(mem16)×imm16 Product≦16 bits: CY←0, V←0 Product>16 bits: CY←1, V←1	0	1	1	0	1	0	0	1	mod		reg			mem			42-48 /46-52	4-6	u	x	x	u	u	u

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Unsigned Division Instructions																										
DIVU	reg8	temp←AW When temp÷reg8≦FFH AH←temp%reg8, AL←temp÷reg8 When temp÷reg8>FFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	1	1	1	1	0	reg			19	2	u	u	u	u	u	u
	mem8	temp←AW When temp÷(mem8)≦FFH AH←temp%(mem8), AL←temp÷(mem8) When temp÷(mem8)>FFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	mod		1	1	0	mem			25	2-4	u	u	u	u	u	u
	reg16	temp←DW, AW When temp÷reg16≦FFFFH DW←temp%reg16, AW←temp÷reg16 When temp÷reg16>FFFFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	1	1	1	1	0	reg			25	2	u	u	u	u	u	u
	mem16	temp←DW, AW When temp÷(mem16)≦FFFFH DW←temp%(mem16), AW←temp÷(mem16) When temp÷(mem16)>FFFFH TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	mod		1	1	0	mem			34	2-4	u	u	u	u	u	u
Signed Division Instructions																										
DIV	reg8	temp←AW When temp÷reg8>0, temp÷reg8≦7FH or temp÷reg8<0, temp÷reg8>0−7FH−1 AH←temp%reg8, AL←temp÷reg8 When temp÷reg8>0, temp÷reg8>7FH or temp÷reg8<0, temp÷reg8≦0−7FH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	1	1	1	1	1	reg			29-34	2	u	u	u	u	u	u

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Signed Division Instructions (cont)																										
DIV	mem8	temp←AW When temp÷(mem8)>0, temp÷(mem8)≦7FH or temp÷(mem8)<0, temp÷(mem8)>0−7FH−1 AH←temp%(mem8), AL←temp÷(mem8) When temp÷(mem8)>0, temp÷(mem8)>7FH or temp÷(mem8)<0, temp÷(mem8)>0−7FH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	0	mod		1	1	1	mem			34-39	2-4	u	u	u	u	u	u
	reg16	temp←DW, AW When temp÷reg16>0, temp÷reg16≦7FFFH or temp÷reg16<0, temp÷reg16>0−7FFFH−1 DW←temp%reg16, AW←temp÷reg16 When temp÷reg16>0, temp÷reg16>7FFFH or temp÷reg16<0, temp÷reg16<0−7FFFH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	1	1	1	1	1	reg			38-43	2	u	u	u	u	u	u
	mem16	temp←DW, AW When temp÷(mem16)>0, temp÷(mem16)≦7FFFH or temp÷(mem16)<0, temp÷(mem16)>0−7FFFH−1 DW←temp%(mem16), AW←temp÷(mem16) When temp÷(mem16)>0, temp÷(mem16)>7FFFH or temp÷(mem16)<0, temp÷(mem16)<0−7FFFH−1 TA←(001H, 000H), TC←(003H, 002H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	1	1	0	1	1	1	mod		1	1	1	mem			43-48 147-52	2-4	u	u	u	u	u	u
BCD Adjust Instructions																										
ADJBA		When (AL AND 0FH)>9 or AC=1, AL←AL+6, AH←AH+1, AC←1, CY←AC, AL←AL AND 0FH	0	0	1	1	0	1	1	1									7	1	x	x	u	u	u	u
ADJ4A		When (AL AND 0FH)>9 or AC=1, AL←AL+6, AC←1, When AL>9FH, or CY=1 AL←AL+60H, CY←1	0	0	1	0	0	1	1	1									3	1	x	x	u	x	x	x
ADJBS		When (AL AND 0FH)>9 or AC=1, AL←AL−6, AH←AH−1, AC←1, CY←AC, AL←AL AND 0FH	0	0	1	1	1	1	1	1									7	1	x	x	u	u	u	u
ADJ4S		When (AL AND 0FH)>9 or AC=1, AL←AL−6, AC←1 When AL>9FH or CY=1 AL←AL−60H, CY←1	0	0	1	0	1	1	1	1									3	1	x	x	u	x	x	x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions																										
ROL	reg, 1	CY←MSB of reg, reg←reg×2+CY MSB of reg≠CY: V←1 MSB of reg=CY:V←0	1	1	0	1	0	0	0	W	1	1	0	0	0	reg			2	2		x	x			
	mem, 1	CY←MSB of (mem), (mem)←(mem)×2+CY MSB of (mem)≠CY:V←1 MSB of (mem)=CY:V←0	1	1	0	1	0	0	0	W	mod		0	0	0	mem			16	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of reg, reg←reg×2+CY temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	0	0	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2+CY temp←temp−1	1	1	0	1	0	0	1	W	mod		0	0	0	reg			19+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB of reg, reg←reg×2+CY temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	0	0	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2+CY temp←temp−1	1	1	0	0	0	0	0	W	mod		0	0	0	mem			19+n	3-5		x	u			
			n: number of shifts																							
ROR	reg, 1	CY←LSB of reg←reg÷2 MSB of reg←CY MSB of reg≠bit following MSB of reg: V←1 MSB of reg=bit following MSB of reg: V←0	1	1	0	1	0	0	0	W	1	1	0	0	1	reg			2	2		x	x			
	mem, 1	CY←LSB of (mem),(mem)←(mem)÷2 MSB of (mem)×CY MSB of (mem)≠bit following MSB of (mem): V←1 MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		0	0	1	mem			16	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, MSB of reg←CY temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	0	1	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, MSB of (mem)←CY temp←temp−1	1	1	0	1	0	0	1	W	mod		0	0	1	mem			19+n	2-4		x	u			
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions (cont)																										
ROR	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, MSB of reg←CY temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	0	1	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	0	0	0	0	W	mod		0	0	1	mem			19+n	3-5		x	u			
			n: number of shifts																							
ROLC	reg, 1	tmpcy←CY, CY←MSB of reg reg←reg×2+tmpcy MSB of reg=CY: V←0 MSB of reg≠CY: V←1	1	1	0	1	0	0	0	W	1	1	0	1	0	reg			2	2		x	x			
	mem, 1	tmpcy←CY, CY←MSB of (mem) (mem)←(mem)×2+tmpcy MSB of (mem)=CY: V←0 MSB of (mem)≠CY: V←1	1	1	0	1	0	0	0	W	mod		0	1	0	mem			16	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, tmpcy←CY, CY←MSB of reg, reg←reg×2+tmpcy temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	1	0	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, tmpcy←CY, CY←MSB of (mem), (mem)←(mem)×2+tmpcy temp←temp−1	1	1	0	1	0	0	1	W	mod		0	1	0	mem			19+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, tmpcy←CY, CY←MSB of reg, reg←reg×2+tmpcy temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	1	0	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, tmpcy←CY, CY←MSB of (mem) (mem)←(mem)×2+tmpcy temp←temp−1	1	1	0	0	0	0	0	W	mod		0	1	0	mem			19+n	3-5		x	u			
			n: number of shifts																							

