NEC

Application Note

μPD784915, 784928, 784928Y Subseries

16-bit Single-chip Microcontrollers

VCR Servo Basics

μPD784915	μPD784927	μPD784927Y
μPD784915A	μPD78F4928	μPD78F4928Y
μPD784916A		
μPD784915B		
μPD784916B		
μPD78P4916		

Document No. U11361EJ3V0AN00 (3rd edition)
Date Published March 1998 N CP(K)

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

The information in this document is subject to change without notice.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated “quality assurance program“ for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.

M7 96.5

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

J98. 2

Major Revisions in This Edition

Page	Description
Throughout	The μPD784928, 784928Y Subseries and the μPD784915B, 784916B are added.
Introduction	Document numbers of related documents are added or corrected.
p. 15	**CHAPTER 1 OUTLINE OF NEC VCR SERVO MICROCONTROLLER PRODUCTS** is added.
p. 19	**Table 2-1 Differences among μPD784915 Subseries Products** is added.
p. 25	**CHAPTER 3 OUTLINE OF μPD784928, 784928Y SUBSERIES** is added.

The mark ★ shows major revised points.

INTRODUCTION

Readers

This application note is intended for user engineers who understand the functions of the μPD784915, 784928, 784928Y Subseries and wish to design and develop its application systems and programs.

Purpose

The purpose of this application note is to help users understand the hardware capabilities of the target device using application examples.

Organization

The main topics of this application note are listed below.

- Outline of μPD784915 Subseries
- Outline of μPD784928, 784928Y Subseries
- Outline of VCR servo
- Servo control examples of stationary VCR
- Analog circuit
- VISS

How to Read This Manual

It is assumed that the readers of this manual have a general knowledge of electronics, logical circuits, and microcontrollers. Moreover, readers should also have a general knowledge of VCRs and servo control.

When there are no functional differences in the products, the application note mentions the μPD784915 Subseries as the representative subseries and the μPD784915 as the representative version, although its descriptions also apply to the versions other than the μPD784915.

Quality Grade

Standard (for general electronic appliances)

Legends

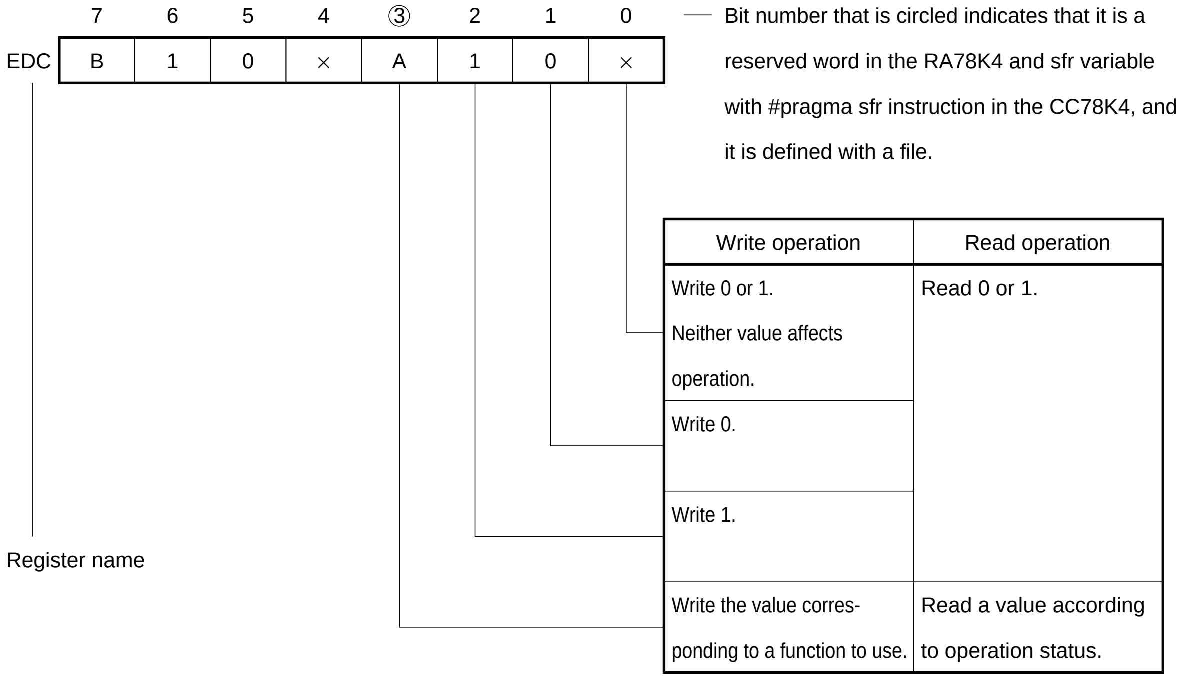
Data significance	: Left: higher digit, right: lower digit
Active low	: $\overline{\times\times\times}$ (top bar over pin or signal name)
Note	: Footnote explaining items marked with “Note” in the text
Caution	: Description of point that requires particular attention
Remark	: Supplementary information
Numerical representation	: Binary ... ××××B or ×××× Decimal ... ×××× Hexadecimal ... ××××H
Easily confused characters	: 0 (zero), O (uppercase letter “o”) 1 (one), l (lowercase of letter “L”), I (uppercase of letter “i”)

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

★ **Device related documents**

Document Name	Document Number	
	Japanese	English
μPD784915 Data Sheet	U11044J	U11044E
μPD784915A, 784916A Data Sheet	U11022J	U11022E
μPD784915B, 784916B Data Sheet	U11930J	To be prepared
μPD78P4916 Data Sheet	U11045J	U11045E
μPD784915 Subseries Special Function Register Table	U10976J	—
μPD784915 Subseries User’s Manual	U10444J	U10444E
μPD784927 Data Sheet	U12255J	To be prepared
μPD78F4928 Preliminary Product Information	U12188J	U12188E
μPD784928 Subseries Special Function Register Table	U12798J	—
μPD784927Y Data Sheet	U12373J	U12373E
μPD78F4928Y Preliminary Product Information	U12271J	U12271E
μPD784928Y Subseries Special Function Register Table	U12719J	—
μPD784928, 784928Y Subseries User’s Manual	U12648J	U12648E
μPD784915, 784928, 784928Y Subseries Application Note — VCR Servo Basics	U11361J	This manual
78K/IV Series User’s Manual — Instruction	U10905J	U10905E
78K/IV Series Instruction Table	U10594J	—
78K/IV Series Instruction Set	U10595J	—
78K/IV Series Application Note — Software Basics	U10095J	U10095E

Register Format



Bit number that is circled indicates that it is a reserved word in the RA78K4 and sfr variable with #pragma sfr instruction in the CC78K4, and it is defined with a file.

Write operation	Read operation
Write 0 or 1. Neither value affects operation.	Read 0 or 1.
Write 0.	
Write 1.	
Write the value corres-ponding to a function to use.	Read a value according to operation status.

Never write a combination of codes marked “setting prohibited” in the register formats in the text.

CONTENTS

LIST OF FIGURES (1/2)

LIST OF FIGURES (2/2)

LIST OF TABLES

[MEMO]

★

CHAPTER 1 OUTLINE OF NEC VCR SERVO MICROCONTROLLER PRODUCTS

1.1 Outline

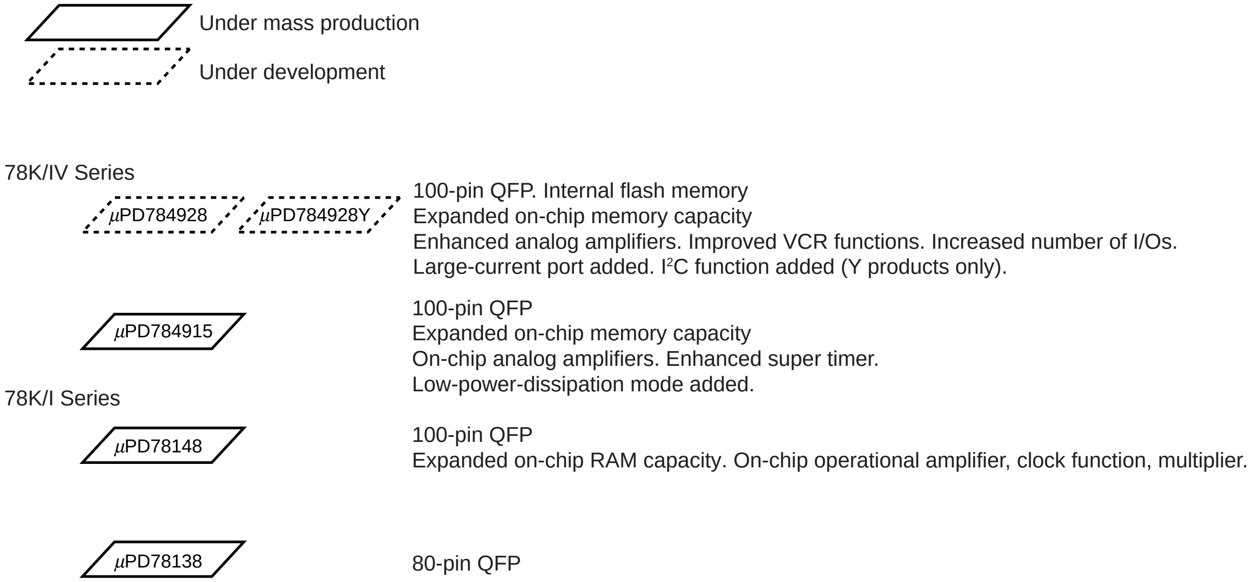
NEC's microcontrollers for VCR servos are 78K/IV Series products featuring a high-speed, high-performance 16-bit CPU that are improved versions of the 78K/I Series of 8-bit single-chip microcontrollers for VCR software servo control.

Microcontrollers for VCR servo control comprise the following three subseries.

- μPD784915 Subseries
- μPD784928 Subseries
- μPD784928Y Subseries

NEC's lineup of microcontrollers for VCR servo control is shown below.

The Y subseries support I^{2}C bus specifications.



100-pin QFP. Internal flash memory
Expanded on-chip memory capacity
Enhanced analog amplifiers. Improved VCR functions. Increased number of I/Os.
Large-current port added. I^{2}C function added (Y products only).

100-pin QFP
Expanded on-chip memory capacity
On-chip analog amplifiers. Enhanced super timer.
Low-power-dissipation mode added.

100-pin QFP
Expanded on-chip RAM capacity. On-chip operational amplifier, clock function, multiplier.

80-pin QFP

● Microcontrollers for VCR Servo Control

• μPD784915 Subseries

Part Number / Parameter	μPD784915, 784915A, μPD784915B	μPD784916A, μPD784916B	μPD78P4916
Internal ROM capacity	Mask ROM		One-time PROM
	48 Kbytes	62 Kbytes	
Internal RAM capacity	1280 bytes		2048 bytes

• μPD784928, 784928Y Subseries

Part Number / Parameter	μPD784927, μPD784927Y	μPD78F4928[Note], μPD78F4928Y[Note]
Internal ROM capacity	Mask ROM	Flash memory
	96 Kbytes	128 Kbytes
Internal RAM capacity	2048 bytes	3584 bytes

Note Under development

1.2 Features

In this section, the μPD784915 Subseries is explained as the representative subseries, which is enhanced, compared with the 78K/I Series, in the points mentioned below.

(1) Equipped with the 78K/IV core, a 16-bit high-performance CPU

The instruction set of the μPD784915 Subseries is perfectly upward-compatible with that of the existing 78K/I series. Therefore, the software assets of the 78K/I Series are effectively utilized.

The 78K/IV Series supports 1-Mbyte linear address space, resulting in improved program handlability. Moreover, the instruction set of the 78K/IV Series has been greatly enhanced, and realizes high-speed servo arithmetic processing by using powerful multiplication and 16-bit transmit instructions.

(2) Enhanced power management function

The μPD784915 Subseries realizes internal 8 MHz (minimum instruction execution time = 250 ns) high-speed operation in 4.5 to 5.5 V voltage range in normal operation. Its CPU guarantees 4.0-V operation.

Moreover, the μPD784915 Subseries is equipped with a low power consumption mode which enables CPU operation using 32.768-kHz subsystem clock. Selection of CPU clock dividing ratio is made possible by on-chip clock frequency dividing circuit. Since operation up to 2.7 V is guaranteed, reduction of the power consumption of the whole system is possible using these functions. The use of these functions in combination with the standby function realizes ultra low power consumption according to the operation conditions, that is, back-up supply voltage operation or battery operation.

(3) Realizes low-frequency/high-speed operation for reducing radiation noise

The μPD784915 Subseries provides a low-frequency oscillation mode which enables internal operation with the clock frequency equal to the external oscillation frequency. It realizes reduction of radiation noise by enabling high-speed operation with a frequency lower than that of conventional products.

(4) On chip VCR servo control timer “Super Timer Unit”

The super timer unit consists of six 16-bit timers, two 8-bit timers, and a 5-bit up/down counter for linear tape in addition to 22-bit free running counter (FRC) to carry out cycle measurement of various VCR motors. Therefore, VCR servo control by software can be performed easily.

The μPD784915 is incorporated with special circuits such as V_{SYNC} and H_{SYNC} separation circuits required for VCR servo control in addition to three 16-bit resolution PWM outputs and three 8-bit resolution PWM outputs required for motor control.

(5) On-chip analog circuits for VCR

The analog circuits for VCR consist of a CTL amplifier to amplify record signals of the tape with any gain, a RECCTL driver required for writing CTL and VISS signals, and other constituents required for VCR servo control such as a drum FG amplifier, drum PG comparator, DPFG separation circuit (three-value separation circuit), CFG amplifier, reel FG comparator (2 channels), and C_{SYNC} comparator.

The CTL amplifier can switch gain in 32 steps by software. In actuality, the CTL amplifier output gain is controlled by setting the CTL detection plug with software. Compared with conventional CTL amplifiers, the circuit configuration is more optimized, which results in a reduction of the number of pins from eleven to six. The analog circuits for VCR have made it possible to largely reduce the number of parts, enabling system cost reduction.

[MEMO]

CHAPTER 2   OUTLINE OF µPD784915 SUBSERIES

The µPD784915 Subseries under the 78K/IV Series consists of products provided with an on-chip high-speed, high-performance 16-bit CPU that are improved versions of the 78K/I Series of 8-bit single-chip microcontrollers for VCR software servo control.

The µPD784915 Subseries provides on chip optimum peripheral hardware for VCR control, including a multifunction timer unit (super timer unit) ideal for software servo control, and analog circuits, thus enabling the realization of VCR system/servo/timer control with a single chip.

Moreover, a product with on-chip one-time PROM, the µPD78P4916, is also available.

This chapter describes the µPD784915 as the representative product.

★ **Table 2-1. Differences among µPD784915 Subseries Products**

Part Number Parameter	µPD784915, 784915A, µPD784915B	µPD784916A, µPD784916B	µPD78P4916
Internal ROM capacity	Mask ROM		One-time PROM
	48 Kbytes	62 Kbytes	
Internal RAM capacity	1280 bytes		2048 bytes
Internal memory capacity selection register (IMS)	Not provided		Provided
IC pin	Provided		Not provided
V_{PP} pin	Not provided		Provided
Electrical characteristics	Refer to data sheet of individual products.		

2.1 Features and Application Fields

(1) Features

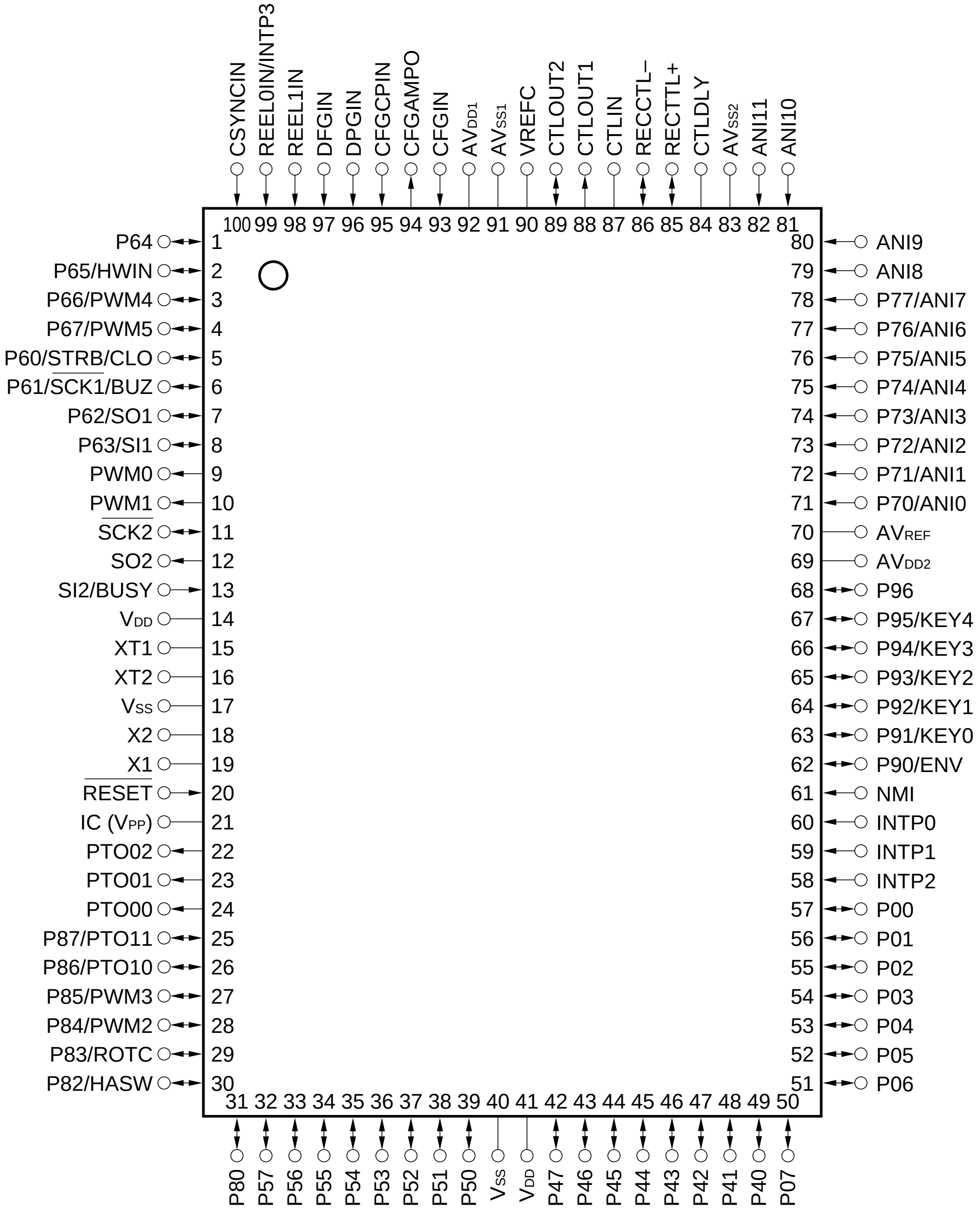
- Minimum instruction execution time: 250 ns (operation when internal clock = 8 MHz)
- On-chip timer unit for VCR servo control (Super timer unit)
- I/O ports: 54
- On-chip VHS-compliant VCR analog circuits
  - CTL amplifier
  - RECCTL driver (rewrite-capable)
  - CFG amplifier
  - DFG amplifier
  - DPG comparator
  - DPFG separation circuit (3-value separation circuit)
  - Reel FG comparator (2 channels)
  - CSYNC comparator
- Serial interface: 2 channels (3-wire serial I/O)
- A/D comparator: 8-bit resolution × 12 channels (conversion time: 10 μs)
- PWM output: 16-bit resolution × 3 channels, 8-bit resolution × 3 channels
- Interrupt functions
  - Vectored interrupt function
  - Macro service function
  - Context switching function
- Low-frequency oscillation mode supported: main system clock frequency = internal clock frequency
- Low-power-dissipation mode supported: CPU operation using subsystem clock possible
- Power supply voltage: V_{DD} = 2.7 to 5.5 V
- On-chip hardware clock function: Low voltage (V_{DD} = 2.7 V (MIN.)), low-current-dissipation clock operation possible

(2) Application fields

System/servo/timer control for VCR (stationary type, camcorder)

2.2 Pin Configuration (Top View)

- **100-pin plastic QFP (14 × 20 mm)**
  μPD784915GF-×××-3BA, 784915AGF-×××-3BA, 784916AGF-×××-3BA,
  ★ **μPD784915BGF-×××-3BA, 784916BGF-×××-3BA, 78P4916GF-3BA**

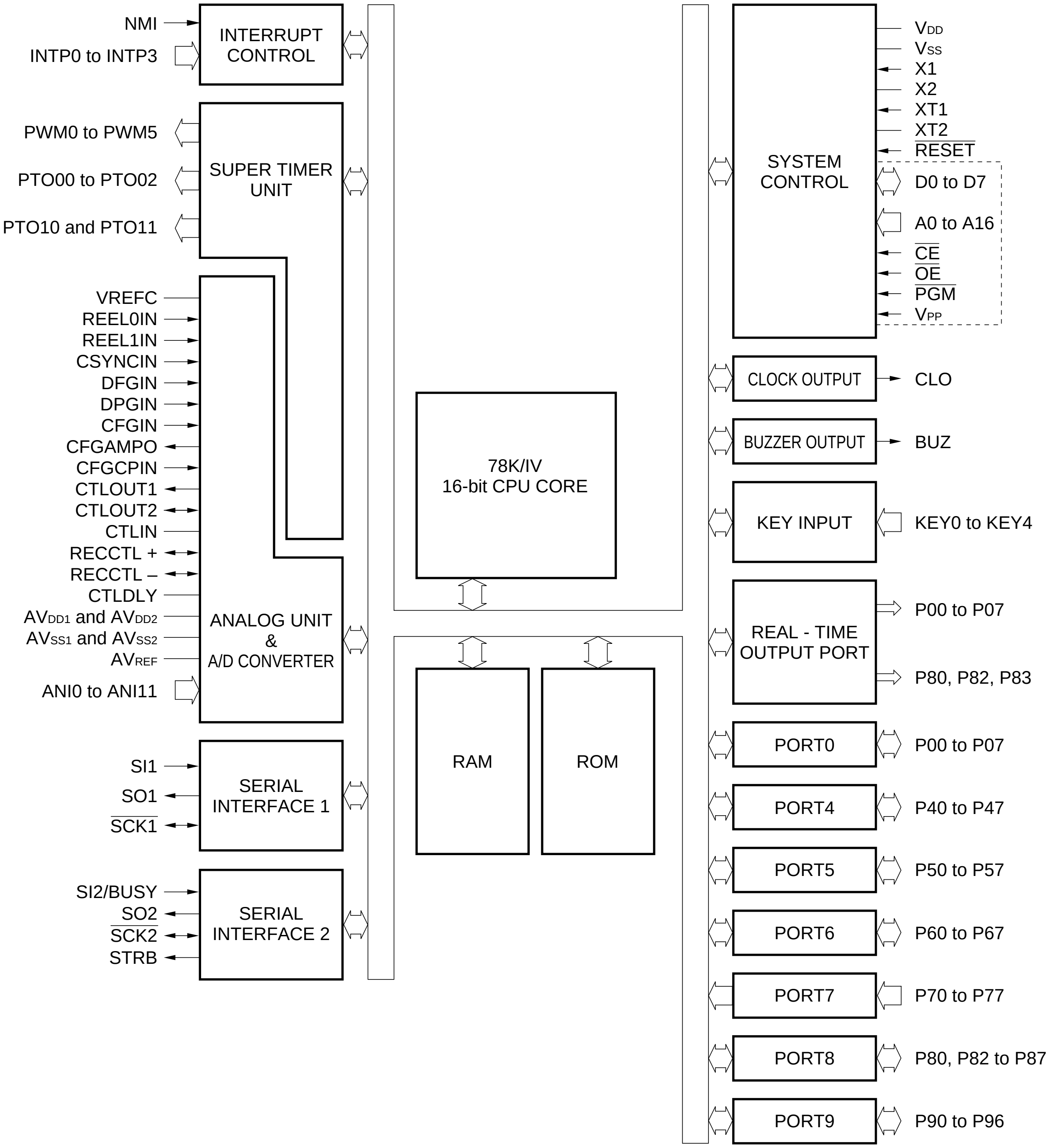


Caution **Connect IC (Internally Connected) pin directly to V_{SS}.**

Remark ( ): μPD78P4916

ANI0 to ANI11	: Analog Input	P00 to P07	: Port0
AV_{DD1}, AV_{DD2}	: Analog Power Supply	P40 to P47	: Port4
AV_{SS1}, AV_{SS2}	: Analog Ground	P50 to P57	: Port5
AV_{REF}	: Analog Reference Voltage	P60 to P67	: Port6
BUSY	: Serial Busy	P70 to P77	: Port7
BUZ	: Buzzer Output	P80, P82 to P87	: Port8
CFGAMPO	: Capstan FG Amplifier Output	P90 to P96	: Port9
CFGCPIN	: Capstan FG Capacitor Input	PTO00 to PTO02,	: Programmable Timer Output
CFGIN	: Analog Unit Input	PTO10, PTO11	
CLO	: Clock Output	PWM0 to PWM5	: Pulse Width Modulation Output
CSYNCIN	: Analog Unit Input	RECCTL+, RECCTL–	: RECCTL Output/PBCLT Input
CTLDLY	: Control Delay Input	REEL0IN, REEL1IN	: Analog Unit Input
CTLIN	: CTL Amplifier Input Capacitor	$\overline{\text{RESET}}$	: Reset
CTLOUT1, CTLOUT2	: CTL Amplifier Output	ROTC	: Chrominance Rotate Output
DFGIN	: Analog Unit Input	$\overline{\text{SCK1}}$, $\overline{\text{SCK2}}$	: Serial Clock
DPGIN	: Analog Unit Input	SI1, SI2	: Serial Input
ENV	: Envelope Input	SO1, SO2	: Serial Output
HASW	: Head Amplifier Switch Output	STRB	: Serial Strobe
HWIN	: Hardware Timer External Input	V_{DD}	: Power Supply
IC	: Internally Connected	VREFC	: Reference Amplifier Capacitor
INTP0 to INTP3	: Interrupt From Peripherals	V_{SS}	: Ground
KEY0 to KEY4	: Key Return	X1, X2	: Crystal (Main System Clock)
NMI	: Non-maskable Interrupt	XT1, XT2	: Crystal (Subsystem Clock)

2.3 Block Diagram



Remarks 1. Internal ROM capacity and RAM capacity differ depending on the product.

2. The broken line indicates the connection in PROM programming mode.

2.4 Outline of Functions

★

Parameter		Part Number: μPD784915, 784915A, μPD784915B	μPD784916A, μPD784916B	μPD78P4916
Instructions		113		
Minimum instruction execution time		250 ns (internal clock: 8 MHz)		
Internal ROM capacity		Mask ROM		One-time PROM
		48 Kbytes	62 Kbytes	
Internal RAM capacity		1280 bytes		2048 bytes
Interrupt		4-level (programmable), vectored interrupts, macro service, context switching		
	External source	9 (including NMI)		
	Internal source	19		
	Macro service available interrupt	25		
	Number of macro service	10 (4 types)		
I/O ports	Input	8		
	I/O	46		
Time-based counter		• 22-bit FRC • Resolution: 125 ns, maximum count time: 524 ms		
Capture register		Input Signal / Number of Bits / Measurement Cycle / Operation Edge CFG / 22 / 125 ns to 524 ms / ↑ ↓ DFG / 22 / 125 ns to 524 ms / ↑ HSW / 16 / 1 μs to 65.5 ms / ↑ ↓ VSYNC / 22 / 125 ns to 524 ms / ↑ CTL / 16 / 1 μs to 65.5 ms / ↑ ↓ TREEL / 22 / 125 ns to 524 ms / ↑ ↓ SREEL / 22 / 125 ns to 524 ms / ↑ ↓		
General-purpose timer		16-bit timer × 3		
PBCTL duty discrimination		• Duty discrimination for Play control signal • VISS detection, wide aspect detection		
Linear time counter		CTL signal counting with 5-bit UDC		
Real-time output port		11		
Serial interface		Clock synchronous (3-wire): 2 channels		
A/D converter		8-bit resolution × 12 channels, conversion time: 10 μs		
PWM output		• 16-bit resolution × 3 channels, 8-bit resolution × 3 channels • Carrier frequency: 62.5 kHz		
Clock function		0.5-second measurement, low-voltage operation possible		
Standby function		HALT mode/STOP mode/Low power dissipation mode/Low power dissipation HALT mode		
Analog circuits		• CTL amplifier • RECCTL driver (rewrite-capable) • CFG amplifier • DFG amplifier	• DPG comparator • DPFG separation circuit (3-value separation circuit) • Reel FG comparator • CSYNC comparator	
Power supply voltage		V_{DD} = 2.7 to 5.5 V		
Package		100-pin plastic QFP (14 × 20 mm)		

★

CHAPTER 3 OUTLINE OF μPD784928, 784928Y SUBSERIES

The μPD784928, 784928Y Subseries under the 78K/IV Series of products with an on-chip high-speed, high-performance 16-bit CPU consists of products for VCR software servo control.

The μPD784928, 784928Y Subseries provides on chip optimum peripheral hardware for VCR control, including a multifunction timer unit (super timer unit) ideal for software servo control, and analog circuits, thus enabling the realization of VCR system/servo/timer control with a single chip.

Moreover, products with on-chip flash memory, the μPD78F4928 and 78F4928Y, are now under development.

This chapter describes the μPD784927 as the representative product.

Table 3-1. Differences among μPD784928, 784928Y Subseries Products

Part Number Parameter	μPD784927, μPD784927Y	μPD78F4928[Note], μPD78F4928Y[Note]
Internal ROM capacity	96 Kbytes (Mask ROM)	128 Kbytes (Flash memory)
Internal RAM capacity	2048 bytes	3584 bytes
Internal memory capacity selection register (IMS)	Not provided	Provided
IC pin	Provided	Not provided
V_{PP} pin	Not provided	Provided
Electrical characteristics	Refer to data sheet of individual products.	

Note Under development

3.1 Features and Application Fields

(1) Features

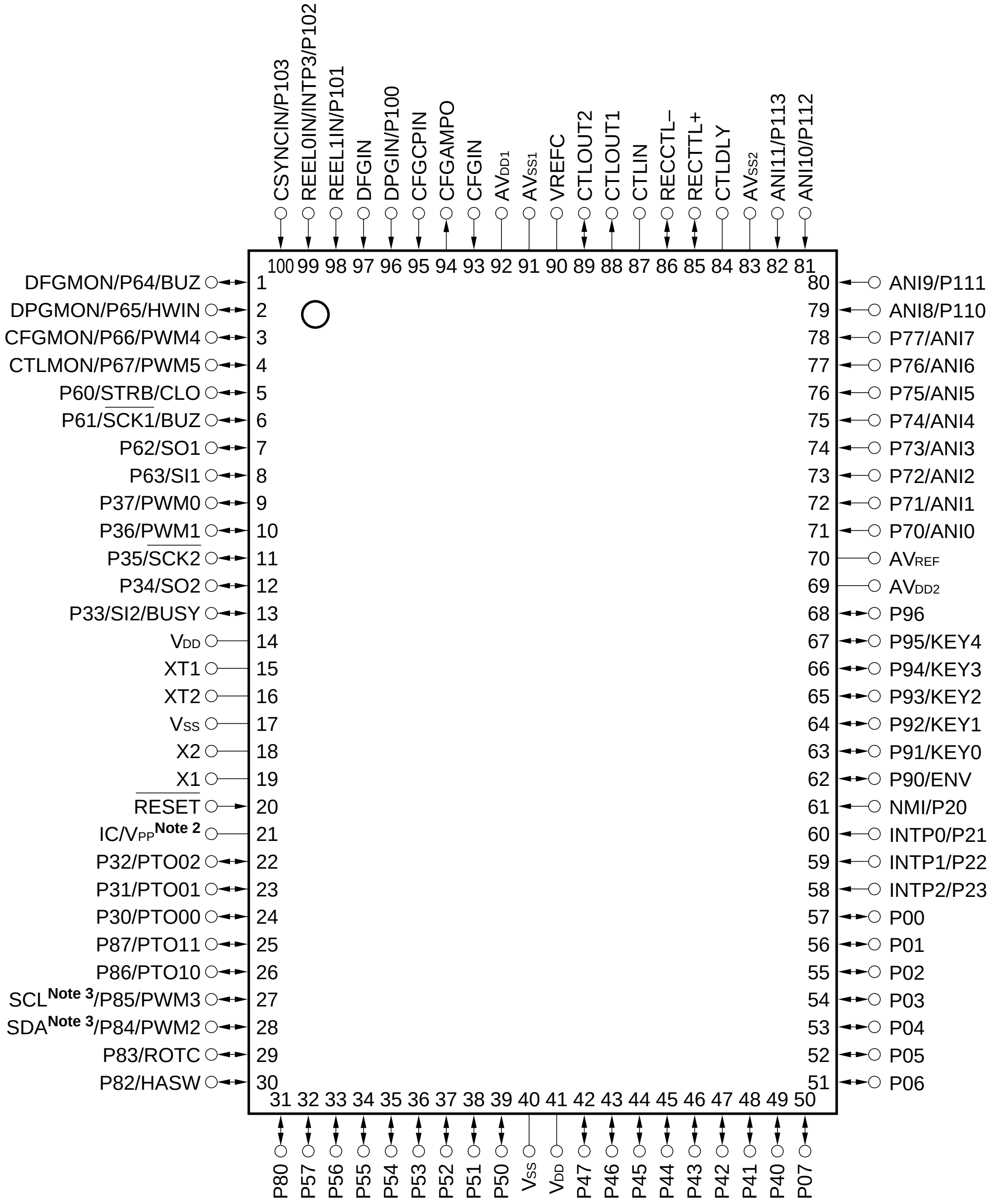
- Minimum instruction execution time: 250 ns (operation when internal clock = 8 MHz)
- On-chip timer unit for VCR servo control (Super timer unit)
- I/O ports: 74
- On-chip VHS-compliant VCR analog circuits
    - CTL amplifier
    - RECCTL driver (rewrite-capable)
    - CFG amplifier
    - DFG amplifier
    - DPG amplifier
    - DPFG separation circuit (3-value separation circuit)
    - Reel FG comparator (2 channels)
    - CSYNC comparator
- Serial interface: 3 channels
    - 3-wire serial I/O: 2 channels
    - I^{2}C bus interface: 1 channel (μPD784928Y Subseries only)
- A/D converter: 12 channels (conversion time: 10 μs)
- PWM output: 16-bit resolution × 3 channels, 8-bit resolution × 3 channels
- Interrupt functions
    - Vectored interrupt function
    - Macro service function
    - Context switching function
- Low-frequency oscillation mode supported: main system clock frequency = internal clock frequency
- Low-power-dissipation mode supported: CPU operation using subsystem clock possible
- Power supply voltage: V_{DD} = 2.7 to 5.5 V
- On-chip hardware clock function: Low voltage (V_{DD} = 2.7 V (MIN.)), low-current-dissipation clock operation possible

(2) Application fields

Stationary type VCRs, camcorders, etc.

3.2 Pin Configuration (Top View)

- **100-pin plastic QFP (14 × 20 mm)**
  μPD784927GF-×××-3BA, 78F4928GF-3BA[Note 1]**,**
  μPD784927YGF-×××-3BA, 78F4928YGF-3BA[Note 1]



Notes 1. Under development

2. The V_{PP} pin is provided only for the μPD78F4928, 78F4928Y.

3. The SCL pin and SDA pin are provided only for the μPD784928Y Subseries.

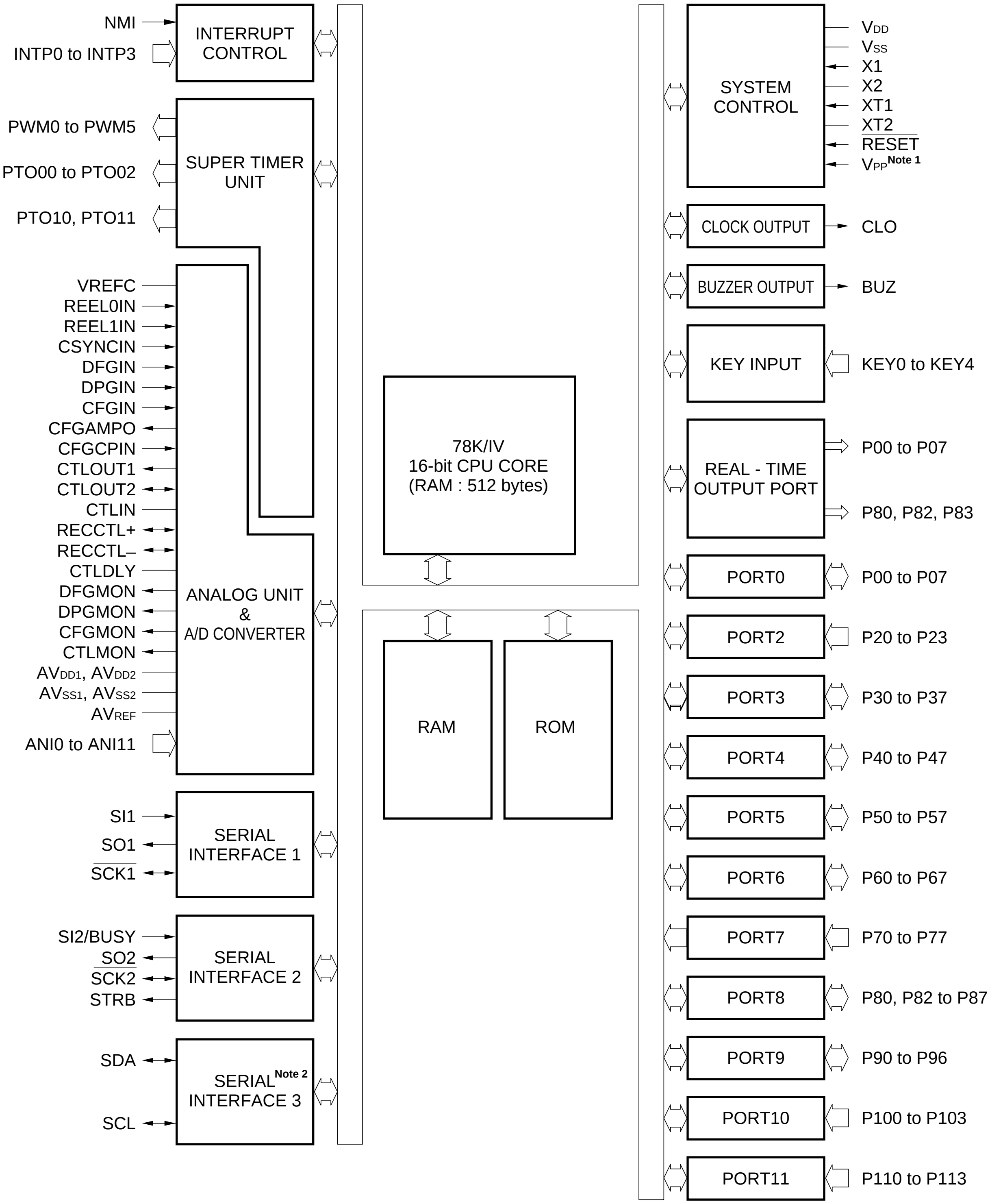
Caution In the normal operation mode, connect the IC (Internally Connected)/V_{PP} pin directly to V_{SS}.

ANI0 to ANI11	: Analog Input	P30 to P37	: Port3
AV_{DD1}, AV_{DD2}	: Analog Power Supply	P40 to P47	: Port4
AV_{SS1}, AV_{SS2}	: Analog Ground	P50 to P57	: Port5
AV_{REF}	: Analog Reference Voltage	P60 to P67	: Port6
BUSY	: Serial Busy	P70 to P77	: Port7
BUZ	: Buzzer Output	P80, P82 to P87	: Port8
CFGAMPO	: Capstan FG Amplifier Output	P90 to P96	: Port9
CFGCPIN	: Capstan FG Capacitor Input	P100 to P103	: Port10
CFGIN	: Analog Unit Input	P110 to P113	: Port11
CFGMON	: Capstan FG Monitor	PTO00 to PTO02,	: Programmable Timer Output
CLO	: Clock Output	PTO10, PTO11	
CSYNCIN	: Analog Unit Input	PWM0 to PWM5	: Pulse Width Modulation Output
CTLDLY	: Control Delay Input	RECCTL+, RECCTL–	: RECCTL Output/PBCLT Input
CTLIN	: CTL Amplifier Input Capacitor	REEL0IN, REEL1IN	: Analog Unit Input
CTLMON	: CTL Amplifier Monitor	$\overline{\text{RESET}}$	: Reset
CTLOUT1, CTLOUT2	: CTL Amplifier Output	ROTC	: Chrominance Rotate Output
DFGIN	: Analog Unit Input	$\overline{\text{SCK1}}$, $\overline{\text{SCK2}}$	: Serial Clock
DFGMON	: DFG Monitor	SCL[Note 1]	: Serial Clock
DPGIN	: Analog Unit Input	SDA[Note 1]	: Serial Data
DPGMON	: DPG Monitor	SI1, SI2	: Serial Input
ENV	: Envelope Input	SO1, SO2	: Serial Output
HASW	: Head Amplifier Switch Output	STRB	: Serial Strobe
HWIN	: Hardware Timer External Input	V_{DD}	: Power Supply
IC	: Internally Connected	V_{PP}[Note 2]	: Programming Power Supply
INTP0 to INTP3	: Interrupt From Peripherals	VREFC	: Reference Amplifier Capacitor
KEY0 to KEY4	: Key Return	V_{SS}	: Ground
NMI	: Non-maskable Interrupt	X1, X2	: Crystal (Main System Clock)
P00 to P07	: Port0	XT1, XT2	: Crystal (Subsystem Clock)
P20 to P23	: Port2		

Notes **1.** The SCL pin and SDA pin are provided only for the μPD784928Y Subseries.

2. The V_{PP} pin is provided only for the μPD78F4928, 78F4928Y.

3.3 Internal Block Diagram



Notes 1. The VPP pin is provided only for the μPD78F4928, 78F4928Y.

2. Provided only for the μPD784928Y Subseries. Supports the I^{2}C bus interface.

Remark The internal ROM and RAM capacities differ according to the product.

3.4 Outline of Functions

(1/2)

<table>
<tr><th colspan="2">Part Number
Parameter</th><th>μPD784927,
μPD784927Y</th><th>μPD78F4928[Note],
μPD78F4928Y[Note]</th></tr>
<tr><td colspan="2">Instructions</td><td colspan="2">113</td></tr>
<tr><td colspan="2">Minimum instruction execution time</td><td colspan="2">250 ns (internal clock: 8 MHz)</td></tr>
<tr><td rowspan="3">Internal memory capacity</td><td>Type</td><td>Mask ROM</td><td>Flash memory</td></tr>
<tr><td>ROM</td><td>96 Kbytes</td><td>128 Kbytes</td></tr>
<tr><td>RAM</td><td>2048 bytes</td><td>3584 bytes</td></tr>
<tr><td rowspan="3">Interrupt sources
(μPD784928 Subseries)</td><td>External</td><td colspan="2">9 (including NMI)</td></tr>
<tr><td>Internal</td><td colspan="2">22 (including software interrupts)</td></tr>
<tr><td></td><td colspan="2">• 4-level programmable priority
• 3 types of servicing:
Vectored interrupts, macro service, context switching</td></tr>
<tr><td rowspan="3">Interrupt sources
(μPD784928Y Subseries)</td><td>External</td><td colspan="2">9 (including NMI)</td></tr>
<tr><td>Internal</td><td colspan="2">23 (including software interrupts)</td></tr>
<tr><td></td><td colspan="2">• 4-level programmable priority
• 3 types of servicing:
Vectored interrupts, macro service, context switching</td></tr>
<tr><td rowspan="2">I/O ports</td><td>Input</td><td colspan="2">20</td></tr>
<tr><td>I/O</td><td colspan="2">54 (including 8 LED direct drive ports)</td></tr>
<tr><td colspan="2">Time-based counter</td><td colspan="2">• 22-bit FRC
• Resolution: 125 ns, maximum count time: 524 ms</td></tr>
<tr><td colspan="2">Capture register</td><td colspan="2">
<table>
<tr><th>Input Signal</th><th>Number of Bits</th><th>Measurement Cycle</th><th colspan="2">Operation Edge</th></tr>
<tr><td>CFG</td><td>22</td><td>125 ns to 524 ms</td><td>↑</td><td>↓</td></tr>
<tr><td>DFG</td><td>22</td><td>125 ns to 524 ms</td><td>↑</td><td></td></tr>
<tr><td>HSW</td><td>16</td><td>1 μs to 65.5 ms</td><td>↑</td><td>↓</td></tr>
<tr><td>VSYNC</td><td>22</td><td>125 ns to 524 ms</td><td>↑</td><td></td></tr>
<tr><td>CTL</td><td>16</td><td>1 μs to 65.5 ms</td><td>↑</td><td>↓</td></tr>
<tr><td>TREEL</td><td>22</td><td>125 ns to 524 ms</td><td>↑</td><td>↓</td></tr>
<tr><td>SREEL</td><td>22</td><td>125 ns to 524 ms</td><td>↑</td><td>↓</td></tr>
</table>
</td></tr>
<tr><td colspan="2">General-purpose timer</td><td colspan="2">16-bit timer × 3</td></tr>
<tr><td colspan="2">PBCTL duty discrimination</td><td colspan="2">• Duty discrimination for Play control signal
• VISS detection, wide aspect detection</td></tr>
<tr><td colspan="2">Linear time counter</td><td colspan="2">CTL signal counting with 5-bit UDC</td></tr>
<tr><td colspan="2">Real-time output port</td><td colspan="2">11</td></tr>
<tr><td colspan="2">Serial interface</td><td colspan="2">• 3-wire serial I/O: 2 channels (including 1 BUSY/STRB function-enabled channel)
• I^{2}C bus interface (multi-master supported): 1 channel (μPD784928Y Subseries only)</td></tr>
<tr><td colspan="2">Buzzer output function</td><td colspan="2">1.95 kHz, 3.91 kHz, 7.81 kHz, 15.6 kHz (Operation when internal clock = 8 MHz)
2.048 kHz, 4.096 kHz, 32.768 kHz (Operation when subsystem clock = 32.768 kHz)</td></tr>
<tr><td colspan="2">A/D converter</td><td colspan="2">8-bit resolution × 12 channels, conversion time: 10 μs</td></tr>
<tr><td colspan="2">PWM output</td><td colspan="2">• 16-bit resolution × 3 channels, 8-bit resolution × 3 channels
• Carrier frequency: 62.5 kHz</td></tr>
<tr><td colspan="2">Clock function</td><td colspan="2">0.5-second measurement, low-voltage operation possible (V_{DD} = 2.7 V)</td></tr>
<tr><td colspan="2">Standby function</td><td colspan="2">HALT mode/STOP mode/Low power dissipation mode/Low power dissipation HALT mode</td></tr>
</table>

Note Under development

(2/2)

Part Number Parameter	µPD784927, µPD784927Y	µPD78F4928[Note], µPD78F4928Y[Note]
Analog circuits	• CTL amplifier • RECCTL driver (rewrite-capable) • CFG amplifier • DFG amplifier	• DPG amplifier • DPFG separation circuit (3-value separation circuit) • Reel FG comparator • CSYNC comparator
Power supply voltage	V_{DD} = 2.7 to 5.5 V	
Package	100-pin plastic QFP (14 × 20 mm)	

Note Under development

3.5 Differences among μPD784928, 784928Y Subseries and μPD784915 Subseries

The μPD784927 is a VCR software servo control product that includes on-chip a high-speed, high-performance 16-bit CPU, enabling the realization of VCR system/servo/timer control with a single chip. The μPD784928 Subseries is an enhanced function version of the μPD784915 Subseries. Moreover, the μPD784928Y Subseries is a product featuring the addition of the I^{2}C bus interface.