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Rotation Instructions (cont)																										
RORC	reg, 1	tmpcy←CY, CY←LSB of reg reg←reg÷2, MSB of reg←tmpcy MSB of reg≠bit following MSB of reg: V←1 MSB of reg=bit following MSB of reg: V←0	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			2	2		x	x			
	mem, 1	tmpcy←CY, CY←LSB of (mem) (mem)←(mem)÷2,MSB of (mem)←tmpcy MSB of (mem)≠bit following MSB of (mem): V←1 MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		0	1	1	mem			16	2-4		x	x			
	reg, CL	temp←CL, while temp≠0, repeat this operation, tmpcy←CY, CY←LSB of reg, reg←reg÷2, MSB of reg←tmpcy, temp←temp−1	1	1	0	1	0	0	1	W	1	1	0	1	1	reg			7+n	2		x	u			
	mem, CL	temp←CL, while temp≠0, repeat this operation, tmpcy←CY, CY←LSB of (mem), (mem)←(mem)÷2 MSB of (mem)←tmpcy, temp←temp−1	1	1	0	1	0	0	1	W	mod		0	1	1	mem			19+n	2-4		x	u			
	reg, imm8	temp←imm8, while temp≠0 repeat this operation, tmpcy←CY, CY←LSB of reg, reg←reg÷2 MSB of reg←tmpcy, temp←temp−1	1	1	0	0	0	0	0	W	1	1	0	1	1	reg			7+n	3		x	u			
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, tmpcy←CY, CY←LSB of (mem), (mem)←(mem)÷2 MSB of (mem)←tmpcy, temp←temp−1	1	1	0	0	0	0	0	W	mod		0	1	1	mem			19+n	3-5		x	u			
			n: number of shifts																							
Subroutine Control Instructions																										
CALL	near-proc	(SP−1, SP−2)←PC, SP←SP−2 PC←PC+disp	1	1	1	0	1	0	0	0									16	3						
	regptr16	(SP−1, SP−2)←PC, SP←SP−2 PC←regptr16	1	1	1	1	1	1	1	1	1	1	0	1	0	reg			14	2						
	memptr16	(SP−1, SP−2)←PC, SP←SP−2 PC←(memptr16)	1	1	1	1	1	1	1	1	mod		0	1	0	mem			23	2-4						
	far-proc	(SP−1, SP−2)←PS, (SP−3, SP−4)←PC SP←SP−4, PS←seg, PC←offset	1	0	0	1	1	0	1	0									21	5						
	memptr32	(SP−1, SP−2)←PS, (SP−3, SP−4)←PC SP←SP−4, PS←(memptr32+2), PC←(memptr32)	1	1	1	1	1	1	1	1	mod		0	1	1	mem			31	2-4						