Table 3-2 shows the differences among these three subseries.

Table 3-2. Differences among μPD784928, 784928Y Subseries and μPD784915 Subseries

Parameter		μPD784928 Subseries, μPD784928Y Subseries	μPD784915 Subseries
Internal ROM capacity		96 Kbytes/128 Kbytes	48 Kbytes/62 Kbytes
Internal RAM capacity		2048/3584 bytes	1280/2084 bytes
I/O ports	Total	74	54
	Input	20	8
	I/O	54	46
Serial interface		• 3-wire serial I/O : 2 channels • I^{2}C bus interface[Note] : 1 channel	• 3-wire serial I/O : 2 channels
Analog circuit	CTL amplifier	√	√
	RECCTL driver	√	√
	DPFG separation circuit	√	√
	DFG amplifier	√	√
	DPG comparator	√	√
	DPG amplifier	√	—
	CFG amplifier	√	√
	Reel FG comparator	√	√
	CSYNC comparator	√	√
Interrupt	External	9 (including NMI)	9 (including NMI)
	Internal	22 (including software interrupts) 23 (including software interrupts)[Note]	19 (including software interrupts)
Flash memory/PROM		μPD78F4928, 78F4928Y	μPD78P4916

Note In the case of the μPD784928Y Subseries

CHAPTER 4 OUTLINE OF VCR SERVO SYSTEM

4.1 Outline of Software Servo

In the current VCR market, software servo has become the mainstream in VCR servo systems in order to reduce the manufacturing process and improve the reliability of sets. High-performance microcontrollers which can control the whole system with a single chip have been called for to simplify the manufacturing process and lower costs by reducing the number of parts. On the other hand, along with the trend toward sets with high performance and a many functions, microcontrollers with larger memory capacity are increasingly being used.

Analog control servo systems, which have conventionally been the mainstream in VCR servos, present the following problems:

(1) Reliability

The analog servo system uses many components whose characteristics are affected by external environment, such as resistors and capacitors, which makes it difficult to keep the characteristics constant over a long period of time. Moreover, analog servo systems tend to have changing characteristics over-time, which affects reliability.

(2) System adjustment

Due to the uneven characteristics of the resistors/capacitors, a lot of adjustments are required prior to shipment in order to gain desired characteristics.

(3) Number of parts

Analog servos have a large number of parts, which makes it difficult to reduce the size of sets.

VCR servo systems, then, have switched over to digital servos with dedicated ICs. However, servos with dedicated ICs cannot perform flexible control because the servo control algorithm is fixed, and it cannot integrate compensation elements such as digital filter into a device.

In order to solve the above problems, software digital servos using single-chip microcontrollers have come into increasing use in recent years.

Realizing VCR servo systems by software offers the following advantages:

<1> Improved reliability

Because software digital servo systems carry out control using the CPU system clock as a reference, stable operation free from environmental conditions can be realized.

Moreover, software digital servos convert all error amounts to digital values and store them in memory, resulting in accurate sample-and-hold operation. Therefore, unlike analog servo systems, hold values do not change due to capacitor leak.

<2> Compactness and light weight

The number of discrete parts is minimized to enable high-density mounting (reduced mounting space). Moreover, compensation filter is realized as a digital filter, resulting in improved reliability as well as reduction of the number of parts.

<3> Flexible of servo control

Software servos can freely change servo system gain according to the amount of speed error/phase error. Moreover, trick plays such as suspension of control and open loop control for a given time period according to the error amount can be easily realized. Also, AI-related functions such as digital tracking can be integrated.

<4> Easy product development of VCR set

Software servos easily keep up with changes of the drum motor and capstan motor to be used simply by changing software. As a result, design with a high degree of freedom is made possible.

Software servos flexibly support various TV broadcasting systems in the world (such as NTSC and PAL), enabling worldwide use of VCR sets.

To realize software servo control of a VCR, the microcontroller to be used is required to have an advanced arithmetic ability and strong timer function.

In order to easily realize servo control, the μPD784915 Subseries incorporates a variety of peripheral hardware such as the super timer unit and analog circuits for VCR, which are ideal for software servo control of VCRs. By incorporating a 16-bit CPU, the μPD784915 supports high-speed arithmetic instructions and large capacity memory. Therefore, it can easily handle servo processing, which must be real-time, and makes system/servo/timer control of VCRs possible with a single chip.

4.2 Servo Control of VCR

A VCR records video signals forming diagonal patterns on magnetic tape (video tape) using a rotary head. This recording method is called rotary head azimuth recording system.

The recorded pattern of the video signals on the magnetic tape is strictly specified with each format such as VHS system and β system.

Figure 4-1 shows the track pattern of video tapes.

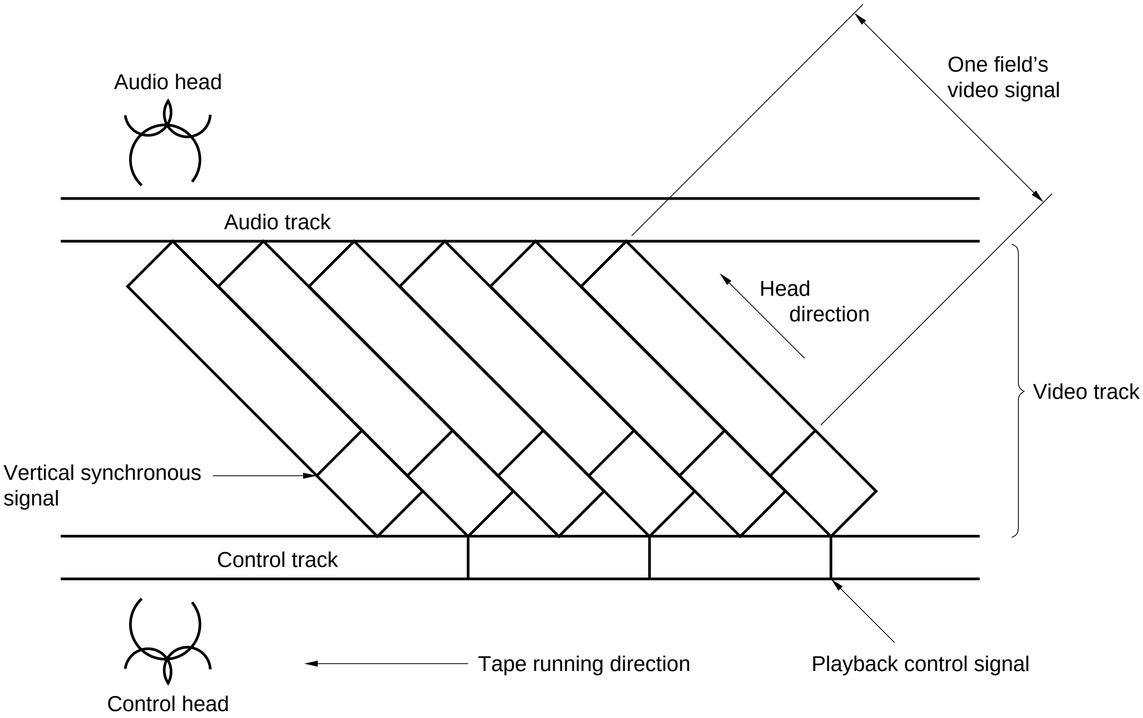
The recording pattern of video signals is as thin as several tens of microns. During VCR playback, the head must accurately trace the recording pattern. This operation is called tracking.

Forming of the recording pattern and playback tracking are controlled by the rotating condition of the rotary head and the running condition of the tape.

A VCR has a drum motor to control the rotation of the rotary head and a capstan motor to control tape running. The VCR carries out record/playback by controlling these two motors.

The servo for recording and playback is explained below.

Figure 4-1. Track Pattern on Video Tape



Remark VHS standard tape speed

Standard mode	: 33.35 mm/s
Triple mode	: 11.12 mm/s

4.3 Servo for Recording

A VCR records exactly one field's video signals on each video track recorded diagonally on a video tape. In TV broadcasting, a frame is composed of two fields.

On a video track, positions on which synchronous signals are recorded are specified. Therefore, control should be made so that the recording drum motor servo synchronizes with the frame cycle of the input video signal and the relation of the position of the video head and vertical synchronous signal are kept constant.

On the other hand, the capstan motor rotates at constant speed because it runs tape accurately at the speed defined in each format.

In addition to these, home VCRs of VHS and β system, etc., record control signals synchronized with the rotation of the drum along with the longer direction of the tape when recording is performed.

Control signal is a pulse signal with 30 [Hz] cycle which is used as a mark when performing playback tracking.

Remarks 1. Video tape running speed of VHS system VCR is 33.35 [mm/s] in standard mode and 11.12 [mm/s] in triple mode.

2. For VHS system VCRs, control signal pulse is normally specified as a signal with 60% high level and 40% low level.

4.4 Servo for Playback

When playing back, rotation of a drum motor and control signals played back from the video tape is synchronized with the reference frame cycle generated in the servo control circuit.

Thereby, the drum motor and the control signals are synchronized indirectly using the reference signal as an intermediary so that the relation between them are made the same as when recording.

As a result, the head is controlled to accurately trace the track on the tape, because the running condition of the tape and the rotation of the head become the same as when recording.

In addition, because the recording condition of the control signals are uneven among sets, it needs to be corrected. Thereby, the relation of the position of the control signals and the video head can be externally adjusted. This is called tracking adjustment. When playing back, the amount of the tracking adjustment is set using an external potentiometer (VR).

4.5 Motor to be Used

Generally, a DC motor is used for VCRs (drum motor and capstan motor). DC motors are motors whose rotation speed varies according to the applied voltage.

Direct drive systems, in which no belts and gears are involved, are becoming the mainstream in the driving method of drum and capstan.

The rotation speed of DC motors fluctuates according to variations in the load and the applied voltage. Therefore, servo systems must control the rotation speed and rotation phase.

Rotation speed control keeps the motor rotation constant. Rotation phase control keeps the relationship between the phase signal and reference phase signal of the motor constant.

4.6 VCR Control Systems

VCRs are mainly composed of the following control systems.

(1) System control
Supervises and controls the whole VCR system.

(2) Servo control
Controls drum motor, capstan motor, and related operations.

(3) Timer control
Performs clock function such as timer reservation, front panel control, and display control.

(4) Camera control (camcorder)
Performs camera section control such as AF and AE.

(5) Others
Blurring correction control, etc. (camcorder).

The μPD784915 is a 16-bit single-chip microcontroller which can perform the three types of control (1) to (3) listed above.

Especially, the super timer unit incorporated in the μPD784915 is designed to easily realize software digital servo control.

4.7 VCR Servo System Control

Servo systems for VCRs control the drum motor for the rotating head and the capstan motor, which runs the tape in low speed.

The VCR elements controlled by a servo system are shown below.

(1) Drum motor speed/phase control
(2) Capstan motor speed/phase control
(3) Generation of head switching signal
(4) Generation of quasi-V_{SYNC} signal for special playback
(5) Generation of recording control signal (RECCTL) (for recording), rewriting (for playback)
(6) Index search control (VISS detection)

Remark This manual mainly explains the method to perform servo control shown in **4.6 (2)** and other controls shown in **4.7 (1)** to **(6)**.

CHAPTER 5   EXAMPLES OF STATIONARY TYPE VCR SERVO CONTROL

This chapter describes examples of stationary type VCR servo system control.

5.1 Examples of System Configuration

In this chapter, the drum motor whose FG wave number (the number of FG signals generated in one rotation of the motor) is 24 poles and the capstan motor whose FG wave number is 36 poles are assumed to be used.

The drum motor is controlled so that the number of rotations is equal to the frame frequency of TV broadcast f_F = 29.97 Hz (NTSC) (the number of rotation is 29.97 r.p.s.) with the servo system locked.

Therefore, the drum FG signal frequency f_{DFG} is as follows:

- $f_{DFG} = f_F \times$ FG wave number = 719.28 [Hz]

Similarly, the capstan motor FG frequency in standard mode (SP mode) is as follows:

- f_{CFGSP} = 1080 [Hz]

The capstan motor FG frequency in triple mode (EP mode), since it is one third the speed of the standard mode, is as follows:

- $f_{CFGSP} = 1080 \div 3 = 360$ [Hz]

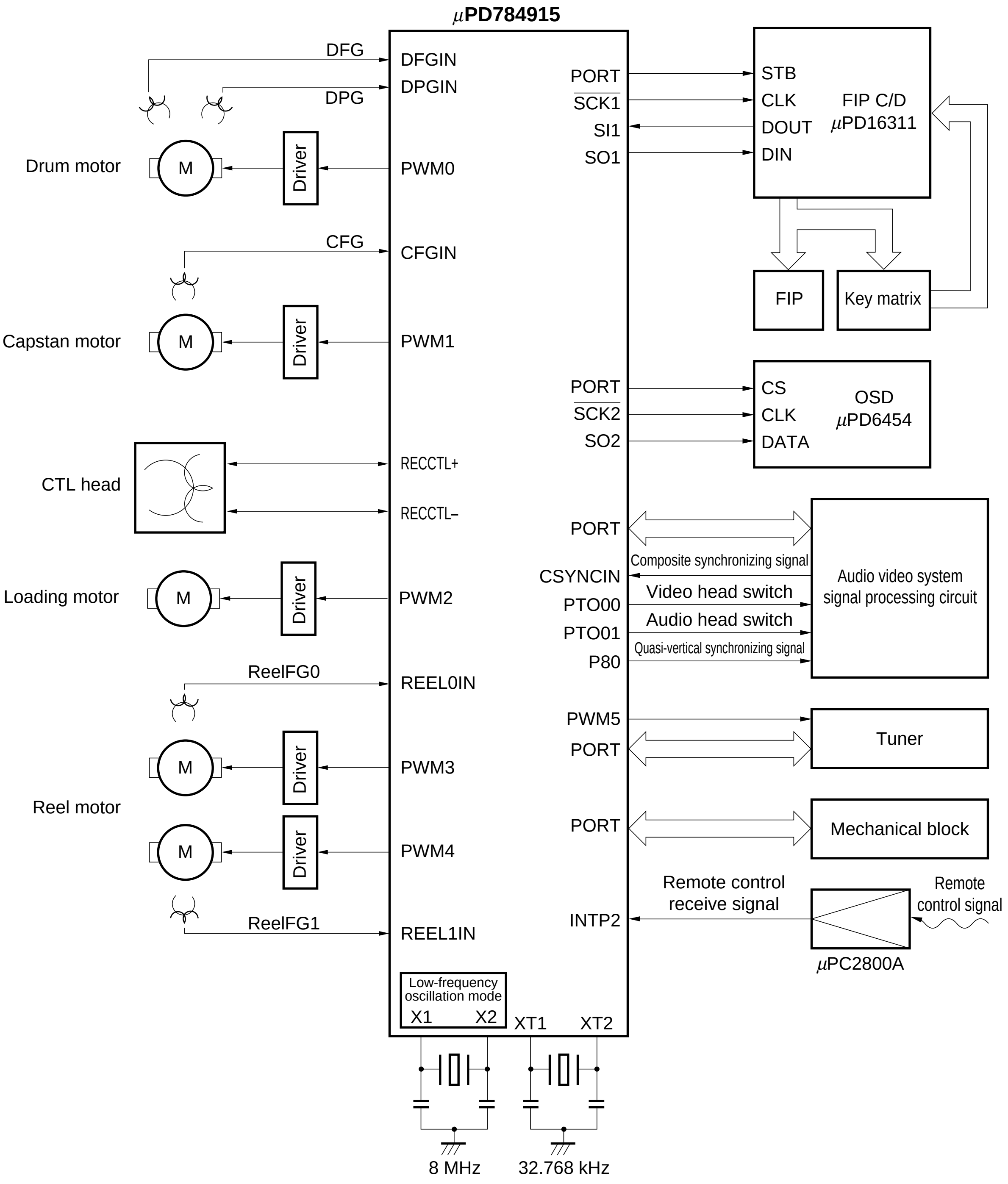
Each motor is driven by PWM output pulse smoothed in external circuit and input to motor driving driver.

PWM0 output is used for driving the drum motor and PWM1 output for capstan motor.

PWM output pulse is smoothed (carrier elimination) through external low pass filter (C-R filter, etc.), impedance converted with operation amplifier, etc., and then input to motor driving driver.

Figure 5-1 shows an example of VCR system configuration to be controlled in this manual.

Figure 5-1. Application to Stationary Type VCR



5.2 Outline of System

This system performs VCR servo control using the μPD784915.

The system executes most of the dedicated digital servo IC functions which have been built in the original sets by software. Moreover the system configures the loop filter, which is a compensation element of the servo system, with a digital filter and realizes it with arithmetic processing by software.

Figure 5-2 shows the processing block diagram in the software digital servo system.

The VCR servo system performs speed/phase control of the drum and capstan motor. Therefore, speed control loop and phase control loop exist in the control loop of each motor.

The FG signal output from the motor is used for detection of the speed error amount and PG signal for detection of the phase error amount. The gains of speed control system and phase control system are set independently from each other.

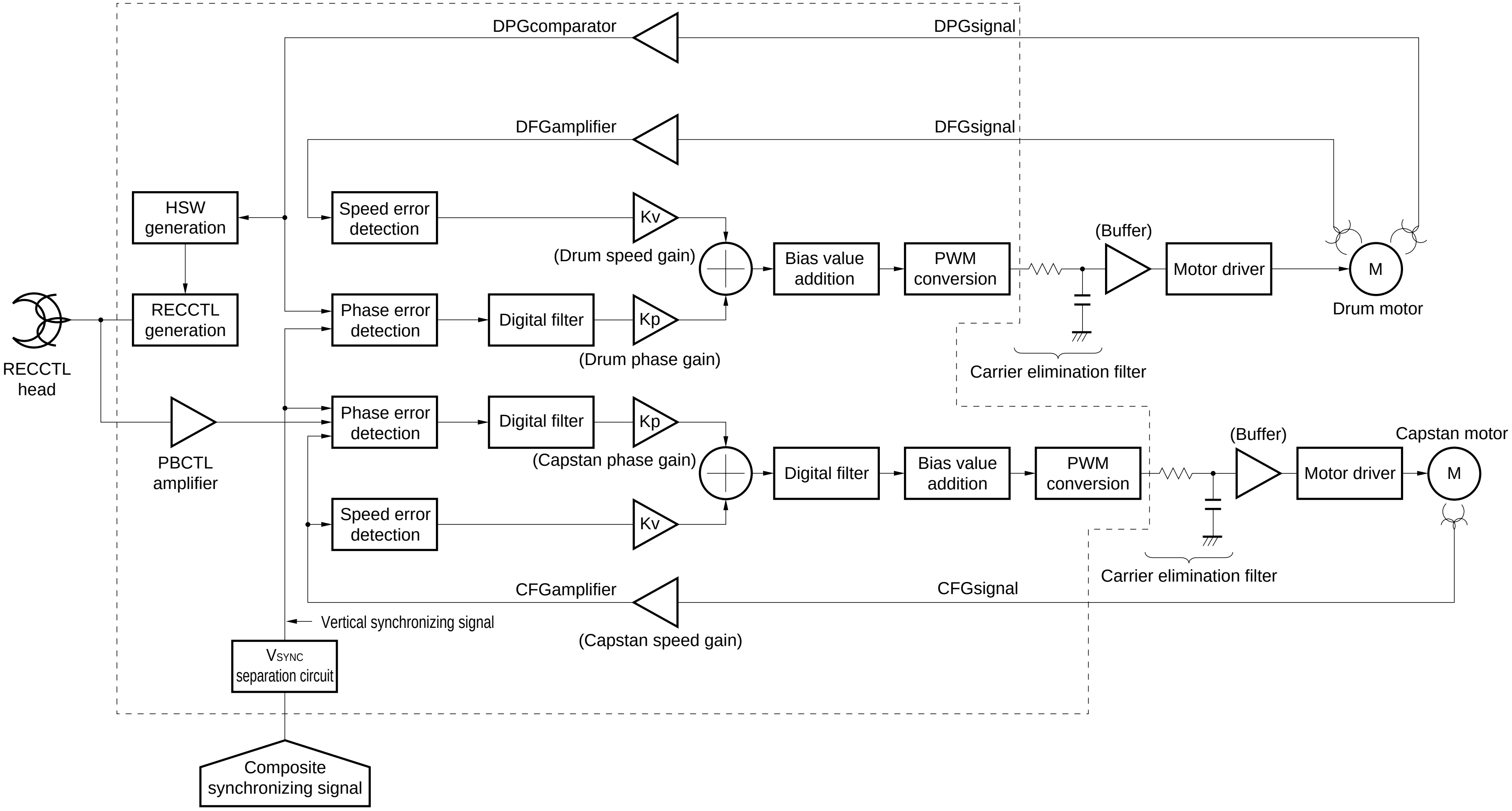
The detected speed and phase error amount are added respectively and then converted to PWM with bias value added. PWM pulse drives each motor after carriers are eliminated through external low pass filter.

The μPD784915 is equipped with analog amplifiers so that amplification of FG and PG signal output from each motor is possible.

The value of the servo circuit built in the set is used as it is for the error amount detection gain and the characteristics of loop filter in the servo system.

In addition to the speed/phase control of drum and capstan motor explained above, the system also generates head switching signal, quasi vertical synchronizing signal, etc.

Figure 5-2. Software Digital Servo System Block Diagram



Remark The process in the μPD784915 is shown in the broken line.

5.3 Using Example of Super Timer Unit

Table 5-1 shows the using examples of the Super Timer Unit, and Figure 5-1 shows the Super Timer Unit block diagram.

Table 5-1. Using Examples of Super Timer Unit

Timer/Counter Name	Register	Use
Event Counter (EC)	ECC0/ECC1/ECC2/ECC3	Generation of internal head switching signal
Timer 0 (TM0)	CR00	Video head switching signal delay control
	CR01	Audio head switching signal delay control
	CR02	Quasi-VSYNC output timing control
Free Running Counter (FRC)	CPT0	Reference phase detection (for drum phase control)
	CPT1	Drum motor phase detection (for drum phase control)
	CPT2	Drum motor speed detection (for drum speed control)
	CPT3	Capstan motor speed detection (for capstan speed control)
	CPT4, CPT5	Tape remain detection by reel FG input
Timer 1 (TM1)	CR10	Generation of internal reference signal (for playback) Buffer oscillator for missing VSYNC (for recording)
	CR11	RECCTL output timing control
	CR12	Capstan motor phase control (for capstan phase controller)
	CR13	Unnecessary VSYNC input mask control
Timer 3 (TM3)	CR30, CR31	PBCTL signal duty detection timing control
	CPT30	PBCTL signal cycle measurement
Timer 2 (TM2)	CR20	Can be used as an interval timer (for system controller)
Timer 4 (TM4)	CR40	Remote control signal duty detection (for remote control decode)
	CR41	Remote control signal cycle measurement (for remote control decode)
Timer 5 (TM5)	CR50	Can be used as internal timer (for system controller)
Up/Down Counter	UDCC	Generation of linear tape counter

Figure 5-3. Super Timer Unit Block Diagram (1/2)

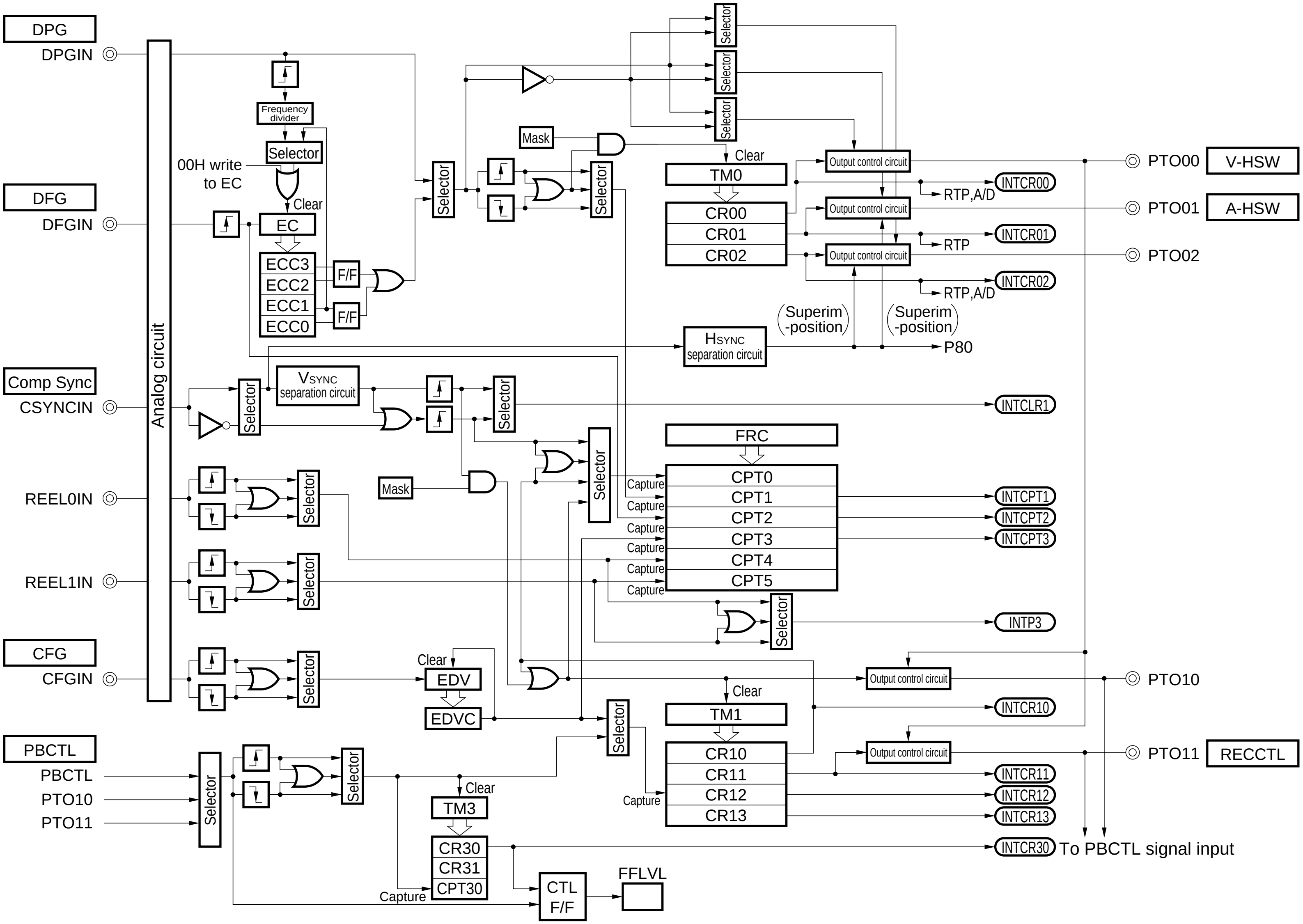
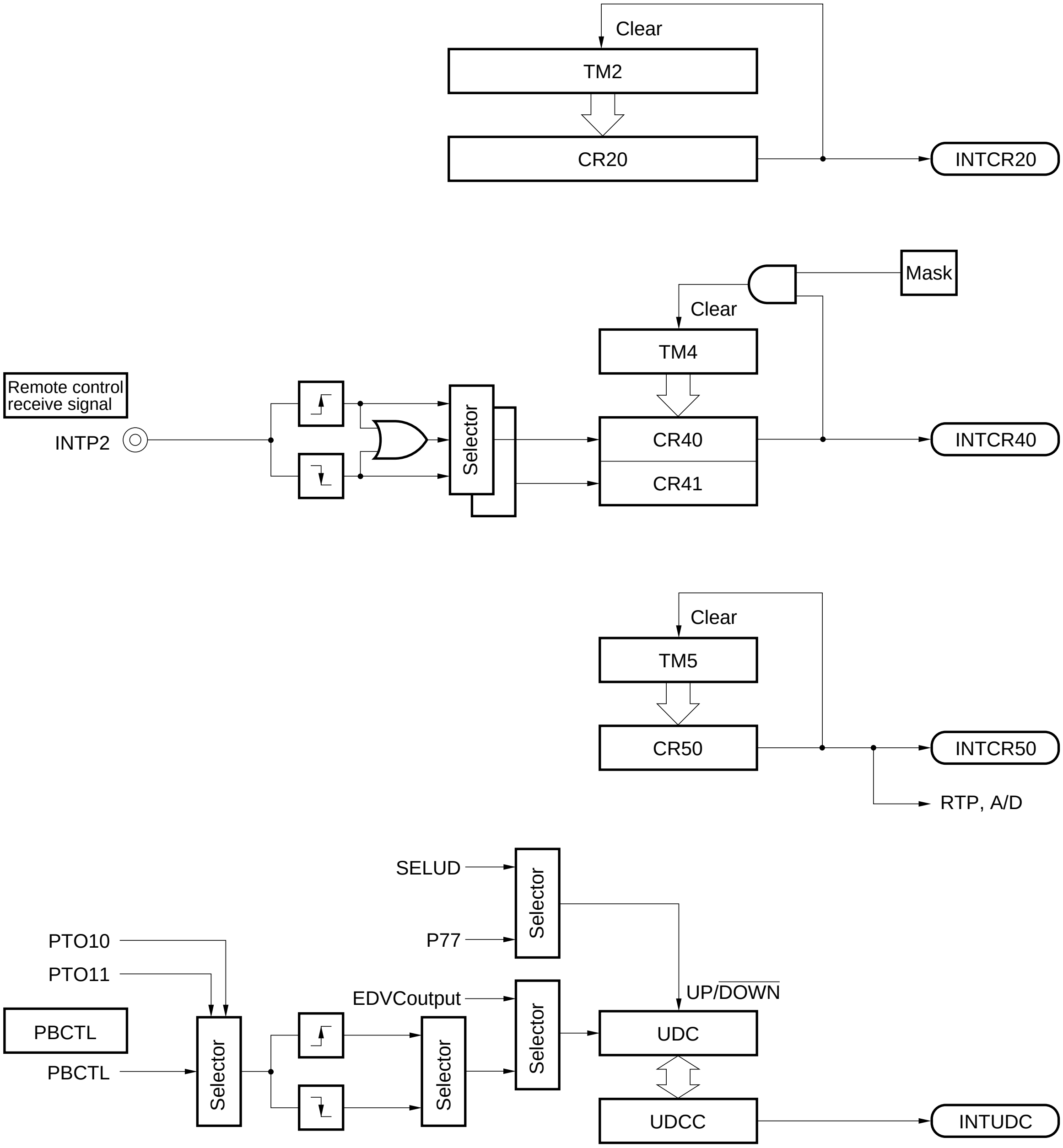


Figure 5-3. Super Timer Unit Block Diagram (2/2)



5.4 Head Switching Signal Generation

5.4.1 Internal head switching signal (HSW-N) generation

(a) HSW-N generation method

This system uses timer 0 clear pulse as phase comparison signal in the drum phase control system. This is called internal head switching signal (HSW-N).

HSW-N is a pulse with 50% duty which is generated from PG and FG signals from the drum motor and synchronizes with the drum rotation.

*μ*PD784915 can generate HSW-N from drum FG signal (DFG signal) and drum PG signal (DPG signal) using event counter (EC).

The DPG signal is input to the DPGIN pin and the DFG signal to the DFGIN pin of the *μ*PD784915.

The pulse generated here is one which is reset, after DPG input, at the rising edge of the second DFG signal and at the falling edge of the fourteenth DFG signal. In this case, the following values are set to the two compare registers of EC.

ECC1...01H
ECC0...0DH

EC output changes at the clock after the clock at which the EC coincides with the compare register. Therefore, the value with 1 subtracted is set as the setting value to the compare register.

Figure 5-4 shows the use of EC, and Figure 5-5 shows the operation timing of EC. Timer 0 is cleared at the rising and falling edges of HSW-N generated in EC.

Figure 5-4. Use of Event Counter (EC)

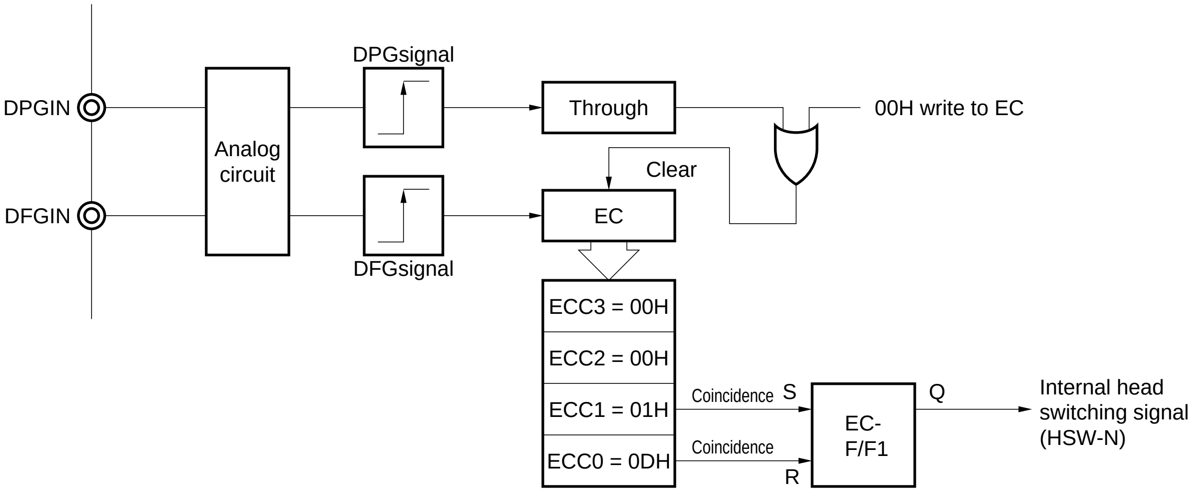
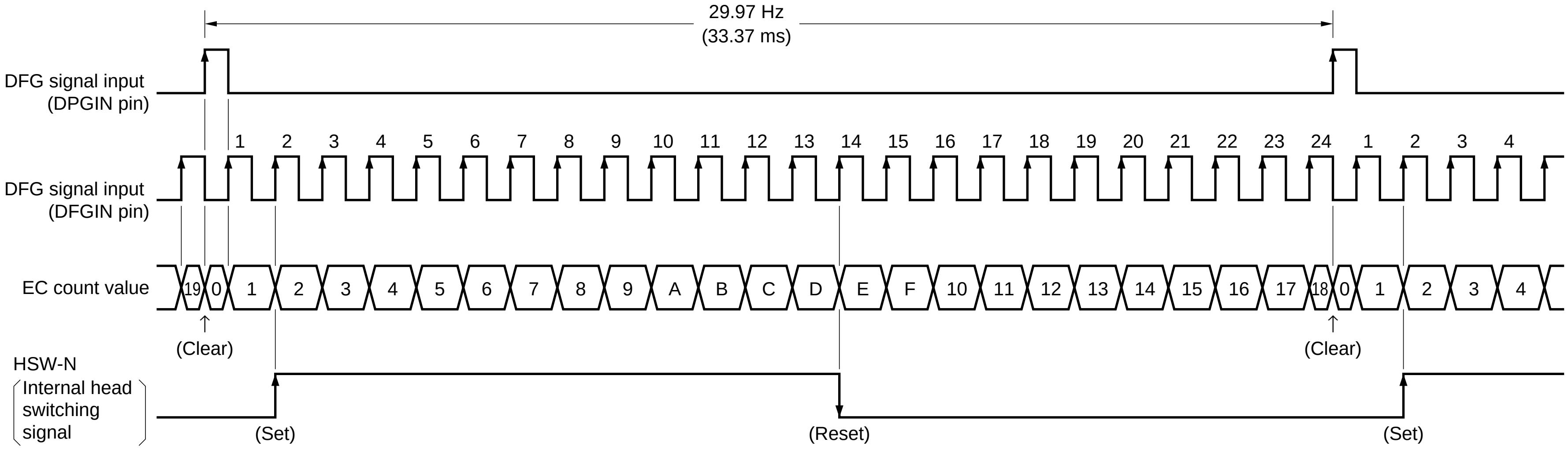


Figure 5-5. Event Counter (EC) Operation Timing



5.4.2 Head switching signal (V-HSW) generation

(a) V-HSW generation method

A VCR is required to externally adjust the head switching signal (V-HSW) and correct the mounting position of the PG signal detector. In order to perform the correction, the internal head switching signal (HSW-N) generated as shown in **5.4.1** is delayed using timer 0 programmable pulse delay circuit.

Figure 5-6 shows the use of timer 0. Figure 5-7 shows the V-HSW timing. Timer 0 is a timer which is cleared at both rising and falling edges of HSW-N.

When a digital value equivalent to the amount of the head switching signal delay is set to compare register 00 (CR00), signals with HSW-N are delayed according to the value set to the compare register are output from the PTO00 output pin. This signal is used as the actual V-HSW.

The relation between the digital value set to CR00 and the delay amount is as follows:

Delay amount = (Setting value to CR00) × 8/f_{CLK}

At 16-MHz operation, 8/f_{CLK} = 1 [μs], then, this is the resolution of timer 0.

In order to correct the positional relation of the PG signal detector and PG magnet, the delay amount to HSW-N should be externally adjustable. Thereby, the data stored in CR00 should be adjustable with analog voltage externally input using the A/D comparator of the μPD784915.

The digital value stored in CR00 is set as follows:

- In EP mode/ LP mode

(Setting value to CR00) = (A/D conversion result) × 11 + 012CH

- In SP mode

(Setting value to CR00) = (A/D conversion result) × 13 + 0190H

When using analog circuit, EC is counted at the reverse edge of the DFG signal, so that correction is required as follows:

- In EP mode/LP mode

(Setting value to CR00) = (A/D conversion result) × 11 + 0120H

- In SP mode

(Setting value to CR00) = (A/D conversion result) × 13 + 0150H

Figure 5-6. Use of Timer 0

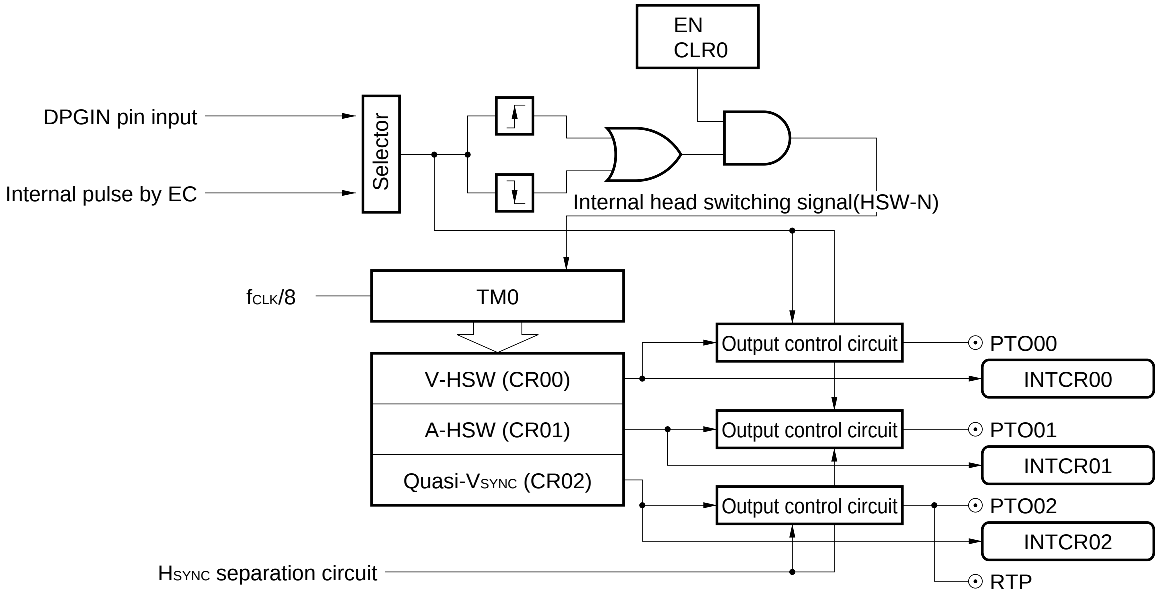
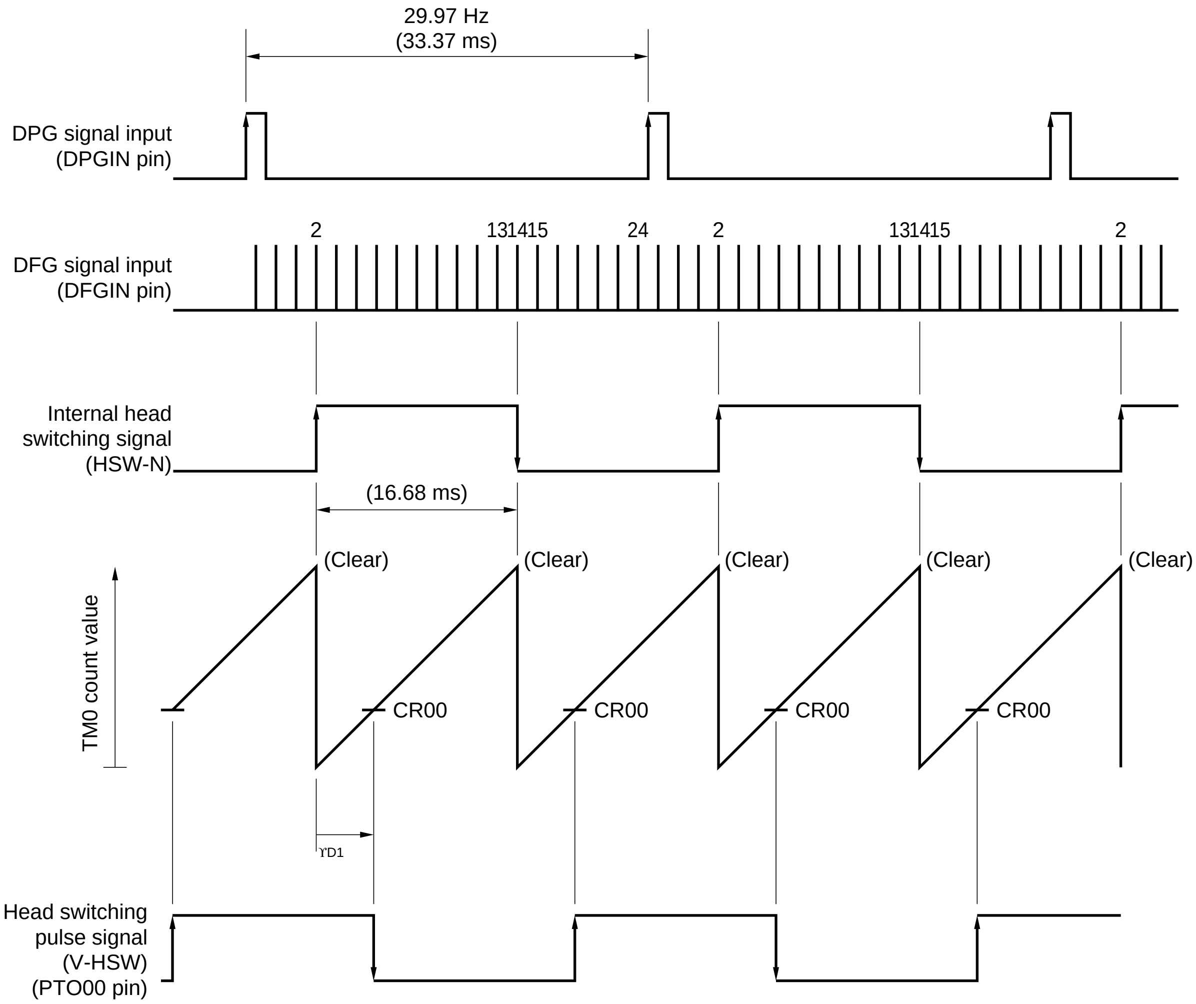


Figure 5-7. Head Switching Signal (V-HSW) Timing (PTO00)

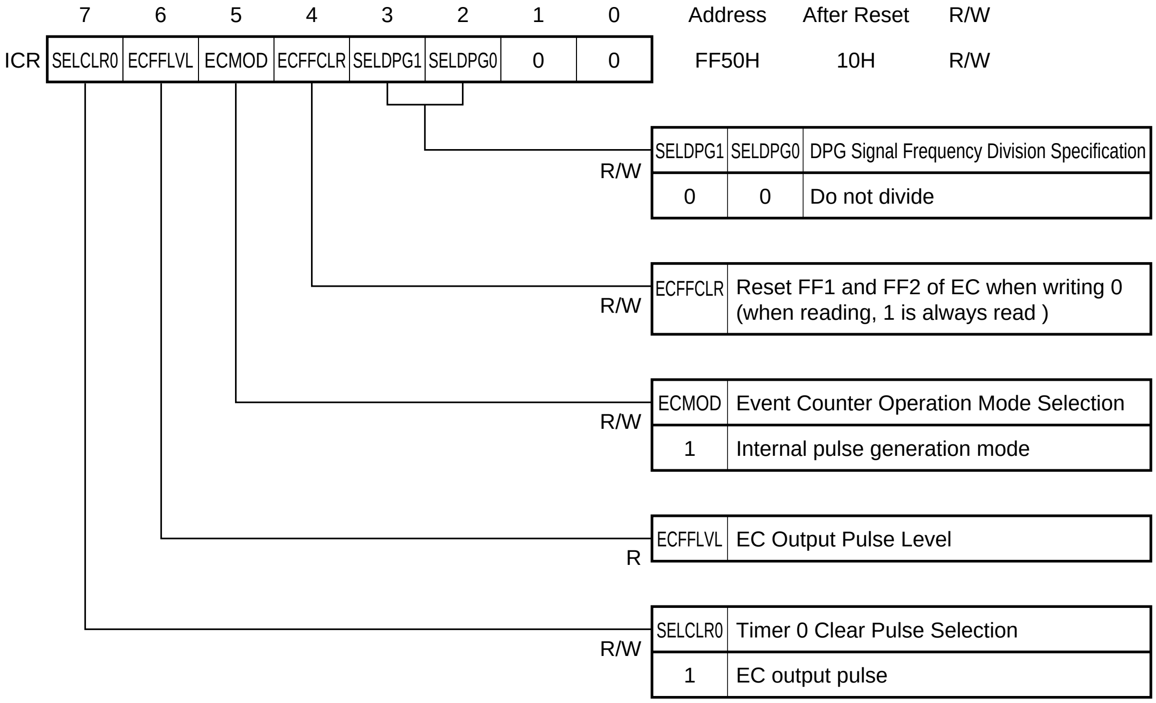


Remark τ_{D1}: Head switching signal delay amount

(b) Timer mode setting

The timer mode setting when generating head switching signal (V-HSW) is as shown in Figure 5-8 to 5-11.