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags					
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC	CY	V	P	S	Z
Subroutine Control Instructions (cont)												
RET		PC←(SP+1, SP), SP←SP+2	1 1 0 0 0 0 1 1		15	1						
	pop-value	PC←(SP+1, SP), SP←SP+2, SP+pop-value	1 1 0 0 0 0 1 0		20	3						
		PC←(SP+1, SP), PS←(SP+3, SP+2) SP←SP+4	1 1 0 0 1 0 1 1		21	1						
	pop-value	PC←(SP+1, SP), PS←(SP+3, SP+2) SP←SP+4, SP←SP+pop-value	1 1 0 0 1 0 1 0		24	3						
Stack Manipulation Instructions												
PUSH	mem16	SP←SP−2 (SP+1, SP)←(mem16)	1 1 1 1 1 1 1 1	mod 1 1 0 mem	18	2-4						
	reg16	SP←SP−2 (SP+1, SP)←reg16	0 1 0 1 0 reg		8	1						
	sreg	SP←SP−2 (SP+1, SP)←sreg	0 0 0 sreg 1 1 0		8	1						
	PSW	SP←SP−2 (SP+1, SP)←PSW	1 0 0 1 1 1 0 0		8	1						
	R	Push registers on the stack	0 1 1 0 0 0 0 0		35	1						
	imm	(SP−1, SP−2)←imm SP←SP−2, When S=1, sign extension	0 1 1 0 1 0 S 0		7 or 8	2-3						
POP	mem16	(mem16)←(SP+1, SP), SP←SP+2	1 0 0 0 1 1 1 1	mod 0 0 0 mem	17	2-4						
	reg16	reg16←(SP+1, SP), SP←SP+2	0 1 0 1 1 reg		8	1						
	sreg	sreg←(SP+1, SP) sreg: SS, DS0, DS1 SP←SP+2	0 0 0 sreg 1 1 1		8	1						
	PSW	PSW←(SP+1, SP),SP←SP+2	1 0 0 1 1 1 0 1		8	1	R	R	R	R	R	R
	R	Pop registers from the stack	0 1 1 0 0 0 0 1		43	1						
PREPARE	imm16, imm8	Prepare new stack frame	1 1 0 0 1 0 0 0 ★: imm8<0: 16 imm8<1: 23+16 (imm8−1)		★	4						
DISPOSE		Dispose of stack frame	1 1 0 0 1 0 0 1		6	1						
Branch Instruction												
BR	near-label	PC←PC+disp	1 1 1 0 1 0 0 1		13	3						
	short-label	PC←PC+ext-disp8	1 1 1 0 1 0 1 1		12	2						
	regptr16	PC←regptr16	1 1 1 1 1 1 1 1	1 1 1 0 0 reg	11	2						
	memptr16	PC←(memptr16)	1 1 1 1 1 1 1 1	mod 1 0 0 mem	20	2-4						
	far-label	PS←seg, PC←offset	1 1 1 0 1 0 1 0		15	5						
	memptr32	PS←(memptr32+2), PS←(memptr32)	1 1 1 1 1 1 1 1	mod 1 0 1 mem	27	2-4						
Conditional Branch Instructions												
BV	short-label	if V=1, PC←PC+ext-disp8	0 1 1 1 0 0 0 0		14/4	2						
BNV	short-label	if V=0, PC←PC+ext-disp8	0 1 1 1 0 0 0 1		14/4	2						
BC, BL	short-label	if CY=1, PC←PC+ext-disp8	0 1 1 1 0 0 1 0		14/4	2						
BNC,BNL	short-label	if CY=0, PC←PC+ext-disp8	0 1 1 1 0 0 1 1		14/4	2						
BE, BZ	short-label	if Z=1, PC←PC+ext-disp8	0 1 1 1 0 1 0 0		14/4	2						
BNE,BNZ	short-label	if Z=0, PC←PC+ext-disp8	0 1 1 1 0 1 0 1		14/4	2						
BNH	short-label	if CY OR Z=1, PC←PC+ext-disp8	0 1 1 1 0 1 1 0		14/4	2						
BH	short-label	if CY OR Z=0, PC←PC+ext-disp8	0 1 1 1 0 1 1 1		14/4	2						
BN	short-label	if S=1, PC←PC+ext-disp8	0 1 1 1 1 0 0 0		14/4	2						
BP	short-label	if S=0, PC←PC+ext-disp8	0 1 1 1 1 0 0 1		14/4	2						
BPE	short-label	if P=1, PC←PC+ext-disp8	0 1 1 1 1 0 1 0		14/4	2						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Conditional Branch Instructions (cont)																										
BPO	short-label	if P=0, PC←PC+ext-disp8	0	1	1	1	1	0	1	1									14/4	2						
BLT	short-label	if S XOR V=1, PC←PC+ext-disp8	0	1	1	1	1	1	0	0									14/4	2						
BGE	short-label	if S XOR V=0, PC←PC+ext-disp8	0	1	1	1	1	1	0	1									14/4	2						
BLE	short-label	if (S XOR V) OR Z=1, PC←PC+ext-disp8	0	1	1	1	1	1	1	0									14/4	2						
BGT	short-label	if (S XOR V) OR Z=0, PC←PC+ext-disp8	0	1	1	1	1	1	1	1									14/4	2						
DBNZNE	short-label	CW←CW−1 if Z=0 and CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	0	0									14/5	2						
DBNZE	short-label	CW←CW−1 if Z=1 and CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	0	1									14/5	2						
DBNZ	short-label	CW←CW−1 if CW≠0, PC←PC+ext-disp8	1	1	1	0	0	0	1	0									13/5	2						
BCWZ	short-label	if CW=0, PC←PC+ext-disp8	1	1	1	0	0	0	1	1									13/5	2						
Interrupt Instructions																										
BRK	3	TA←(00DH, 00CH), TA←(00FH, 00EH) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	0	0									38	1						
	imm8 (≠3)	TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=Imm8 SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	0	1									38	2						
BRKV		When V=1 TA←(011H, 010H), TC←(013H, 012H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1	1	0	0	1	1	1	0									40/3	1						
RETI		PC←(SP+1, SP), PS←(SP+3, SP+2), PSW←(SP+5, SP+4), SP←SP+6	1	1	0	0	1	1	1	1									27	1	R	R	R	R	R	R
CHKIND	reg16, mem32	When (mem32)>reg16 or (mem32+2)<reg16 TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=imm8 SP←SP−2, (SP+1, SP)←PSW, MD←0 MD Bit Write Enable SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	0	1	1	0	0	0	1	0	mod			reg			mem		53-56/ 18	2-4						
BRKEM	imm8	TA←(015H, 014H), TC←(017H, 016H) SP←SP−2, (SP+1, SP)←PSW, IE←0, BRK←0 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	38	3						