Figure 5-8. Input Control Register (ICR) Format (when generating V-HSW)



	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ICR	SELCLR0	ECFFLVL	ECMOD	ECFFCLR	SELDPG1	SELDPG0	0	0	FF50H	10H	R/W

R/W

SELDPG1	SELDPG0	DPG Signal Frequency Division Specification
0	0	Do not divide

R/W

ECFFCLR	Reset FF1 and FF2 of EC when writing 0 (when reading, 1 is always read)

R/W

ECMOD	Event Counter Operation Mode Selection
1	Internal pulse generation mode

R

ECFFLVL	EC Output Pulse Level

R/W

SELCLR0	Timer 0 Clear Pulse Selection
1	EC output pulse

Figure 5-9. Timer 0 Output Mode Register (TOM0) Format (when generating V-HSW)

	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TOM0	–	–	MOD021	MOD020	MOD011	MOD010	MOD001	MOD000	FF58H	××000000	W

MOD001	MOD000	PTO00 Output Mode Specification
1	1	Delay pulse output mode 2

MOD0n1	MOD0n0	PTO0n Output Mode Specification (n=1, 2)
0	0	General-purpose output mode
0	1	RS output mode
1	0	Delay pulse output mode 1
1	1	Delay pulse output mode 2

Figure 5-10. Timer 0 Output Control Register (TOC0) Format (when generating V-HSW)

	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TOC0	ENHSY	SELPTO	ENTO02	ALV02	ENTO01	ALV01	ENTO00	ALV00	FF59H	00H	W

ALV00	PTO00 Timer Output Active Level Specification
1	Active high

ENTO00	PTO00 Timer Output Enable Specification
1	Output enabled

ALV0n	PTO0n Timer Output Active Level Specification (n = 1, 2)
0	Active low
1	Active high

ENTO0n	PTO0n Timer Output Enable Specification
0	Output disabled (fixed to inactive level)
1	Output enabled

SELPTO	HSYNC Superimposition Pin Specification
0	Superimpose to PTO02
1	Superimpose to PTO01

ENHSY	HSYNC Superimposition Enable to PTO0n Pin Specification
0	Do not superimpose
1	Superimpose

Figure 5-11. Timer Control Register 0 (TMC0) Format (when generating V-HSW)

	⑦	⑥	5	4	③	②	1	0	Address	After Reset	R/W
TMC0	CS1	SELFFLG	ATMSK	ENCLR1	CS0	INTTMSK	0	ENCLR0	FF38H	00H	R/W

R/W

ENCLR0	Timer 0 Clear Control
1	Clear TM0 with TM0 clear signal

R/W

INTTMSK	HSYNC Separation Circuit Initialization Flag Initialize HSYNC separation circuit mask period measurement counter when writing 1. When reading, 0 is always read.

R/W

CS0	Timer 0 Operation
1	Count operation

R/W

ENCLR1	Timer 1 Clear Control
0	Mask CSYNC signal input. TM1 is not cleared.
1	TM1 is cleared with CSYNC signal input

R/W

ATMSK	CSYNC Signal Mask Auto Cancellation Control
0	CSYNC signal mask is not canceled with TM1-CR13 coincidence signal
1	CSYNC signal mask is canceled (set ENCLR1) with TM1- CR13 coincidence signal

R

SELFFLG	HSYNC Self Generation Condition
0	Self generation pulse is not output
1	Self generation pulse is output

R/W

CS1	Timer 1 Operation Control
0	Clear and stop counting
1	Count operation

5.4.3 Audio head switching signal (A-HSW) generation

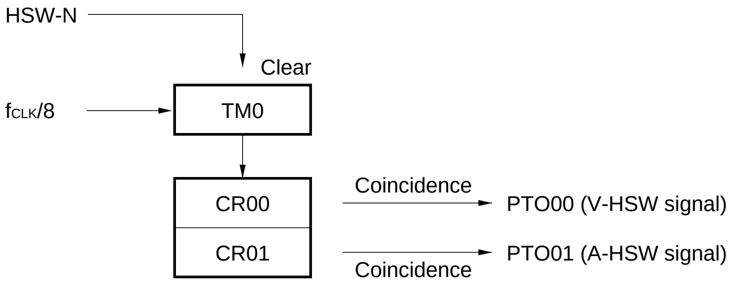
(a) A-HSW generation method

A Hi-Fi VCR requires audio head switching signal (A-HSW) because it records audio signals on the video track with a rotating head.

The audio head is tilted at 270° degrees against the video head, so that A-HSW is output at 270° degrees against the head switching signal (V-HSW).

A-HSW is generated, as well as V-HSW, using timer 0 pulse delay circuit. The compare register uses CR01.

Figure 5-12. Assigning A-HSW to Timer 0



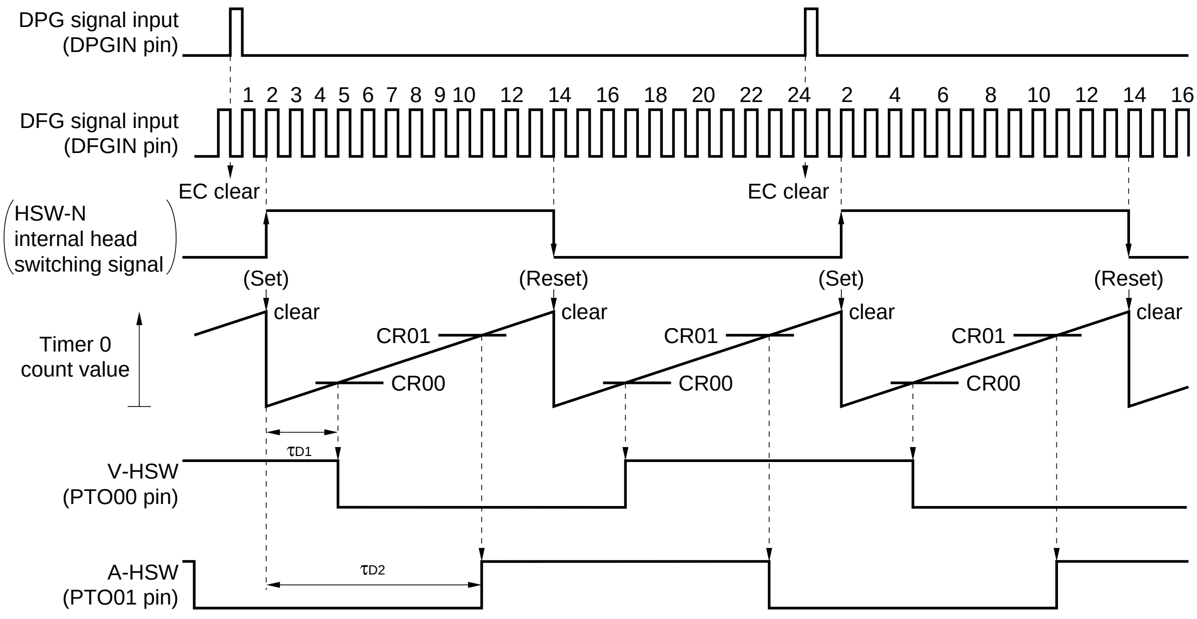
A-HSW is tilted at 270° degrees, so that correction of more than 180° degrees is necessary. The delay pulse output mode 1 is used for 180°-degree correction of A-HSW while the delay pulse output mode 2 is used for V-HSW. Therefore, the delay amount to CR01 is set for the remaining 90° degrees.

The value of V-HSW delay amount with one fourth of a cycle (90° degrees) added is set as the digital value to CR01.

CR01 = CR00 + 1/4 of one V-HSW cycle (1/4 of frame cycle)

Figure 5-13 shows the V-HSW and A-HSW timings.

Figure 5-13. V-HSW and A-HSW Timings



(b) Timer mode settings

Figures 5-14 to 5-16 show the timer mode settings when generating audio head switching signal (A-HSW).

Figure 5-14. Timer 0 Output Mode Register (TOM0) Format (when generating A-HSW)

	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TOM0	–	–	MOD021	MOD020	MOD011	MOD010	MOD001	MOD000	FF58H	××000000	W

MOD0n1	MOD0n0	PTO0n Output Mode Specification (n = 0, 2)
0	0	General-purpose output mode
0	1	RS output mode
1	0	Delay pulse output mode 1
1	1	Delay pulse output mode 2

MOD011	MOD010	PTO01 Output Mode Specification
1	0	Delay pulse output mode 1

Figure 5-15. Timer 0 Output Control Register (TOC0) Format (when generating A-HSW)

	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TOC0	ENHSY	SELPTO	ENTO02	ALV02	ENTO01	ALV01	ENTO00	ALV00	FF59H	00H	W

ALV0n	PTO0n Timer Output Active Level Specification (n = 0, 2)
0	Active low
1	Active high

ENTO0n	PTO0n Timer Output Enable Specification (n = 0, 2)
0	Output disabled (fixed to inactive level)
1	Output enabled

ALV01	PTO01 Timer Output Active Level Specification
1	Active high

ENTO01	PTO01 Timer Output Enable Specification
1	Output enabled

SELPTO	HSYNC Superimposition Pin Specification
0	Superimpose to PTO02
1	Superimpose to PTO01

ENHSY	HSYNC Superimposition Enable to PTO0n Pin Specification
0	Do not superimpose
1	Superimpose

Figure 5-16. Timer Control Register 0 (TMC0) Format (when generating A-HSW)

	⑦	⑥	5	4	③	②	1	0	Address	After Reset	R/W
TMC0	CS1	SELFFLG	ATMSK	ENCLR1	CS0	INTTMSK	0	ENCLR0	FF38H	00H	R/W

R/W

ENCLR0	Timer 0 Clear Control
1	Clear TM0 with TM0 clear signal

R/W

INTTMSK	HSYNC Separation Circuit Initialization Flag Initialize mask period measurement counter of HSYNC separation circuit when writing 1. When reading, 0 is always read.

R/W

CS0	Timer 0 Operation Control
1	Count Operation

R/W

ENCLR1	Timer 1 Clear Control
0	Mask CSYNC signal input. TM1 is not cleared.
1	TM1 is cleared with CSYNC signal input

R/W

ATMSK	CSYNC Signal Mask Auto Cancellation Control
0	CSYNC signal mask is not canceled with TM1-CR13 coincidence signal
1	CSYNC signal mask is canceled (set ENCLR1) with TM1-CR13 coincidence signal

R

SELFFLG	HSYNC Self Generation Condition
0	Self generation pulse is not output
1	Self generation pulse is output

R/W

CS1	Timer 1 Operation Control
0	Clear and stop counting
1	Count operation

5.5 Drum Speed Control

The drum FG signal (DFG) from the drum motor is input to DFGIN input pin of the μPD784915. The value of the free running counter (FRC) is captured to capture register 2 (CPT2H and CPT2L) at the rising edge of DFG and INTCPT2 interrupt request is generated.

Since the FRC of the μPD784915 is 22-bit configuration and has 6 CPTs (22-bit), the measurement of generation cycle can be carried out for 6 types of capture trigger.

The CPT is configured with CPT2H, which captures the higher 6 bits, and the CPT2L, which captures the lower 16 bits.

The FRC value is stored in CPT2H and CPT2L respectively with DFG input.

This program uses the FRC as speed control information by DFG input.

The drum speed error amount is calculated in INTCPT2 interrupt processing routine. In INTCPT2 interrupt processing routine, the cycle of FG signal is measured by subtracting the current capture value. Then, the speed error amount is detected by comparing the cycle data when the speed control system is locked. The concrete method of finding the drum speed error amount is shown below. Figure 5-18 shows an example of drum speed control timings.

Figure 5-17. Drum Speed Error Amount Detection Method

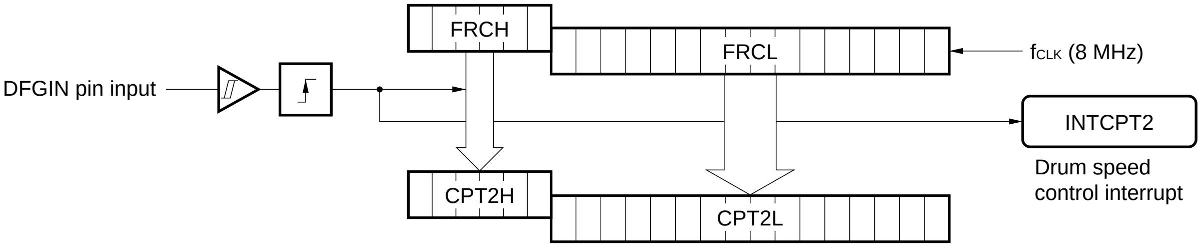
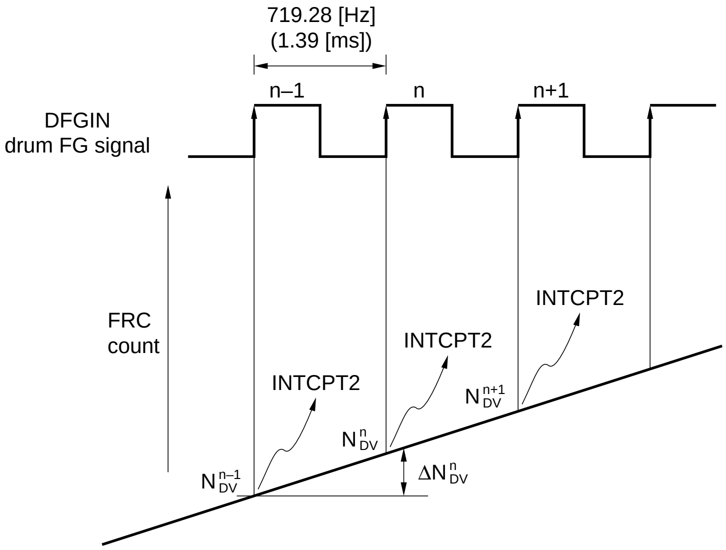


Figure 5-18. Drum Speed Control Timings



When the frame frequency of TV broadcast is assumed as f_F, the f_F is as follows:

- $f_F = 29.97$ [Hz]

Then, since the drum FG wave number is 24 poles, the drum FG signal frequency in the standard playback is as follows:

- $f_{DFG} = 24 \times f_F = 719.28$ [Hz]

Therefore, the drum FG signal cycle NDFG becomes as follows:

$$T_{DFG} = \frac{1}{f_{DFG}}$$

$$N_{DFG} = \frac{T_{DFG}}{T_{FRC}} = \frac{1}{T_{FRC} \times f_{DFG}} = 11122.2 = 2B72H \text{ [Count]}$$

Where: $T_{FRC} = 125$ [ns]

The drum speed error amount E_{DV} is represented by the following expression:

$$E_{DV} = (N_{DV}^{n} - N_{DV}^{n-1}) - N_{DFG}$$
$$= \Delta N_{DV}^{n} - N_{DFG}$$

In the above expression, N_{DV}^n represents the value of the free running counter (FRC) captured at the n-th FG pulse. The meanings of the signs for the drum speed error amount E_{DV} calculated from the expression above are as follows:

(1) When DFG cycle is longer than the target value...+
(when the rotation of the drum motor is slow)
(2) When DFG cycle is shorter than the target value...–
(when the rotation of the drum motor is fast)

5.6 Drum Phase Control

The drum phase error amount is detected by comparing the capture value (CPT1) of the free running counter (FRC) by the internal head switching signal (HSW-N) and the FRC capture value (CPT0) by the reference frame cycle (V_{SYNC} for recording, the coincidence of timer 0 and compare register 10 (CR10) for playback).

Basically, the only difference between the processing for recording and for playback is that the capture source of the capture 0 (CPT0) of FRC is switched.

5.6.1 Phase reference

Timer 1 of the Super Timer Unit is used to generate the phase/reference signal of the servo system in all the modes. The TM1 operation differs for recording and for playback.

Figure 5-19 shows the TM1 peripheral circuit. The setting of selectors differs for recording and playback.

[For recording]

When recording, TM1 is operated as an interval timer synchronized with the frame cycle of TV broadcast. Composite synchronizing signal is input for CSYNCIN input pin. TM1 is cleared at the rising edge of the composite synchronizing signal using a digital noise elimination circuit incorporated in the CPU. Thus, timer 1 is operated as a frame synchronous interval timer synchronized with vertical synchronous signal input externally.

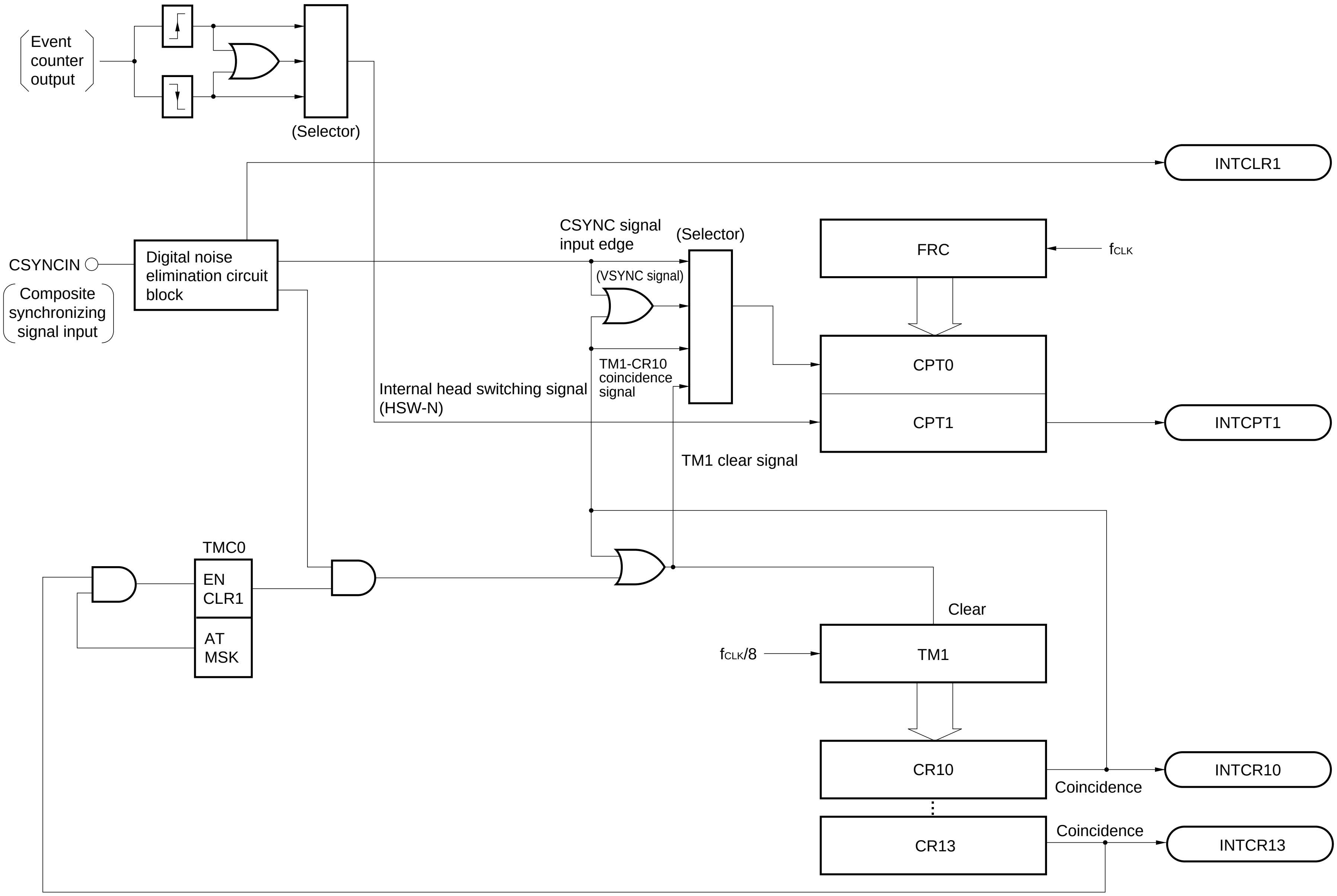
Approximately 90% of the frame sync of the CSYNCIN pin input should be masked so that misoperation caused by noise, etc., is prevented.

[For playback]

When playing back, TM1 is operated as a free running interval timer which has the frequency equal to the frame cycle of TV broadcast. The value corresponding to the frame cycle is stored in CR10 of TM1 because vertical synchronous signal is not externally input when playing back.

When playing back, TM1 clear timing is the reference signal of phase control. The phase reference signal is the TM1 clear timing.

Figure 5-19. Timer 1 Peripheral Circuit



(1) Phase reference for playback

When playing back, ENCLR1 flag is reset and timer 1 (TM1) clear by CLR1 input is always disabled. Data which makes the TM1 clear interval equal to the frame cycle is set to compare register 10 (CR10). Thereby, TM1 is operated as a free running interval timer having the frequency equal to the frame cycle.

Figure 5-20 shows the TM1 operation timings for playback.

Since the reference frame cycle is 33.366 [ms], the set value of CR10 is as follows:

$$\text{CR10} = \frac{33.366\ [\text{ms}]}{1.0\ [\mu\text{s}]} = 33366 = 8256\text{H}$$

Figure 5-21 shows the mode settings of timer 1 for playback.

Figure 5-20. Example of Timer 1 Operation Timings (for playback)

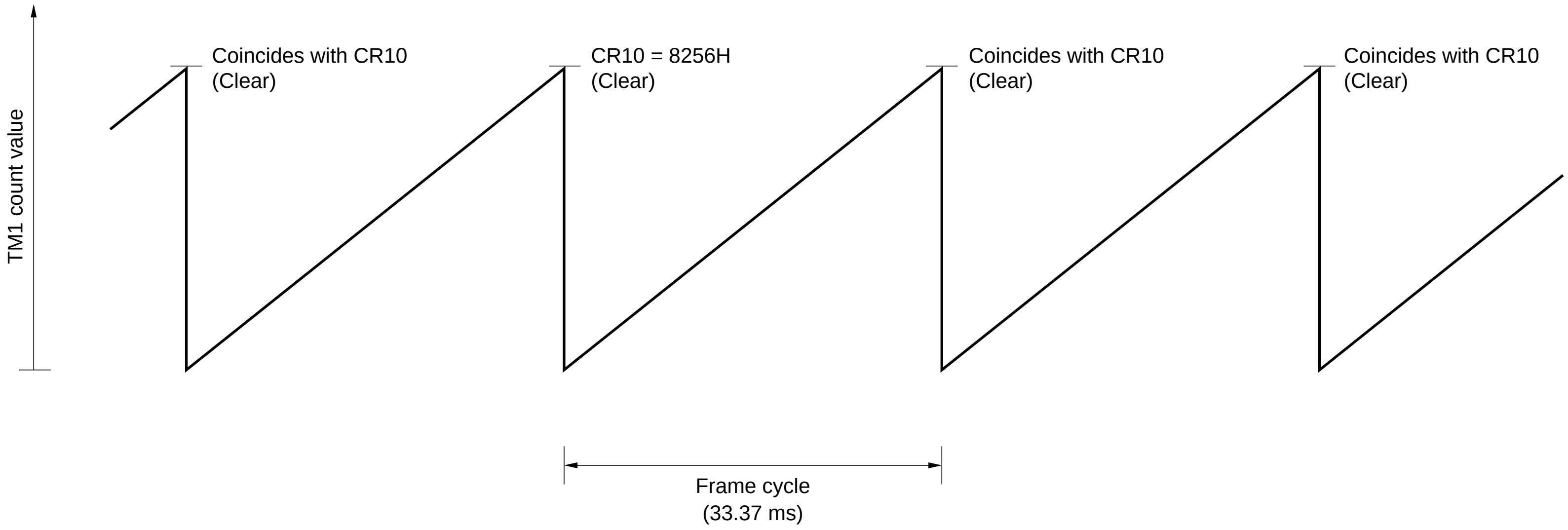


Figure 5-21. Timer Control Register 0 (TMC0) Format (drum phase control for playback)

	⑦	⑥	5	4	③	②	1	0	Address	After Reset	R/W
TMC0	CS1	SELFFLG	ATMSK	ENCLR1	CS0	INTTMSK	0	ENCLR0	FF38H	00H	R/W

R/W

ENCLR0	Timer 0 Clear Control
0	Mask timer 0 clear signal. TM0 is not cleared.
1	TM0 is cleared by TM0 clear signal

R/W

INTTMSK	HSYNC Separation Circuit Initialization Flag When writing 1, mask period measurement counter of HSYNC separation circuit is initialized. When reading, 0 is always read.

R/W

CS0	Timer 0 Operation Control
0	Clear and stops counting
1	Count operation

R/W

ENCLR1	Timer 1 Clear Control
0	Mask CSYNC signal input. TM1 is not cleared.

R/W

ATMSK	CSYNC Signal Mask Auto Cancellation Control
0	CSYNC signal mask is not canceled by TM1-CR13 coincidence signal
1	CSYNC signal mask is canceled by TM1-CR13 coincidence signal (ENCLR1 is set).

R

SELFFLG	HSYNC Self Generation Condition
0	Self generation pulse is not output
1	Self generation pulse is output

R/W

CS1	Timer 1 Operation Control
1	Count operation

(2) Phase reference for recording

When recording, timer 1 (TM1) is operated as an interval timer synchronized with a vertical synchronizing signal. Figure 5-22 shows an example of TM1 operation timings for recording.

Composite synchronizing signals are input from the video processing circuit to the CSYNCIN input pin. The composite synchronizing signal includes cut-in pulse, equalizing pulse, and horizontal synchronizing signal, as well as vertical synchronizing signal, so that vertical synchronizing signal needs to be separated from these signals. The digital noise elimination circuit incorporated in the μPD784915 is used for this purpose.

By using the digital noise elimination circuit, it is possible to clear TM1 at the rising edge of the vertical synchronizing signal included in the composite synchronizing signal and generate interrupt for INTCLR1.

TM1 is cleared in synchronization with the falling edge of the vertical synchronizing signal in the composite synchronizing signal input to the CSYNCIN pin. If noise is mixed in the composite synchronizing signal input to the CSYNCIN pin, TM1 clear may be mistakenly carried out. TM1 clear by CSYNCIN input is disabled for the certain period of time using the ENCLR1 flag and compare register 13 (CR13) in CSYNCIN input which controls TM1 clear enable/disable. Figure 5-23 shows the timer 1 mode setting for recording.

In this program, TM1 clear input disabled time is set to approximately 90% of the frame cycle after inputting a separated vertical synchronizing signal.

Since the frame cycle is 33.36 [ms], the time 90% of it is calculated as follows:

$$33.36 \times 0.9 = 30.03 \text{ [ms]}$$

In this program, the mask period is set with CR13 so that interrupt is generated at a point which is 90% of a field cycle. Since the TM1 count clock frequency is $f_{CLK}/8$ = (1.0 [μs]), the value set for CR13 is as follows:

$$\text{CR13} = \frac{30.03 \text{ [ms]}}{1.0 \text{ [}\mu\text{s]}} = 30030 = 754\text{EH}$$

When the ATMSK flag, which controls CSYNCIN signal mask auto cancellation, is set, 754EH is set to CR13, and the timer is started, ENCLR1 control bit is set when 90% of a field cycle is passed. If ENCLR1 control bit is controlled, CSYNCIN pin input is masked for the 90% period of time of a frame cycle.

If, for some reason, vertical synchronizing signal is not input, the compare register 10 (CR10) value is set so that clear is executed at a cycle approximately equal to the frame cycle by coincidence signal of CR10 of timer 1 and timer 1. In this case, CR10 is set with additional 3% of the frame cycle. Therefore, as the frame frequency is 29.97 [Hz] (33.36 [μs]), the set cycle is as follows:

$$33.36 \times 1.03 = 34.37 \text{ [ms]}$$

Since the count clock frequency of timer 1 is $f_{CLK}/8$ = (1.0 [μs]), the set value to CR10 is as follows:

$$\boxed{\text{CR10} = \frac{34.37\ [\text{ms}]}{1.0\ [\mu\text{s}]} = 34370 = 8642\text{H}}$$

Figure 5-22. Example of Timer 1 Operation Timings (for recording)

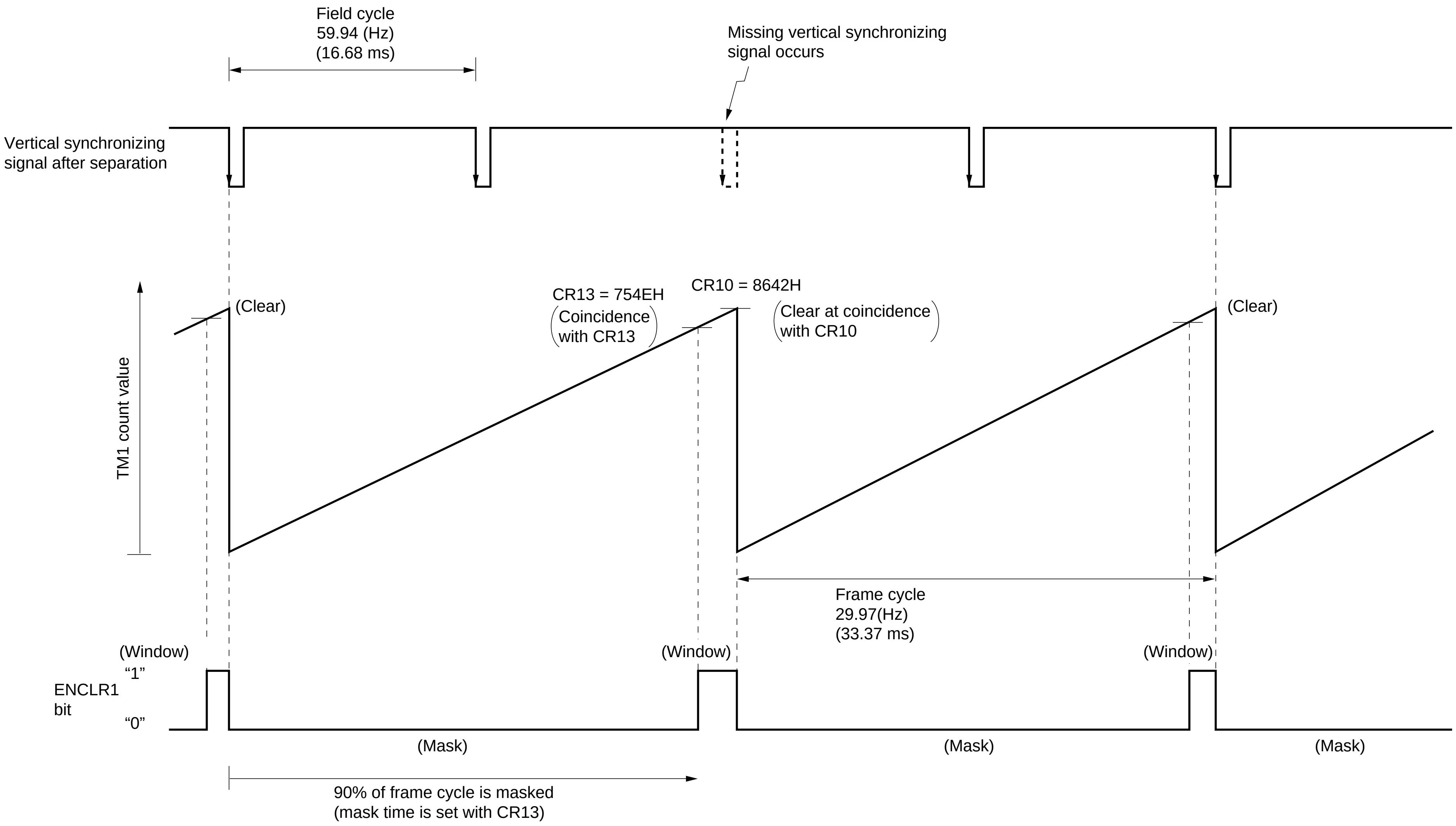


Figure 5-23. Timer Control Register 0 (TMC0) Format (drum phase control for recording)

	⑦	⑥	5	4	③	②	1	0	Address	After Reset	R/W
TMC0	CS1	SELFFLG	ATMSK	ENCLR1	CS0	INTTMSK	0	ENCLR0	FF38H	00H	R/W

R/W

ENCLR0	Timer 0 Clear Control
0	Mask timer 0 clear signal. TM0 is not cleared.
1	TM0 is cleared by TM0 clear signal

R/W

INTTMSK	HSYNC Separation Circuit Initialization Flag When writing 1, HSYNC separation circuit mask period measurement counter is initialized. When reading, 0 is always read.

R/W

CS0	Timer 0 Operation Control
0	Clear and stops counting
1	Count operation

R/W

ENCLR1	Timer 1 Clear Control
1	TM1 is cleared by CSYNC signal input

R/W

ATMSK	CSYNC Signal Mask Auto Cancellation Control
1	CSYNC signal mask is canceled by TM1-CR13 coincidence signal (ENCLR1 is set)

R

SELFFLG	HSYNC Self Generation Condition
0	Self generation pulse is not output
1	Self generation pulse is output

R/W

CS1	Timer 1 Operation Control
1	Count operation

5.6.2 Drum phase control for playback

For drum phase control for playback, drum motor rotation phase is synchronized with a reference timer which has the frequency equal to the TV broadcast frame cycle f_F.

Timer 1 (TM1) of Super Timer Unit is used for the reference timer as mentioned earlier.

For VHS standards, the locking point of the drum phase in this program is specified as 6.5H before the phase reference signal.

1H, here, shows one cycle of horizontal synchronizing signal (1H = 63.56 [μs]).

Figure 5-24 shows the use of timer for drum phase control for playback. Figure 5-25 shows the drum phase control timing chart for playback.

Figure 5-24. Use of Timer for Drum Phase Control (for playback)

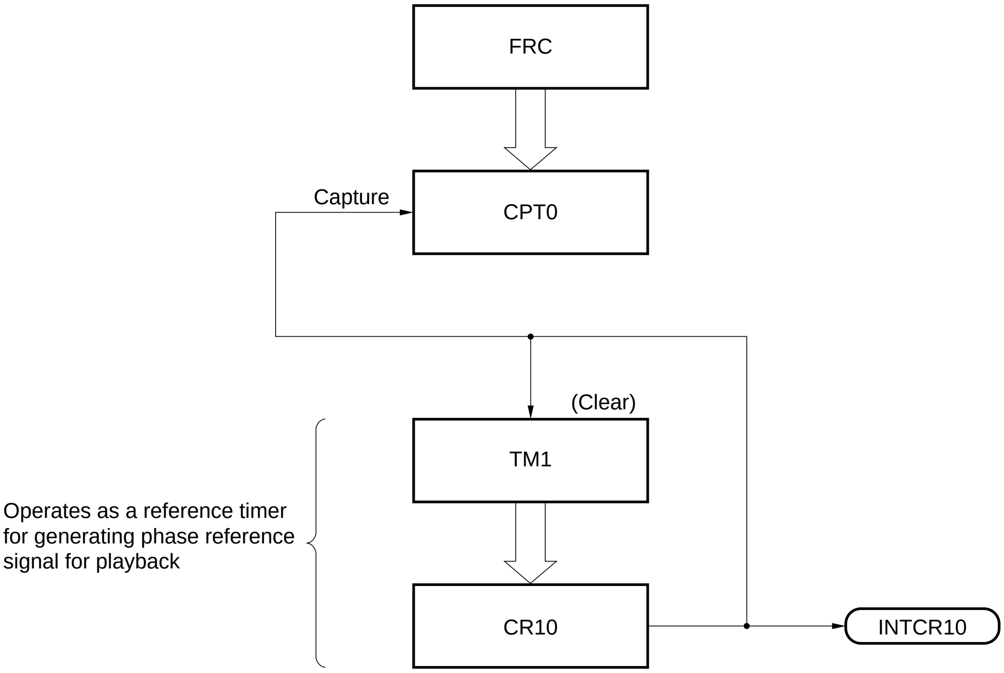
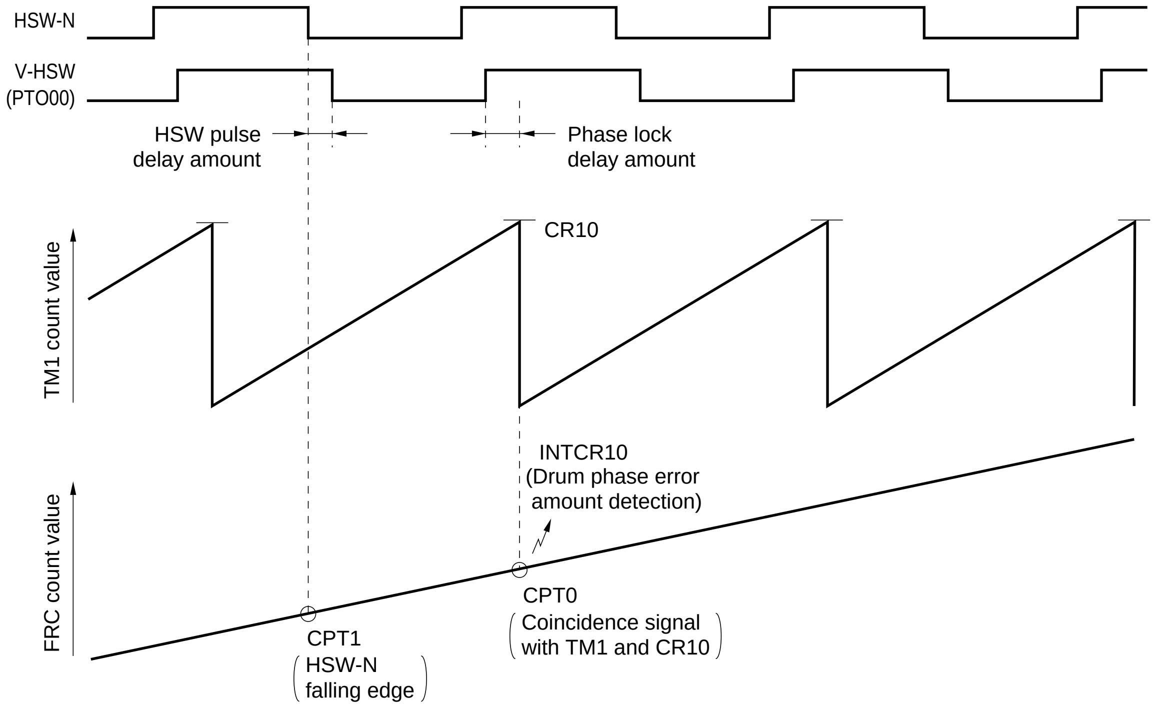

FRC
Capture
CPT0
(Clear)
TM1
Operates as a reference timer for generating phase reference signal for playback
CR10
INTCR10

Figure 5-25. Drum Phase Control Timing (for playback)



The drum phase error amount detection method is shown below.

The count value of free running counter (FRC) is stored in the capture register 1 (CPT1) at the falling edge of internal head switching signal (HSW-N).

INTCR10 interrupt is generated at the coincidence timing with the count value of timer 1 (TM1) and compare register 10 (CR10). The value of FRC is, at the same time, stored in CPT0.

The drum phase error amount E_{DP} is shown in the expression below. Figure 5-26 shows the method to set CPT0 and CPT1 capture trigger source.

$$E_{DP} = (\text{(CPT0 value)} - \text{(CPT1 value)}) - N_{DPL}$$

N_{DPL}, here, is the target value of drum phase control.

The count clock of the capture registers of CPT0 and CPT1 is 125 [ns] of FRC. However, since timer 0 (TM0) to generate head switching signal (V-HSW), which is the object of comparison, is 1 [μs], CPT1 is subtracted from CPT 0, and then the result is made 1/4 so that data can be handled in 16 bits.

The sampling clock cycle of drum phase error, hereafter, is calculated as 0.5 [μs].

The target value of the drum phase control N_{DPL} is the remainder of the subtraction between CPT0 and CPT1, therefore, calculated with the following expression.

$$N_{DPL} = \begin{pmatrix}\text{Video head}\\ \text{switching pulse}\\ \text{delay amount}\end{pmatrix} + \begin{pmatrix}\text{Delay amount}\\ \text{for half a frame}\\ \text{cycle}\end{pmatrix} + \begin{pmatrix}\text{Delay amount}\\ \text{for 6.5H}\end{pmatrix} + \begin{pmatrix}\text{Delay for } V_{SYNC}\\ \text{separation}\end{pmatrix}$$

Each value of the above expression is calculated here.

(1) The digital value equivalent to the head switching signal (V-HSW) delay amount

The digital value equivalent to the head switching signal (V-HSW) is calculated. The V-HSW delay is stored in compare register 00 (CR00) of timer 0 as the delay amount from HSW-N. The TM0 sampling clock frequency is twice as large as that of the drum phase error. Therefore, the value equivalent to V-HSW delay amount when counted with FRC is twice as large as the value set in CR00.

(2) The delay amount for half a frame cycle

The half of a frame cycle $T_F/2$ is stored in CR10, therefore:

$$CR10/2 = 16.68 \text{ [ms]}$$

The above value is counted with the sampling clock cycle 0.5 [μs] of drum phase error as follows:

$$\frac{\text{CR10/2}}{0.5\ [\mu\text{s}]} = \text{CR10}$$

(3) The delay amount for 6.5H

1H = 63.56 [μs], therefore, the time for 6.5 H is:

$$63.56 \times 6.5 = 413.14\ [\mu\text{s}]$$

Therefore, if counted with the drum phase error sampling clock:

$$\frac{413\ [\mu\text{s}]}{0.5\ [\mu\text{s}]} = 826$$

(4) The delay for V_{SYNC} separation

Drum control system for recording uses the vertical synchronizing signal V_{SYNC} as the phase reference signal. V_{SYNC}, here, is acquired by being separated from the composite synchronizing signal in order to make the phase control system program for playback and recording equal.

The delay time for V_{SYNC} separation is with the time for the separation is considered.

μPD784915 is equipped with digital noise elimination circuit, so that 13.5 [μs] (INTTM2.4 = 0) of the delay is the delay for V_{SYNC} separation.

If the time for 13.5 [μs] necessary for V_{SYNC} separation is counted with the sampling clock of the drum phase error, the value is as follows:

$$\frac{13.5\ [\mu\text{s}]}{0.5\ [\mu\text{s}]} = 27$$

From above, N_{DPL} is calculated as follows:

$$\begin{aligned}\text{NDPL} &= (\text{CR00} \times 2) + (\text{CR10}) + 826 + 27\\ &= (\text{CR00} \times 2) + (\text{CR10}) + 355\text{H}\end{aligned}$$

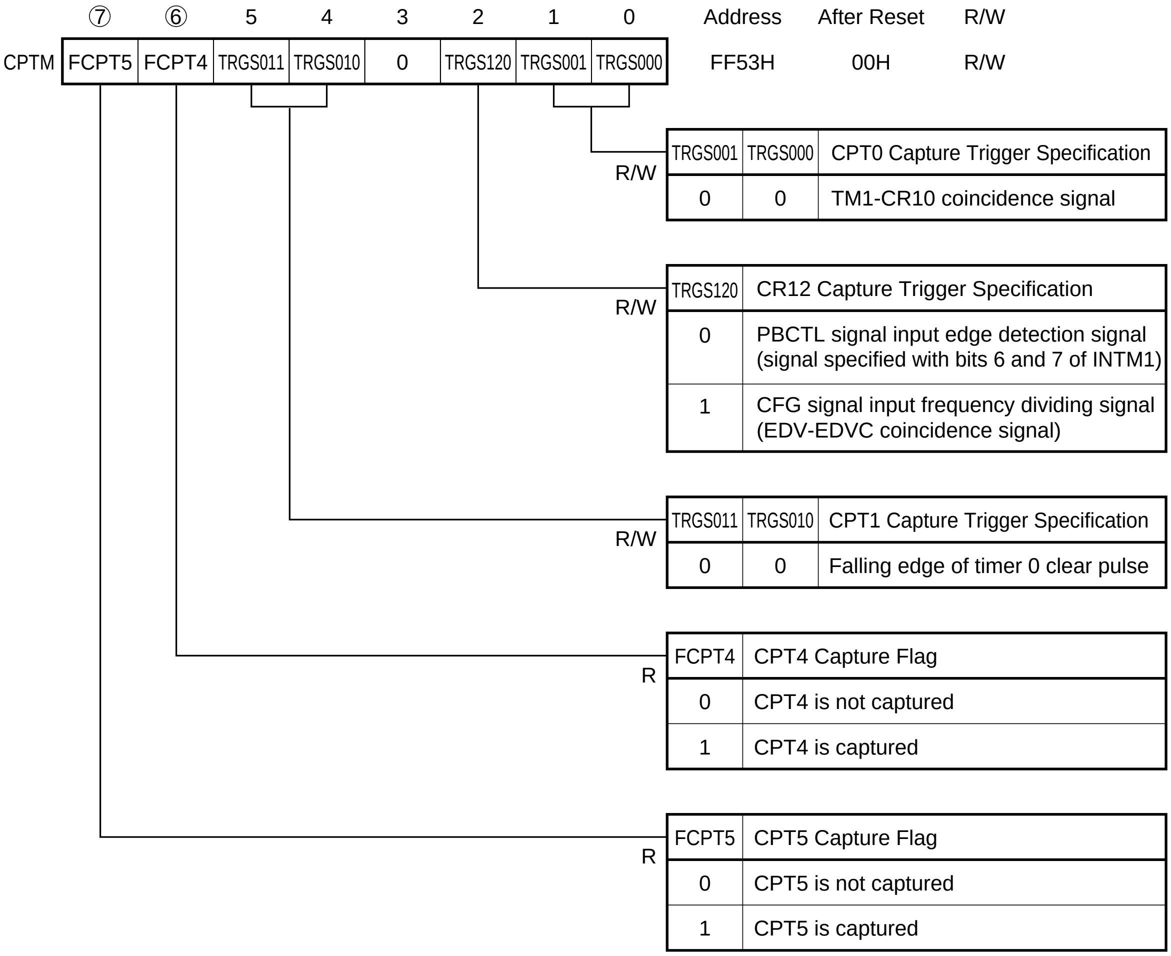
The drum phase error amount is calculated from the free running counter (FRC) value (CPT1) captured at the falling edge of the internal head switching signal (HSW-N) and the FRC value (CPT0) captured at the timer 1 (TM1) clear timing.

Therefore, the capture trigger selector of CPT0 is switched so that the CPT0 capture trigger source becomes the coincidence signal of the TM1 value and compare register 10 (CR10).

Timer 1 (TM1) is used as a reference timer which is cleared with frame cycle. The value set to CR10 of TM1 is as follows:

$$\text{CR10} = \frac{33.366\ [\text{ms}]}{1.0\ [\mu\text{s}]} = 33366 = 8256\text{H}$$

Figure 5-26. Capture Mode Register (CPTM) Format



	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
CPTM	FCPT5	FCPT4	TRGS011	TRGS010	0	TRGS120	TRGS001	TRGS000	FF53H	00H	R/W

R/W

TRGS001	TRGS000	CPT0 Capture Trigger Specification
0	0	TM1-CR10 coincidence signal

R/W

TRGS120	CR12 Capture Trigger Specification
0	PBCTL signal input edge detection signal (signal specified with bits 6 and 7 of INTM1)
1	CFG signal input frequency dividing signal (EDV-EDVC coincidence signal)

R/W

TRGS011	TRGS010	CPT1 Capture Trigger Specification
0	0	Falling edge of timer 0 clear pulse

R

FCPT4	CPT4 Capture Flag
0	CPT4 is not captured
1	CPT4 is captured

R

FCPT5	CPT5 Capture Flag
0	CPT5 is not captured
1	CPT5 is captured

5.6.3 Drum phase control for recording

For drum phase control for recording, the rotational phase of the drum motor is synchronized with vertical synchronizing signal externally input.

The point of phase lock is where the rising and falling edges of the head switching signal (V-HSW) are 6.5H before the vertical synchronizing signal, which complies with VHS standard. However, the time required for V_{SYNC} separation, as mentioned in the section about the control for playback, must be taken in consideration.

Figure 5-27 shows the use of the timer in drum phase control for recording. Figure 5-28 shows the drum phase control timing chart for recording.

When recording, the phase error amount is calculated from the free running counter (FRC) value (CPT1) captured at the falling edge of internal head switching signal (HSW-N) and the FRC value (CPT0) captured at the falling edge of the vertical synchronizing signal input from the CSYNCIN pin.

The phase error amount detection method is described below.

The method to capture the FRC value in CPT1 only at the falling edge of HSW-N is the same as for playback.

On the other hand, the CPT0 capture operation is performed when the vertical synchronizing signal is input to the CSYNCIN input pin, and TM1 is cleared simultaneously. In fact, the phase error detection is performed in frame cycle, so that TM1 clearance by inputting CSYNC is masked for the 90% time period of a frame cycle.

Compare register 13 (CR13) is used for the setting of the mask time.

Figure 5-29 shows the CPT0 and CPT1 capture trigger source setting method for recording.

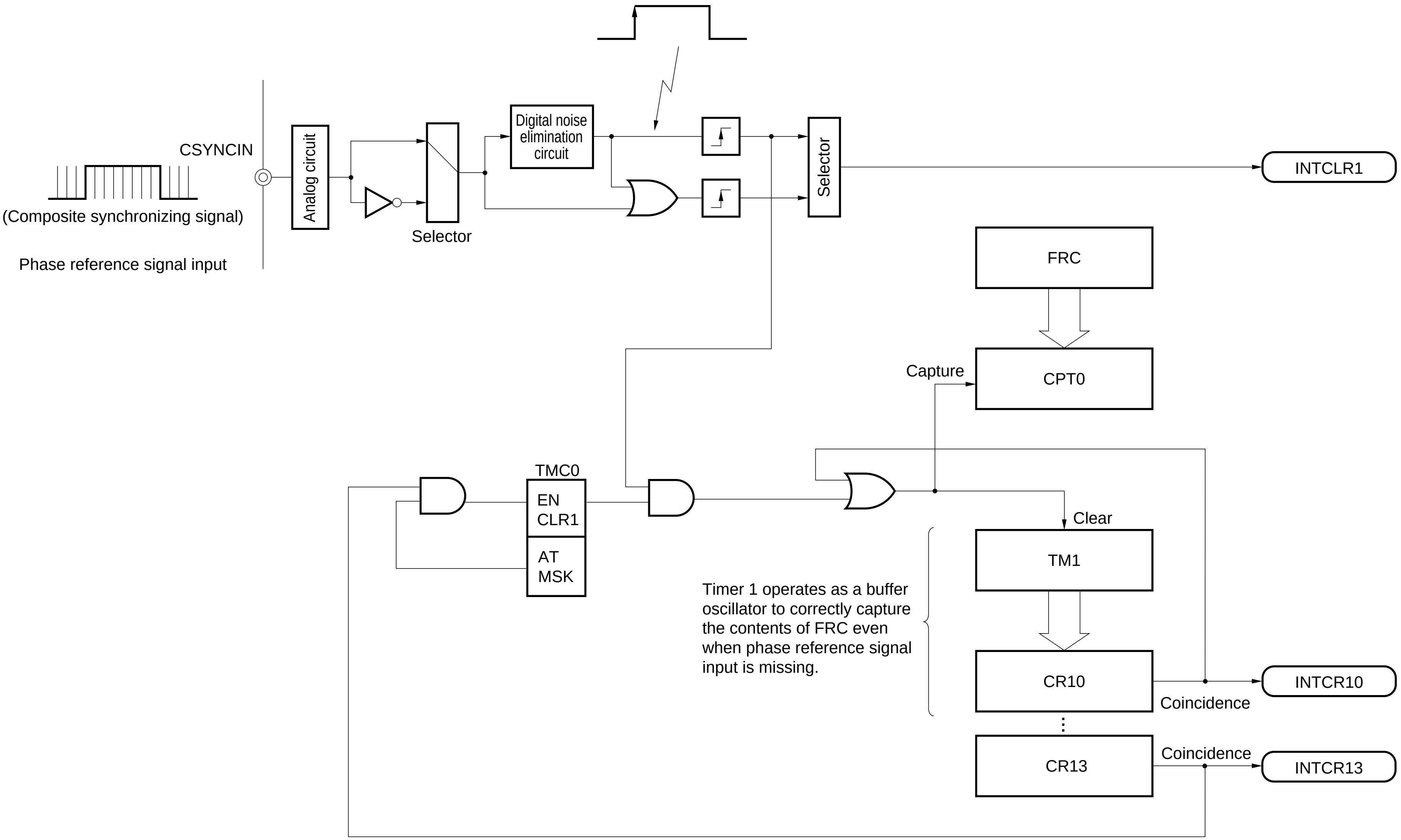
The phase error amount is detected in INTCLR1 interrupt processing. The phase error amount E_{DP} is calculated as follows:

$$E_{DP} = (\text{ (CPT0 value)} - \text{(CPT1 value) }) - N_{DPL}$$

N_{DPL}, here, is the difference between CPT0 and CPT1 when the drum phase control is locked, that is, the target value of the phase control.

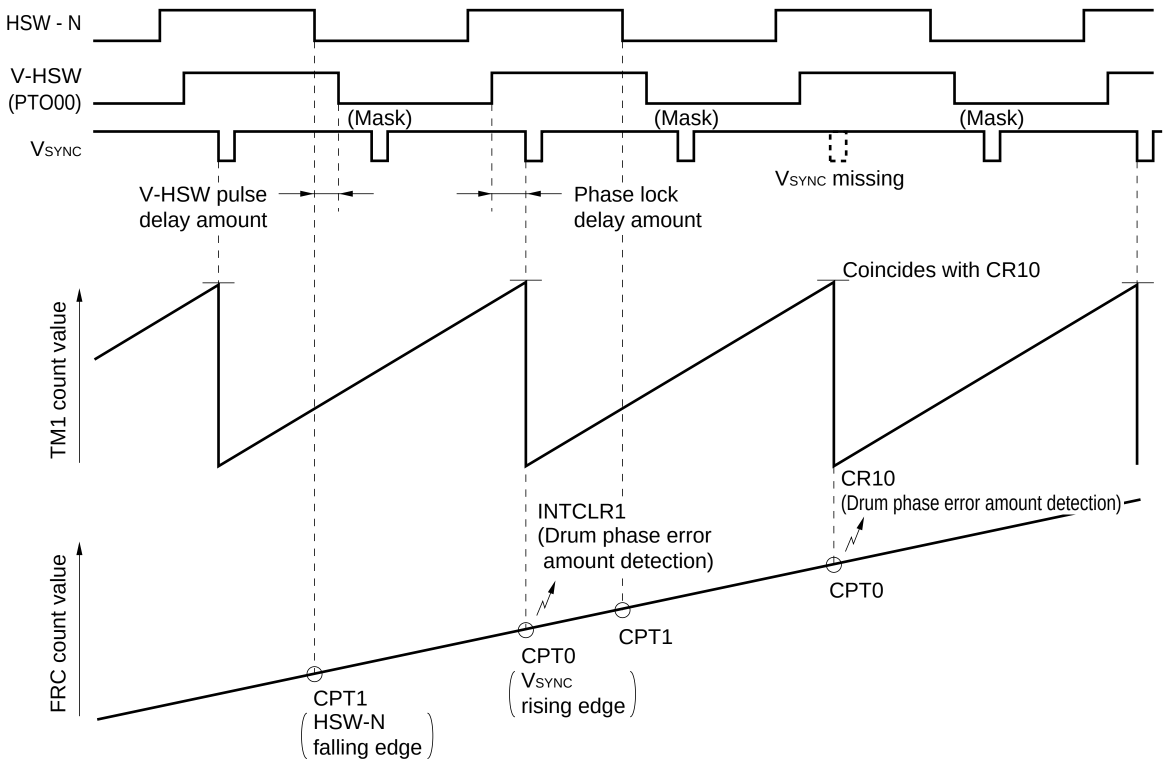
The difference between CPT0 and CPT1 when the phase is locked, is the sum of head switching signal delay amount, frame half cycle, VHS standard 6.5H delay, and the delay amount for V_{SYNC} separation.

Figure 5-27. Use of Timer for Drum Phase Control (for recording)



Timer 1 operates as a buffer oscillator to correctly capture the contents of FRC even when phase reference signal input is missing.

Figure 5-28. Drum Phase Control Timing (for recording)



Remark Phase lock delay amount: value determined by VCR standard

Figure 5-29. Capture Mode Register (CPTM) Format

	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
CPTM	FCPT5	FCPT4	TRGS011	TRGS010	0	TRGS120	TRGS001	TRGS000	FF53H	00H	R/W

R/W

TRGS001	TRGS000	CPT0 Capture Trigger Specification
1	0	TM1 Clear Signal

R/W

TRGS120	CR12 Capture Trigger Specification
0	PBCTL signal input edge detection signal (signal specified with bits 6 and 7 of INTM1)
1	CFG signal input frequency dividing signal (EDV-EDVC coincidence signal)

R/W

TRGS011	TRGS010	CPT1 Capture Trigger Specification
0	0	Falling edge of timer 0 clear pulse

R

FCPT4	CPT4 Capture Flag
0	CPT4 is not captured
1	CPT4 is captured

R

FCPT5	CPT5 Capture Flag
0	CPT5 is not captured
1	CPT5 is captured

The μPD784915 uses vertical synchronizing signal (V_{SYNC}) as phase reference signal. Therefore, digital noise elimination circuit is used to separate only vertical synchronizing signals from composite synchronizing signals. In this application example, the amount of time required for V_{SYNC} of digital noise elimination circuit is 13.5 [μs].

The count value of drum phase error for 6.5H sampling clock is 826.

The count value for 13.5 [μs] required for V_{SYNC} separation is shown in the following expression:

$$\frac{13.5\ [\mu s]}{0.5\ [\mu s]} = 27$$

Therefore, the target value is represented with the expressions as follows:

$$N_{DPL} = \left(\begin{array}{c}\text{Digital value}\\ \text{equivalent to}\\ \text{video head}\\ \text{switching pulse}\\ \text{delay amount}\end{array}\right) + \left(\begin{array}{c}\text{Delay amount}\\ \text{for half a frame}\\ \text{cycle}\end{array}\right) + \left(\begin{array}{c}\text{Delay amount}\\ \text{for 6.5H}\end{array}\right) + \left(\begin{array}{c}\text{Delay for } V_{SYNC}\\ \text{separation}\end{array}\right)$$

$$= (CR00 \times 2) + (CR10) + 826 + 27$$

$$= (CR00 \times 2) + (CR10) + 355H$$

This expression is the same as that for phase error amount for playback.

5.7 Capstan Speed Control

Delay of capstan speed error amount is carried out, as well as drum speed control, by capturing the free running counter (FRC) value at the capstan FG signal input edge.

The capstan FG signal (CFG) is input to the DFGIN input pin. INTCPT3 interrupt request occurs simultaneously with the capture of the FRC value to CPT3 at the CFG edge input.

The difference from drum speed control is that CFG frequency f_{CF} varies according to the tape running mode. In this set, the CFG frequency for normal playing back is as follows:

- SP mode : 1080.00 [Hz]
- LP mode : 540.00 [Hz]
- EP mode : 360.00 [Hz]

In order to equalize error detection gain in the servo system according to each running mode, CFG is divided with the 8-bit event divider control register (EDVC) incorporated in the CFGIN pin input of the Super Timer Unit.

Since this counter operates as the event divider of the DFGIN pin input pulse, the detection cycle in the SP mode becomes the same as that in the EP mode if CFG is divided by one third.

Figure 5-30 shows the capstan speed detection method. Figure 5-31 shows the capstan speed control timing chart.

The capstan speed error ECV is calculated in the INTCPT3 interrupt request processing routine.

The expression is as follows:

$$\begin{aligned} \Delta N_{CV}{}^{n} &= \Delta N_{CV}{}^{n} - N_{CV}{}^{n} \\ E_{CV}{}^{n} &= N_{CVL}{}^{n} - \Delta N_{CV}{}^{n} \end{aligned}$$

N_{CVL}, here, is the target value of capstan speed control.

The CFG frequency f_{CF} in the EP mode is as follows:

$$\begin{aligned} f_{CF} &= 360.00 \text{ [Hz]} \\ f_{CFEP} &= 360.00 \text{ [Hz]} \end{aligned}$$

Therefore, it becomes the CFG frequency in the SP mode, and the CFG cycle f_{CF} is calculated with the following expressions:

$$\begin{aligned} f_{CFSP}/3 &= 360.00 \\ f_{CF}/3 &= 239.7602093 \text{ [Hz]} \\ T_{CF} &= 2.7778 \text{ [ms]} \end{aligned}$$

The time interval counted by the free running counter (FRC) becomes the target value N_{CFL} of capstan FG signal (CFG). Since the FRC count pulse cycle (T_{FRC}) is 125 [ns], N_{CFL} is as follows:

$$\boxed{N_{CFL} = \frac{2.778\ [\text{ms}]}{125\ [\text{ns}]} = 22222 = 56\text{CEH}}$$

The meaning of the signs for capstan speed error amount E_{CV} is shown below:

(1) When CFG cycle is longer than the target value...–
(when the rotation of capstan motor is slow)

(2) When CFG cycle is shorter than the target value...+
(when the rotation of capstan motor is fast)

Figure 5-30. Capstan Speed Detection Method

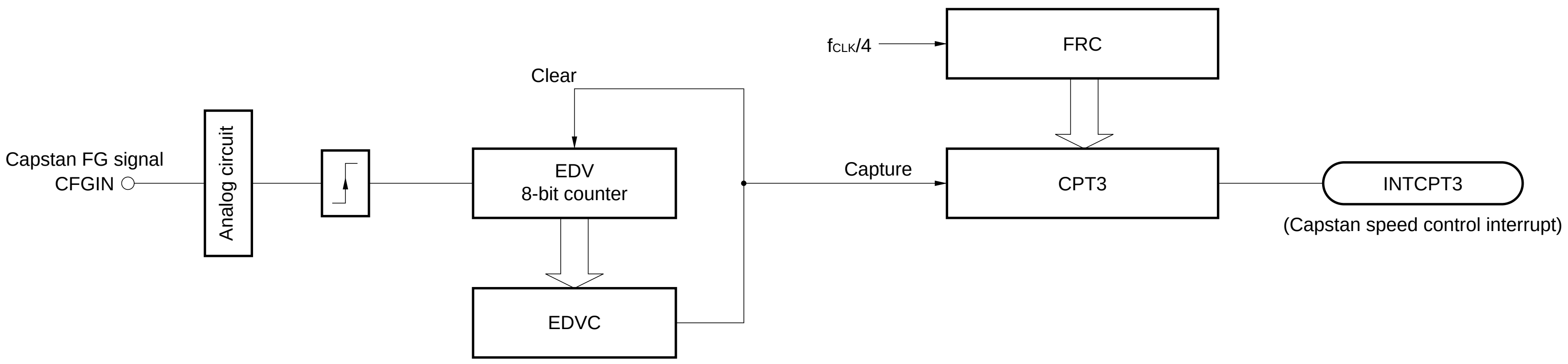
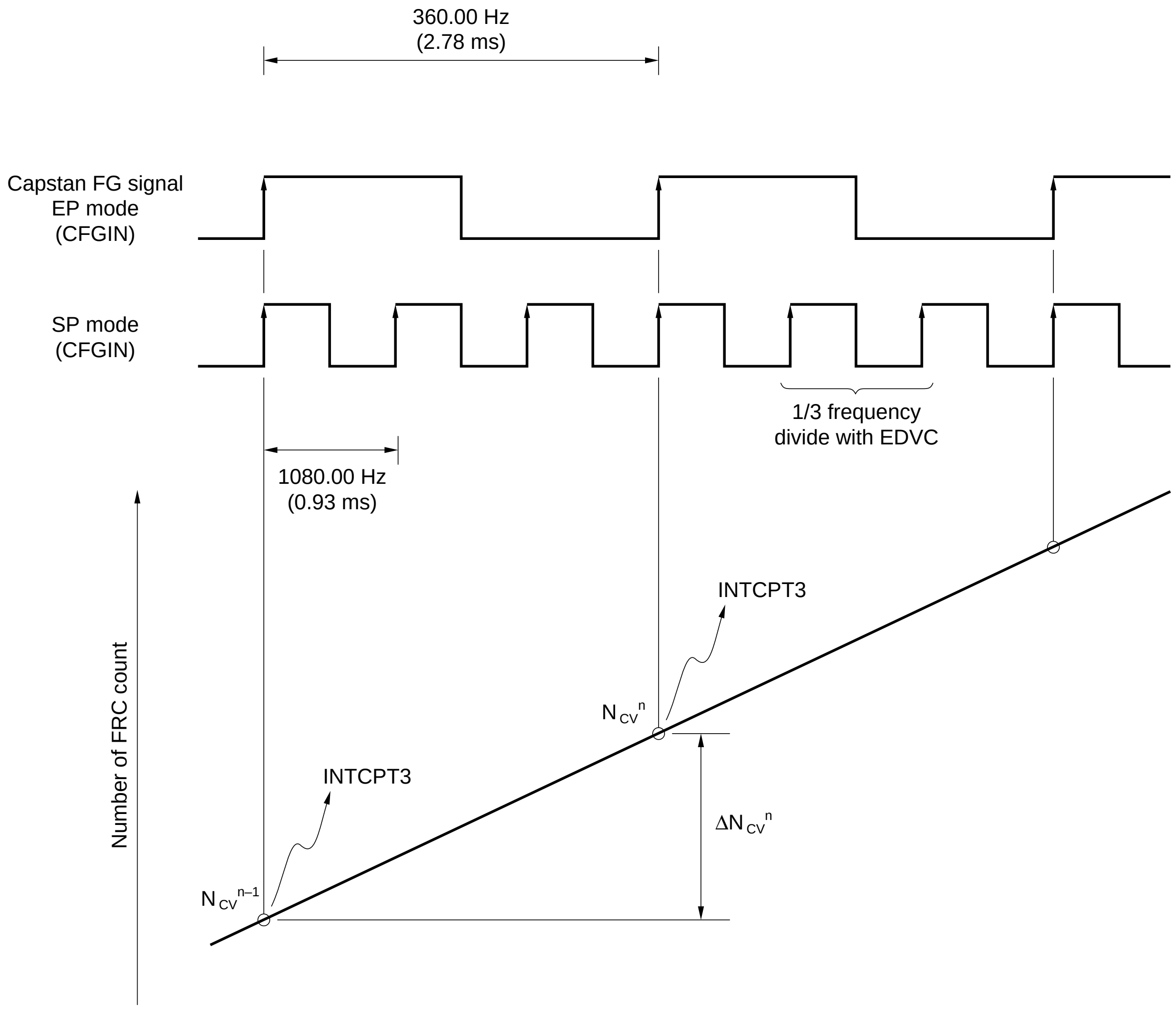


Figure 5-31. Capstan Speed Control Timing



5.8 Capstan Phase Control

The phase detection of the capstan motor is performed by compare register 12 (CR12), and capstan phase control is performed by the INTCR12 interrupt routine.

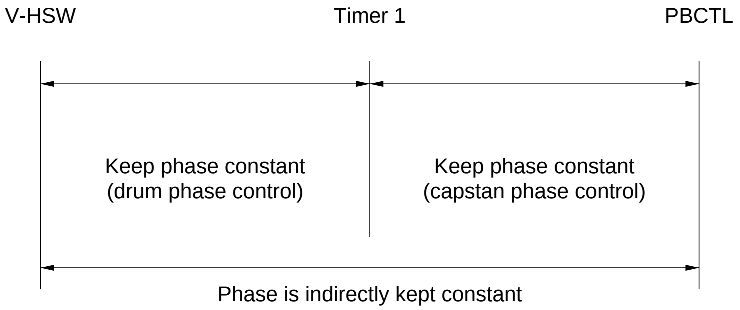
The control method differs for playback and recording. The control method for each case is explained below.

5.8.1 Capstan phase control for playback

The purpose of capstan phase control for playback is to keep the phase relation constant between the playback control signal (PBCTL) and the head switching signal (V-HSW) acquired when playing back.

The relation between timer 1 (TM1), which is the reference timer in drum phase control, and V-HSW is already kept constant (refer to **5.6.2 Drum phase control for playback**). Therefore, the phase relation between V-HSW and PBCTL is indirectly kept constant by keeping the phase relation between TM1 and PBCTL constant (refer to Figure 5-32).

Figure 5-32. Model of Capstan Phase Control



In capstan phase control for playback, the selector is selected so that PBCTL signal is set to CR12 of timer 1. Figure 5-33 shows the capstan phase error detection method for playback. Figure 5-34 shows how to set the capture trigger source of CR12 for playback.

In capstan phase control, the lock point is the point tilted for the amount of time from video head to control head (x value correction amount). The x value correction amount differs according to the VCR sets and the tape running mode.

Figures 5-35 to 5-37 show the phase control timing charts. The phase error amount E_{CP} is calculated from the following expression.

$$\boxed{\begin{aligned} E_{CP} &= \text{(Value captured by PBCTL signal)} - N_{CPL} \\ &= \text{(CR12 value)} - N_{CPL} \end{aligned}}$$

N_{CPL} : capstan phase control target value

The capstan phase control target value (N_{CPL}), here, is the TM1 value captured in CR12 when the phase is locked. Taking it into account that the phase between the video head position (V-HSW timing) and the reference timer is 6.5H, N_{CPL} is calculated as follows:

$$N_{CPL} = \frac{\text{(x value correction amount)} - 6.5\text{H}}{T_{TM1}} + \text{tracking adjustment amount}$$

$$= \frac{\text{(x value correction amount)} - 6.5\text{H}}{8/f_{CLK}} + \text{tracking adjustment amount}$$

$$= \frac{\text{(x value correction amount)} - 6.5\text{H} \times 63.55\ [\mu\text{sec}]}{8/8[\text{MHz}]} + \text{tracking adjustment amount}$$

If the phase control range is set equally for both advance/delay directions centered in the phase lock, the amount of time before the phase lock is shortened. In other words, in the condition of Figure 5-36, the distance to the lock point is shorter if the condition is considered as advanced rather than delayed so that the amount of time before the phase lock is shorter. Concretely, the $N_F/2$ range (N_F: the full count value of timer 1: CR10 set value) is regarded as it is, centered in the locking point, and the value corrected by adding N_F to (or subtracting N_F from) the value captured is used for the other range.

In addition, digital filter arithmetic is performed to the capstan phase error amount acquired here so that the result is used for the phase control.

Figure 5-33. Capstan Phase Error Detection Method (for playback)

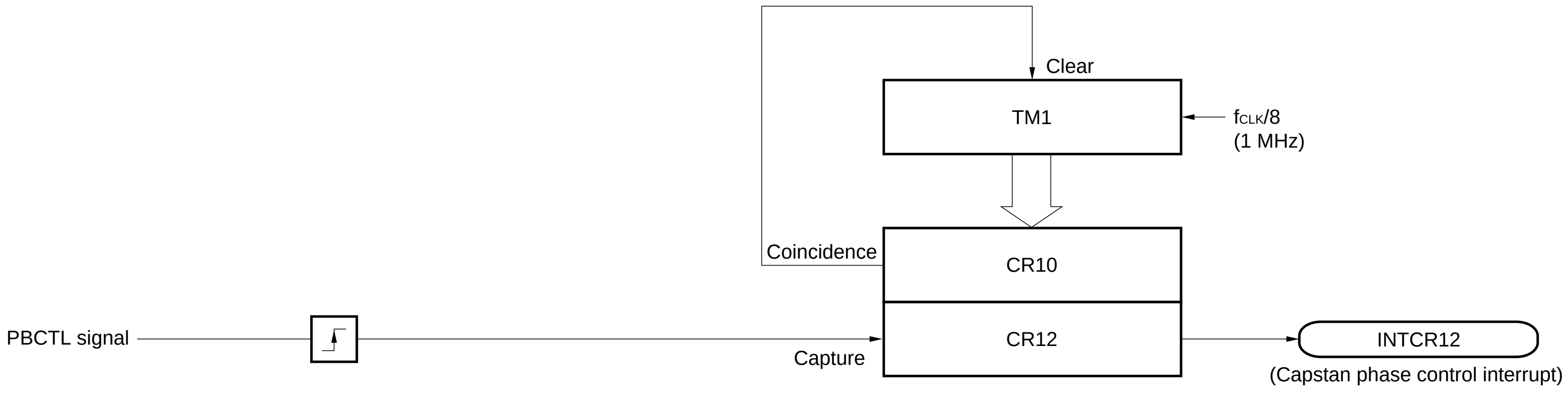


Figure 5-34. Capture Mode Register (CPTM) Format

	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
CPTM	FCPT5	FCPT4	TRGS011	TRGS010	0	TRGS120	TRGS001	TRGS000	FF53H	00H	R/W

R/W

TRGS001	TRGS000	CPT0 Capture Trigger Specification
0	0	TM1-CR10 coincidence signal
0	1	CSYNC signal input edge detection signal
1	0	TM1 clear signal
1	1	OR of TM1-CR10 coincidence signal and CSYNC signal input edge detection signal

R/W

TRGS120	CR12 Capture Trigger Specification
0	PBCTL signal input edge detection signal

R/W

TRGS011	TRGS010	CPT1 Capture Trigger Specification
0	0	Falling edge of timer 0 clear pulse
0	1	Rising edge of timer 0 clear pulse
1	0	Setting prohibited
1	1	Both falling/rising edge of timer 0 clear pulse

R

FCPT4	CPT4 Capture Flag
0	CPT4 is not captured
1	CPT4 is captured

R

FCPT5	CPT5 Capture Flag
0	CPT5 is not captured
1	CPT5 is captured

Figure 5-35. Capstan Phase Control Timing (playback mode, phase locked)

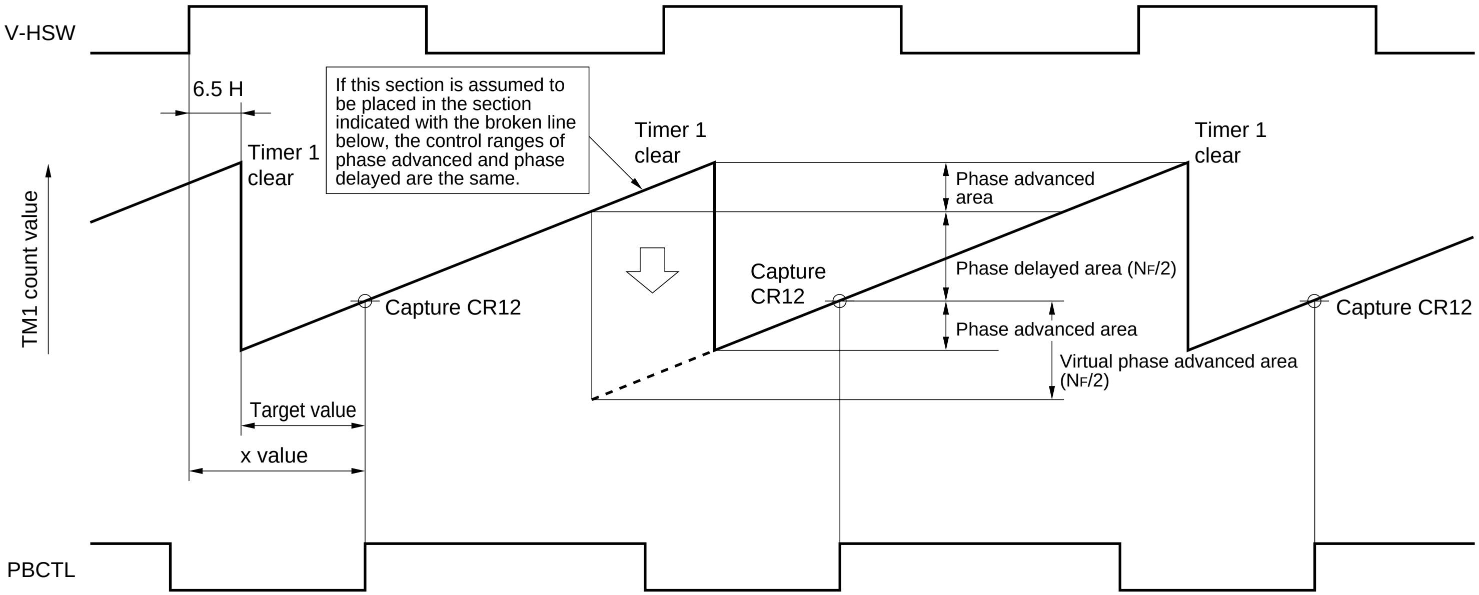

V-HSW
6.5 H
If this section is assumed to be placed in the section indicated with the broken line below, the control ranges of phase advanced and phase delayed are the same.
Timer 1 clear
Timer 1 clear
Timer 1 clear
TM1 count value
Phase advanced area
Phase delayed area (NF/2)
Phase advanced area
Virtual phase advanced area (NF/2)
Capture CR12
Capture CR12
Capture CR12
Target value
x value
PBCTL

Figure 5-36. Capstan Phase Control Timing (playback mode, phase delayed)

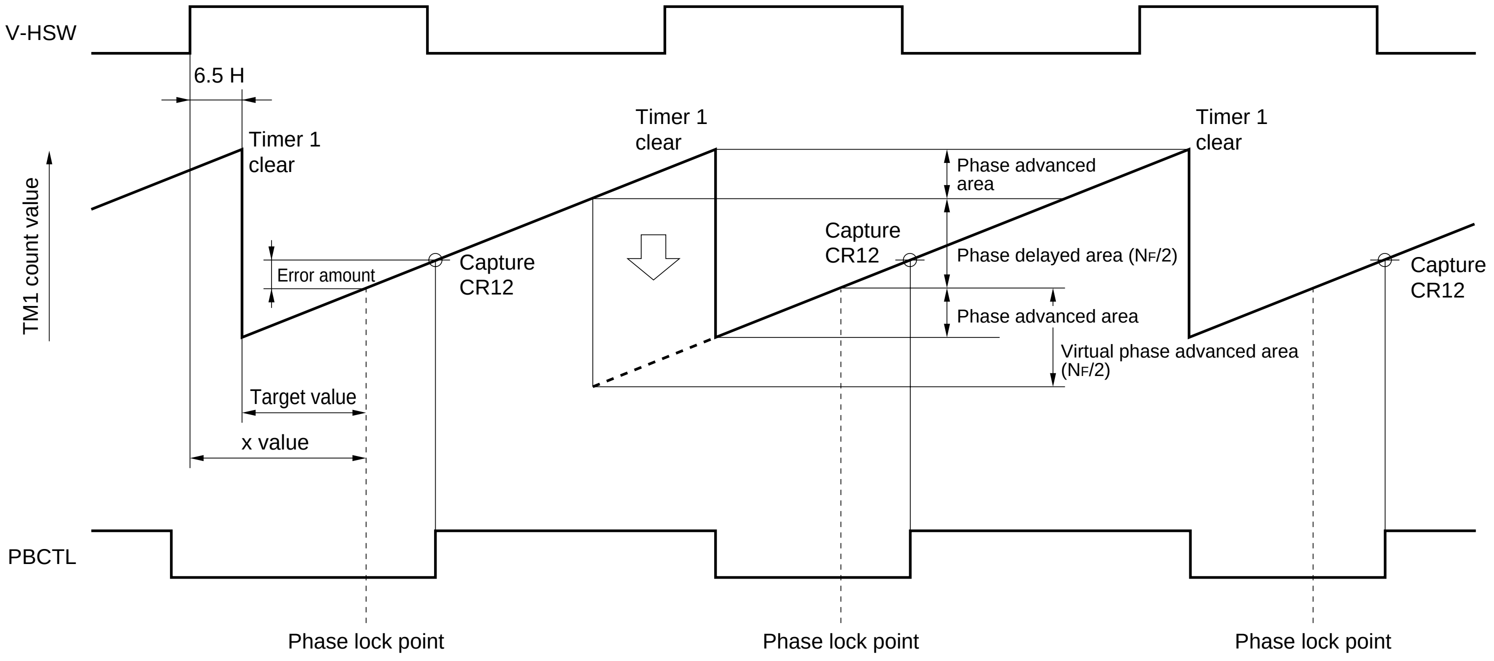
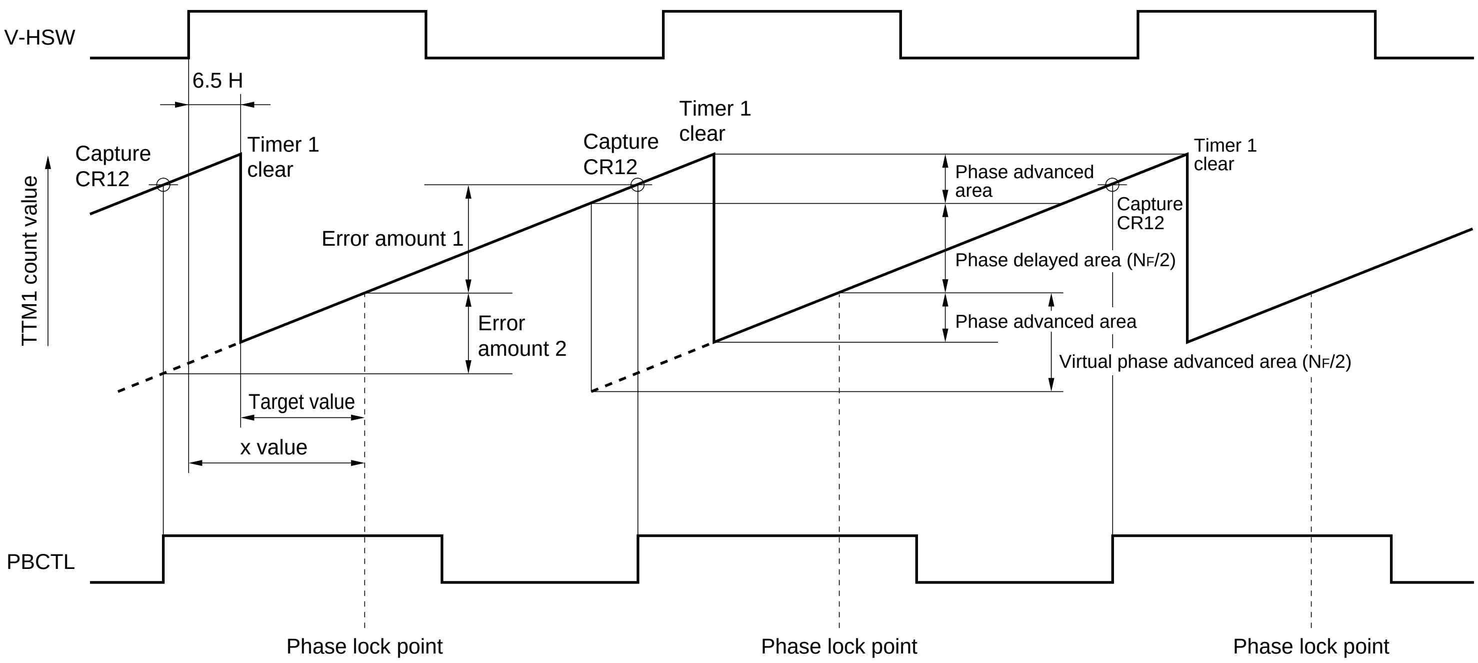


Figure 5-37. Capstan Phase Control Timing (playback mode, phase advanced)



Remark Error amount 1 > error amount 2, therefore, error amount 2 is used as the error amount.

5.8.2 Capstan phase control for recording

The capstan control for recording is performed by dividing capstan FG signal (CFG). As long as the capstan motor is steadily rotating while recording, it is not necessary to consider absolute phase.

Figure 5-38 shows the capstan phase error detection method for recording. Figure 5-39 shows the method of setting the capture trigger source of compare register 12 (CR12) for recording. The capstan phase control for recording, as well as for playback, uses the CR12 and INTCR12 interrupts. Although a value is captured in CR12 every time CFG is input, CFG has to be divided because only one captured value is needed for one frame. In actuality, the input is not divided but interrupt is divided using the macro service counter mode, and the CR12 value when a vectored interrupt is generated is used. The number of frequency division is the same as the FG wave number (in EP mode). CFG input is triple divided by event divider control register (EDVC) in SP mode. Therefore, the required value is acquired by dividing the interrupt for the capstan FG wave number in EP mode.

Figure 5-40 shows the phase control timing chart.

Capstan phase error amount is calculated in the same way as for playback.

$$\begin{aligned} E_{CP} &= \text{(Captured value by CFG frequency dividing signal)} - N_{CPL} \\ &= \text{(CR12 value)} - N_{CPL} \end{aligned}$$

Remark N_{CPL}: capstan phase control target value

However, since there is not an absolute phase for recording, target value N_{CPL} can be any value.

Capstan phase control range is determined in the same way as for playback.

Further, digital filter arithmetic is performed to the capstan phase error amount acquired from the above calculation, and the result is used for phase control.

Figure 5-38. Capstan Phase Error Detection Method (for recording)

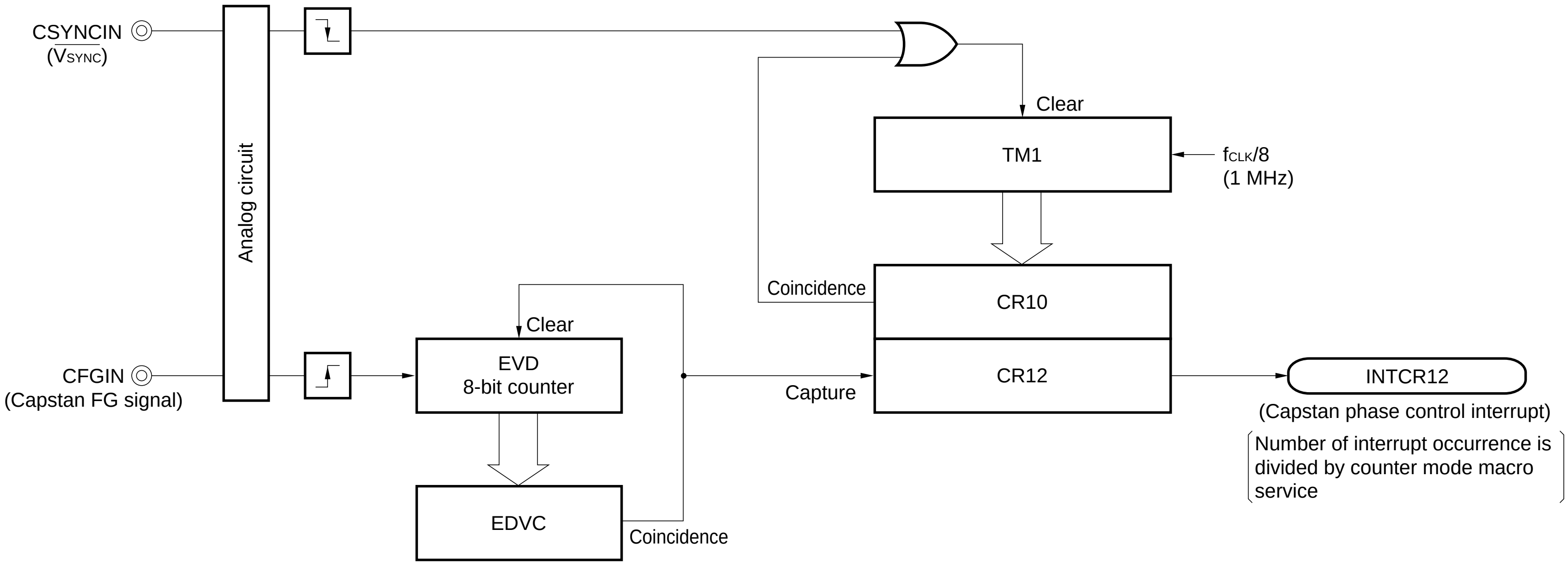


Figure 5-39. Capture Mode Register (CPTM) Format

	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
CPTM	FCPT5	FCPT4	TRGS011	TRGS010	0	TRGS120	TRGS001	TRGS000	FF53H	00H	R/W

R/W

TRGS001	TRGS000	CPT0 Capture Trigger Specification
0	0	TM1-CR10 coincidence signal
0	1	CSYNC signal input edge detection signal
1	0	TM1 clear signal
1	1	OR of TM1-CR10 coincidence signal and CSYNC signal input edge detection signal

R/W

TRGS120	CR12 Capture Trigger Specification
1	CFG signal input dividing signal (EDV-EDVC coincidence signal)

R/W

TRGS011	TRGS010	CPT1 Capture Trigger Specification
0	0	Falling edge of timer 0 clear pulse
0	1	Rising edge of timer 0 clear pulse
1	0	Setting prohibited
1	1	Both falling/rising edges of timer 0 clear pulse

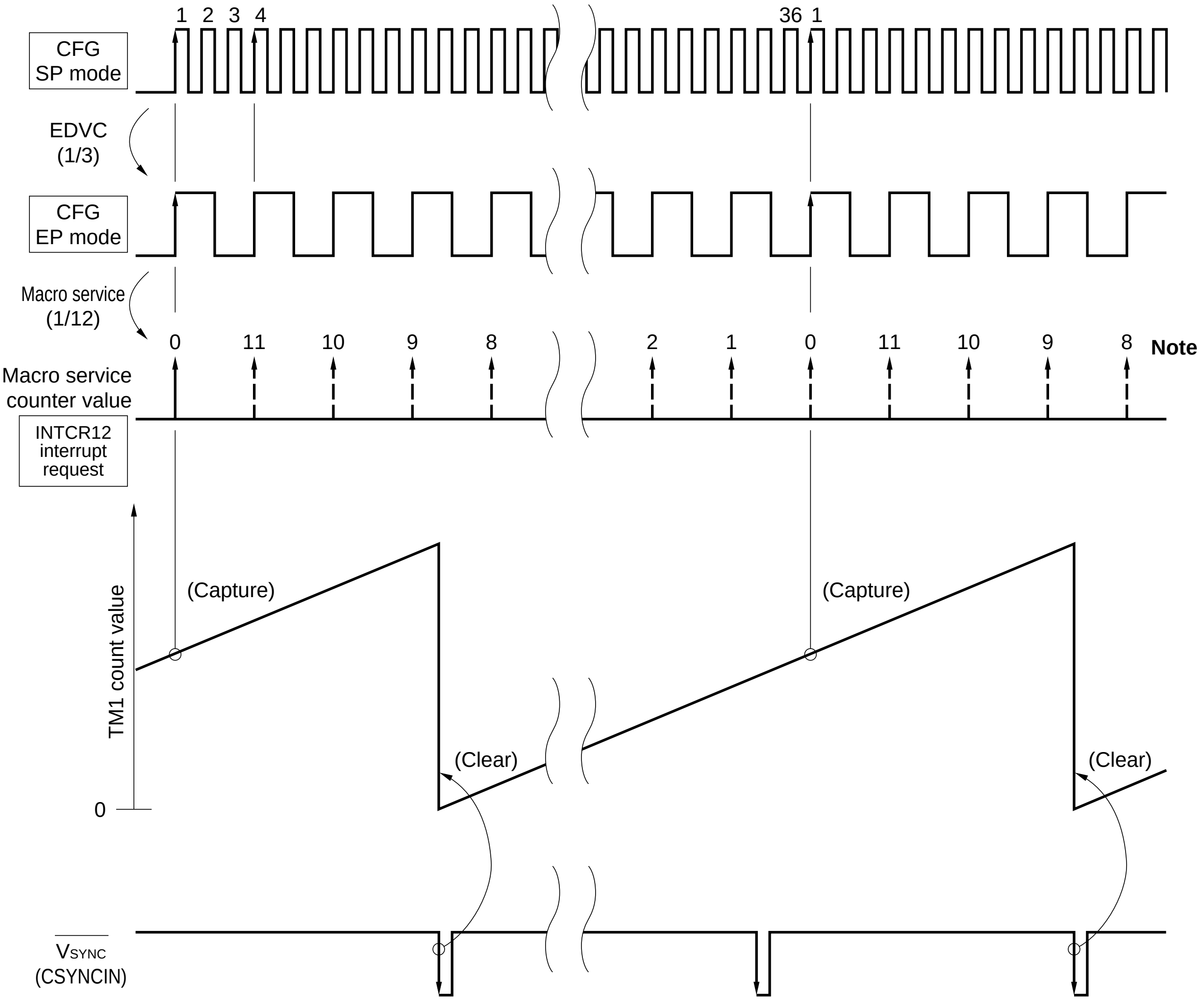
R

FCPT4	CPT4 Capture Flag
0	CPT4 is not captured
1	CPT4 is captured

R

FCPT5	CPT5 Capture Flag
0	CPT5 is not captured
1	CPT5 is captured

Figure 5-40. Capstan Phase Control Timing (for recording)



Note The arrows have the following meanings.

⟶ (Solid line) : vectored interrupt

– – → (Broken line): macro service

5.9 Recording Control Signal Generation

Recording control signal (RECCTL) is a signal synchronized with head switching signal (V-HSW) and recorded on the control track for recording. The RECCTL cycle is equal to T_F and the duty is normally 60% (27.5% for index signal, may become other duty).

RECCTL rising timing t_{RECR} is the point tilted for the amount of time from video head to control head (x value correction amount).

The x value correction amount differs according to the VCR sets and the tape running mode.