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC CY V P S Z
CPU Control Instructions							
HALT		CPU Halt	1 1 1 1 0 1 0 0		2	1	
BUS-LOCK		Bus Lock Prefix	1 1 1 1 0 0 0 0		2	1	
FP01	fp-op	No Operation	1 1 0 1 1 X X X	1 1 Y Y Y Z Z Z	2	2	
	fp-op, mem	data bus←(mem)	1 1 0 1 1 X X X	mod Y Y Y mem	11	2-4	
FPO2	fp-op	No Operation	0 1 1 0 0 1 1 X	1 1 Y Y Y Z Z Z	2	2	
	fp-op, mem	data bus←(mem)	0 1 1 0 0 1 1 X	mod Y Y Y mem	11	2-4	
POLL		Poll and wait	1 0 0 1 1 0 1 1 n: number of times POLL pin is sampled		2+5n	1	
NOP		No Operation	1 0 0 1 0 0 0 0		3	1	
DI		IE←0	1 1 1 1 1 0 1 0		2	1	
EI		IE←1	1 1 1 1 1 0 1 1		2	1	
8080 Mode Instructions							
RETEM		PC←(SP+1, SP), PS←(SP+3, SP+2), PSW←(SP+5, SP+4), SP←SP+6, MD Bit Write Disable	1 1 1 0 1 1 0 1	1 1 1 1 1 1 0 1	27	2	R R R R R R
CALLN	imm8	TA←(4n+1, 4n), TC←(4n+3, 4n+2) n=imm8 SP←SP−2, (SP+1, SP)←PSW, MD←1 SP←SP−2, (SP+1, SP)←PS, PS←TC SP←SP−2, (SP+1, SP)←PC, PC←TA	1 1 1 0 1 1 0 1	1 1 1 0 1 1 0 1	38	3	
Data Conversion Instructions							
CVTBD		AH←AL=0AH, AL←AL%0AH	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0	15	2	u u u x x x
CVTDB		AH←0, AL←AH×0AH+AL	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0	7	2	u u u x x x
CVTBW		When AL<80H, AH←0, all other times AH←FFH	1 0 0 1 1 0 0 0		2	1	
CVTWL		When AL<8000H, DW←0, all other times DW←FFFFH	1 0 0 1 1 0 0 1		4-5	1	
Comparaision Instructions							
CMP	reg, reg	reg−reg	0 0 1 1 1 0 1 W	1 1 reg reg	2	2	x x x x x x
	mem, reg	(mem)−reg	0 0 1 1 1 0 0 W	mod reg mem	11/15	2-4	x x x x x x
	reg, mem	reg−(mem)	0 0 1 1 1 0 1 W	mod reg mem	11/15	2-4	x x x x x x
	reg, imm	reg−imm	1 0 0 0 0 0 S W	1 1 1 1 1 reg	4	3-4	x x x x x x
	mem, imm	(mem)−imm	1 0 0 0 0 0 S W	mod 1 1 1 mem	13/17	3-6	x x x x x x
	acc, imm	When W=0, AL−imm When W=1, AW−imm	0 0 1 1 1 1 0 W		4	2-3	x x x x x x
Complement Instructions							
NOT	reg	reg←$\overline{\text{reg}}$	1 1 1 1 0 1 1 W	1 1 0 1 0 reg	2	2	
	mem	(mem)←$\overline{\text{(mem)}}$	1 1 1 1 0 1 1 W	mod 0 1 0 mem	16	2-4	
NEG	reg	reg←$\overline{\text{reg}}$+1	1 1 1 1 0 1 1 W	1 1 0 1 1 reg	2	2	x x x x x x
	mem	(mem)←$\overline{\text{(mem)}}$+1	1 1 1 1 0 1 1 W	mod 0 1 1 mem	16/24	2-4	x x x x x x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Logical Operation Instructions																										
TEST	reg, reg	reg AND reg	1	0	0	0	0	1	0	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg or reg, mem	(mem) AND reg	1	0	0	0	0	1	0	W	mod			reg			mem		10/14	2-4	u	0	0	x	x	x
	reg, imm	reg AND imm	1	1	1	1	0	1	1	W	1	1	0	0	0		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem) AND imm	1	1	1	1	0	1	1	W	mod		0	0	0		mem		11/15	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL AND imm8 When W=1, AW AND imm8	1	0	1	0	1	0	0	W									4	2-3	u	0	0	x	x	x
AND	reg, reg	reg←reg AND reg	0	0	1	0	0	0	1	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) AND reg	0	0	1	0	0	0	0	W	mod			reg			mem		16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg AND (mem)	0	0	1	0	0	0	1	W	mod			reg			mem		11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg AND imm	1	0	0	0	0	0	0	W	1	1	1	0	0		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) AND imm	1	0	0	0	0	0	0	W	mod		1	0	0		mem		18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL AND imm8 When W=1, AW←AW AND imm16	0	0	1	0	0	1	0	W									4	2-3	u	0	0	x	x	x
Logical Operation Instructions (cont)																										
OR	reg, reg	reg←reg OR reg	0	0	0	0	1	0	1	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) OR reg	0	0	0	0	1	0	0	W	mod			reg			mem		16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg OR (mem)	0	0	0	0	1	0	1	W	mod			reg			mem		11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg OR imm	1	0	0	0	0	0	0	W	1	1	0	0	1		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) OR imm	1	0	0	0	0	0	0	W	mod		0	0	1		mem		18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL←AL OR imm8 When W=1, AW←AW OR imm16	0	0	0	0	1	1	1	W									4	2-3	u	0	0	x	x	x
XOR	reg, reg	reg←reg XOR reg	0	0	1	1	0	0	1	W	1	1		reg			reg		2	2	u	0	0	x	x	x
	mem, reg	(mem)←(mem) XOR reg	0	0	1	1	0	0	0	W	mod			reg			mem		16/24	2-4	u	0	0	x	x	x
	reg, mem	reg←reg XOR (mem)	0	0	1	1	0	0	1	W	mod			reg			mem		11/15	2-4	u	0	0	x	x	x
	reg, imm	reg←reg XOR imm	1	0	0	0	0	0	0	W	1	1	1	1	0		reg		4	3-4	u	0	0	x	x	x
	mem, imm	(mem)←(mem) XOR imm	1	0	0	0	0	0	0	W	mod		1	1	0		mem		18/26	3-6	u	0	0	x	x	x
	acc, imm	When W=0, AL←AL XOR imm8 When W=1, AW←AW XOR imm16	0	0	1	1	0	1	0	W									4	2-3	u	0	0	x	x	x
Bit Operation Instructions																										
			2nd byte★								3rd byte★															
TEST1	reg8, CL	reg8 bit no, CL=0: Z←1 reg8 bit no, CL=1: Z←0	0	0	0	1	0	0	0	0	1	1	0	0	0		reg		3	3	u	0	0	u	u	x
	mem8, CL	(mem8) bit no. CL=0: Z←1 (mem8) bit no. CL=1: Z←0	0	0	0	1	0	0	0	1	mod		0	0	0		mem		8	3-5	u	0	0	u	u	x
	reg16, CL	reg16 bit no. CL=0: Z←1 reg16 bit no. CL=1: Z←0	0	0	0	1	0	0	0	1	1	1	0	0	0		reg		3	3	u	0	0	u	u	x
	mem16, CL	(mem16) bit no. CL=0: Z←1 (mem16) bit no. CL=1: Z←0	0	0	0	1	0	0	0	0	mod		0	0	0		mem		8/12	3-5	u	0	0	u	u	x
	reg8, imm3	reg8 bit no. imm3=0: Z←1 reg8 bit no. imm3=1: Z←0	0	0	0	1	1	0	0	0	1	1	0	0	0		reg		4	4	u	0	0	u	u	x