The RECCTL write timing is calculated as follows:

(1) [RECCTL rising timing (t_{RECR})]

RECCTL rising timing (t_{RECR}) is the point tilted for the amount of time from the video head position to control head (x value correction amount). The video head position has the phase tilted for 6.5H from the reference timer. Therefore, t_{RECR} is represented as follows:

$$t_{RECR} = \text{(x value correction amount)} - 6.5H - \tau d$$

Remark τd : digital noise elimination circuit (V_{SYNC} separation) delay time ($80/f_{CLK}$ or $128/f_{CLK}$)
0 if not using digital noise elimination circuit

(2) RECCTL falling timing (t_{RECF})

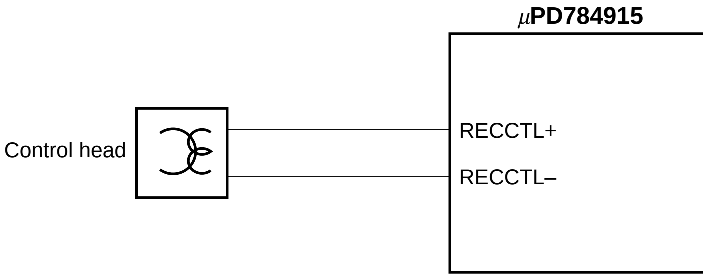
RECCTL falling timing (t_{RECF}) is the point tilted for 60% of the frame frequency from rising timing.

$$\begin{aligned} t_{RECF} &= t_{RECR} + \text{(60\% of frame frequency)} \\ &= [\text{(x value correction amount)} - 6.5H - \tau d] + (T_F \times 0.6) \end{aligned}$$

Remark τd : digital noise elimination circuit (V_{SYNC} separation) delay time ($80/f_{CLK}$ or $128/f_{CLK}$)
0 if not using digital noise elimination circuit
T_F : TV broadcast frame frequency (33.36 msec: NTSC)

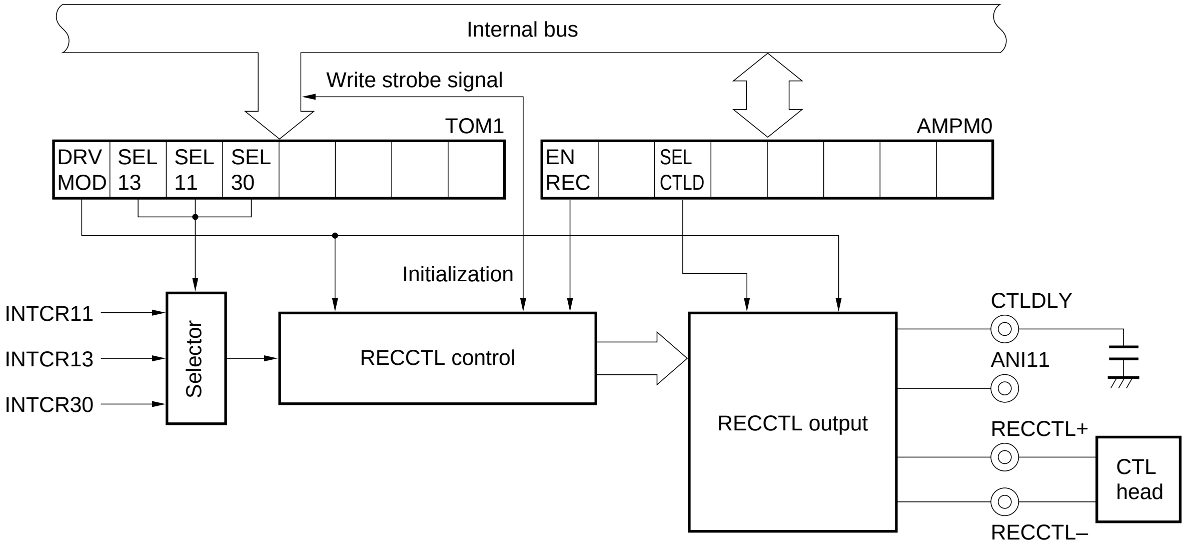
First, connect directly the μPD784915 and control head as shown in Figure 5-41 and set registers so that RECCTL write circuit is used.

Figure 5-41. Connection of μPD784915 and Control Head



The recording control signal (RECCTL) driver of the μPD784915 has the REC mode, which is used for RECCTL signal write. Figure 5-42 shows RECCTL driver configuration.

Figure 5-42. RECCTL Driver Block Diagram



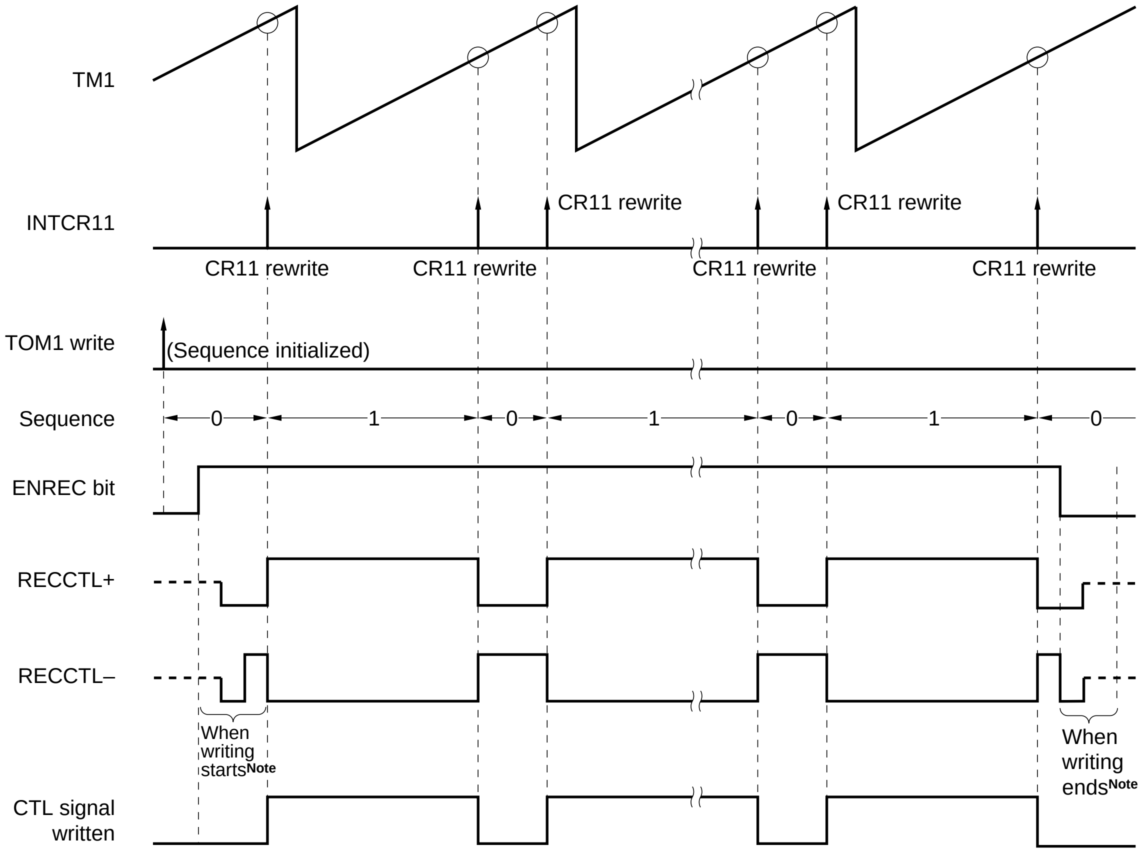
The REC mode sequence of RECCTL driver operates RECCTL+ pin and RECCTL– pin as shown in Table 5-2. Therefore, RECCTL signal write is realized taking only the interrupt occurrence timing, which is a trigger signal, into consideration.

Table 5-2. RECCTL Driver REC Mode Sequence

Sequence	RECCTL+	RECCTL–
0	Low level	High level
1	High level	Low level

Figure 5-43 shows an example of RECCTL signal writing operation timings.

Figure 5-43. Example of RECCTL Signal Writing Operation Timings



Note R/W to TOM1 register is not executed until approximately 800 μs from the setting of ENREC = 1 (start of REC mode), or ENREC = 0 (end of REC mode).

Caution Keep CTL amplifier in operation (ENCTL (AMPC.1) = 1) even while REC driver is operating.

The case using only compare register 11 (CR11) as a register for setting timing is explained here.

Set Timer 1 output mode register (TOM1) to use CR11 as shown in Figure 5-44. Then, write the value corresponding to rising timing to CR11. When timer register 1 (TM1) coincides with CR11, the rising edge is recorded and INTCR11 interrupt occurs. Rewrite the value to the value corresponding to the falling timing. And the falling edge is recorded at the next coincidence of TM1 and CR11, then write again the value corresponding to the rising timing to CR11. By repeating this procedure, RECCTL can be recorded.

Figure 5-45 shows the timing chart.

Figure 5-44. Timer 1 Output Mode Register (TOM1) Format

	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TOM1	DRVMOD	SEL13	SEL11	SEL30	MOD111	MOD110	MOD101	MOD100	FF5AH	80H	R/W

R/W

MOD101	MOD100	PTO10 Output Mode Specification
0	0	General purpose output mode
0	0	Setting prohibited
1	1	Delay pulse output mode 1
1	1	Delay pulse output mode 2

W

MOD111	MOD110	PTO11 Output Mode Specification
0	0	General purpose output mode
0	0	Setting prohibited
1	0	Delay pulse output mode 1
1	1	Delay pulse output mode 2

W

SEL30	RECCTL Write Circuit Operation Trigger Setting
0	TM3-CR30 coincidence signal is not selected as a trigger
1	TM3-CR30 coincidence signal is selected as a trigger

W

SEL11	RECCTL Write Circuit Operation Trigger Setting
1	TM1-CR11 coincidence signal is selected as a trigger

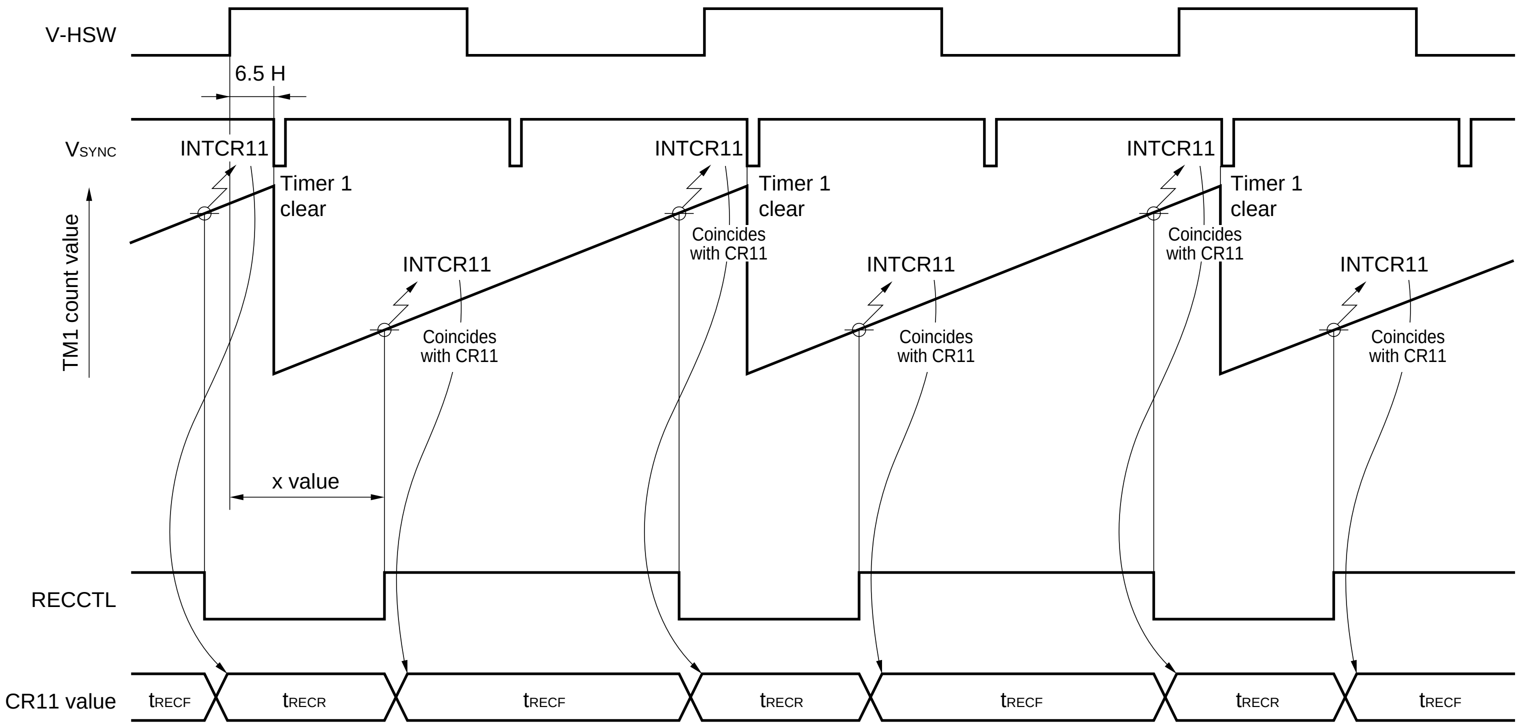
W

SEL13	RECCTL Write Circuit Operation Trigger Setting
0	TM1-CR13 coincidence signal is not selected as a trigger
1	TM1-CR13 coincidence signal is selected as a trigger

W

DRVMOD	RECCTL Write Circuit Operation Mode Setting
0	REC mode

Figure 5-45. RECCTL Write Timing Using CR11



5.10 Quasi Vertical Synchronizing Signal (Quasi-V_{SYNC}) Generation

The method to generate quasi vertical synchronizing signal (quasi-V_{SYNC}) for special playback is explained.

There are several types of wave forms for quasi-V_{SYNC} depending on the signal processing circuit to be used. The method to output the wave form shown in Figure 5-48 is explained here.

The wave form shown in Figure 5-46 requires not only "H" and "L" but also "M" (middle level) outputs.

Real-time output port RTP80 incorporated with the μPD784915 is used, for this port can output "H", "L", and "Hi-Z". For the middle level, the level of Hi-Z output is set with external pull-up and pull-down resistors (refer to **Figure 5-47**).

RTP80 can superimpose H_{SYNC} pulse during Hi-Z period, so that H_{SYNC} does not need to occur for every H_{SYNC}. The procedure is shown below:

<1> Set P80 in real time output port mode. And select TM0-CR02 coincidence signal as the RTP8 output trigger.

<2> Set H_{SYNC} output timing (rising (Figure 5-48 <1>)) to CR02. And set 00010001B (superimpose high-level H_{SYNC} to Hi-Z) to P8L.

<3> The wave form with Hi-Z that have high-level HSYNC superimposed is output from P80 pin by the coincidence of TM0 and CR02. INTCR02 occurs simultaneously.

<4> Set V_{SYNC} output timing (rising (Figure 5-48 <2>)) to CR02 by the INTCR02 interrupt routine. Set 00000001B (high-level output) to P8L.

<5> High level is output from P80 pin by the coincidence of TM0 and CR02. INTCR02 occurs simultaneously.

<6> Set V_{SYNC} output timing (falling (Figure 5-48 <3>)) by the INTCR02 interrupt routine. Set 00000000B (low-level output) to P8L.

<7> Low level is output from P80 pin by the coincidence of TM0 and CR02. INTCR02 occurs simultaneously.

<8> Set H_{SYNC} output timing (rising (Figure 5-48 <1>)) to CR02 by the INTCR02 interrupt routine. Set 00010001B (superimpose high-level H_{SYNC} to Hi-Z) to P8L.

<9> Go back to <3> (repeat this procedure).

Figure 5-46. Quasi-V_{SYNC} Waveform

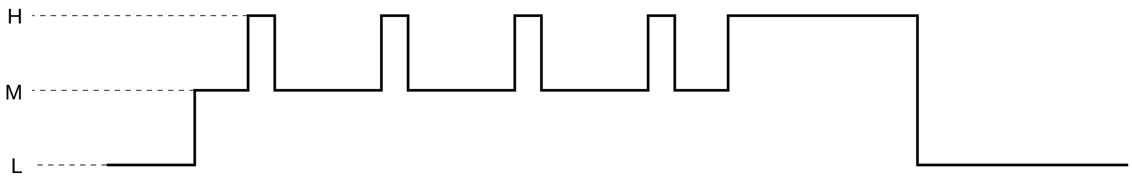
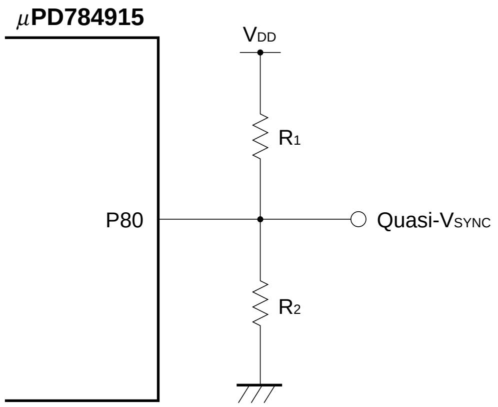
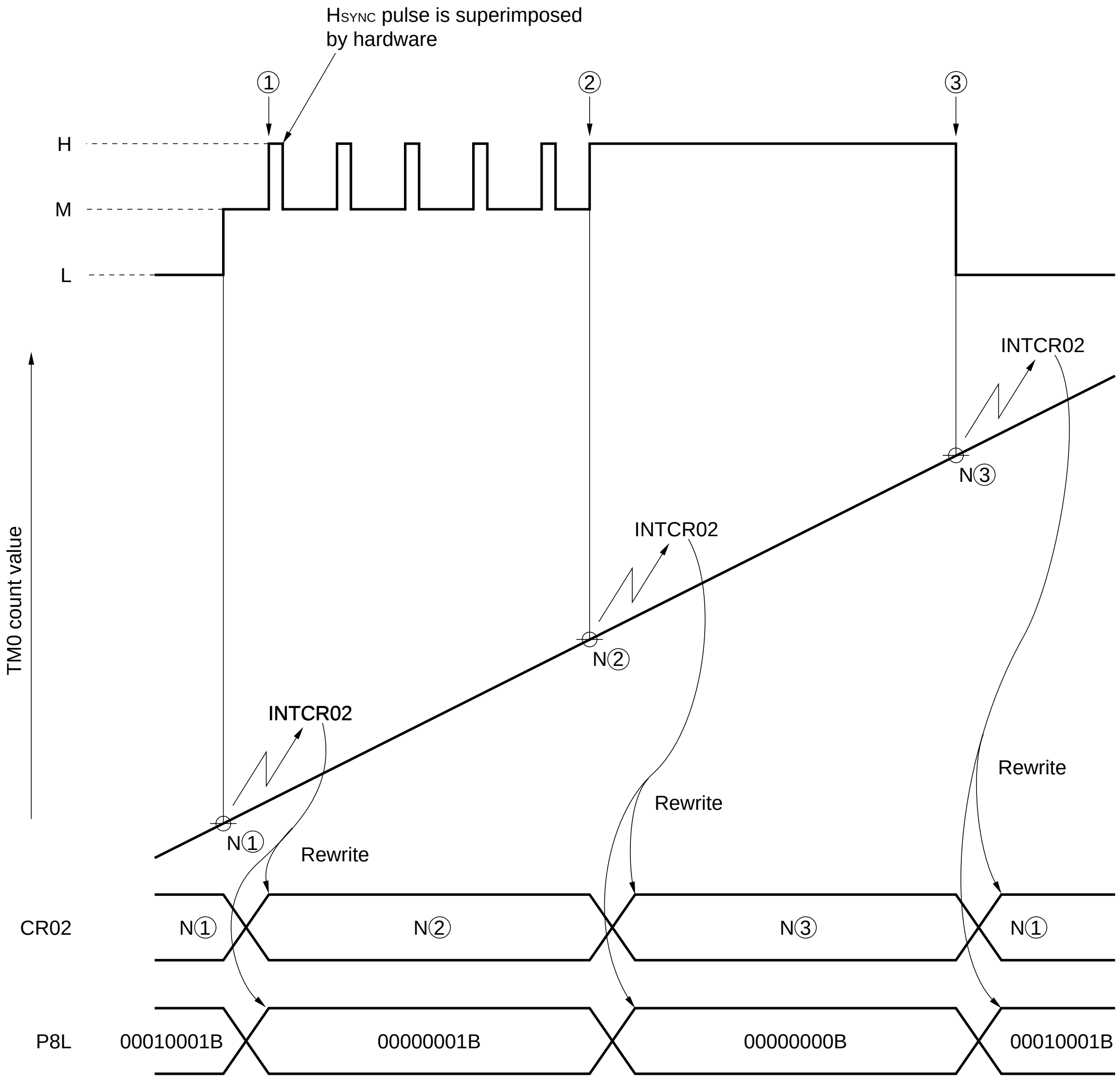


Figure 5-47. Middle Level Generation



Remark When P80 is Hi-Z, the output level is $\frac{R2}{R1 + R2} V_{DD}$

Figure 5-48. Quasi-VSYNC Generation Timing



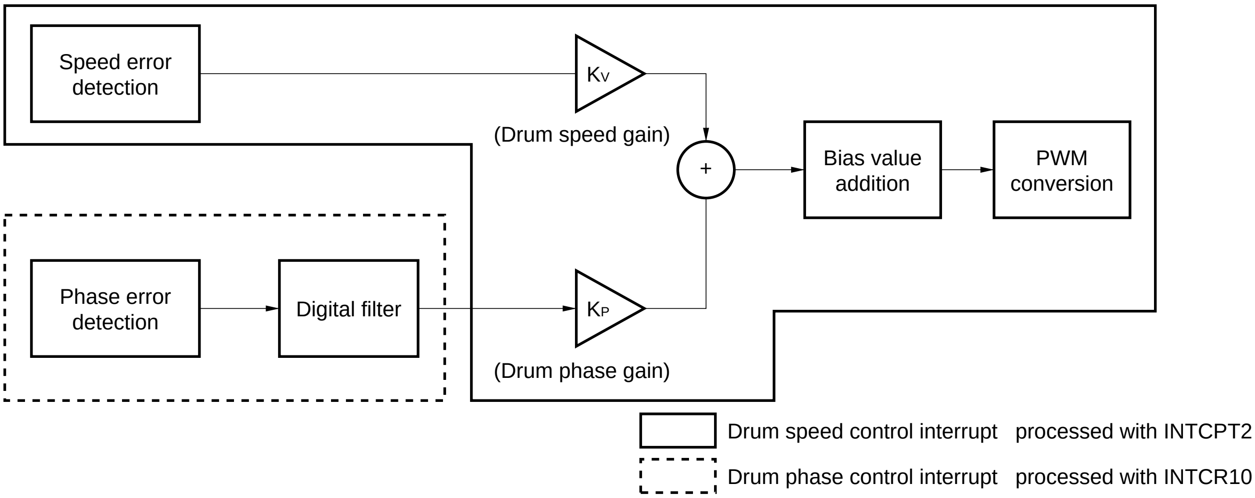
5.11 Treatment of Servo Error Amount

5.11.1 Drum control system processing

Drum control system performs the calculation and filtering processing of drum speed error and phase error and output of drum motor control signal (PWM0).

Figure 5-49 shows the drum control system configuration.

Figure 5-49. Drum Control System Configuration



As shown in Figure 5-49, the drum speed control system performs only phase error calculation and phase control system filtering. The drum phase control system reads out the filtered phase error, adds it with speed system error, and performs PWM output.

First, the total error amount of drum motor is calculated from the speed error amount and phase error amount. The speed error amount E_{DV} acquired from drum speed control interrupt and the phase error amount E_{DP} (digital filter arithmetic result) acquired from drum phase control interrupt are multiplied with gains, respectively (the gains are defined as K_{DV} and K_{DP}, respectively). The sum of these results is defined as the drum error amount E_D.

$$E_D = K_{DV} \cdot E_{DV} + K_{DP} \cdot E_{DP}$$

The sum of the drum total error amount and the bias value is PWM output.

The bias value is PWM output to control the motor in open loop and output the voltage required to rotate the motor in the approximate target rotation. However, the bias is not necessarily a strict value because the actual control is carried out by feedback control and errors are automatically corrected to a certain extent.

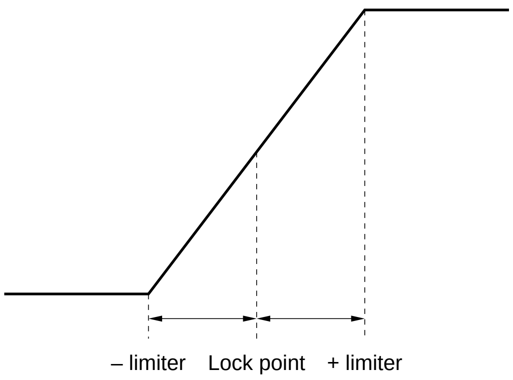
In addition, the servo system characteristics can be improved, such as reduction of motor rising time and improvement of locking stability, by changing the gain according to the speed error amount and phase error amount and canceling phase control.

The speed error gain K_{DV} and phase error gain K_{DP} vary according to the motor characteristics. Set these values according to the characteristics of the motor to be used (in fact, find the value that has the best characteristics in cut and try).

(1) Error amount maximum control processing (limit limiter)

Error amount maximum control processing is the same as trapezoidal pattern for servo IC error value detection, and it controls the maximum of internal error value (error amount) to input to digital filter.

Figure 5-50. Trapezoidal Pattern for Error Value Detection (drum control system)



In this application example, the control range is specified also from loop gain so as to prevent data overflow in the arithmetic processing of digital filter. A maximum limit is set for speed error and phase error, respectively, and each control range is set as follows:

- Drum speed control range ±735H (1 count = 125 nsec) ±230.625 μsec
- Drum phase control range ±2220H (1 count = 500 nsec) ±4368 μsec

(2) Special processing in drum control system

(a) Special processing for error amount calculation

By checking the drum speed error amount detected in the drum speed error amount calculation routine, if the drum speed deviates ±5% or more from the target value, 0 is set to the drum phase error amount. The purpose of this processing is to prohibit addition of the drum phase error amount while the drum speed control system is not in operation and to reduce the lock time of the motor by operating the drum motor only with the speed control system.

(b) Special processing in drum phase system digital filter

This special processing limits the maximum of Y_{n-1} data value in the arithmetic processing of drum phase system digital filter. Y_{n-1} is the data to reflect the past output data of the filter. If the drum phase becomes out of phase for a long period of time (when applying load by lightly holding the drum manually, etc.), Y_{n-1} data keeps increasing. The increased Y_{n-1} data will start decreasing gradually when the applied load is removed. However, the lock time is affected because it takes an extremely long time before Y_{n-1} is decreased. In order to avoid this, the lock time should be reduced by setting the maximum limit for Y_{n-1} data.

The limit value for Y_{n-1} is set as follows:

Y_{n-1} maximum : 13FH

The Y_{n-1} limit value setting method is adopted according to experimental values.

(3) Loop gain multiplication

K_V and K_P shown in Figure 5-50 are loop gains in speed control system and phase control system, respectively. When handling the error amount data (calculated from FRC) which is digital filtering processed as PWM data, the variable range of PWM data is small (the dynamic range is narrow) because the range available for the data is narrow.

Loop gain also has the functions to widen the dynamic range by amplifying the filtered data and to adjust the addition rate of speed system and phase system.

The loop gains of the speed system and phase system are as follows:

Speed system loop gain	: K_V = 17.76 times
Phase system loop gain	: K_P = 46.0 times
Speed/phase addition rate	: K_V : K_P = 4.74 :1

(4) Bias value addition

Bias value addition is carried out to keep the motor control voltage at the lock point during servo lock (speed/ phase error amount 0).

The bias value setting method is adopted according to experimental values. PWM output data in the condition that drum is controlled only with speed control and stabilized at the drum speed target value is adopted as the bias value.

The bias value in drum control system is as follows:

Bias value in NTSC : 66F0 [HEX]

Bias value is added to the sum of speed correction amount and phase correction amount. However, the arithmetic result may overflow, so that overflow check is carried out. The arithmetic result is fixed to the maximum if overflow occurs and to the minimum if borrow occurs.

(5) PWM output for drum motor control

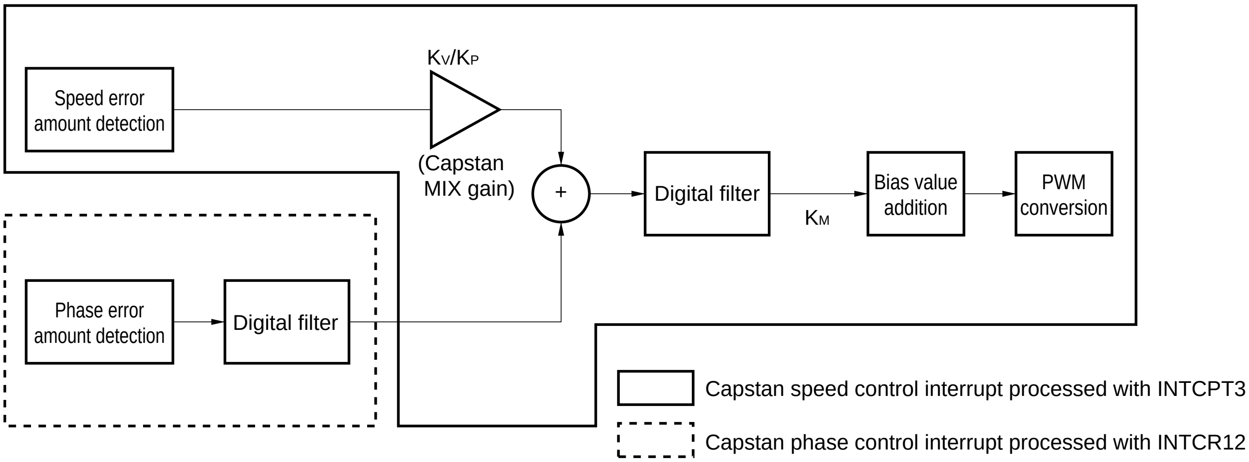
The PWM of the data which is the addition of speed/phase correction amount and bias value is output. The data is processed as 16-bit data. However, since the operation range of PWM output unit is 0FF00H to 0100H, when operating outside this range, the maximum or the minimum is written. The PWM for drum motor control is output in the drum speed control interrupt routine.

5.11.2. Capstan control system processing

Capstan control system performs the calculation and filtering processing of capstan speed error and phase error and the output of capstan motor control signal (PWM1).

Figure 5-51 shows the capstan control system configuration.

Figure 5-51. Capstan Control System Configuration


K_V/K_P
Speed error amount detection
(Capstan MIX gain)
+
Digital filter
K_M
Bias value addition
PWM conversion
Phase error amount detection
Digital filter
Capstan speed control interrupt processed with INTCPT3
Capstan phase control interrupt processed with INTCR12

As shown in Figure 5-51, capstan phase control system performs the calculation of phase error amount and filtering of phase system. Capstan speed control system reads out the filtered phase error, adds it with speed system error, and performs PWM output.

First, the capstan motor total error amount is calculated from the speed error amount and phase error amount. Speed error amount E_{CV} acquired from the capstan speed control interrupt and phase error amount E_{CP} (digital filter arithmetic result) acquired from the capstan phase control interrupt are multiplied with gains, respectively (the gains are defined as K_{CV} and K_{CP}, respectively). The sum of these results are defined as capstan error amount E_C.

$$E_C = K_{CV} \bullet E_{CV} + K_{CP} \bullet E_{CP}$$

The sum of the capstan total error amount and bias value is PWM output.

The bias value is PWM output to control the motor in open loop and output the voltage required to rotate the motor in the approximate target rotation. However, the bias is not necessarily a strict value because the actual control is carried out by feedback control and errors are automatically corrected to a certain extent.

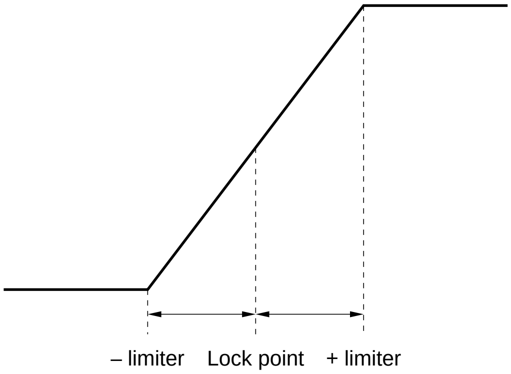
In addition, the servo system characteristics can be improved, such as reduction of motor rising time and improvement of locking stability, by changing the gain according to the speed error amount and phase error amount and canceling phase control.

The speed error gain K_{CV} and phase error gain K_{CP} vary according to the motor characteristics. Set these values according to the characteristics of the motor to be used (in fact, find the value that has the best characteristics in cut and try).

(1) Error amount maximum control processing (limit limiter)

Error amount maximum control processing is the same as trapezoidal pattern for servo IC error value detection, and it controls the maximum of internal error value (error amount) to input to digital filter.

Figure 5-52. Trapezoidal Pattern for Error Value Detection (capstan control system)



This also prevents data overflow in digital filter arithmetic processing.

The limit range is specified by limiting the error maximum.

The maximum limit is set for speed error and phase error, respectively, and each control range is set as follows:

- Capstan speed control range ±1E79H (1 count = 125 nsec) ±975.125 μsec
- Capstan phase control range ±15A0H (1 count = 1 μsec) ±5536 μsec

(2) Special processing in capstan control system

(a) Special processing for error amount calculation

<1> Relation with drum speed error amount

In drum speed error calculation, if the drum speed deviates ±10% or more from the target value, 0 is set to the capstan phase error amount. The purpose of this processing is to prohibit addition of the capstan phase error amount while the drum speed control system is not in operation and to reduce the lock time of the motor by operating the capstan motor only with the speed control system.

<2> Relation with capstan speed error amount

By checking the capstan speed error amount detected in the capstan speed error amount calculation routine, if the capstan speed deviates ±5% or more from the target value, 0 is set to the capstan phase error amount. The purpose of this processing is to prohibit addition of the capstan phase error amount while the capstan speed control system is not in operation and to reduce the lock time of the motor by operating the capstan motor only with the speed control system.

<3> Relation with playback control (PBCTL) signal missing

When playback control (PBCTL) signal missing occurs, 0 is set to the capstan phase error amount. PBCTL signal missing is detected by PBCTL signal missing counter. Normally, while PBCTL signal is input, the PBCTL signal missing counter is incremented in the capstan speed control interrupt (INTCPT3), and the PBCTL signal missing counter is reset to 0 in the capstan phase control interrupt (INTCR12).

However, the PBCTL signal missing counter is not reset if the PBCTL signal misses. Therefore, if PBCTL signal missing counter is 28H (40d) by checking during INTCPT3 interrupt processing, it is judged that PBCTL signal missing has occurred and a flag is set.

This processing prevents PBCTL signal missing due to tape damage and misdetection of the phase error amount for playback non-recorded tapes, etc. and keeps the tape speed at the target value.

(b) Capstan extreme high-speed rotation processing

When the capstan rotates in an extremely high speed (when motor control shorts to 5 V, etc.), CFG interrupt occurs extremely frequently, interrupt processing gets behind, and runs out of time to return to main routine. Once lapsed into this condition, even the short circuit is repaired, the speed error amount detection continues to misdetect, keeps high speed rotation, and is unable to return to main routine, so that pushing keys has no effect.

To avoid this, when the capstan rotates faster than at a certain speed, the interrupt processing thereafter is not performed and interrupt processing ends by lowering PWM data (to shorten the processing time). In this program, the processing becomes effective when the CFG cycle becomes 600 μs or higher.

(3) Loop gain multiplication for each running mode

K_V and K_P shown in Figure 5-51 are, as well as drum control system, loop gains for speed control system and phase control system, respectively. K_M is the gain correction coefficient corresponding to each operation mode and changes according to VCR playback modes.

As discussed in drum control system, K_V and K_P are for adjusting the addition rate of speed system and phase system.

In the drum control system, there is little speed difference among the operation modes. Accordingly, the entire loop gain is not varied. However, in capstan control system, there is a large difference between the SP and EP modes even in standard playback, and speed difference exists in special playback such as CUE/REVIEW. Therefore, the gain also varies according to each operation mode. K_M is set as the correction coefficient to correct the variation.

K_V and K_P are used only as adjustment of addition rate, so that they are represented as K_V/K_P.

Table 5-3 shows the capstan loop gain in each operation mode.

Table 5-3. Capstan Loop Gain in Each Operation Mode

Operation Mode		K_V/K_P	K_M
Standard playback (PB)	SP	4.2	6.0
	LP	1.1	5.25
	EP	1.1	4.25
Fast forward search 1 (CUE1)	SP	4.2	6.0
	LP	3.3	6.0
	EP	3.3	6.0
Fast forward search 2 (CUE2)	SP	4.2	6.0
	LP	3.3	6.0
	EP	3.3	6.0
Rewind search 1 (REV1)	SP	4.2	6.0
	LP	3.3	6.0
	EP	3.3	6.0
Rewind search 2 (REV2)	SP	4.2	6.0
	LP	3.3	6.0
	EP	3.3	6.0
Still, frame (STILL, FRAME)	SP	4.2	6.0
	LP	3.3	5.25
	EP	3.3	4.25

(4) Bias value addition

Bias value addition is carried out to keep the motor control voltage at the lock point during servo lock (speed/ phase error amount 0).

The bias value setting method is adopted according to the experimental values. PWM output data in the condition that capstan is controlled by only speed control and stabilized at the capstan speed target value is adopted as the bias value.

Since capstan speed differs according to each operation mode, different bias value is required in each operation mode.

Table 5-4 shows the capstan bias value in each operation mode.

Table 5-4. Capstan Bias Value in Each Operation Mode

Operation Mode		Bias Value
Standard playback (PB)	SP	85E0H
	LP	8695H
	EP	86DFH
Fast forward search 1 (CUE1)	SP	8678H
	LP	8695H
	EP	86DFH
Fast forward search 2 (CUE2)	SP	8678H
	LP	8695H
	EP	86DFH
Rewind search 1 (REV1)	SP	8678H
	LP	8695H
	EP	86DFH
Rewind search 2 (REV2)	SP	8678H
	LP	8695H
	EP	86DFH
Still, frame (STILL, FRAME)	SP	8678H
	LP	8695H
	EP	86DFH

Bias value is added to the sum of speed correction amount and phase correction amount. However, the arithmetic result may overflow, so that overflow check is carried out. The arithmetic result is fixed to the maximum if overflow occurs and to the minimum if borrow occurs.

(5) PWM output for capstan motor control

PWM output of the data which is the addition of speed/phase correction amount and bias value is performed. The data is processed as 16-bit data. However, since the operation range of the PWM output unit is 0FF00H to 0100H, when operating outside this range, the maximum or the minimum is written. The PWM for capstan motor control is output in the capstan speed control interrupt routine.

5.12 Compensation Filter

The digital servo system only with proportional control element requires a digital filter in the control system for steady-state deviation elimination. The configuration method of the lag-lead type digital filter, which is often used in VCR servo systems, is discussed here.

5.12.1 Filter types

Filters are divided into analog filter and digital filter by the difference of operational principle. Analog filters are configured with circuits such as capacitors (C) and resistors (R) and realize filter characteristics electronically.

Digital filters are configured also with microcontrollers and signal processors, and realize the characteristics equal to analog filters by performing various arithmetic processing on the input signals which are sampled and quantized.

Digital filters are divided into FIR type and IIR type by the difference of filter configurations.

(1) FIR type (Finite Impulse Response) filter

The finite impulse response filter is also called acyclic filter.

FIR has finite response and no feedback loop due to its filter configuration, that is, the filter output value is determined only with the input value of the present and the past.

(2) IIR type (Infinite Impulse Response) filter

The Infinite impulse response filter is also called cyclic filter.

Since FIR has feedback loop due to its filter configuration, impulse response continues infinitely. Therefore, the filter output value is determined not only with the input of the present and the past but also with the output value of the past. This type of filter realizes steep cut-off characteristics in much lower degree than that of FIR type filter. The VCR servo system mainly uses the IIR type filter.

5.12.2 Biprimary conversion method

(1) Sampling theorem

When using a digital filter, sampling processing is required in the course of converting analog input signals to digital values.

That is, analog signals are converted to discrete numeric sequences in certain constant time intervals T_S.

However, if the sampling cycle T_S is made too long, restoration of the original analog signal is impossible.

The limit of the cycle that the original analog signal can be restored is described with the well-known sampling theorem below:

$$2f_{max.} \leq f_S = \frac{1}{T_S}$$

$\left(\begin{array}{l} f_{max.} \text{ .... The maximum frequency included in the original analog signal} \\ f_S \text{ .... Sampling frequency} \end{array} \right.$

That is, unless the frequency twice or more of the maximum frequency included in the original analog signal is selected for the sampling frequency f_S, it is impossible to restore the original analog signal from the sampled digital signal.

Figure 5-53 shows sampling theorem observed on frequency spectrum.

Figure 5-53 (1) shows the case the maximum frequency component $f_{max.}$ satisfies

$$2f_{max.} < f_S$$

that is, the original signal is band limited.

In such case, the original signal can be completely restored if the sideband component is eliminated, extracting only the basic spectral component using ideal low-pass filter whose cut-off frequency f_C is $f_C = f_S/2$.

However, in the case that $f_{max.}$ does not satisfy sampling theorem, that is,

$$2f_{max.} > f_S$$

sections where the original signal spectrum and fold spectrum are overlapped, that is, fold error is generated as shown in Figure 5-53 (2).

In this case, the restoration of the original signal is impossible even if ideal low-pass filter is used.

Next, sampling theorem is examined on s planar.

The time function of the original signal is defined as f (t), and the result of Laplace transform of the function is defined as F (s).

Further, F (s) is sampled with sampling cycle T_S. This is defined as F* (s).

Now, assuming the maximum frequency $f_{max.}$ of the original signal satisfies sampling theorem, the pole of F (s), as shown in Figure 5-54 (1), is in basic band. The basic band is folded as shown in Figure 5-54 (2) by sampling processing, as a result, sideband whose width is $2\pi/T_S$ is generated, and the pole of F (s) is also folded.

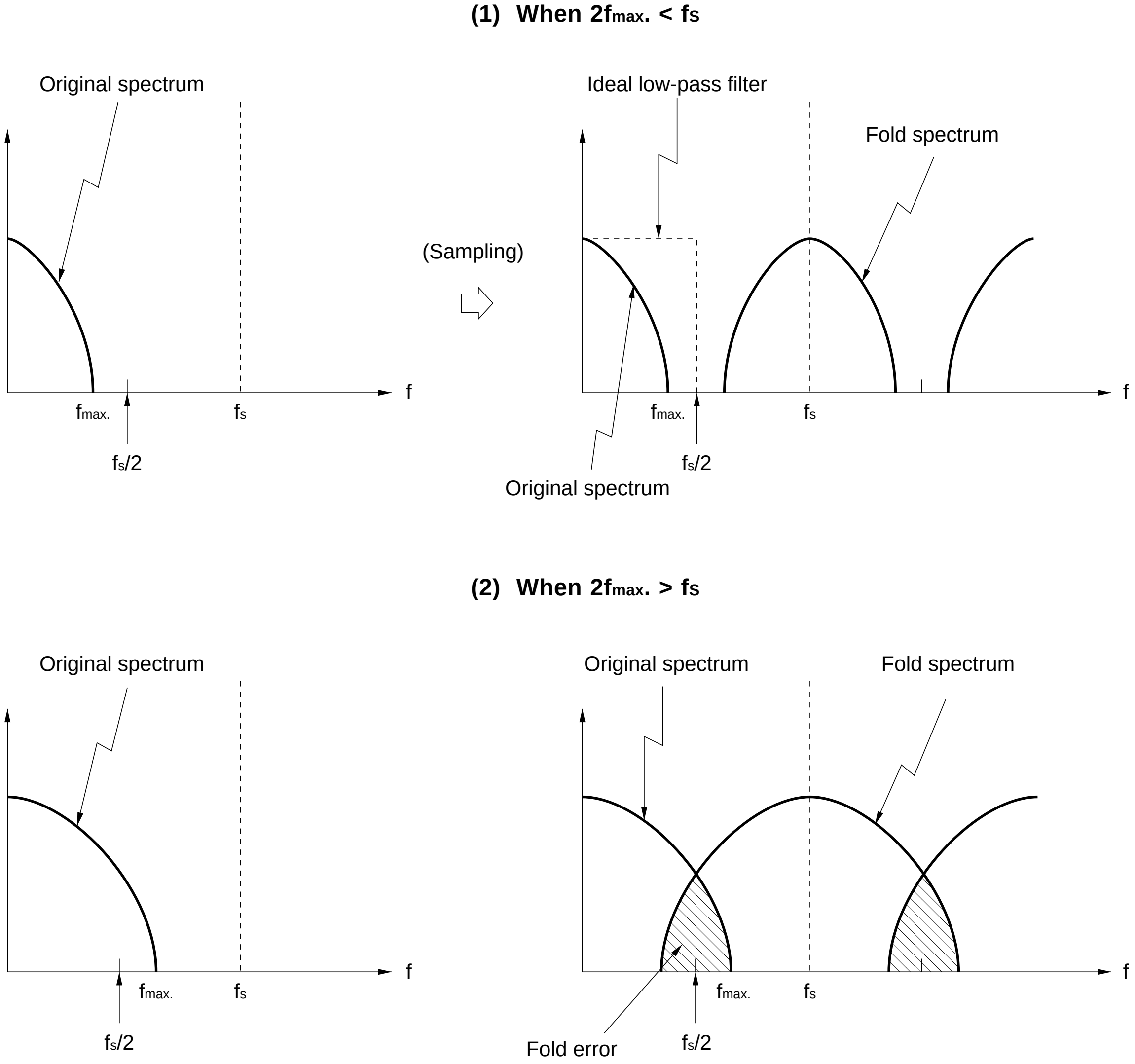
Since the basic band and sideband are exactly the same, the original signal can be completely restored if only the basic band component is extracted using ideal low-pass filter.

On the other hand, the case that the original signal does not satisfy sampling theorem is shown in Figure 5-55.

In this case, the pole of F (s) is located out of the basic band. Therefore, if pole is folded by sampling processing, pole is generated in the basic band, where pole is not originally located.

Once this happens, the restoration of the original signal is impossible even if only the basic band component is extracted using ideal low-pass filter, since the basic band component is different from the original one.

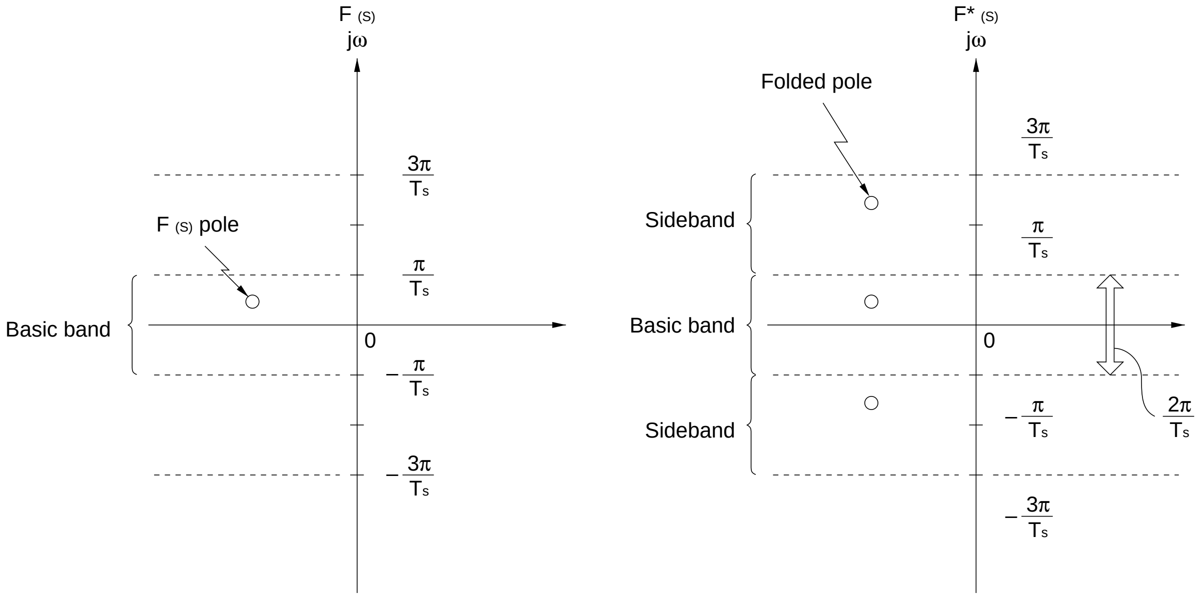
Figure 5-53. Fold Error



(1) When $2f_{max.} < f_S$

(2) When $2f_{max.} > f_S$

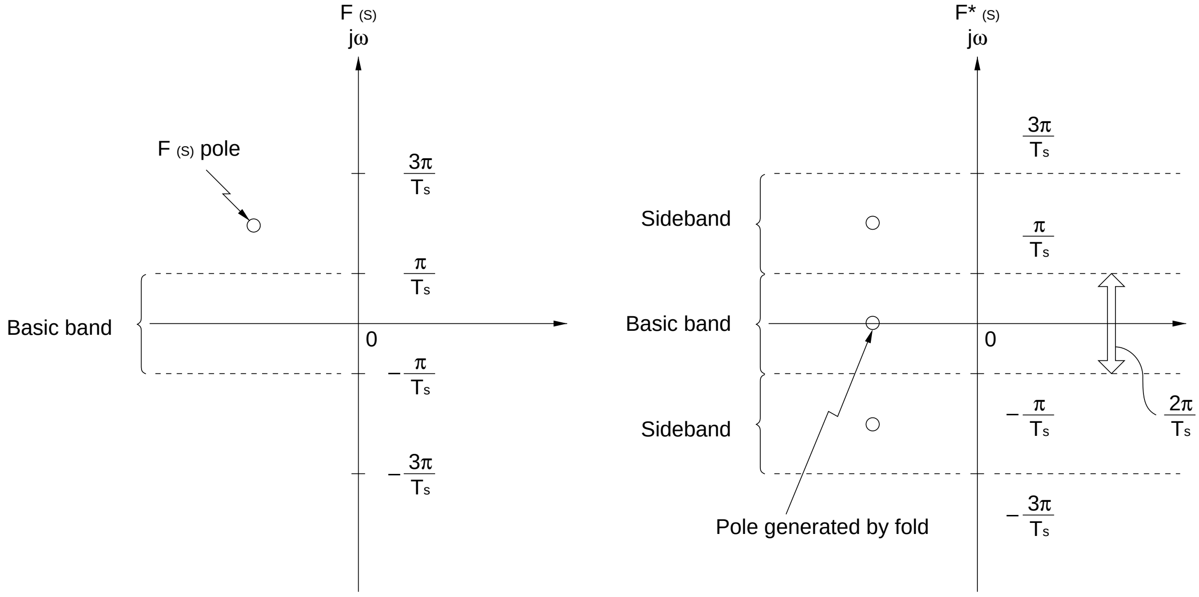
Figure 5-54. Pole Location when Sampling Theorem is Satisfied



(1) Pole location of F (s) before sampling processing

(2) Pole location of F (s) after sampling processing

Figure 5-55. Pole Location when Sampling Theorem is Not Satisfied



(1) Pole location of F (s) before sampling processing

(2) Pole location of F (s) after sampling processing

(2) Biprimary transform method

Biprimary transform is a transform method to prevent intrusion of fold errors in the standard z function for analysis of control.

Generally, when analyzing a control system, the analysis in a continuous system is performed on s planar using Laplace transform and the analysis in a discrete system on z planar using z transform. Transform of s planar to z planar is called standard z transform.

When configuring a digital filter, it is easier if the filter is designed in a continuous system before transforming to a discrete system. The issue here is the effect by sampling processing.

That is, if T_S is made too long when transforming analog signal to discrete numeric sequence in certain time interval T_S, the restoration of original signal is impossible.

The limit sampling frequency fS to restore the original analog signal can be described with the well-known sampling theorem below:

$$2 \bullet f_{max.} \leq f_S = \frac{1}{T_S}$$

$f_{max.}$: The maximum frequency included in the original analog signal

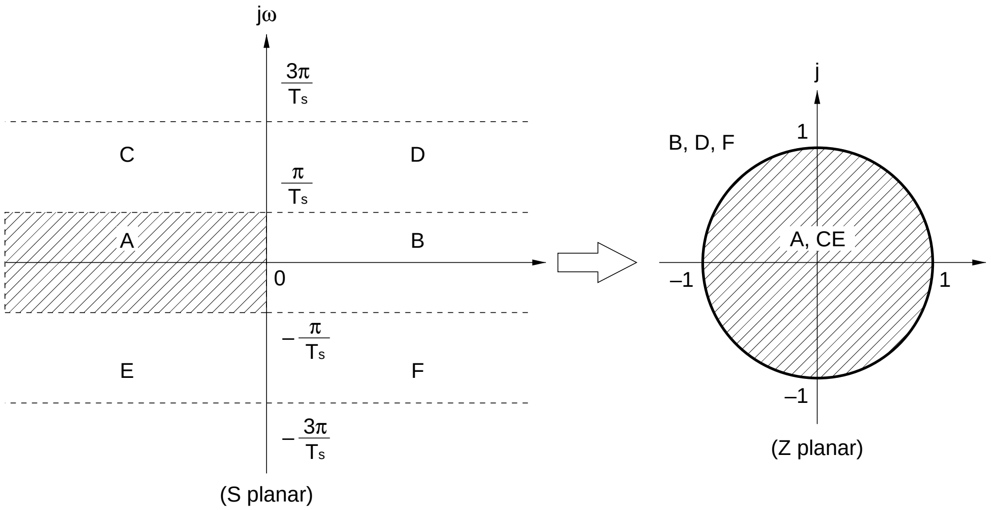
f_S : Sampling frequency

T_S : Sampling cycle

The expression above shows that the restoration of the original analog signal from the sampled digital signal unless setting the sampling frequency f_S to twice or more of the frequency included in the original analog signal. In the following paragraph, this is considered in the corresponding relation of s planar and z planar.

Figure 5-56 shows the mapping by standard z transform. The band of $2\pi/T_S$ width on s planar is generated with sampling processing. The area corresponding to the width of this band is mapped on the whole z planar. That is, block A on s planar (shaded area) is mapped to inside the unit circle on z planar and block B on s planar is mapped to outside the unit circle on z planar, respectively. Therefore, if a pole by fold error exists in the $2\pi/T_S$ band on s planar, the fold error is also mapped on z planar.

Figure 5-56. Mapping by Standard z Transform



Biprimary transform is one of the processing methods to prevent fold errors from intruding into z planar. In biprimary conversion, when sampling processing is performed, the whole s planar is transformed to 2π/Ts band area before performing standard z transform so that fold errors are not generated. The planar where the whole s planer is transformed to is defined as s planar.
Figure 5-57 shows the mapping by biprimary transform. Since s transform is a cyclic function consisting of the band with 2π/Ts width, s-z transform with no fold error is acquired if standard z transform is carried out after s transform.
In biprimary transform, the relation between s operator, which is the parameter of a continuous system, and z operator, which is the parameter of a discrete system, is represented in the following expression:

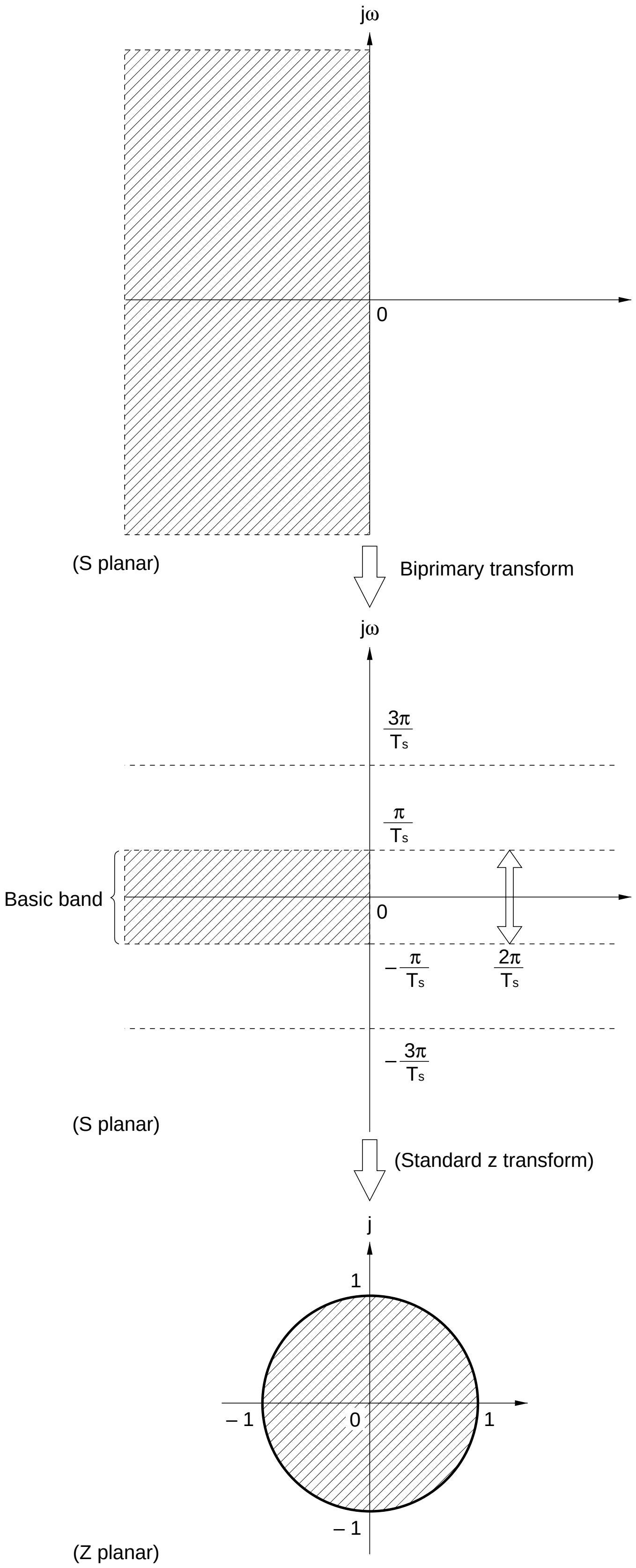
$$s = \frac{2}{T_S} \times \frac{1 - Z^{-1}}{1 + Z^{-1}}$$

T_S : Sampling cycle

From the above expression, the following operation is performed to transform transfer function G (s) expressed in a continuous system to transfer function G (z) of a discrete system with no fold error.

$$G\ (z) = G\ (s)\ \ |\ s = \frac{2}{T_S} \times \frac{1 - Z^{-1}}{1 + Z^{-1}}$$

Figure 5-57. Mapping by Biprimary Transform



5.12.3 Digital filter designing method

An example of digital filter designing methods is shown below.

(1) Determination of specification

Determine the specification of the digital filter to realize, such as frequency characteristics, cut-off frequency, time area response, and sampling cycle.

(2) Configuration on analog circuit

Design the analog filter satisfying the specification in (1).

At this time, transform operation to digital filter is made easier if the analog circuit is configured with passive filter using LCR.

(3) Calculation of transfer function

Find the transfer function G (s) in continuous time area of the analog filter found in (2).

(4) Biprimary transform processing

Transform the analog filter transfer function G (s) to discrete time sequence transfer function G (z). At this time, perform biprimary transform to (s) so that fold errors by sampling processing are avoided.

(5) Determination of filter constant

Calculate digital filter constant from the specification in (1) and quantize the filter coefficient.

The setting of the coefficient word length of the digital filter is determined according to the filter cut-off frequency, sampling cycle, and dynamic range.

(6) Program generation of digital filter

(7) Measurement of characteristics

Measure whether the digital filter generated in (6) is operating or not as specified using servo analyzer.

Also, measure the dynamic range of the digital filter. The dynamic range refers to the maximum digital value which will not cause overflow if input to the filter.

(8) Improvement of characteristics

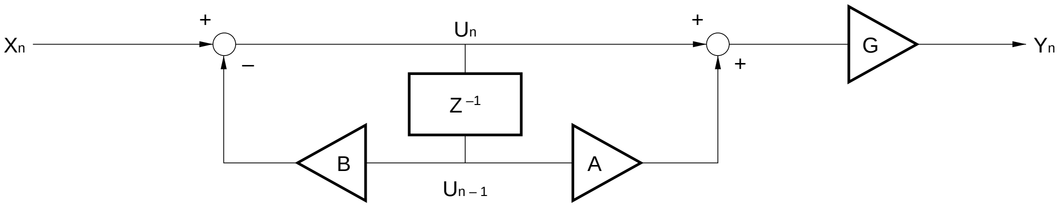
Change the arithmetic word length and filter configuration to improve the characteristics acquired in (7).

Also, shorten the arithmetic word length and change algorithm if the calculation time of the digital filter is too long.

5.12.4 Primary IIR type digital filter transfer function

Figure 5-58 shows primary IIR type digital filter block diagram.

Figure 5-58. Primary IIR Type Digital Filter Block Diagram



In Figure 5-58, A, B, and G are filter constants and the meanings of them are as follows:

A : non-cyclic filter constant
B : cyclic filter constant
G : filter gain constant

Assume n-th input value of this filter as X_n, output value as Y_n, calculation value in n-th filter arithmetic process as U_n, calculation value of n–1-th filter arithmetic process as U_{n-1}.

From the block diagram in Figure 5-58 , the following expression is found:

$$U_n = X_n - B \times U_{n-1}$$
$$Y_n = (U_n + A \times U_{n\text{-}1}) \times G \qquad \text{(Expression 5-1)}$$

If the expression above is solved for X_n:

$$X_n = U_n + B \times U_{n-1}$$
$$Y_n = (U_n + A \times U_{n\text{-}1}) \times G \qquad \text{(Expression 5-2)}$$

Both parts in the expressions above are z transformed to acquire the following expression:

$$X\ (z) = U\ (z) + Bz^{-1}U\ (z)$$
$$Y\ (z) = G\ (U\ (z) + Az^{-1}U\ (z)) \qquad \text{(Expression 5-3)}$$

From the expression above, the transfer function G (z) in the system is as follows:

$$G\ (z) = \frac{Y\ (z)}{X\ (z)} = \frac{G\ (U\ (z) + Az^{-1}U\ (z))}{U\ (z) + Bz^{-1}U\ (z)}$$
$$= G \times \frac{1 + Az^{-1}}{1 + Bz^{-1}} \qquad \text{(Expression 5-4)}$$

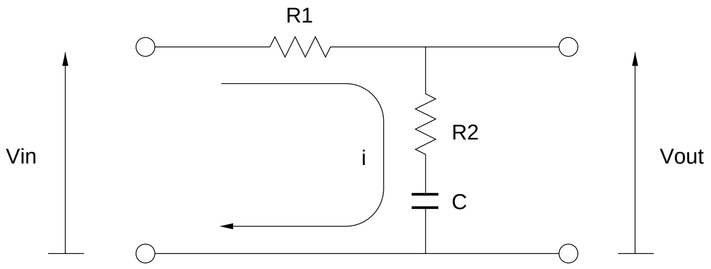
5.12.5 Lag-lead filter configuration method

The lag-lead filter is often used as the drum phase control system compensation filter for VCRs. The purpose is to eliminate the constant deviation and improve the accuracy of the system.

Figure 5-59 shows the lag-lead filter configuration and characteristics.

Figure 5-59. Lag-lead Filter Configuration and Characteristics

(a) Lag-lead filter configuration



Cut-off frequency

$$f1 = \frac{1}{2\,\pi\,C\,(R1+R2)}$$

$$f2 = \frac{1}{2\,\pi\,CR2}$$

(b) Lag-lead filter board line graph

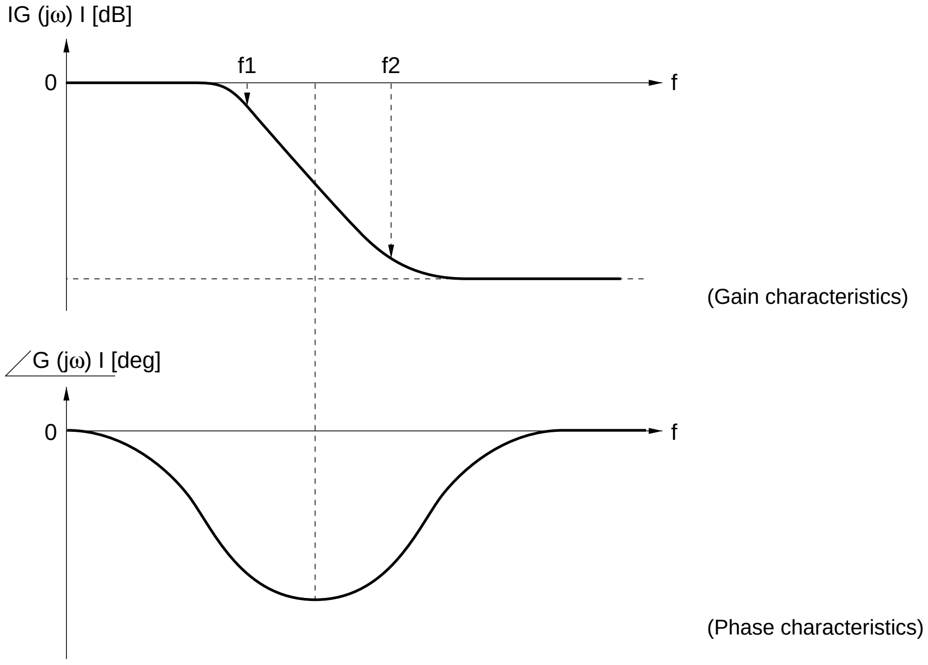


Figure 5-59 (b) shows lag-lead filter gain characteristics and phase characteristics (board line graph) of the analog circuit configuration shown in Figure 5-59 (a).

Lag-lead filter has two segmented point frequencies f1 and f2. By freely setting these, the filter gain characteristics and phase characteristics can be changed.

The method to find the constants (A, B, and G) used in primary IIR type digital filter to realize lag-lead filter characteristics is as follows:

In the case of the filter shown in Figure 5-59 (a), the segmented point frequencies f1 and f2 are found in the following expression:

$$f1 = \frac{1}{2\pi C\ (R1 + R2)} \qquad \text{(Expression 5-5)}$$

$$f2 = \frac{1}{2\pi CR2} \qquad \text{(Expression 5-6)}$$

The transfer function of the filter in Figure 5-59 (a) is found as follows (the transfer function is find by plus transforming the relational expressions for Vin and Vout, respectively):

$$G\ (s) = \frac{1 + sCR2}{1 + sC\ (R1 + R2)} = \frac{1 + \frac{1}{2\pi f1}S}{1 + \frac{1}{2\pi f2}S} \qquad \text{(Expression 5-7)}$$

$$a = \frac{1}{2\pi f2} \qquad b = \frac{1}{2\pi f1} \qquad \text{(Expression 5-8)}$$

Now, if parameter a and b are assumed and assigned as the expression above, the transfer function of lag-lead filter is as follows:

$$G\ (s) = \frac{1 + bS}{1 + aS} \qquad \text{(Expression 5-9)}$$

Since the transfer function in the expression above is represented in continuous time system, this is transformed to be represented in discrete time system.

In this case, biprimary transform is used because fold error is generated if standard z transform is performed. That is, S arithmetic operator is replaced as follows:

$$S = \frac{2}{T_S} \times \frac{1 - Z^{-1}}{1 + Z^{-1}} \qquad T_S : \text{Sampling cycle} \qquad \text{(Expression 5-10)}$$

The S operator is assigned to the transfer function, the expression is reorganized.

$$G(z) = \frac{1 + \dfrac{2a}{\mathrm{Ts}} \times \dfrac{1 - Z^{-1}}{1 + Z^{-1}}}{1 + \dfrac{2b}{\mathrm{Ts}} \times \dfrac{1 - Z^{-1}}{1 + Z^{-1}}} = \frac{\mathrm{Ts}\,(1 + Z^{-1}) + 2a\,(1 - Z^{-1})}{\mathrm{Ts}\,(1 + Z^{-1}) + 2b\,(1 - Z^{-1})}$$

$$= \frac{\mathrm{Ts} + 2a}{\mathrm{Ts} + 2b} \times \frac{1 + \dfrac{\mathrm{Ts} - 2a}{\mathrm{Ts} + 2a} Z^{-1}}{1 + \dfrac{\mathrm{Ts} - 2b}{\mathrm{Ts} + 2b} Z^{-1}} = G \times \frac{1 + AZ^{-1}}{1 + BZ^{-1}}$$

(Expression 5-11)

The above transfer function found here has the same configuration as the one found from the primary IIR type digital filter block diagram in Figure 5-58.

G, A, and B in the expression above are filter coefficients, and turn out as follows:

$$G = \frac{\mathrm{Ts} + 2a}{\mathrm{Ts} + 2b}$$

$$A = \frac{\mathrm{Ts} - 2a}{\mathrm{Ts} + 2a}$$

$$B = \frac{\mathrm{Ts} - 2b}{\mathrm{Ts} + 2b}$$

(Expression 5-12)

If G, A, and B are found from sampling cycle Ts [sec] and two segmented point frequencies, f1 and f2, which are filter characteristics, primary IIR type digital filter coefficient can be found. An example of this is shown below.

- Filter design specification

Sampling cycle	Ts	: 4.0 [msec] (sampling frequency 250 Hz)
Segmented point frequency	f1	: 1.0 [Hz]
	f2	: 10.0 [Hz]

Filter coefficients, G, A, and B are found from the above filter design specification.
a and b are found from Expression 5-8.

$$a = \frac{1}{2\pi\ f2} = \frac{1}{2\pi \times 10} = 0.01591549$$

$$b = \frac{1}{2\pi\ f1} = \frac{1}{2\pi \times 1} = 0.15915494$$

If a, b, and T_S found from the above are assigned to Expression 5-12, filter coefficients G, A, and B are found as follows:

$$G = 0.111169375$$
$$A = –0.77672957$$
$$B = –0.97517917$$

5.12.6 Filter processing method

Lag-lead filter is configured with product-sum instruction.

Lag-lead filter propagation function is as follows:

$$\begin{aligned} Y_n &= G (X_n + AX_{n-1}) - BY_{n-1} \\ &= G \bullet X_n + AG \bullet X_{n-1} + (-B) \bullet Y_{n-1} \end{aligned}$$

The operation of product-sum instruction when the number of operations is two is as follows:

AXDE ← (B) × (C) + (B + 2) × (C + 2) + AXDE

Then each parameter of lag-lead filter is assigned as follows:

AXDE (Left part)	: Y_n	signed 32 bits
(B)	: G	signed 16 bits
(C)	: X_n	signed 16 bits
(B + 2)	: AG	signed 16 bits
(C + 2)	: X_{n-1}	signed 16 bits
AXDE (Right part)	: $(-B) \bullet Y_{n-1}$	signed 32 bits

Signed multiplication is considered here.

First, the coefficients G, AG, and (–B) of lag-lead filter are designed with gain of 1, so that they become values with an absolute value of 1 or less.

$| G | <1, | AG | <1, | (-B) | <1$

Therefore, the values multiplied with the 15th power of 2 (32768) are actually used for operation.

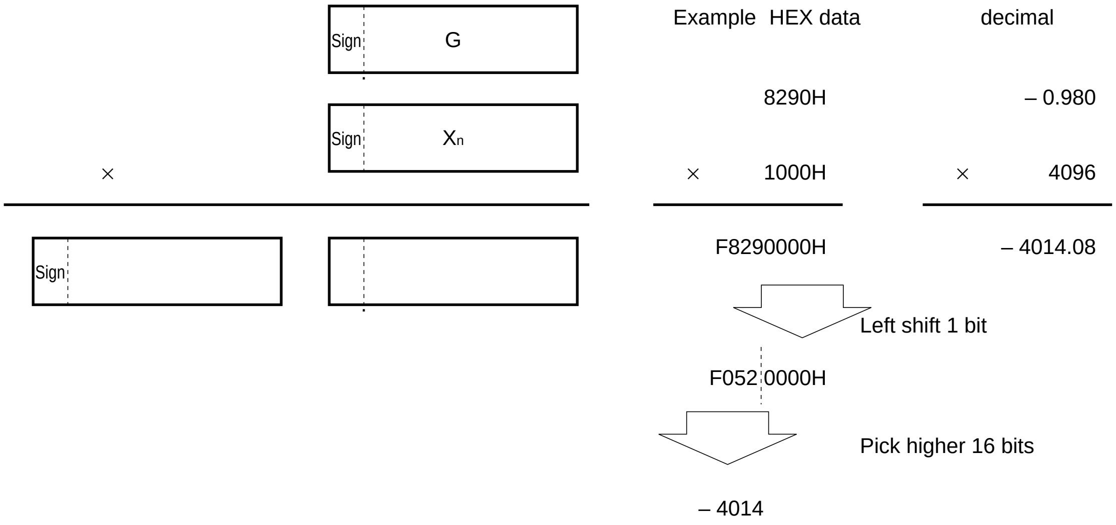
For example,

$0.980 \rightarrow 0.98 \times 32768 = 32112.64 \rightarrow$ 7D70H
$-0.980 \rightarrow -0.98 \times 32768 = -32112.64 \rightarrow$ 8290H

The data has the following range if the error amount is also dealt with signs.

8000H–7FFFH (–32768 to +32767)

If this is calculated with signed multiplier:



The 1/2 of the actual calculation result enters higher 16 bits of the arithmetic result of signed multiplication.

Therefore, the result is doubled (right shift) after executing product-sum instruction. However, gain multiplication is normally performed after digital filter calculation, so that there is a method which abbreviates the shift processing if the gain is doubled.

The propagation function becomes as follows:

$Y_n' = G \bullet X_n + AG \bullet X_{n-1} + (-B) \bullet Y_{n-1}'$

Where

$Y_n' = Y_n/2$

$(-B) \bullet Y_{n-1}' = (-B) \bullet Y_{n-1}/2$

This enables filter calculation only with product-sum instruction.

However, calculation of (–B) Y_{n-1} is necessary for the following sampling timing, then, singed multiplication is performed again. Since the result is made 1/2 as it is, right shift processing is performed before multiplication.

Filter calculation is summarized as follows:

(1) Set values for each register

(B) : G
(C) : Xn
(B + 2) : AG
(C + 2) : X_{n-1}
AXDE : (–B) • Y_{n-1}' (Already stored in memory at the previous sampling timing)

(2) Execute product-sum instruction

```
MACSW 2
```

The value stored in calculation result AXDE is "Y_n"

(3) Right shift AXDE 1 bit to "Y_n"

```
SHLW    DE
ROLC    X
ROLC    A
```

(4) Find (–B) • Yn´

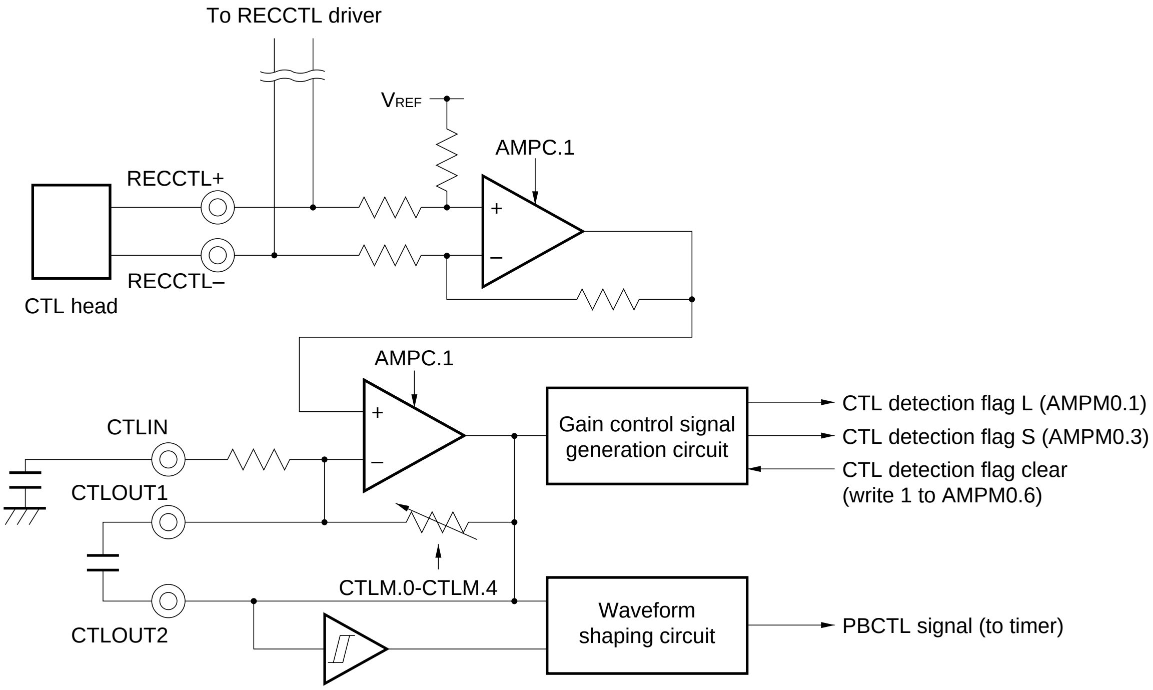
```
MULW    DE          ; DE ← (–B)
```

Store calculation result AXDE in memory and use it as (–B) • Y_{n-1}'

6.1 CTL Amplifier Auto Gain Control Processing

CTL amplifier is used for amplifying the playback control (PBCTL) signal which is the playback of the CTL signal recorded on VCR tape. Figure 6-1 shows the CTL amplifier configuration.

Figure 6-1. CTL Amplifier Configuration



CTL amplifier is configured with two OP amplifiers and the forestage amplifier is fixed to 20 dB. Therefore, gains are adjusted by changing the gain of the second stage amplifier.

The gain setting of CTL amplifier can be changed with CTLM register in 32 steps (by 1.78 dB).

Caution Changing of the gain setting should be avoided while CTL signal is being input.

The μPD784915 has a gain control signal generation circuit which uses CTL detection flags to discriminate the amplifying state of CTL amplifier output.

CTL detection flags are divided into CTL detection flag S and CTL detection flag L according to the detection level.

CTL detection flags S and L can be cleared by writing “1” to FLGCLR (AMPC0.6).

Using these two detection flags, auto gain control of CTL amplifier is carried out.

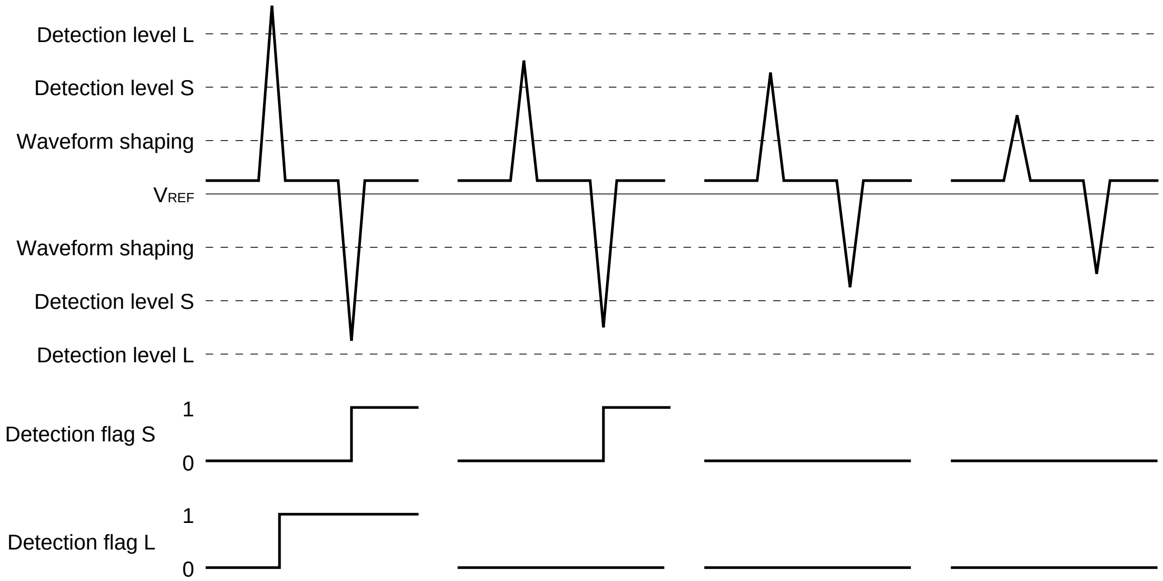
Table 6-1 shows the relation between the CTL detection flag read value and CTL amplifier gain adjustment.

Table 6-1. CTL Detection Flag Read Value and CTL Amplifier Gain Adjustment

CTL Detection Flag Read		Discrimination	CTL Amplifier Gain Adjustment
Flag L	Flag S		
1	1	Gain large	Lower gain
0	1	Gain optimum	No change
0	0	Gain small	Raise gain

Figure 6-2 shows the relationship between CTL amplifier output and each detection level/flag.

Figure 6-2. Relationship between CTL Amplifier Output and Each Detection Level/Flag



6.1.1 CTL amplifier auto gain control method

CTL amplifier auto gain control is performed with the timings of CTL detection flag read and the amplifier gain setting which are determined by the playback control (PBCTL) signal edge interrupt.

- **Timing of CTL detection flag read and CTL amplifier gain setting**

As mentioned earlier, change of the gain setting must be done avoiding PBCTL signal input (rising and falling edges of amplifier amplifying point).

Moreover, since CTL detection flag S and L are specified at the rising and falling edges of PBCTL signal, so that after changing CTL amplifier gain, the both edge must be passed more than once before flag is read.

In order to pass both edges avoiding PBCTL signal input, the timings of the CTL detection flag read and the amplifier gain setting are determined by the PBCTL signal edge interrupt (one edge).

- **For PLAY, CUE/REV**

  <In forward direction> (refer to **Figure 6-3**)
  Gain is changed at 70% point of PBCTL signal.
  <In reverse direction> (refer to **Figure 6-4**)
  Gain is changed at 30% point of PBCTL signal.

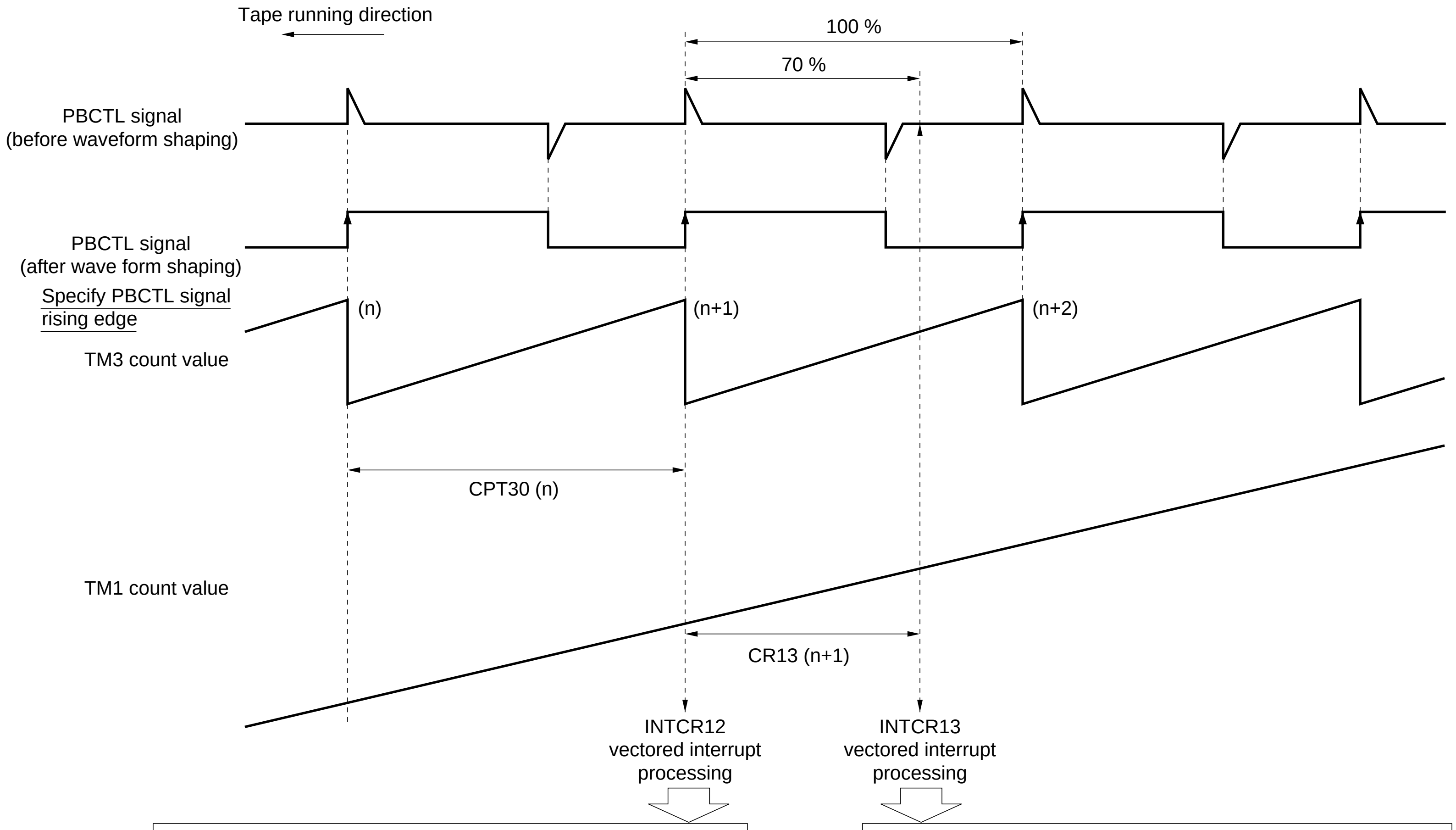
- **For FF/REW**

  <In forward direction> (refer to **Figure 6-5**)
  Gain is changed at 180% point of PBCTL signal.
  <In reverse direction> (refer to **Figure 6-6**)
  Gain is changed at 120% point of PBCTL signal.

Remark For PLAY or CUE/REV, gain is changed 65% or more of PBCTL signal in forward direction and less than 35% in reverse direction and the order of the signal input has been set so that the rising edge of the PBCTL signal is input first and then the falling edge is input.

In addition, for FF/REW, CTL signal input is the fastest, approx. 130 μs (= 33.37 ms/256) in 256-time speed (when tape mode is EP), and it would take 198 μs Max. before PBCTL signal input INTCR12 interrupt (due to other priority interrupt), so that the timing has been set at +100%.

Figure 6-3. Gain Change Timing for PLAY or CUE/REV in Forward Direction



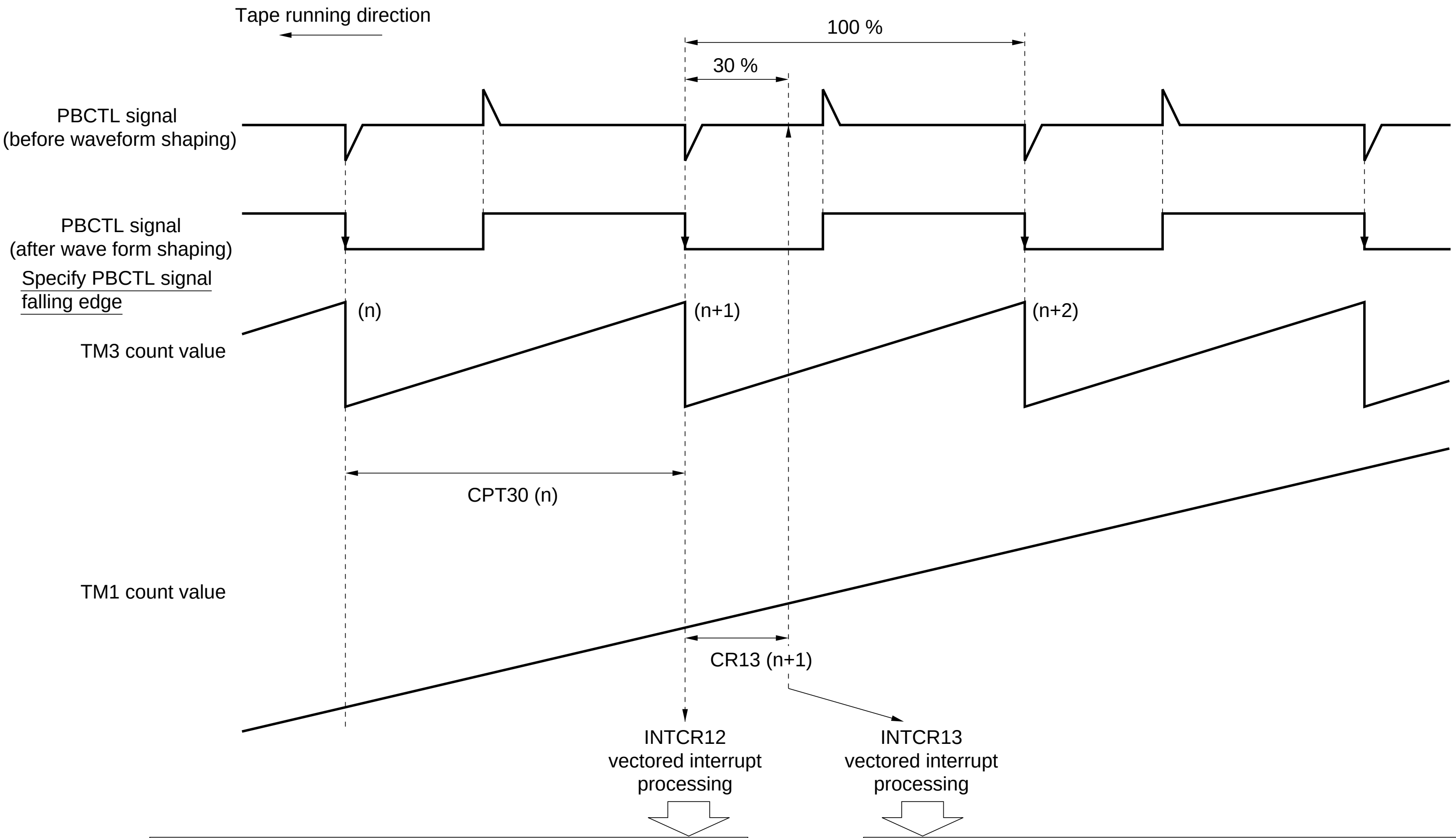
Execute the following processings by INTCR12 vectored interrupt

- Capture current TM1 count value to CR12 (automatic)
- CR13 (n+1) ← CR12 + (CPT30 (n) × 0.7)
- INTCR13 interrupt request clear and interrupt enabled

Execute the following processings by INTCR13 vectored interrupt

- Read CTL detection flag
- If CTL detection flag S and L are both “ 1 ”, the gain is large; therefore, decrease the gain for 1.78 dB
- If CTL detection flag S and L are both “ 0 ”, the gain is small; therefore, increase the gain for 1.78 dB
- INTCR13 interrupt disabled

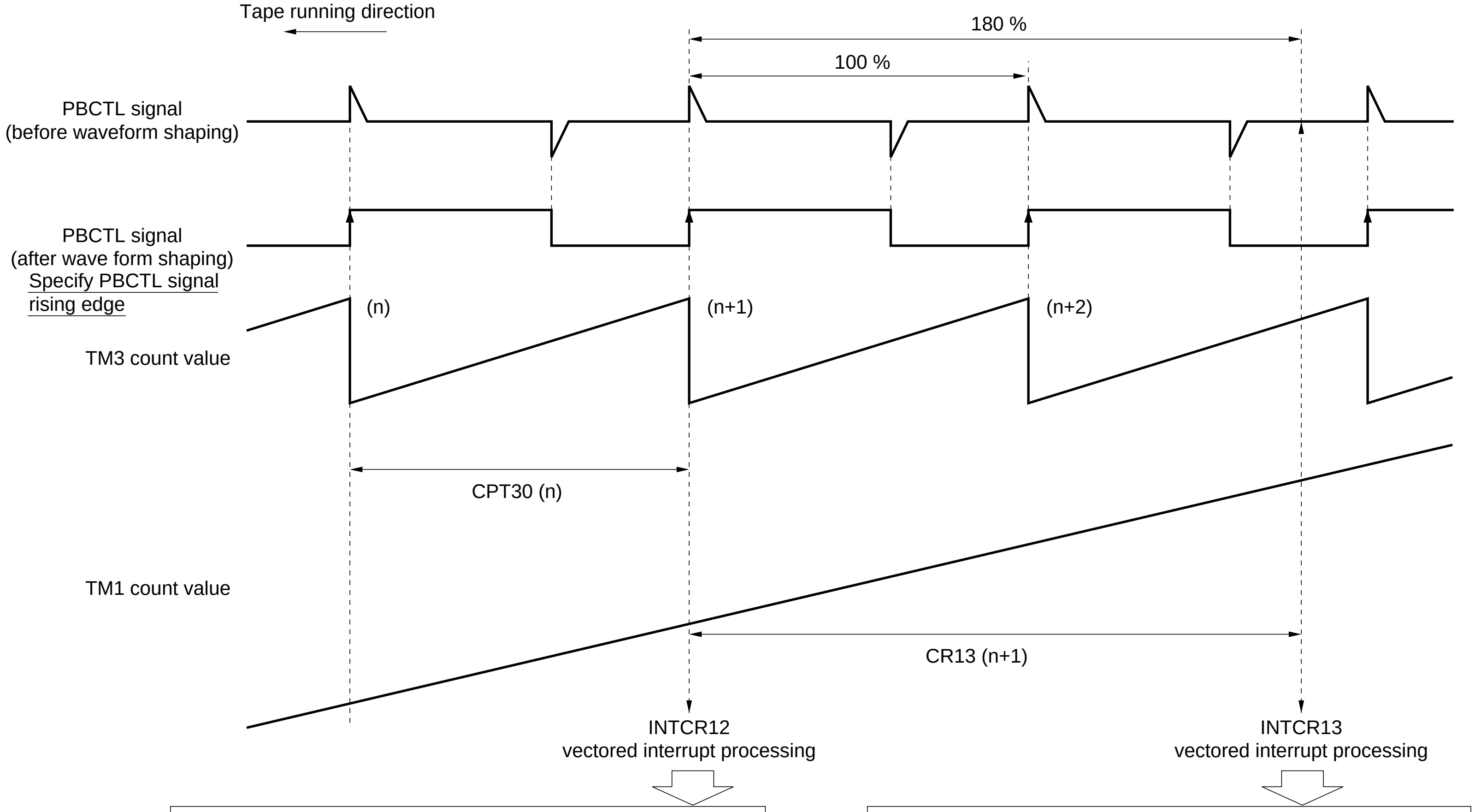
Figure 6-4. Gain Change Timing for PLAY or CUE/REV in Reverse Direction



Execute the following processings by INTCR12 vectored interrupt
- Capture current TM1 count value to CR12 (automatic)
- CR13 (n+1) ← CR12 + (CPT30 (n) × 0.3)
- INTCR13 interrupt request clear and interrupt enabled

Execute the following processings by INTCR13 vectored interrupt
- Read CTL detection flag
- If CTL detection flag S and L are both “ 1 ”, the gain is large; therefore, decrease the gain for 1.78 dB
- If CTL detection flag S and L are both “ 0 ”, the gain is small; therefore, increase the gain for 1.78 dB
- INTCR13 interrupt disabled

Figure 6-5. Gain Change Timing for FF/REW in Forward Direction



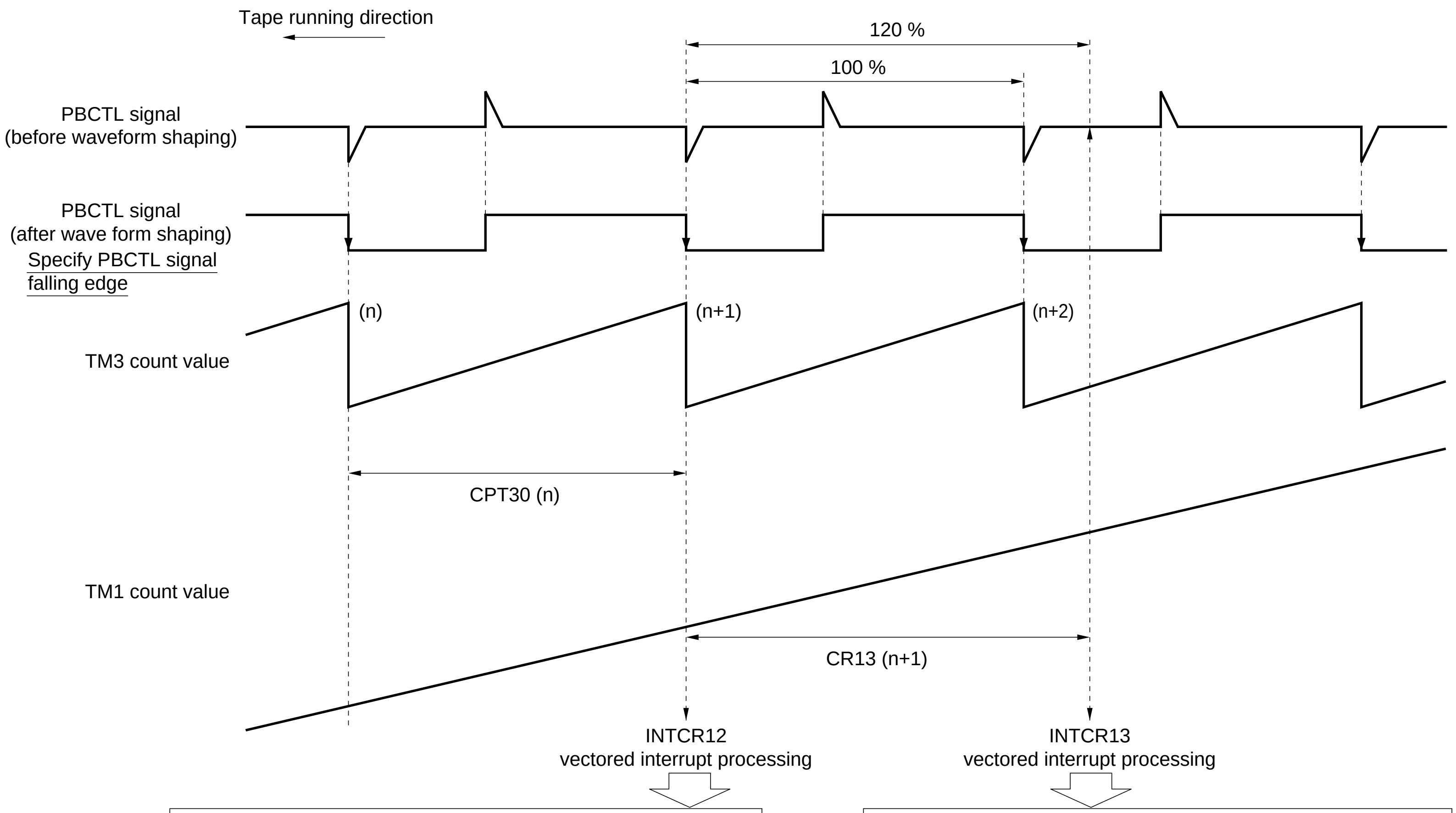
Execute the following processings by INTCR12 vectored interrupt

- Capture current TM1 count value to CR12 (automatic)
- CR13 (n+1) ← CR12 + (CPT30 (n) × 1.8)
- INTCR13 interrupt request clear and interrupt enabled

Execute the following processings by INTCR13 vectored interrupt

- Read CTL detection flag
- If CTL detection flag S and L are both “ 1 ”, the gain is large; therefore, decrease the gain for 1.78 dB
- If CTL detection flag S and L are both “ 0 ”, the gain is small; therefore, increase the gain for 1.78 dB
- INTCR13 interrupt disabled

Figure 6-6. Gain Change Timing for FF/REW in Reverse Direction



Execute the following processings by INTCR12 vectored interrupt
- Capture current TM1 count value to CR12 (automatic)
- CR13 (n+1) ← CR12 + (CPT30 (n) × 1.2)
- INTCR13 interrupt request clear and interrupt enabled

Execute the following processings by INTCR13 vectored interrupt
- Read CTL detection flag
- If CTL detection flag S and L are both “ 1 ”, the gain is large; therefore, decrease the gain for 1.78 dB
- If CTL detection flag S and L are both “ 0 ”, the gain is small; therefore, increase the gain for 1.78 dB
- INTCR13 interrupt disabled

6.1.2 CTL amplifier auto gain control processing

The following setting and processing are carried out to perform CTL amplifier auto gain control.