Mnemonic	Operand	Operation	Operation Code		No. of Clocks	No. of Bytes	Flags					
			7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			AC	CY	V	P	S	Z
Bit Operation Instructions(cont)												
TEST1	mem8, imm3	(mem8) bit no. imm3=0: Z←1 (mem8) bit no. imm3=1: Z←0	0 0 0 1 1 0 0 0	mod 0 0 0 mem	9	4-6	u	0	0	u	u	x
	reg16, imm4	reg16 bit no. imm4=0: Z←1 reg16 bit no. imm4=1: Z←0	0 0 0 1 1 0 0 1	1 1 0 0 0 reg	4	4	u	0	0	u	u	x
	mem16, imm4	(mem16) bit no. imm4=0: Z←1 (mem16) bit no. imm4=1: Z←0	0 0 0 1 1 0 0 1 2nd byte★ ★Note: First byte=0FH	mod 0 0 0 mem 3rd byte★	9/13	4-6	u	0	0	u	u	x
NOT1			2nd byte★	3rd byte★								
	reg8, CL	reg8 bit no. CL←$\overline{\text{reg8 bit no. CL}}$	0 0 0 1 0 1 1 0	1 1 0 0 0 reg	4	3						
	mem8, CL	(mem8) bit no. CL←$\overline{\text{(mem8) bit no. CL}}$	0 0 0 1 0 1 1 0	mod 0 0 0 mem	13	3-5						
	reg16, CL	reg16 bit no. CL←$\overline{\text{reg16 bit no. CL}}$	0 0 0 1 0 1 1 1	1 1 0 0 0 reg	4	3						
	mem16, CL	(mem16) bit no. LC←$\overline{\text{(mem16) bit no. CL}}$	0 0 0 1 0 1 1 1	mod 0 0 0 mem	13/21	3-5						
	reg8, imm3	reg8 bit no. imm3←$\overline{\text{reg8 bit no. imm3}}$	0 0 0 1 1 1 1 0	1 1 0 0 0 reg	5	4						
	mem8, imm3	(mem8) bit no. imm3←$\overline{\text{(mem8)bit no. imm3}}$	0 0 0 1 1 1 1 0	mod 0 0 0 mem	14	4-6						
	reg16, imm4	reg16 bit no. imm4←$\overline{\text{(reg16) bit no.imm4}}$	0 0 0 1 1 1 1 1	1 1 0 0 0 reg	5	4						
	mem16, imm4	(mem16) bit no. imm4←$\overline{\text{(mem16) bit no. imm4}}$	0 0 0 1 1 1 1 1 2nd byte★ ★Note: First byte=0FH	mod 0 0 0 mem 3rd byte★	14/22	4-6						
	CY	CY←$\overline{\text{CY}}$	1 1 1 1 0 1 0 1		2	1		x				
CLR1			2nd byte★	3rd byte★								
	reg8, CL	reg8 bit no. CL←0	0 0 0 1 0 0 1 0	1 1 0 0 0 reg	5	3						
	mem8, CL	(mem8) bit no. CL←0	0 0 0 1 0 0 1 0	mod 0 0 0 mem	14	3-5						
	reg16, CL	reg16 bit no. CL←0	0 0 0 1 0 0 1 1	1 1 0 0 0 reg	5	3						
	mem16, CL	(mem16) bit no. CL←0	0 0 0 1 0 0 1 1	mod 0 0 0 mem	14/22	3-5						
	reg8, imm3	reg8 bit no. imm3←0	0 0 0 1 1 0 1 0	1 1 0 0 0 reg	6	4						
	mem8, imm3	(mem8) bit no. imm3←0	0 0 0 1 1 0 1 0	mod 0 0 0 mem	15	4-6						
	reg16, imm4	reg16 bit no. imm4←0	0 0 0 1 1 0 1 1	1 1 0 0 0 reg	6	4						
	mem16, imm4	(mem16) bit no. imm4←0	0 0 0 1 1 0 1 1 2nd byte★ ★Note: First byte=0FH	mod 0 0 0 mem 3rd byte★	15/23	4-6						
	CY	CY←0	1 1 1 1 1 0 0 0		2	1		0				
	DIR	DIR←0	1 1 1 1 1 1 0 0		2	1						