(1) The following setting is carried out at every forward/reverse direction change

- PBCTL signal input edge is set as follows:
  <In forward direction>
  The input edge is generated at rising edge of PBCTL signal.
  <In reverse direction>
  The input edge is generated at falling edge of PBCTL signal.

Remark When the tape mode is EP, INTCR12 vectored interrupt is generated with every PBCTL signal for PLAY, every nine PBCTL signals for CUE/REV, and every eight PBCTL signals for FF/REW at the edge shown above.

(2) The following setting and processing are carried out by INTCR12 vectored interrupt

- The following time is set to compare register 13 (CR13) by INTCR12 vectored interrupt

  - **For PLAY/CUE/REV**
    <In forward direction>
    Set time 70% of PBCTL signal cycle to CR13
    CR13 = CR12 + (CPT30 × 70%)
    <In reverse direction>
    Set time 30% of PBCTL signal cycle to CR13
    CR13 = CR12 + (CPT30 × 30%)

  - **For FF/REW**
    <In forward direction>
    Set time 180% of PBCTL signal cycle to CR13
    CR13 = CR12 + (CPT30 × 180%)
    <In reverse direction>
    Set time 120% of PBCTL signal cycle to CR13
    CR13 = CR12 + (CPT30 × 120%)

Explanation

- CR12 : TM1 count value is captured with every INTCR12 vectored interrupt by PBCTL signal
- CPT30 : TM3 count value is captured with every INTCR12 vectored interrupt by PBCTL signal
  Since CR12 captures TM1 count value and CR13 captures TM3 count value, value need to be set to CR13 after adding up the input clock ratio of TM1 and TM3 (however, in this time, the setting is unnecessary because both have the same clock [f_{CLK}/8]).

- INTCR13 interrupt request clear and interrupt enabled

(3) Processing at INTCR13

- CTL gain control signal detection and gain change
  According to the status of CTL detection flag S and L, gain is changed by ±1 step as follows:
  - If CTL detection flag S and L are both “1”, gain is large; therefore, decrease the gain for 1 step (1.78 dB)
  - If CTL detection flag S and L are both “0”, gain is small; therefore, decrease the gain for 1 step (1.78 dB)
  - If CTL detection flag S is “1” and L is “0”, gain is optimum; therefore, no change is made for the gain.

- INTCR13 interrupt disabled

(4) Processing at PBCTL

- Increase the gain by +5 steps, every time there is no CTL signal and 40 interrupts does not occur continuously at INTCPT13 interrupt (capstan FG interrupt)
- The gain is maximum (1FH) on non-recorded tape

(5) Processing in each mode transition

- Set the optimum gain previously measured in each mode in every mode transition of each mode (equipment operation such as PLAY and CUE, and tape mode such as EP and SP)
  (this processing is optional; however, it has the advantage that the optimum gain can be quickly achieved.)

[MEMO]

CHAPTER 7 VISS DETECTION

The following shows the VISS detection method.

7.1 What is VISS

VISS stands for “VHS Index Search System”. In VHS, cue code is set by varying the duty ratio of control signal to be recorded on control track.

Each VISS data is specified as shown in Table 7-1. The cue code as index information is set by data sequence of control signal as shown in Figure 7-1.

Table 7-1. VISS Data

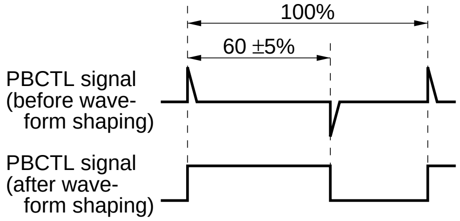
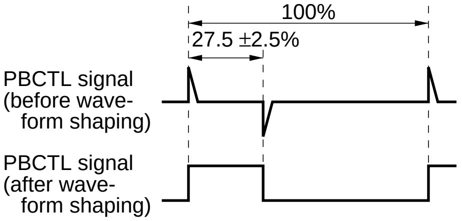
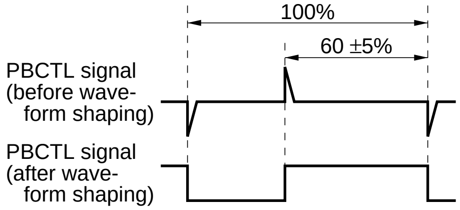
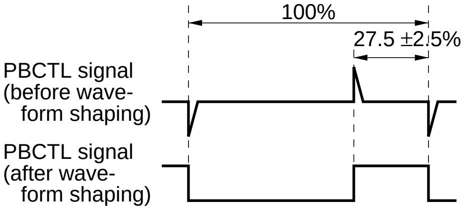
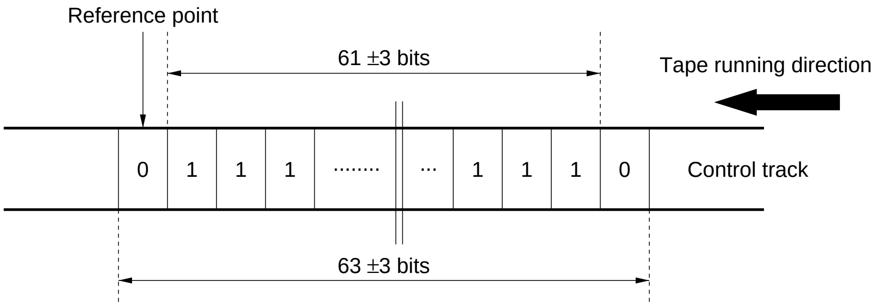
Data	“ 0 ”	“ 1 ”
Waveform in forward direction	 100% 60 ±5% PBCTL signal (before wave-form shaping) PBCTL signal (after wave-form shaping)	 100% 27.5 ±2.5% PBCTL signal (before wave-form shaping) PBCTL signal (after wave-form shaping)
Waveform in reverse direction	 100% 60 ±5% PBCTL signal (before wave-form shaping) PBCTL signal (after wave-form shaping)	 100% 27.5 ±2.5% PBCTL signal (before wave-form shaping) PBCTL signal (after wave-form shaping)

Figure 7-1. VISS Cue Code



VISS write (cue code write) is carried out at the following timings.

- When starting recording (except joint recording)
- When starting programmed recording
- When index writing by pushing down INDEX key

7.2 VISS Detection

7.2.1 VISS detection method

VISS detection is performed using macro service in data pattern discrimination mode by playback control (PBCTL) signal edge interrupt (INTCR12).

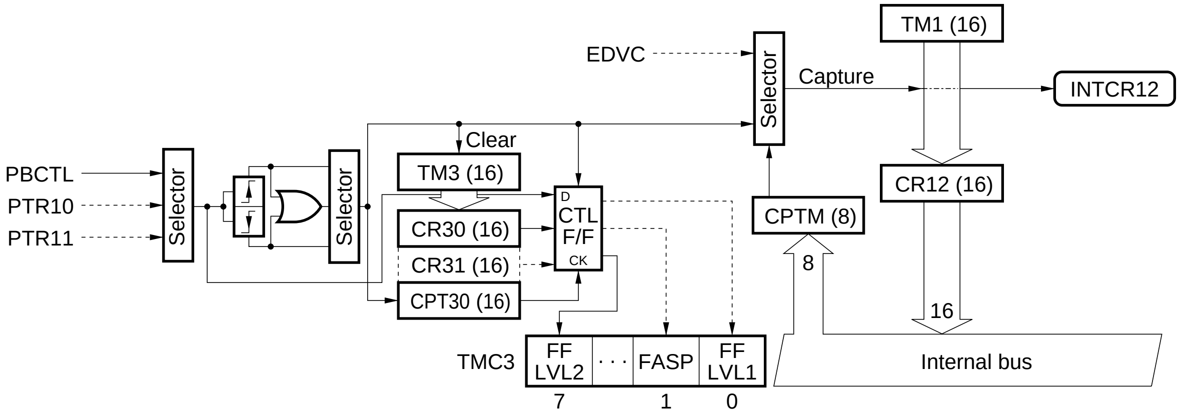
INTCR12 interrupt also performs PBCTL signal frequency division.

(1) About VISS detection method

In VISS detection, PBCTL signal level is taken at 43.75% (in forward direction) or 56.25% (in reverse direction) of one PBCTL signal cycle as VISS detection point. According to the level, if the level is high, “0” is set, if it is low, “1” is set. It is judged that VISS signal exists when “0” is detected 10 times after “1” is consecutively detected 15 times (According to the specification of system controller, it may be judged VISS signal exists if “1” is consecutively detected several times).

The μPD784915 is provided with timer 3 (TM3) and capture register 30 (CPT30) to find a cycle, compare register (CR30) to store VISS detection point, and control flip flop (CTL F/F) to take in control signal level at detection point, in order to keep up with the change of tape running speed, so that it can perform VISS detection.

Figure 7-2. VISS Detection Circuit (Pulse Width Detection Circuit) Configuration



μPD784915 uses macro service in data pattern discrimination mode to perform VISS detection.

The comparison data to perform comparison with the data stored in buffer area is set to an address indicated with comparison area pointer (not only program space in memory but also internal RAM space can be specified as the comparison area).

(2) About macro service in data pattern discrimination mode (VISS detection mode)

This is a macro service to sequentially store the output from control flip flop (CTL F/F) in the pulse detection circuit (timer 3) in the Super Timer Unit into the buffer set in the RAM area with left shift.

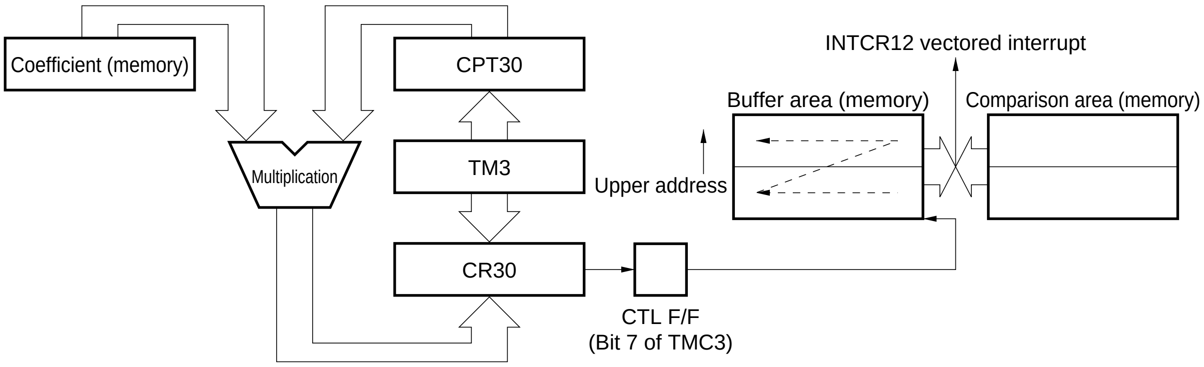
The timer measures the PBCTL signal pulse duty from the CTL amplifier circuit, and latches “1” to CTL F/F if the duty is larger than the value previously set, and “0” if the duty is smaller.

Caution Take note that “1” and “0” of the VISS signal are reversed.

The contents of SFR (bit 7 of timer control 3) specified with SFR pointer 1 is buffer area left shifted at interrupt generation. At the same time, the data of buffer area and comparison area are compared, and a vectored interrupt is generated if they coincide (macro service counter is decremented and if it becomes 0, a vectored interrupt is also generated).

By option specification (bit 5 of macro service mode register = “1”), the operation is made so that the contents of SFR [capture trigger 30 (CPT30)] specified with SFR pointer 2 is multiplied with the coefficient and stored to SFR [compare register 30 (CR30)] specified with SFR pointer 3 (automatic updating of discrimination threshold when tape speed is varying).

Figure 7-3. Data Pattern Discrimination Mode Block Configuration



- **Explanation**

CPT30 : PBCTL signal cycle enters

Coefficient : multiplier to find detection point enters

CR30 : detection point (the result of CPT30 multiplied with coefficient) enters

Buffer area : VISS signal data with left shift enters

Comparison area : VISS detection pattern enters

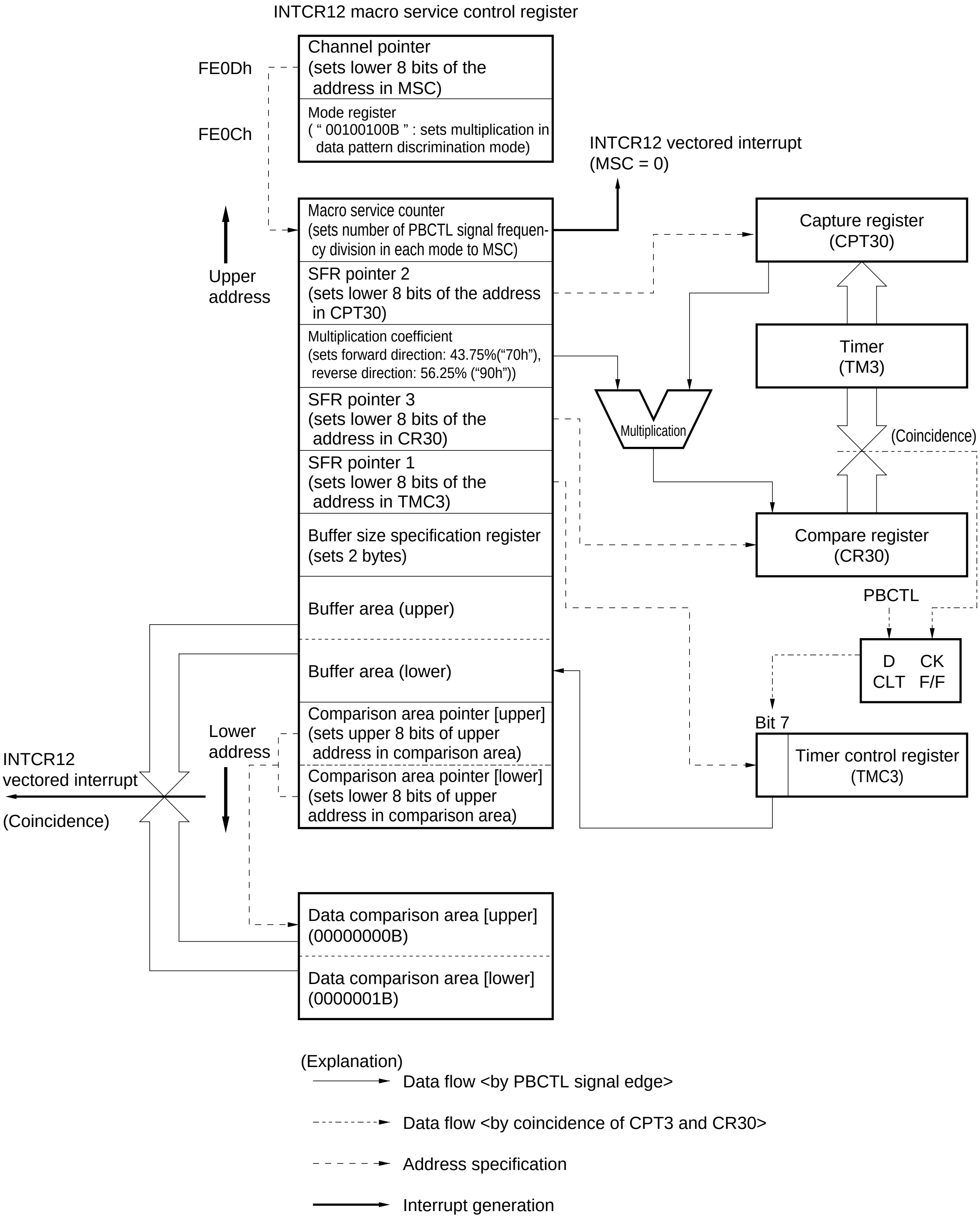
Vectored interrupt is generated when either one of the following conditions is satisfied.

<1> If the contents of macro service counter 8 (MSC) is 0 (if interrupt request is generated for the number of times set in MSC).

<2> If the data stored in buffer area coincides with the data in the comparison area separately set.

Figure 7-4 shows the addressing and data setting in data pattern discrimination mode.

Figure 7-4. Addressing and Data Setting in Data Pattern Discrimination Mode


INTCR12 macro service control register
FE0Dh
FE0Ch
Channel pointer (sets lower 8 bits of the address in MSC)
Mode register (" 00100100B " : sets multiplication in data pattern discrimination mode)
INTCR12 vectored interrupt (MSC = 0)
Upper address
Macro service counter (sets number of PBCTL signal frequency division in each mode to MSC)
SFR pointer 2 (sets lower 8 bits of the address in CPT30)
Multiplication coefficient (sets forward direction: 43.75%("70h"), reverse direction: 56.25% ("90h"))
SFR pointer 3 (sets lower 8 bits of the address in CR30)
SFR pointer 1 (sets lower 8 bits of the address in TMC3)
Buffer size specification register (sets 2 bytes)
Buffer area (upper)
Buffer area (lower)
Comparison area pointer [upper] (sets upper 8 bits of upper address in comparison area)
Comparison area pointer [lower] (sets lower 8 bits of upper address in comparison area)
Lower address
INTCR12 vectored interrupt
(Coincidence)
Capture register (CPT30)
Timer (TM3)
Multiplication
(Coincidence)
Compare register (CR30)
PBCTL
D CK
CLT F/F
Bit 7
Timer control register (TMC3)
Data comparison area [upper] (00000000B)
Data comparison area [lower] (0000001B)
(Explanation)
Data flow <by PBCTL signal edge>
Data flow <by coincidence of CPT3 and CR30>
Address specification
Interrupt generation

7.2.2 VISS detection processing

The following setting and processing are carried out to perform VISS detection.

(1) Macro service initialization is performed before starting VISS detection

- Set data with data pattern discrimination mode multiplication (“14H”) to mode register
- Set lower 8 bits of the address in macro service counter (MSC) to channel pointer
- Set buffer area size specification register to 2 bytes (“02H”)
- Set clear (“0FFFFH”) to 2 bytes of buffer area
- Set lower 8 bits (“3BH”) of the address in timer control register 3 (TMC3) to SFR pointer 1
- Set lower 8 bits (“56H”) of the address in timer 3 capture register 0 (CPT30) to SFR pointer 2
- Set lower 8 bits (“5CH”) of the address in timer 3 compare register 0 (CR30) to SFR pointer 3
- Set data comparison area address to comparison area pointer
- Set comparison data (“0001H”) in comparison area

Remark The value in the comparison setting area (“0001H”) means that VISS data “0” is entered once after VISS data “1” is entered 15 times.

Caution Duty detection malfunction prevent circuit control (TMC3.6) is made operation enable for preventing VISS signal malfunction. Therefore, take note that unless VISS data “0” is entered twice consecutively, it is not judged that data “0” is entered (if VISS “1” data is entered once, it is judged that “1” is entered).

(2) The frequency division of CTL signal is also set in each mode transition (equipment operation such as PLAY/CUE, and macro service counter (MSC) of tape mode such as EP and SP)

Caution PBCTL signal frequency division is also performed in INTCR12 interrupt.
In sets not provided with VISS detection, the counter mode macro service is used for PBCTL signal frequency division while, in sets provided with VISS detection, the data pattern discrimination mode macro service is used.

(3) The following setting is carried out at every forward/reverse direction change

- Set each multiplication coefficient as follows:
  <In forward direction>
  Set 0.4375 time multiplier (70H) which is the value of the 43.75% position of PBCTL signal.
  <In reverse direction>
  Set 0.5625 time multiplier (90H) which is the value of the 56.25% position of PBCTL signal.

Remark <1> The middle point of the percentage of VISS data “0” and “1” is adopted for multiplication coefficient.
In forward direction ... (60% + 27.5%) ÷ 2 = 43.75%
In reverse direction ... (40% + 72.5%) ÷ 2 = 56.25%

<2> The multiplication coefficient set value is set as follows:
In forward direction ... 0.4375 × 256 = 112 (70H)
In reverse direction ... 0.5625 × 256 = 144 (90H)

- Set PBCTL signal input 4 edge as follows:
  <In forward direction>
  Generated at PBCTL signal rising edge
  <In reverse direction>
  Generated at PBCTL signal falling edge

(4) INTCR12 macro service processing (automatically executed by macro service)

The following INTCR12 macro service processing is automatically executed with trigger by PBCTL signal edge.

- The result of the automatic multiplication of the value of CPT30 and multiplication coefficient is set to CR30.

  (VISS detection setting)

  <In forward direction> (refer to **Figure 7-5**)

  Set 43.75% time of PBCTL signal cycle to CR30.

  CR30 ← CPT30 × 43.75%

  <In reverse direction> (refer to **Figure 7-6**)

  Set 56.25% time of PBCTL signal cycle to CR30.

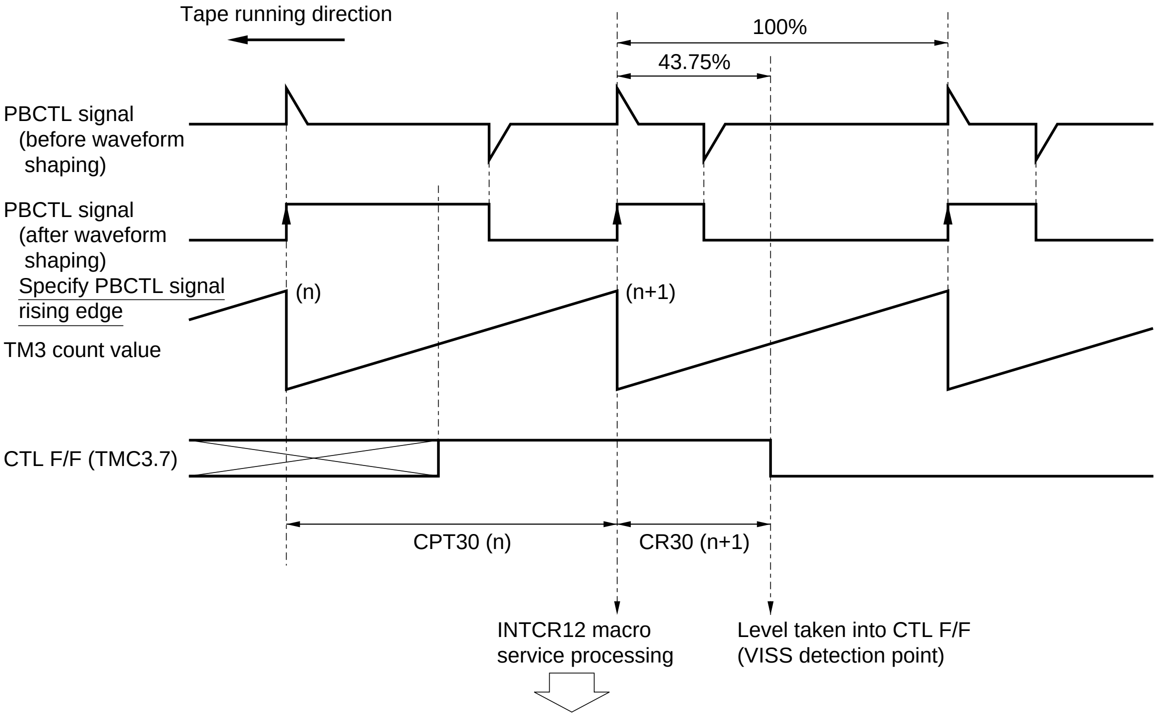
  CR13 ← CPT30 × 56.25%

  (Explanation) CPT30:

  TM3 count value (a cycle of PBCTL signal) is captured at every PBCTL signal edge interrupt.

- Value of CLT F/F (TMC3.7) is left shifted to buffer area (the entire buffer is also left shifted).
- Compared with the value in comparison area, and if they coincide, INTCR12 vectored interrupt is generated. INTCR12 vectored interrupt is also generated when macro service counter is “0”.

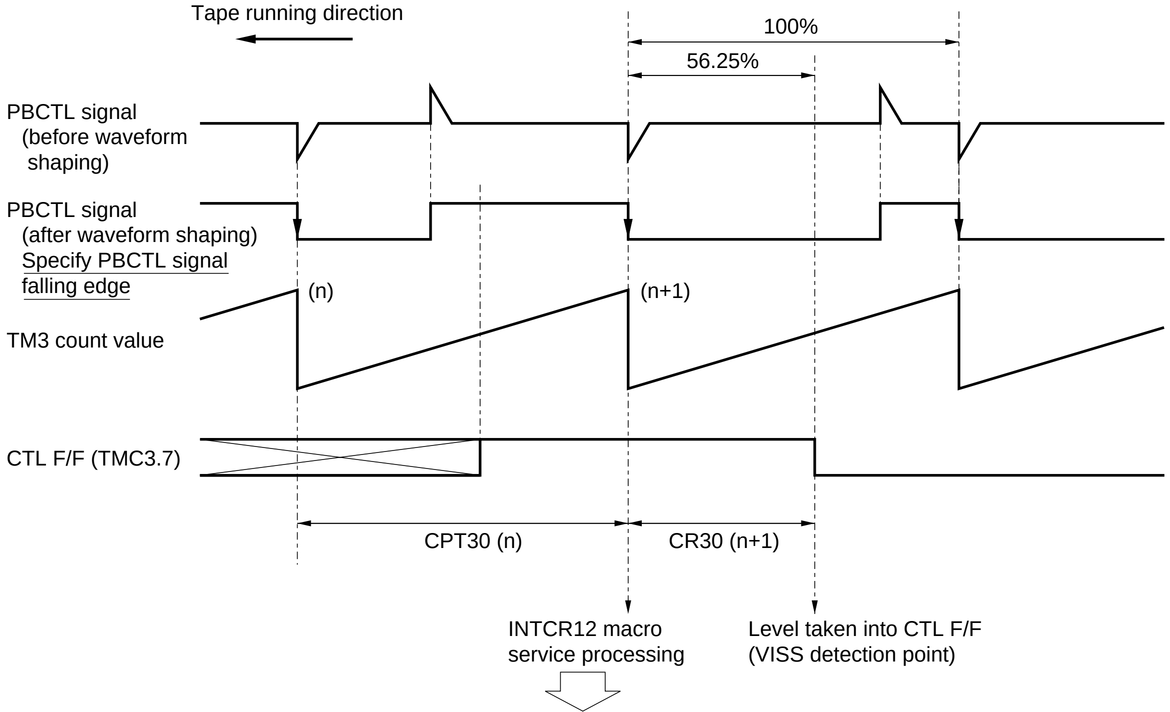
Figure 7-5. INTCR12 Macro Service Processing in Forward Direction


Tape running direction
100%
43.75%
PBCTL signal (before waveform shaping)
PBCTL signal (after waveform shaping)
Specify PBCTL signal rising edge
(n)
(n+1)
TM3 count value
CTL F/F (TMC3.7)
CPT30 (n)
CR30 (n+1)
INTCR12 macro service processing
Level taken into CTL F/F (VISS detection point)

The following processings are automatically performed with INTR12 macro service

· CR30 (n+1) ← CPT30 (n) × 0.4375
· Left shift CTL F/F (TMC3.7) value to buffer area
· Compare the values in comparison area and buffer area if they coincide,
  INTCR12 vectored interrupt is generated
  (if MSC is “ 0 ”, INTCR12 vectored interrupt is also generated).

Figure 7-6. INTCR12 Macro Service Processing in Reverse Direction


Tape running direction
100%
56.25%
PBCTL signal (before waveform shaping)
PBCTL signal (after waveform shaping)
Specify PBCTL signal falling edge
(n)
(n+1)
TM3 count value
CTL F/F (TMC3.7)
CPT30 (n)
CR30 (n+1)
INTCR12 macro service processing
Level taken into CTL F/F (VISS detection point)

The following processings are automatically performed with INTR12 macro service
· CR30 (n+1) ← CPT30 (n) × 0.5625
· Left shift CTL F/F (TMC3.7) value to buffer area
· Compare the values in comparison area and buffer area if they coincide, INTCR12 vectored interrupt is generated (if MSC is “ 0 ”, INTCR12 vectored interrupt is also generated).

(5) The following processing is performed at INTCR12 vectored interrupt

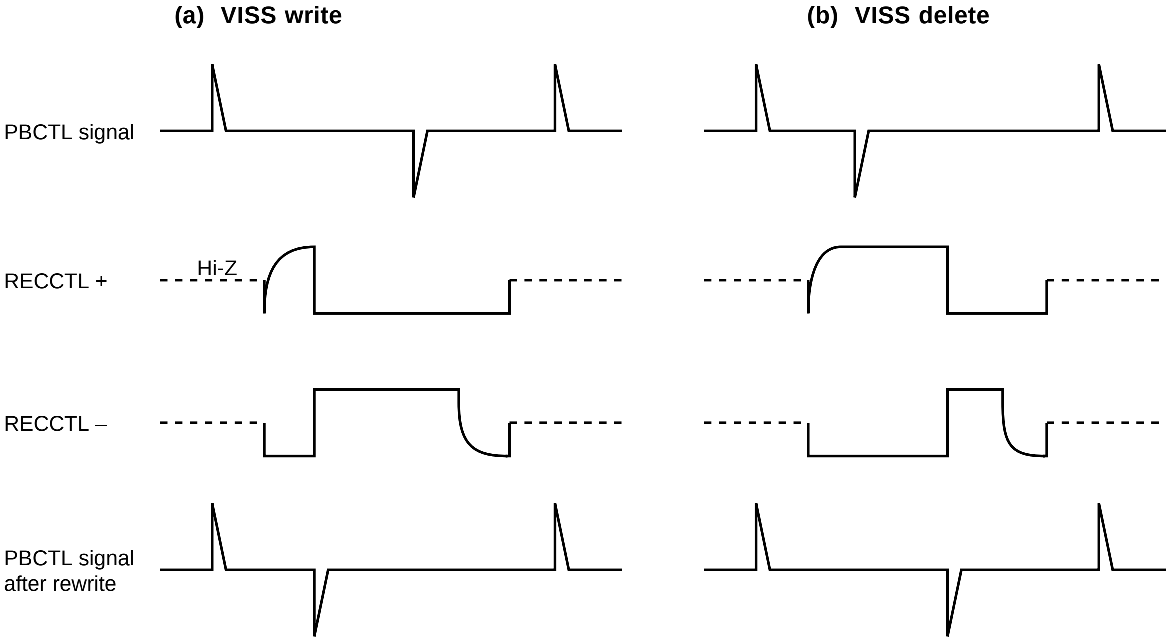
- Since interrupt is generated either at every CTL signal frequency division or coincidence of data comparison, the following method is taken to judge which one is generated.
  - If macro service counter (MSC) is not “00H”, it is judged as <interrupt by coincidence of VISS data comparison>
  - If macro service counter (MSC) is “00H” and the contents of buffer area is “0001H” (the same value as that of comparison discrimination area) , it is judged as <interrupt by coincidence of VISS data comparison>
- <In the case of interrupt by coincidence of VISS data comparison>
  - Set macro service counter value again
  - Set macro service interrupt enable
  - VISS signal is detected. Set VISS detection flag and notify system controller processing
- <In the case of interrupt not by coincidence of VISS data comparison>
  - Set macro service counter value again
  - Set macro service interrupt enable

7.3 VISS Rewrite

7.3.1 VISS rewrite method

Newly writing VISS signal on recorded tape or erasing VISS signal already written is called VISS rewrite. Rewrite is performed by rewriting PBCTL signal as shown in Figure 7-7.

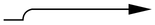
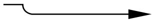
Figure 7-7. VISS Rewrite



The recording control signal (RECCTL) driver of μPD784915 has rewrite mode used for rewriting VISS signal. RECCTL driver internally holds sequence data, and the sequence is updated with specific interrupt as trigger.

RECCTL driver sequence in rewrite mode operates RECCTL+ Pin and RECCTL– pin as shown in Table 7-2. Therefore, VISS signal rewrite is realized considering only the interrupt generation timing, which is a trigger signal.

Table 7-2. RECCTL Driver Rewrite Mode Sequence

Sequence	RECCTL+	RECCTL–
0	High impedance	
1		Low level
2	Low level	High level
3	High level	

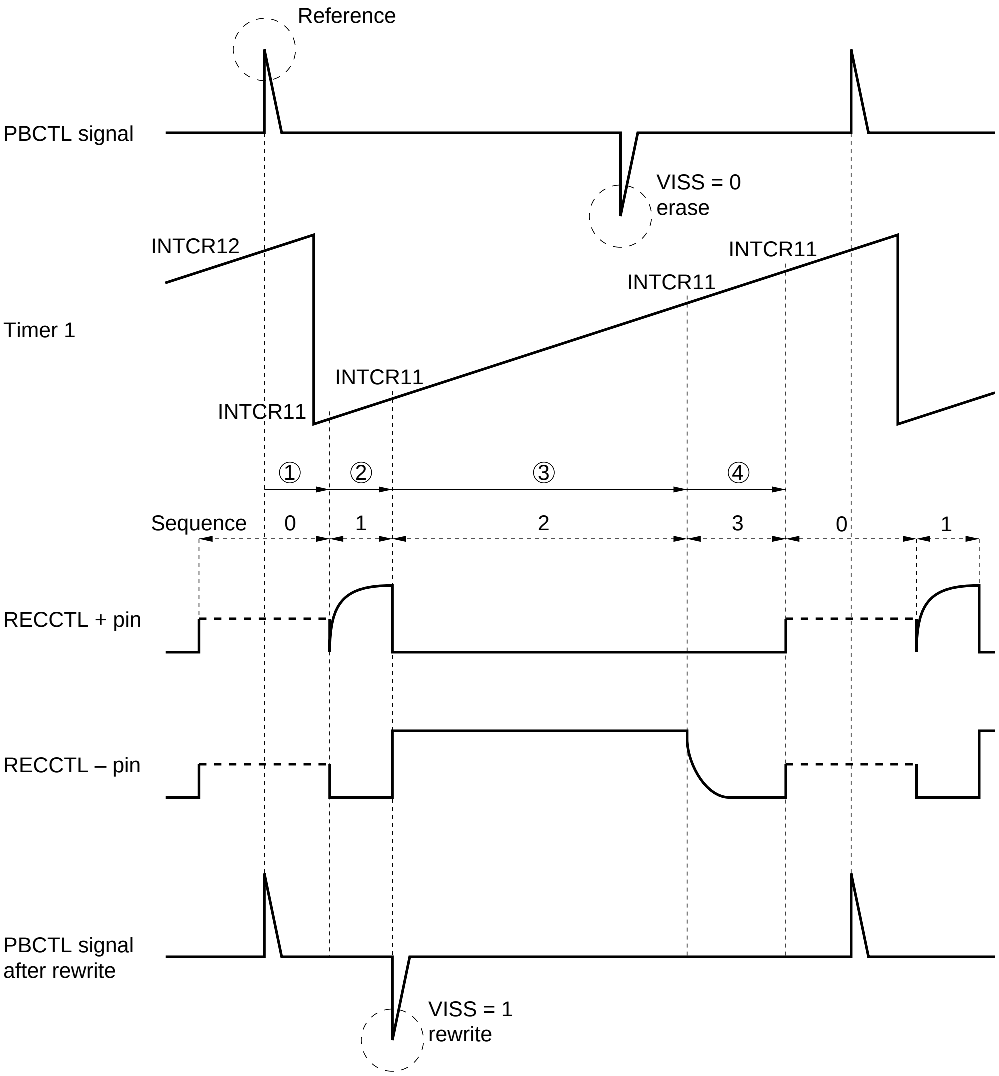
7.3.2 VISS rewrite processing

Rewrite processing is realized using INTCR11 and INTCR12.

For rewrite timing, trigger timing is set to compare register 11 (CR11) using PBCTL signal interrupt INTCR12 as reference.

Figure 7-8 shows VISS = 1 signal rewrite operation timing.

Figure 7-8. VISS = 1 Signal Rewrite Operation Timing Chart



Timing <1> in Figure 7-8 is PBCTL signal rising. At this point, the sequence is initialized, and the changing point from sequence 0 to sequence 1 is found with timer 1 (TM1) using the captured value as reference value, stored in compare register 11 (CR11), and INTCR11 interrupt is enabled.

For the following timing <2> and <3> INTCR11 interrupt, each changing point to sequence 2 and 3 is found on timer 1 and stored in CR11. Rewrite is completed at timing <4> and INTCR11 interrupt is disabled.

Each timing in NTSC is as shown in Table 7-3. VISS = 1 signal and VISS = 0 signal have different timings.

Table 7-3. VISS Write Operation Timings

Timings	VISS = 1 Write	VISS = 0 Write
PBCTL rising <1> → <2>	5 ms	5 ms
<2> → <3>	4.176 ms	20.021 ms
<3> → <4>	16.192 ms	5.345 ms
<4> → PBCTL rising	5 ms	5 ms

Hi-Z cancellation timing from PBCTL rising is not specified for the period between the timing to become Hi-Z <4> and PBCTL rising. However, it is set as 5 ms for the sake of convenience, assuming it as the timing approximately a half way from PBCTL high pulse to PBCTL low pulse when VISS is 1.