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Bit Operation Instructions (cont)																										
SET1	reg8, CL	reg8 bit no. CL←1	0	0	0	1	0	1	0	0	1	1	0	0	0	reg			4	3						
	mem8, CL	(mem8) bit no. CL←1	0	0	0	1	0	1	0	0	mod		0	0	0	mem			13	3-5						
	reg16, CL	reg16 bit no. CL←1	0	0	0	1	0	1	0	1	1	1	0	0	0	reg			4	3						
	mem16, CL	(mem16) bit no. CL←1	0	0	0	1	0	1	0	1	mod		0	0	0	mem			13/21	3-5						
	reg8, imm3	reg8 bit no. imm3←1	0	0	0	1	1	1	0	0	1	1	0	0	0	reg			5	4						
	mem8, imm3	(mem8) bit no. imm3←1	0	0	0	1	1	1	0	0	mod		0	0	0	mem			14	4-6						
	reg16, imm4	reg16 bit no. imm4←1	0	0	0	1	1	1	0	1	1	1	0	0	0	reg			5	4						
	mem16, imm4	(mem16) bit no. imm4←1	0	0	0	1	1	1	0	1	mod		0	0	0	mem			14	4-6						
			2nd byte★								3rd byte★															
			★Note: First byte=0FH																							
	CY	CY←1	1	1	1	1	1	0	0	1									2	1		1				
	DIR	DIR←1	1	1	1	1	1	1	0	1									2	1						
Shift Instructions																										
SHL	reg, 1	CY←MSB or reg, reg←×2 When MSB of reg≠CY, V←1 When MSB of reg=CY, V←0	1	1	0	1	0	0	0	W	1	1	1	0	0	reg			2	2	u	x	x	x	x	x
	mem, 1	CY←MSB or (mem),(mem)←(mem)×2 When MSB of (mem)≠CY, V←1 When MSB of (mem)=CY, V←0	1	1	0	1	0	0	0	W	mod		1	0	0	mem			16/24	2-4	u	x	x	x	x	x
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB or reg, reg←reg×2, temp←temp−1	1	1	0	1	0	0	1	W	1	1	1	0	0	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←MSB of (mem), (mem)←(mem)×2, temp←temp−1	1	1	0	1	0	0	1	W	mod		1	0	0	mem			19/27+n	2-4	u	x	u	x	x	x
	reg, imm8	temp←imm8, While temp≠0, repeat this operation, CY←MSB of reg, reg←reg×2, temp←temp−1	1	1	0	0	0	0	0	W	1	1	1	0	0	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←MSB or (mem), (mem)←(mem)×2, temp←temp−1	1	1	0	0	0	0	0	W	mod		1	0	0	mem			19/27+n	3-5	u	x	u	x	x	x
			n: number of shifts																							
SHR	reg, 1	CY←LSB of reg, ←reg÷2 When MSB of reg≠bit following MSB of reg: V←1 When MSB of reg=bit following MSB of reg: V←0	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			2	2	u	x	x	x	x	x

Mnemonic	Operand	Operation	Operation Code																No. of Clocks	No. of Bytes	Flags					
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			AC	CY	V	P	S	Z
Shift Instructions (cont)																										
SHR	mem, 1	CY←LSB of (mem), (mem)←(mem)÷2 When MSB of (mem)≠bit following MSB of (mem): V←1 When MSB of (mem)=bit following MSB of (mem): V←0	1	1	0	1	0	0	0	W	mod		1	0	1	mem			16/24	2-4	u	x	x	x	x	x
	reg, CL	temp←CL, while time≠0, repeat this operation, CY←LSB or reg, reg←reg÷2, temp←temp−1	1	1	0	1	0	0	0	W	1	1	1	0	1	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB or (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	1	0	0	1	W	mod		1	0	1	mem			19/27+n	2-4	u	x	u	x	x	x
	reg,imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, temp←temp−1	1	1	0	0	0	0	0	W	1	1	1	0	1	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1	1	1	0	0	0	0	0	W	mod		1	0	1	mem			19/27+n	3-5	u	x	u	x	x	x
			n: number of shifts																							
SHRA	reg, 1	CY←LSB of reg, reg←reg÷2, V←0 MSB of operand does not change	1	1	0	1	0	0	0	W	1	1	1	1	1	reg			2	2	u	x	0	x	x	x
	mem, 1	CY←LSB of (mem), (mem)←(mem)÷2, V←0, MSB of operand does not change	1	1	0	1	0	0	0	W	mod		1	1	1	mem			16	2-4	u	x	0	x	x	x
	reg, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, temp←temp−1 MSB of operand does not change	1	1	0	1	0	0	1	W	1	1	1	1	1	reg			7+n	2	u	x	u	x	x	x
	mem, CL	temp←CL, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1 MSB of operand does not change	1	1	0	1	0	0	1	W	mod		1	1	1	mem			19+n	2-4	u	x	u	x	x	x
	reg, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of reg, reg←reg÷2, temp←temp−1 MSB of operand does not change	1	1	0	0	0	0	0	W	1	1	1	1	1	reg			7+n	3	u	x	u	x	x	x
	mem, imm8	temp←imm8, while temp≠0, repeat this operation, CY←LSB of (mem), (mem)←(mem)÷2, temp←temp−1 MSB of operand does not change	1	1	0	0	0	0	0	W	mod		1	1	1	mem			19+n	3-5	u	x	u	x	x	x
			n: number of shifts																							

SHARP

SHARP CORPORATION Japan
IC SALES DEPARTMENT
INTERNATIONAL SALES & MARKETING GROUP
–IC/ELECTRONIC COMPONENTS
2613-1 ICHINOMOTO-CHO TENRI-CITY NARA 632, JAPAN
PHONE: (07436) 5-1321
TELEX: LABOMETA-B J63428
FACSIMILE: (07436) 5-1532

NORTH AMERICA: SHARP ELECTRONICS CORPORATION
Microelectronics Division
Sharp Plaza, Mahwah, New Jersey 07430-2135, U.S.A.
PHONE: (201) 529-8757
TELEX: 134327 (SHARPAM MAWA)/426903 (SHARPAM MAWA)
FACSIMILE: (201) 529-8759

EUROPE: SHARP ELECTRONICS (EUROPE) GmbH
Microelectronics Division
Sonninstraße 3, 2000 Hamburg 1, F.R. Germany
PHONE: (40) 23-775-216
TELEX: 2161867 (HEEG D)
FACSIMILE: (40) 23-775-232

HONG KONG: SHARP-ROXY (HONG KONG) LTD.
1701-1710, Admiralty Centre, Tower 1,
Harcourt Road, Hong Kong
PHONE: (5) 8229379
TELEX: 61950 (SRHL HX)
FACSIMILE: (5) 8660779

SINGAPORE: SHARP-ROXY SALES (SINGAPORE) PTE. LTD.
100G Pasir Panjang Road, Singapore 0511
PHONE: 4731911
TELEX: 55504 (SRSSIN RS)
FACSIMILE: 4794105

KOREA: SHARP ELECTRONICS INDUSTRIAL CORPORATION
4F. Dae San Bldg. 14-27 Yeoyido-dong,
Young Deung Po-ku, Seoul, Korea
PHONE: 782-8637~40
TELEX: SHARPEI K26754
FACSIMILE: (02) 782-5070

© SHARP CORP. Oct. '90 S8EI Printed in Japan

MICROCOMPUTERS DATA BOOK

SHARP

NORTH AMERICA: SHARP ELECTRONICS CORPORATION
Microelectronics Group
5700 Northwest, Pacific Rim Boulevard – Suite 20
Camas, Washington 98607, U.S.A.
PHONE: (206) 834-2500
TELEX: 49608472 SHARPCAM
FAX: (206) 834-8903

EUROPE: SHARP ELECTRONICS (EUROPE) GmbH
Microelectronics Division
Sonninstraße 3, 2000 Hamburg 1, F.R. Germany
PHONE: (040) 23775-286
TELEX: 2161867 (HEEG D)
FAX: (40) 231480

HONG KONG: SHARP-ROXY (HONG KONG) LTD.
3rd Business Division,
Room 1701-1710, Admiralty Centre, Tower 1,
Harcourt Road, Hong Kong
PHONE: 8229311/8229348
TELEX: 74258 SRHL HX
FAX: 5297561/8660779

SINGAPORE: SHARP-ROXY SALES (SINGAPORE) PTE. LTD.
100G Pasir Panjang Road, Singapore 0511
PHONE: 4731911
TELEX: 55504 (SRSSIN RS)
FAX: 4794105

KOREA: SHARP ELECTRONICS INDUSTRIAL CORPORATION
4th Floor Daesan Bldg., 14-27, Yeo Yido-dong,
Youngdungpo-gu, Seoul 150-010, Korea
PHONE: (782) 8637-40
TELEX: SHARPEI K26754
FAX: (782) 5070

REF. NO. WT243C