CHAPTER 8 PROGRAM LIST

This chapter lists programs of this application software.

```
$     DEBUG
swOLD EQU     0

;----------------------------
   PUBLIC, EXTRN Declaration
;----------------------------

      ;------------------------------
              PUBLIC
      ;------------------------------

      ;/////  PROCESS ///////

      PUBLIC  VPT2_000                ; INTCPT2   DRUM FG INTERRUPTION
                                      ; PROCESS ROUTINE
      PUBLIC  VR10_000                ; INTCR10   DRUM PHASE ERROR DETECTION
                                      ; INTERRUPTION PROCESS
      PUBLIC  VPT3_000                ; INTCRP3   CAPSTAN FG
                                      ; INTERRUPTION
      PUBLIC  VR12_000                ; INTCR12   CAPSTAN PHASE ERROR
                                      ; DETECTION INTERRUPT
      PUBLIC  VR00_000                ; INTCR00   QUASI Vsync
                                      ; TIMING SETTING
      PUBLIC  VR02_000                ; INTCR02   QUASI Vsync
                                      ; TIMING SETTING
      PUBLIC  VR02_000                ; INTCR02   QUASI Vsync
                                      ; TIMING SETTING
      PUBLIC  VR13_000                ; %INTCR13  CTL DETECTION & OUTPUT SETTING
                                      ; %CTL

      ;/////  SUBROUTINE /////

      PUBLIC  YVTBL_00                ; SERVO DATA SETTING SUB

      ;/////  RAM ///////////////

      PUBLIC  RVSCR12                 ; INTCR12 MACRO SERVICE MODE REGISTER
      PUBLIC  RVCCR12                 ; INTCR12 MACRO SERVICE
                                      ; CHANNEL POINTER
      PUBLIC  RVMCMPP                 ;%INTCR12 MACRO SERVICE COMPARE AREA
                                      ; POINTER
      PUBLIC  RVB2CR12                ; INTCR12 MACRO SERVICE BUFFER
                                      ; AREA 2(L)
      PUBLIC  RVB1CR12                ; INTCR12 MACRO SERVICE BUFFER
                                      ; AREA 1(H)
      PUBLIC  RVBFREG                 ; INTCR12 MACRO SERVICE BUFFER SIZE REG
      PUBLIC  RVMSFRP1                ;%INTCR12 MACRO SERVICE SFR POINTER 1
      PUBLIC  RVMSFRP3                ;%INTCR12 MACRO SERVICE SFR POINTER 3
      PUBLIC  RVMKEISU                ;%INTCR12 MACRO SERVICE KEISU AREA
      PUBLIC  RVMSFRP2                ;%INTCR12 MACRO SERVICE SFR POINTER 2
      PUBLIC  RVMCCR12                ; INTCR12 MACRO SERVICE COUNTER AREA
      PUBLIC  RVMCMPD                 ;%INTCR12 MACRO SERVICE COMPARE DATA

      ;////   MACRO SERVICE DATA ////  ;%
      PUBLIC  PTN_FF                  ;%POSITIVE DIRECTION MULTIPLIER
                                        COEFFICIENT (0.4375) DATA
      PUBLIC  PTN_REW                 ;%REVERSE DIRECTION MULTIPLIER
                                        COEFFICIENT (0.5625) DATA
      PUBLIC  DT_CMP                  ;%CR12 COMPARISON DATA
                                      ;%VISS
      PUBLIC   RVCPT3   ;_L               ; CPT3 LOW DATA MEMORY
      PUBLIC   RVSRVCD                    ; SERVO CODE AREA
      PUBLIC   RVCPRF   ;_L               ; CAPSTAN PHASE REFERENCE LOW

      PUBLIC   RVCPT2   ;_L               ; CPT2 LOW DATA MEMORY
      PUBLIC   RVCPT2   ;_M               ; CPT2 MIDDLE & HIGH DATA MEMORY

      PUBLIC   RVCPT1                     ; % CPT1
      PUBLIC   RVCPT0                     ; % CPT0

      PUBLIC   RVFSRV_2                   ; SERVO DATA FLAG AREA 2

      PUBLIC   RVCEVFG                    ; SP/LP/EP AUTO DETECT CFG DIVIDE
                                          ; COUNTER

      PUBLIC   RVPSVCNT                   ; QUASI V SIGNAL COUNTER

      PUBLIC   RVCRAM                     ; MACRO SERVICE COUNT DATA

      PUBLIC   RVBCR10                    ; CR10 DATA BUFFER AREA

      ;/////   BIT ///////////////

      PUBLIC   FVPBLP                     ; RUNNING MODE FPBLP FPBSEP
      PUBLIC   FVPBSEP                    ;    SP :        0      0
                                          ;    LP :        1      0
                                          ;    EP :        0      1
                                          ;    PAL:        1      1

      PUBLIC   FVDOUT                     ; QUASI Vsync OUTPUT

      PUBLIC   FVFLCTL                    ; SET PBCTL MISSING FLAG

      ;/////   CONSTANT /////

      PUBLIC   VSP
      PUBLIC   VLP
      PUBLIC   VSLP
      PUBLIC   VPAL

$     EJECT

      ;------------------------------
      ;        EXTRN
      ;------------------------------

      ;/////   PROCESS ///////

      EXTRN    SR12_000

      ;/////   SUB ///////////////
      EXTRN    YPGADCHG                   ; SET PG VALUE

      EXTRN    YSA01_R1                   ; 1SEC TIMER START FOR AUTO-TRACKING

      ;/////   RAM ///////////////

      EXTRN    RSNOW                      ; TRANSITION NOW MODE
```

```
        EXTRN    RSNEXT                     ; TRANSITION NEXT MODE

        EXTRN    RSCFG90C                   ; CFG 90 PULSE COUNTER CHECK

        EXTRN    RNSTIMO                    ; TRANSITION TIMER AREA

        EXTRN    RSFRSPED                   ; CAPSTAN FF/REW SPEED LEVEL

        ;/////   BIT ///////////////

        EXTBIT   FSVMOFRQ                   ; V-MUTE OFF REQUEST

        EXTBIT   FSMDCHG                    ; FLAG DURING MODE TRANSITION

        EXTBIT   FSEICPT2                   ; INTCPT2 ENABLE REQUEST FLAG

        EXTBIT   FSDRMON                    ; DRUM ON/OFF FLAG

        EXTBIT   FHIFIM                     ; Hi-Fi MODE FLAG

        EXTBIT   FSVISSI                    ; INDEX SEARCH   MODE FLAG

        EXTBIT   FSVISSO                    ; ONCE MORE SEARCH MODE FLAG

        EXTBIT   FSVISSME                   ; VISS MARK/ERASE MODE FLAG

        EXTBIT   FSVISTR                    ; VISS SEARCH START FLAG

        EXTBIT   FSVISSOK                   ; VISS DETECTION FLAG

        EXTBIT   FSAFRQ                     ; RFS DOWN EDGE FOR AUTO-TRACKING

        EXTBIT   FSCAPON                    ; CAPSTAN ON FLAG

        EXTBIT   FSCRRFRQ                   ; CAPSTAN REVERSE RFS EDGE ON REQUEST
                                            ; FLAG

        EXTBIT   FNSTENA                    ; SEARCH DETECT DI TIMER END FLAG

        EXTBIT   FSAEND                     ; AUTO TRACKING END FLAG

        EXTBIT   FNSTENO                    ; TRANSITION TIMER END FLAG

        EXTBIT   FSDFG                      ; DFG EDGE DETECTION FLAG

        EXTBIT   FSSPDCHG                   ; TAPE SPEED CHANGE FLAG

        ;/////   PORT ///////////////

        EXTBIT   PRFS                       ; RF SWITCHING PULSE

        EXTBIT   PQVD                       ; V-MUTE

        EXTBIT   PCAPFWD                    ; CAPSTAN FORWARD/REVERSE

        EXTBIT   PCAPF_R                    ; CAPSTAN FORWARD/REVERSE (to DECK)

$       INCLUDE  (PORT. INC)                ; %

        EXTBIT   FPCAPF_R                   ; FLAG FOR PORT REFRESH
```

```
      EXTBIT   FPQVD                    ; FLAG FOR PORT REFRESH

      ;/////   CODE /////////////

      EXTRN    CSMLOAD                  ; TAPE LOADING
      EXTRN    CSMPLAY                  ; PLAY

      EXTRN    CVPLAY                   ; PLAY
      EXTRN    CVFFRW2H                 ; 2Hrs PLAY (PH FIX)
      EXTRN    CVFFRW6H                 ; 6Hrs PLAY (PH FIX)
      EXTRN    CVFFRWX3                 ; 3Hrs PLAY x3 (PH FIX)
      EXTRN    CVFFREW                  ; FF/REW (PH FIX)
      EXTRN    CVCUE                    ; CUE        (VD OUT)
      EXTRN    CVREV                    ; REVIEW (VD OUT)
      EXTRN    CVSTILL                  ; STILL      (VD OUT)
      EXTRN    CVCUPL                   ; CUE → PLAY (VD OUT)
      EXTRN    CVRVS                    ; RVS PLAY  (VD OUT & PH FIX)
      EXTRN    CVFR6HVD                 ; 6Hrs PLAY (VD OUT & PH FIX)

$     EJECT

;--------------------------------
;★★★  SERVO RELATED EQU AREA
;--------------------------------

      ;*** SERVO DATA AREA ***

VSEQU1         DSEG    SADDR            ;%

      ;*** SERVO REFERENCE DATA ***
RVDFRF:                DS      3        ; DRUM SPEED REFERENCE
RVCPT2:                DS      3        ; CPT2 DATA MEMORY
RVCPT22:               DS      3        ; FOR DEBUG

      ;*** CAPTURE DATA MEMORY ***

RVCPT0:                DS      3        ; CPT0 DATA MEMORY
RVCPT1:                DS      3        ; CPT1 LOW DATA MEMORY

      ;*** SERVO ERROR DATA ***

RVERDF:                DS      2        ; DRUM SPEED ERROR
RVERDF_1:              DS      2        ; DRUM SPEED ERROR(–1)

RVERDP:                DS      2        ; DRUM PHASE ERROR
RVERDP_1:              DS      2        ; DRUM PHASE ERROR(–1)

      ;--- PWM OUTPUT BIAS DATA AREA ---

RVDBAS:                DS      2
      ;RVDBAS_L:       DS      1        ; DRUM BIAS LOW BYTE
      ;RVDBAS_H:       DS      1        ; DRUM BIAS HIGH BYTE

      ;--- DRUM PHASE FILTER UNKNOWN-QUANTITY ---

RVERDP_Y:              DS      2
RVERDP_bY:             DS      4
RVERDF_Y               DS      2
RVERDF_bY              DS      4
```

```
$     EJECT
      ;*** FILTER MEMORY ***
      ;*** FILTER COEFFICIENT DATA *** ; %FILTER PRODUCT-SUM OPERATION WORK AREA

B_buf:                DS      2        ; (LOOP GAIN)
                      DS      2        ; (FILTER COEFFICIENT ″ a ″) x
                                       ; (LOOP GAIN)
                      DS      2        ; (FILTER COEFFICIENT ″ b ″) x
                                       ; (OUTPUT) x (–1)

RVC_Kmp:              DS      2        ; CAPSTAN LOOP GAIN ″ G1 ″
$EJECT

      ;*** CAPSTAN DATA ***

RVCFRF:               DS      3        ; CAPSTAN SPEED REFERENCE
RVCPRF:               DS      2        ; CAPSTAN PHASE REFERENCE

      ;*** SERVO ERROR DATA ***

RVERCF:               DS      2        ; CAPSTAN SPEED ERROR

RVERCP:               DS      2        ; CAPSTAN PHASE ERROR
RVERCP_1:             DS      2        ; CAPSTAN PHASE ERROR(–1)

RVERCMX:              DS      2        ; CAPSTAN SPEED & PHASE MIXED
                                       ; ERROR
RVERCMX_1             DS      2        : CAPSTAN SPEED & PHASE MIXED
                                       ; ERROR(–1)
      ;*** CAPTURE DATA MEMORY ***

RVCPT3 :              DS      3        ; CPT3 DATA MEMORY

      ;--- PWM OUTPUT BIAS DATA AREA ---

RVCBAS:               DS      2        ; CAPSTAN BIAS LEVEL

      ;--- CAPSTAN PHASE FILTER UNKNOWN-QUANTITY ---

RVERCP_Y:             DS      2        ; CAPSTAN PHASE ″ Y ″
RVERCP_by:            DS      4        ; CAPSTAN PHASE ″ b x Y ″

      ;--- CAPSTAN SPEED/PHASE MIX FILTER UNKNOWN-QUANTITY ---

RVERCMX_Y:            DS      2        ; CAPSTAN SPEED/PHASE MIX
                                       ; ″ Y ″
RVERCMX_bY:           DS      4        ; CAPSTAN SPEED/PHASE MIX
                                       ; ″ b x Y ″

RVSRVCD:              DS      1        ; SERVO CODE AREA

      FVDOUT   EQU    RVSRVCE.7        ; QUASI Vsync OUTPUT
      FVPHFX   EQU    RVSRVCD.6        ; PHASE CONTROL IS NOT PERFORMED

RVCRAM:               DS      1        ; MACRO SERVICE COUNT DATA

RVCEVFG:              DS      1        ; SP/LP/EP AUTO DETECT CFG
                                       ; DIVIDE COUNTER
```

```
RVSLPCH:               DS      1        ; SP/LP/EP CHATTERING COUNT
                                        ; AREA

RVFSRV_2:              DS      1        ; SERVO DATA FLAG AREA 2

        FVDFE10 EQU    RVFSRV_2.7       ; DRUM SPEED ERROR 10% OVER FLAG
        FVCFERR EQU    RVFSRV_2.6       ; CAPSTAN SPEED ERROR (+/-)SIGN FLAG
        FVCPLCK EQU    RVFSRV_2.5       ; CAPSTAN PHASE LOCK FLAG
                                        ; 0:LOCK 1: UNLOCK
        FVCFE05 EQU    RVFSRV_2.4       ; CAPSTAN SPEED ERROR 5% OVER FLAG
        FVHQVDT EQU    RVFSRV_2.3       ; QVD HIGH TIMING FLAG
        FVFLCTL EQU    RVFSRV_2.2       ; PBCTL ERROR FLAG
        FVPBLP  EQU    RVFSRV_2.1       ; PB LP DATA FLAG
        FVPBSEP EQU    RVFSRV_2.0       ; PB SP/EP DATA FLAG

;                      FPBLP FPBSEP
;               SP :     0      0
;               LP :     1      0
;               EP :     0      1
;               PAL:     1      1

RVPSVCNT:              DS      1        ; QUASI V SIGNAL COUNTER

        ;///// CAPSTAN KV/KP ////////////////

RVC_Kvp:               DS      2        ;
                                        ; Kv/Kp REAL NUMBER (1 BYTE) + DECIMAL
                                          FRACTION (1 BYTE)

        ;*** CR10 DATA BUFFER AREA ***

                                        ; %930
RVBCR10:               DS      2        ; CR10 BUFFER REG LOW

        ;--- MODE NTSC/MODE PAL FOR ROM READ ---

        MODE NTSC      EQU     0        ;
        MODE PAL       EQU     2        ;
        MODE LP        EQU     4        ;

NTSC_PAL:              DS      1        ; FOR SPEED UP

        ;*** INTCR12 MACRO SERVICE DATA ***

MCRAREA          DSEG  AT 0FE0CH        ;%
RVSCR12:         DS    1                ; INTCR12 MACRO SERVICE MODE REGISTER
RVCCR12:         DS    1                ; INTCR12 MACRO SERVICE CHANNEL
                                        ; POINTER

RVMCMPP:         DS    2                ;%INTCR12 MACRO SERVICE COMPARE AREA
                                        ; POINTER
RVB2CR12:        DS    1                ; INTCR12 MACRO SERVICE BUFFER AREA 2(L)
RVB1CR12:        DS    1                ; INTCR12 MACRO SERVICE BUFFER AREA 1(H)
RVBFREG:         DS    1                ; INTCR12 MACRO SERVICE BUFFER SIZE REG
RVMSFRP1:        DS    1                ;%INTCR12 MACRO SERVICE SFR POINTER 1
RVMSFRP3:        DS    1                ;%INTCR12 MACRO SERVICE SFR POINTER 3
RVMKEISU:        DS    1                ;%INTCR12 MACRO SERVICE KEISU AREA
RVMSFRP2:        DS    1                ;%INTCR12 MACRO SERVICE SFR POINTER 2
RVMCCR12:        DS    1                ; INTCR12 MACRO SERVICE COUNTER
RVMCMPD:         DS    2                ;%INTCR12 MACRO SERVICE COMPARE AREA
```

```
PTN_FF         EQU    70H              ; %POSITIVE DIRECTION MULTIPLIER
                                       ; COEFFICIENT (0.4375)
PTN_REW        EQU (PTN_FF XOR 0FFH)+1 ; %REVERSE DIRECTION MULTIPLIER
                                       ; COEFFICIENT (0.5625)
DT_CMP         EQU    0000H or 0001H   ; %CR30 COMPARISON DATA
                                       ; %VISS
       ;--- FOR DEBUG ---- %%%%

DEB    DSEG    UNIT

       SAVE_CNT:      DS      2        ;
       SAVE_AREA:     DS      256      ;

       ;--- SP/LP/EP PAL MODE CODE ----

CVSP   EQU     00H                     ; SP MODE
CVSLP  EQU     01H                     ; EP MODE
CVLP   EQU     02H                     ; LP MODE
CVPAL  EQU     03H                     ; SP (PAL) MODE

$   EJECT

;----------------------------
      SERVO DATA TABLE
;----------------------------

VtSRVO CSEG    UNIT

       ;////   5% of maximum drum speed error amount ////

tDF_5per:
       DW      022CH          ;NTSC 1.3903ms / 125ns * 0.05 = 556.1
       DW      029AH          ;PAL  1.6667ms / 125ns * 0.05 = 666.7

       ;////   10% of maximum drum speed error amount ////

tDF_10per:
       DW 0458H               ;NTSC 1.3903ms / 125ns * 0.1 = 1112.2
       DW 0535H               ;PAL  1.6667ms / 125ns * 0.1 = 1333.3

       ;////   Drum speed gain ////

tDF_Kv:
       DW 011C2H              ;NTSC 32767 / (0.23066ms / 125ns)= 17.76
       DW 011C2H              ;PAL

       ;////   Maximum drum speed error amount xx.xx ////

tDF_max:
       DW 0735H               ;NTSC 8388607 / 4546 = 1845.3
       DW 0735H               ;PAL

       ;////   Minimum drum speed error amount ////

tDF_min:
       DW 10000H - 0735H           ;NTSC
       DW 10000H - 0735H           ;PAL

       ;////   Filter coefficient of drum speed (G) ////

tDF_fG:
      DW      015FDH                  ; NTSC
      DW      015A0H                  ; PAL

      ;////   Filter coefficient of drum speed (aG) ////

tDF_fAG:
      DW      0F2A8H                  ; NTSC
      DW      0F22CH                  ; PAL

      ;////   Filter coefficient of drum speed (-b) ////

tDF_fB:
      DW      0775AH                  ; NTSC
      DW      07832H                  ; PAL

      ;////   Drum phase gain ////

tDP_Kp:
                                      ; NTSC 32767 / (5.710ms / 125ns) = 0.717
      DW 0FC40H                       ; NTSC %8/18 ADJUSTMENT
      DW 0FC40H                       ; PAL

      ;////   Maximum drum phase error amount ////

tDP_max:
      DW 2220H                        ; NTSC 8386607 / 0B7 = 45839.4 > 32767
      DW 2220H                        ; NTSC

      ;////   Minimum drum phase error amount ////

tDP_min:
      DW 10000H - 2220H               ; NTSC
      DW 10000H - 2220H               ; PAL

      ;////   Filter coefficient of drum phase (G) ////

tDP_fG:
      DW 006D2H                       ; NTSC
      DW 0086EH                       ; PAL

      ;////   Filter coefficient of drum phase (aG) ////

tDP_fAG:
      DW 0F987H                       ; NTSC
      DW 0F815H                       ; PAL

      ;////   Filter coefficient of drum phase (-b) ////

tDP_fB:
      DW 07FA5H                       ; NTSC
      DW 07F7BH                       ; PAL

      ;////   10% of maximum capstan speed error amount ////

tCF_10per:
      DW 0458H                        ; NTSC 2.7778ms / 125ns * 0.1 = 2222.2
      DW 0C60H                        ; PAL  3.9602ms / 125ns * 0.1 = 3668.2
```

```
;      ;////    Capstan speed gain ////
;
;tCF_Kv, tCP_Kp are given in table per mode.

       ;////    Maximum capstan speed error amount ////

tCF_max:
       DW       1E79H              ; NTSC 8386607 / 433(1075) = 7801.5...4.2
       DW       1E79H              ; PAL  8386607 / 433(1075) = 7801.5

       ;////    Minimum capstan speed error amount ////

tCF_min:
       DW       10000H - 1E79H     ; NTSC
       DW       10000h - 1E79H     ; PAL

       ;////    Filter coefficient of capstan speed phase composite (G) ////

tCMX_fG:
       DW       0205FH             ; NTSC
       DW       01478H             ; PAL
       DW       01796H             ; LP

       ;////    Filter coefficient of capstan speed phase composite (aG) ////

tCMX_fAG:
       DW       0E0A1H             ; NTSC
       DW       0ECDCH             ; PAL
       DW       0E9C0H             ; LP

       ;////    Filter coefficient of capstan speed phase composite (-b) ////

tCMX_fB:
       DW       07EFEH             ; NTSC
       DW       07EA9H             ; PAL
       DW       07EA9H             ; LP

       ;////    Maximum capstan phase error amount ////

tCP_max:
       DW       15A0H              ; NTSC 8386607 / 0BD5 = 2768.8
       DW       15A0H              ; PAL  8386607 / 0BD5 = 2768.8

       ;////    Minimum capstan phase error amount ////

tCP_min:
       DW       10000H - 15A0H     ; NTSC
       DW       10000H - 15A0H     ; PAL

       ;////    Filter coefficient of capstan phase control (G) ////

fCP_fG:
       DW       01264H             ; 01164H %% ; NTSC
       DW       010D8H             ; PAL

       ;////    Filter coefficient of capstan phase control (aG) ////

fCP_fAG:
       DW       0EE74H             ; NTSC
       DW       0EFF1H             ; PAL
```

```
        ;////  Filter coefficient of capstan phase control (–b) ////

tCP_fB:
        DW     07F57H         ; NTSC
        DW     07F35H         ; PAL

        ;------- Data store subroutine --------------------
        ;             FOR DEBUG

SAVE_AX:
        XCH    B,!SAVE_CNT
        ADD    B, #2
        MOVW   SAVE_AREA[B], AX
        XCH    B,!SAVE_CNT
        RET

$       EJECT

VDFG    CSEG   UNIT

;---------------------------------------------------
;★★★ INTCPT2 DRUM FG INTERRUPTION PROCESS ROUTINE
;---------------------------------------------------
;
;       VPT2_000 : Interruption initial processing
;       VPT2_100 : Drum speed error calculation
;       VPT2_200 : Drum phase error x loop gain (Kp)
;       VPT2_300 : Drum speed error x loop gain (Kv)
;       VPT2_400 : Drum speed adjustment amount + drum
;                  phase adjustment degree + bias value
;       VPT2_500 : Save capture value (next CPT2n - 1)
;       VPT2_600 : Processing after interruption
;
;---------------------------------------------------
;      Interruption initial process
;---------------------------------------------------

VPT2_000 :;V
        ://///  Register setting /////////

        SEL    RB2
VPT2_010:
        ;/////  Highest order interruption enable ///

        MOVW   AX,MK0                 ; SAVE MASK REGISTER
        PUSH   AX
        MOVW   AX,MK1                 ;%SAVE MASK REGISTER
        NOP    ;%%%
        PUSH   AX                     ;%

        OR     MK0L,#11101111b        ;%INTCR00 ENABLE
        OR     MK0H,#01111100b        ;%INTP2, INTCR02, INTCR11 ENABLE
        OR     MK1L,#11110111b        ;%INTCR13 ENABLE  ;%ctl
        OR     MK1H,#11111110b        ;%INTP3 ENABLE

        EI

        SET1   FSDFG                  ; SET DFG EDGE DETECTION FLAG
        MOVG    RVCPT22,WHL           ; FOR DEBUG %%%

        MOVG    TDE,RVDFRF            ; LIMIT THE MAXIMUM VALUE
        ADDG    TDE,#7FFFH            ;
        SUBG    TDE,WHL               ; SET THE MAXIMUM VALUE TO 7FFFH + TARGET VALUE
        BH      $VPT2_101             ;

        ADDG    WHL,TDE               ;

VPT2_101:
VPT2_110:
        ;/////  E DV CALCULATION ///////////

        SUBG    WHL,RVDFEF            ; E DV = ΔNDF – NDFL
                                      ; DRUM SPEED ERROR = MEASURED SPEED –
                                        TARGET SPEED

VPT2_120:;B
        ;/////  Get error amount /////////

        MOVW    AX,HL                 ; ABSOLUTE VALUE CALCULATION
        BF      A.7,$VPT2_128         ;

        MOVW    HL,#0                 ;
        SUBW    HL,AX                 ; HL ← ABSOLUTE VALUE

VPT2_128:
        MOVW    VP,AX                 ; VP ← ERROR AMOUNT

        MOV     A,RVFSRV_2
        AND     A,#00000011b          ; READ RUN MODE

        MOV     B,#MODE NTSC          ;

        CMP     A,#CVPAL              ; PAL?
        BNE     $VPT2_129             ; No

        MOV     B,#MODE PAL           ;

VPT2_129:
        MOV     NTSC PAL,B            ; STORE NTSC = 0/PAL = 2

VPT2_130:;B

        ;/////  Check error amount 5% ///

        MOVW    AX, tDF_5per[B]       ;

        ;       DW      022CH         ;
        ;       DW      029AH         ;

        CMPW    HL,AX                 ;
        BC      $VPT2_140             ;

        MOVW    RVERDP_1,#0000H       ;%
        MOVW    RVERDP_Y,#0000H       ; DRUM PHASE ERROR ← 0000H
        MOVW    RVERDP_bY,#0000H      ;%
        MOVW    RVERDP_bY+2,#0000H     ;%

VPT2_140 :;B

        ;/////  Check error 10% ///

        MOVW    AX,tDF_10per[B]        ;

        ;       DW      0458H          ;
        ;       DW      0535H          ;

VPT2_140_10 :

        CMPW    AX,HL                  ;
        MOV1    FVDFE10,CY             ; SET FLAG IF DRUM SPEED ERROR IS 10% OR MORE

VPT2_150 :;B

        ;/////  Check maximum error value ///

        MOVW    AX,tDF_MAX[B]          ; LIMIT MAXIMUM VALUE OF DRUM SPEED ERROR

        ;       DW      0735H          ;
        ;       DW      0735H          ;

        CMPW    AX,HL                  ; MAXIMUM VALUE: DRUM SPEED ERROR
        BNC     $VPT2_160              ;              >=


        CMPW    VP,#8000H              ; SIGN
        BC      $VPT2_151              ; CY = 1 POSITIVE NUMBER
                                       ; CY = 0 NEGATIVE NUMBER

        MOVW    AX,tDF_MIN[B]          ; MINIMUM VALUE: DRUM SPEED ERROR

        ;       DW      10000H-0735H   ;
        ;       DW      10000H-0735H   ;

VPT2_151 :;B

        MOVW    VP,AX                  ;

VPT2_160 :;B

        ;/////  Save speed error ///////

        XCHW    VP,RVERDF              ; RVERDF ← ERROR AMOUNT OF THIS TIME
        MOVW    RVERDF_1,VP            ; RVERDF_1 ← ERROR AMOUNT OF LAST TIME
```

```
;-------------------------------------------------------------------------------
;      Lag read filter processing
;-------------------------------------------------------------------------------
;
;    0. Set filter coefficient
;
;        NTSC:
;            f1  =    8                   □ □ □
;            f2  =  56                      □ calculation from
;            DFG = 719.28Hz               □ □ □
;
;        MAL  a = –0.60695401
;        MBL  b = –0.93247632
;        MGL  g =  0.17179585
;
;        PAL:
;            f1  =    6                   □ □ □
;            f2  =  42                      □ calculation from
;            DFG = 600.00Hz               □ □ □
;
;        MAL  a = –0.63946320
;        MBL  b = –0.93908194
;        MGL  g =  0.16896488
;
;
--------------------------------------------------------------------------------

VPT2_170 :
        ;/////Set filter coefficient ///

        MOV    B,NTSC_PAL                 ;

        MOVW   AX,tDF_fG[B]               ;
        MOVW   B_buf,AX                   ;
        MOVW   AX,tDF_fAG[B]              ;
        MOVW   B_buf+2,AX                 ;
        MOVW   AX,tDF_fB[B]               ;
        MOVW   B_buf+4,AX                 ;

        ;/////Filter calculation processing ///

        MOV    B,#LOW(B_buf)              ;
        MOV    C,#LOW(RVERDF)             ;
        MOVW   DE,RVERDF_bY               ;
        MOVW   AX,RVERDF_bY+2             ;

        MACSW  2                          ; LAG READ FILTER PROCESSING

        MOVW   RVERDF_Y,AX                ; DRUM SPEED ERROR AMOUNT
                                            (AFTER FILTER CALCULATION)

        MOVW   DE,B_buf+4                 ;

        MULW   DE                         ;
        SHLW   DE,1                       ;
        ROLC   X,1                        ;
        ROLC   A,1                        ;

        MOVW   RVERDF_bY,DE               ; (–b)•Y
        MOVW   RVERDF_bY+2,AX             ;
```

```
;-------------------------------------------------------------------------------
;                       Drum phase error x Loop gain (Kp)
;-------------------------------------------------------------------------------
;
;   0. Get Kp (Decimal fraction is ignored)
;
;     NTSC  Kp = –3.75      Real number   : 0FC40H
;
;     PAL   Kp = –3.75      Real number   : 0FC40H
;
;   1. Phase error x Loop gain
;
;     HL ← HL + AX
;
;-------------------------------------------------------------------------------

VPT2_200 :
        ;///// Get Kp ///////////

        MOVW  B,NTSC_PAL                ;
        MOVW  AX,tDP_Kp[B]              ;

        ;     DW      11C2H             ;
        ;     DW      11C2H             ;

        ;///// Error x Kp ////////

        MOVW  DE,RVERDP_Y               ; DRUM PHASE ERROR AMOUNT
                                          (AFTER FILTER CALCULATION)
        MULW  DE                        ;

        CMPW  AX,#0FF80H                ; ZERO CHECK
        BNC   $VPT2_230                 ;

        CMP   A,#80H                    ;
        BNC   $VPT2_229                 ; UNDERFLOW

        CMPW  AX,#0080H                 ;
        BC    $VPT2_230

        MOVW  AX,#7FFFH                 ; 7FFFH ← OVERFLOW
        BR    VPT2_231                  ;
VPT2_229 :
        MOVW  AX,#8000H                 ; 8000H ← UNDERFLOW
        BR    VPT2_231                  ;

VPT2_230 :
        MOV   A,X                       ; A(XD)E
        MOV   X,D                       ;

VPT2_231 :
        MOVW  HL,AX                     ; HL ← DRUM PHASE ERROR AMOUNT
                                          (AFTER GAIN ADDITION)

VPT2_220 :
```

```
;-----------------------------------------------------------------------------
;                      Drum speed error x Loop gain (Kv)
;-----------------------------------------------------------------------------
;
;   0. Get Kv (Decimal fraction is ignored)
;
;     NTSC  Kv = 17.76      Real number   : 11C2H
;
;     PAL   Kv = 17.76      Real number   : 08E1H
;
;   1. Speed error x Loop gain
;
;     AX ← AX + DE
;
;;----------------------------------------------------------------------------

VPT2_300:;B
        ;/////Get Kv ///////////

        MOV   B,NTSC_PAL                ;
        MOVW  AX,tDF_Kv[B]              ;

        ;/////Error x Kv ////////

        MOVW  DE,RVERDF_Y               ; DRUM SPEED ERROR AMOUNT
                                          (AFTER FILTER OPERATION)
        MULW  DE                        ; AX ← DRUM SPEED ERROR AMOUNT
                                          (AFTER GAIN ADDITION)

        MOV   A,X                       ; A(XD)E
        MOV   X,D                       ;

;-----------------------------------------------------------------------------
; Drum speed adjustment + Drum phase adjustment + bias value → PWM 0
;-----------------------------------------------------------------------------
;
VPT2_400:;B
        ;/////Speed total + bias + phase total gain /////

        ADDW  AX,HL                     ; DRUM PHASE ERROR ADDITION
        BNV   $VPT2_411                 ;

        MOVW  AX,#07FFFH                ; 7FFFH ← OVERFLOW
        ADDC  X,#0                      ;
        ADDC  A,#0                      ; 8000H ← UNDERFLOW

VPT2_411:

        BT    A.7,$VPT2_412             ;

        ADDW  AX,RVDBAS                 ; BIAS ADDITION (<7FFF)
        BR    VPT2_420                  ;

VPT2_412:
        ADDW  AX,RVDBAS                 ; BIAS ADDITION (<7FFF)
        BC    $VPT2_420                 ;

        MOVW  AX,#0000H                 ; 0000H ← OVERFLOW
```

```
VPT2_420 :;B
        ;/////PWM 0 Output //////////

;v% PWM limitation items
        CMPW  AX,#0100H
        BC    $VPT2_421

        CMPW  AX,#0FF00H
        BNH   $VPT2_422

        MOVW  AX,#0FF00H
        BR    VPT2_422
VPT2_421 :
        MOVW  AX,#0100H

VPT2_422 :

;^% PWM limitation items

        MOVW  PWM0,AX                  ; SET DRUM PWM DATA


;-------------------------------------------------------------------------------
;                     Save capture value (Next CPT2n - 1)
;-------------------------------------------------------------------------------

VPT2_500 :
        ;/////Save CPT2 ///////

        MOVG  RVCPT2,UUP               ;

;-------------------------------------------------------------------------------
;                          Processing after interrupt
;-------------------------------------------------------------------------------

VPT2_600 :
        ;///// Multiple interrupt disable /////

        DI

        POP   AX                       ; %RETURN MASK REGISTER
        MOVW  MK1,AX                   ; %SET MASK REGISTER

        POP   AX                       ; RETURN MASK REGISTER

        MOV1  CY,CRMK02                ; LOAD INTCR0 2 INTERRUPT MASK FLAG
        MOV1  A.0,CY                   ; SAVE INTCR0 2 INTERRUPT MASK FLAG

        MOV1  CY,PMK2                  ; %LOAD INTP 2 INTERRUPT MASK FLAG
        MOV1  A.7,CY                   ; %SAVE INTP 2 INTERRUPT MASK FLAG

        MOVW  MK0,AX                   ; SET MASK REGISTER

VPT2_EXT :
        RETI

$       EJECT

VDPGP   CSEG  UNIT
```

```
;---------------------------------------------------------------------------------
;★★★ INTCR10 Drum phase error detection interruption processing
;---------------------------------------------------------------------------------
;
;       VR10_000 : Interrupt initial processing
;       VP10_100 : Calculation of phase control target value
;       VP10_200 : Calculation of drum phase error
;       VP10_300 : Lag read filter processing
;       VP10_400 : Processing after interrupt
;
;---------------------------------------------------------------------------------
;                          Interrupt initial processing
;---------------------------------------------------------------------------------

VR10_000 :;V
        ;///// Register setting /////////

        SEL    RB2                      ; HIGH-ORDER INTERRUPT

VR10_010 :
        ;///// Read CPT0 ////

        MOVW   AX,CPT0L                 ;
        MOVW   HL,AX                    ;
        MOV    A,CPT0H                  ;
        MOV    W,A                      ;
        MOVG   UUP,WHL                  ;

VR10_020 :
        ;///// Drum on check ///

        BT     FSDRMON,$VR10_030
        RETI

VR10_030 :;B
        ;///// Check speed error /////

        BF     FVDFE10,$VR10_040        ; 10% OR MORE DRUM SPEED ERROR? NO
        RETI                            ; Yes INTERRUPT END

VR10_040 :;B
        ;///// High-order interrupt enable ///

        MOVW   AX,MK0                   ; SAVE MASK REGISTER
        PUSH   AX
        MOVW   AX,MK1                   ;%SAVE MASK REGISTER
        NOP                             ; %%%
        PUSH   AX                       ;%

        OR     MK0L,#11101111B          ;%INTCR00 ENABLE
        OR     MK0H,#01111100B          ;%INTP2, INTCR02, INTCR11 ENABLE
        OR     MK1L,#11110111B          ;%INTCR13 ENABLE ;%CTL
        OR     MK1H,#11111110B          ;%INTP3 ENABLE

        EI
```

```
;-----------------------------------------------------------------------------
;                   Calculation of phase control target value
;-----------------------------------------------------------------------------
;
;   <1>Digital value equal to HSW pulse delay amount
;
;           CR00 x 4 times (difference between timer 0 and FRC clock frequency)
;
;   <2>Delay amount for half cycle of frame
;
;           CR10 value
;
;   <3>Delay amount for 6.5 Hrs
;
;           6.5 Hrs (0.41 msec) ÷ 125 ns (FRC clock) = 616d
;
;   <4>Delay amount for Vsync separation
;           SOFT execution time ÷ 125 ns (FRC clock)
;
;-----------------------------------------------------------------------------

VR10_100:
        ;/////Phase target value calculation ///////

                                      ; SET MINIMUM UNIT TO 0.500 us
        MOVG   WHL,#0                 ;
        MOVW   HL,CR00                ; CR00*2
        SHLW   HL,1                   ;
        ADDW   HL,CR10                ; + CR10

        ADDG   WHL,#0355H             ; + (<3> + <4>) 413.14 + 13.5
                                      ; = 426.64 µs (853)

        MOVG   VVP,WHL                ; VVP ← PHASE TARGET VALUE

;-----------------------------------------------------------------------------
;                         Drum phase error calculation
;-----------------------------------------------------------------------------
;
; 0. Phase error calculation
;
;   EDP = [(CPT0 value) – (CPT1 value)] – NDPL
;
;   EDP         :Drum phase error amount
;   NDPL        :Phase control target value
;
;   MCPT1       :Capture value of internal HSW falling edge only
;   MCPT0       :Capture at CR10 match
;
; 1. Check phase error maximum value
;
;   Assume NTSC error ≥ 06B1H → error = 06B1H (maximum)
;
;   Assume PAL error ≥ 06B1H → error = 06B1H (maximum)
;
;-----------------------------------------------------------------------------
```

```
VR10_200 :
        ;///// Phase error calculation /////////

        MOVG   WHL,UUP                  ; (MCPT0 – MCPT1)
        MOVG   RVCPT0,WHL               ;
        SUBG   WHL,RVCPT1               ; DRUM PHASE ERROR

        MOV    A,W                      ; DIVIDE DRUM PHASE ERROR INTO 1/2
        AND    A,#03FH                  ;
        SHR    A,1                      ;
;%%     MOV    W,A                      ;
        RORC   H,1                      ;
        RORC   L,1                      ;
        SHR    A,1                      ;%%
        MOV    W,A                      ;%%
        RORC   H,1                      ;%%
        RORC   L,1                      ;%%

        MOV    B,NTSC_PAL               ; LIMIT MAXIMUM VALUE
        MOVW   AX,tDP_MAX[B]            ;
        MOV    T,#0                     ;
        MOVW   DE,AX                    ;
        ADDG   TDE,VVP                  ; SET MAXIMUM VALUE TO TARGET +
                                          MAXIMUM LIMITATION VALUE
        SUBG   TDE,WHL                  ;
        BH     $VR10_201                ;

        ADDG   WHL, TDE                 ;
VR10_201 :

        SUBG   WHL,VVP                  ; DRUM PHASE ERROR –
                                        ; E DP (TARGET VALUE OF DRUM PHASE ERROR)
        BNC    $VR10_220                ; IS ERROR AMOUNT NEGATIVE VALUE?

        ;///// Check maximum error value ///

                                        ; NEGATIVE VALUE
;;      MOV    B,NTSC_PAL               ;
        MOV    T,#0FFH                  ; LESS THAN MINIMUM VALUE?
        MOVW   AX,tDP_MIN[B]            ;
        MOVW   DE,AX                    ;
        SUBG   TDE,WHL                  ; MINIMUM VALUE – ERROR AMOUNT
        BC     $VR10_220                ;

        MOVW   HL,AX                    ;

VR10_220 :
        XCHW   HL,RVERDP                ; RVERDP ← DRUM PHASE ERROR OF THIS TIME
        MOVW   RVERDP_1,HL              ; RVERDP_1 ← DRUM PHASE ERROR OF LAST TIME
```

```
;-------------------------------------------------------------------------------
;      Lag read filter processing
;-------------------------------------------------------------------------------
;
;    0. Set filter coefficient
;
;      NTSC:
;          f1  = 0.013Hz                 □□□
;          f2  = 0.25 Hz                   □ calculation from
;          DPG = 30Hz                    □□□
;
;      MAL a  =–0.94897592
;      MBL b  =–0.99728098
;      MGL g  =  0.05328881
;
;      PAL:
;          f1  = 0.016Hz                 □□□
;          f2  = 0.25 Hz                   □ calculation from
;          DPG = 25Hz                    □□□
;
;      MAL a  =–0.93908194
;      MBL b  =–0.99598683
;      MGL g  =  0.06587816
;
;;------------------------------------------------------------------------------

VR10_300:;B
        ;/////Set filter coefficient ///

        MOV     B,NTSC_PAL            ;

        MOVW    AX,tDP_fG[B]          ;
        MOVW    B_buf,AX              ;
        MOVW    AX,tDP_fAG[B]         ;
        MOVW    B_buf+2,AX            ;
        MOVW    AX,tDP_fB[B]          ;
        MOVW    B_buf+4,AX            ;

VR10_310:;B
        ;/////  Filter calculation processing /////

        MOV     B,#LOW(B_buf)         ;
        MOV     C,#LOW(RVERDP)        ;
        MOVW    DE,RVERDP_bY          ;
        MOVW    AX,RVERDP_bY+2        ;

        MACSW   2                     ;

        MOVW    RVERDP_Y,AX           ; DRUM PHASE ERROR AMOUNT
                                        (AFTER FILTER CALCULATION)

        MOVW    DE,B_buf+4            ;

        VR10_YL EQU     13FH

        BT    A.7,$VR10_312           ;;;;; MAXIMUM LIMITATION PROCESSING Yn – 1
        CMPW  AX,#VR10_YL             ; POSITIVE NUMBER
        BC    $VR10_314               ;
        MOVW  AX,#VR10_YL             ;
        BR    VR10_314
```

```
VR10_312 :
        CMPW   AX,#10000H - VR10_YL      ; NEGATIVE NUMBER
        BNC    $VR10_314                 ;
        MOVW   AX,#10000H - VR10_YL      ;
VR10_314 :                               ;;;;;

        MULW   DE                        ;
        SHLW   DE,1                      ;
        ROLC   X,1                       ;
        ROLC   A,1                       ;

        MOVW   RVERDP_bY,DE              ; (–b)•Y
        MOVW   RVERDP_bY+2,AX            ;


;-------------------------------------------------------------------------------
;                            CR10 Revision processing
;-------------------------------------------------------------------------------
VR10_320:
        MOVW   AX,RVBCR10                ; CR10 ← CR10 DATA BUFFER AREA
        MOVW   CR10,AX                   ;
                                         ; Note: When write CR10, perform in
                                         ;       INTCR10 routine. (Unless, TM1
                                         ;       may overflow depending on timing
                                         ;       of writing!)

;-------------------------------------------------------------------------------
;                           Processing after interrupt
;-------------------------------------------------------------------------------

VR10_400 :
        ;/////Multiple interrupt disable /////

        DI

        POP    AX                        ; %RETURN MASK REGISTER
        MOVW   MK1,AX                    ; %SET MASK REGISTER

        POP    AX                        ; RETURN MASK REGISTER

        MOV1   CY,CRMK02                 ; LOAD INTCR02 INTERRUPT MASK FLAG
        MOV1   A.0,CY                    ; SAVE INTCR02 INTERRUPT MASK FLAG

        MOV1   CY,PMK2                   ; %LOAD INTP2 INTERRUPT MASK FLAG
        MOV1   A.7,CY                    ; %SAVE INTP2 INTERRUPT MASK FLAG

        MOVW   MK0,AX                    ; SET MASK REGISTER

VR10_EXT :
        RETI

$       EJECT

VCFG    CSEG   UNIT
```

```
;--------------------------------------------------------------------------------
;★★★                      INTCPT3 Capstan FG interrupt
;--------------------------------------------------------------------------------

;
;       VRT3_000 : Interrupt initial processing
;       VRT3_100 : PBCTL signal missing detection
;       VRT3_200 : Capstan speed error calculation
;       VRT3_300 : Error amount calculation special processing
;       VRT3_400 : Speed error x loop gain (Kv/Kp)
;       VRT3_500 : Capstan speed adjustment amount + Capstan phase
;                  adjustment amount
;       VRT3_600 : MIX error amount digital filter processing
;       VRT3_700 : Capstan speed/phase MIX (Yn) x gain adjustment
;       VRT3_800 : Capstan speed/phase MIX adjustment value + bias value
;       VPT3_900 : Capstan PWM output
;       VRT3_A00 : Processing after interrupt
;

;--------------------------------------------------------------------------------
;                           Interrupt initial processing
;--------------------------------------------------------------------------------

VPT3_000:;V
        ;/////Register setting /////////

        SEL    RB2                      ; HIGH ORDER INTERRUPT

VPT3_010:
        ;/////Multiple interrupt enable ////

        MOVW   AX,MK0                   ; SAVE MASK REGISTER
        PUSH   AX
        MOVW   AX,MK1                   ;%SAVE MASK REGISTER
        NOP    ;%%%
        PUSH   AX                       ;%

        OR     MK0L,#11101111B          ;%INTCR00 ENABLE
        OR     MK0H,#01111100B          ;%INTP2,INTCR02, INTCR11 ENABLE
        OR     MK1L,#11110111B          ;%INTCR13 ENABLE ;%ctl
        OR     MK1H,#11111110B          ;%INTP3 ENABLE

        EI

VPT3_015:
        ;/////Check CFG 90 pulse counter ////

        CMP    RSCFG90C,#00
        BZ     $VPT3_100

        DEC    RSCFG90C

;--------------------------------------------------------------------------------
;       Increment play run mode automatic judgment counter
;
;       PBCTL signal missing detection
;--------------------------------------------------------------------------------

VPT3_100:

        INC    RVCEVFG                  ; CFG counter increment @@@ change
```

```
VPT3_110 :

        BT     FSMDCHG,$VPT3_200          ; AT TRANSITION? Yes

VPT3_130 :

        CMP    RVCEVFG,#40                ; PBCTL SIGNAL MISSING? @@@ CHANGE
        BC     $VPT3_200                  ; No

VPT3_140 :

        BT     FVPHFX,$VPT3_141           ;%PH FIX ON? Yes (FF/REW) ;%ctl
        SET1   FVFLCTL                    ; SET PBCTL SIGNAL MISSING FLAG
VPT3_141 :                                ;%CTL
        MOV    RVCEVFG,#00                ; CLEAR PLAY MODE JUDGMENT COUNTER ;@@@
                                          ; CHANGE
;%ctl v
        ;%
        ;%% CTL AMP GAIN INC(+5)
        ;% AMPLIFY CTL AMP GAIN BY +5 DURING PBCTL SIGNAL MISSING
        MOV    A,CTLM                     ;%
        ADD    A,#05H                     ;%
        CMP    A,#1FH                     ;%
        BC     $VPT3_150                  ;%

        MOV    A,#1FH                     ;%
VPT3_150 :;B                              ;%
        MOV    CTLM.A                     ;%
;%ctl ^

;-----------------------------------------------------------------------
;       Calculate capstan speed error
;-----------------------------------------------------------------------
;
;       ΔNCF = CPT3n – CPT3n-1
;
;       ΔNCF: Capture value of this time – capture value of last time
;
;       ECV = ΔNCF – NCFL
;
;             ECV                         : Capstan speed error amount
;             NCFL                        : Capstan speed target value
;-----------------------------------------------------------------------

VPT3_200 :;B

        MOVW   AX,CPT3L                   ;
        MOVW   HL,AX                      ;
        MOV    A,CPT3H                    ;
        MOV    W,A                        ;
        MOVG   UUP,WHL                    ;

        SUBG   WHL,RVCPT3                 ; ΔNCF = CPT3n – CPT3n-1
        MOV    A,W                        ;
        AND    A,#003FH                   ;
        MOV    W,A                        ;
        MOVG   VVP,WHL                    ;
        SUBG   WHL,#12C0H                 ; Check capstan abnormal high speed rotating
        BNC    $VPT3_201                  ; CFG is within 600 μsec?
        MOVW   AX,#1FFFH                ; Yes

        BR     VPT3_820                 ; TO AVOID OCCURRING CFG FREQUENT INTERRUPT
                                        ; DUE TO MOTOR RUNAWAY,
                                        ; AND MICRO CONTROLLER'S RUNAWAY
VPT3_201 :;B

        MOVG   WHL,VVP                  ;
        MOVG   TDE,RVCFRF               ; LIMIT MAXIMUM VALUE
        ADDG   TDE,#7FFFH               ;
        SUBG   TDE,WHL                  ; SET MAXIMUM VALUE TO 7FFFH + TARGET VALUE
        BH     $VPT3_202                ;

        ADDG   WHL,TDE                  ;

VPT3_202 :
        SUBG   WHL,RVCFRF               ; ECV = ΔNCF – NCFL

        MOVW   AX,HL                    ; CALCULATION OF ABSOLUTE VALUE
        BF     A.7,$VPT3_203            ; NEGATIVE VALUE?

        MOVW   HL,#0                    ;
        SUBW   HL,AX                    ; HL ← ABSOLUTE VALUE
VPT3_203 :
        MOVW   VP,AX                    ; VP ← ERROR AMOUNT

;-----------------------------------------------------------------------------
;  Error amount calculation special processing
;-----------------------------------------------------------------------------
;
;       When set capstan phase error amount to 0
;
;             • When drum speed error amount is more than ± 10%
;
;               Flag more than 10%: FSDP10=1
;
;             • When capstan speed error amount is more than ± 10%
;
;               Error amount is calculated by NTSC/PAL PLAY target value.
;
;               NTSC5% : 56CEH x 0.10 = 0458H
;               PAL 5% : 7BC1H x 0.10 = 0C60H
;
;             • When PBCTL signal missing is detected in play
;             • At FF/REW mode
;             • When capstan phase servo disabled (during loading)
;
;-----------------------------------------------------------------------------

VPT3_300 :;B

        MOV    A,RVFSRV_2
        AND    A,#00000011b             ; READ RUN MODE

        MOV    B,#MODE NTSC             ;

        CMP    A,#CVPAL                 ; PAL?
        BNE    $VPT3_321                ; No
        MOV    B,#MODE PAL              ;

VPT3_321 :
        MOV    NTSC_PAL,B               ;
        MOVW   AX,tCF_MAX[B]            ; LESS THAN MAXIMUM VALUE?

        ;      DW      1E79H
        ;      DW      1E79H

        CMPW   HL,AX                    ; ERROR AMOUNT (ABSOLUTE VALUE):
                                        ; MAXIMUM VALUE
        BC     $VPT3_320                ;      =<

        CMPW   VP,#8000H                ; SIGN
        BC     $VPT3_311                ; CY=1 POSITIVE
                                        ; CY=0 NEGATIVE
        MOVW   AX,tCF_MIN[B]            ;

VPT3_311 :
        MOVW   VP,AX                    ; SET MAXIMUM VALUE AS SPEED ERROR AMOUNT

VPT3_320 :;B
        MOVW   RVERCF,VP                ; CAPSTAN SPEED ERROR
        MOVW   AX,tCF_10per[B]          ;

        ;      DW 0458H                 ;
        ;      DW 0C60H                 ;

        CLR1   FVCFE05                  ;

        CMPW   AX,HL                    ;
        BNC    $VPT3_330                ;

        SET1   FVCFE05                  ;

        BR     VPT3_340                 ;

VPT3_330 :;b

        BT     FVPHFX,$VPT3_340         ; PH FIX ON? YES(FF/REW)

        BT     FVDFE10,$VPT3_340        ; IS DRUM SPEED ERROR MORE THAN 10%? YES

        BF     FVFLCTL,$VPT3_400        ; PBCTL SIGNAL MISSING? NO

VPT3_340 :;B
        ;///// Phase error amount to 0 //////

        MOVW   RVERCP_Y,#0              ; CAPSTAN PHASE ERROR ← 0000H
        MOVW   RVERCP,#0                ;
        MOVW   RVERCP_1,#0              ;
        MOVW   RVERCP_bY,#0             ; CAPSTAN PHASE FILTER
        MOVW   RVERCP_bY+2,#0           ; CLEAR MEMORY
```

```
;------------------------------------------------------------------
;  Capstan speed error x Loop gain (Kv/Kp)
;------------------------------------------------------------------
;
;       AX ← AX + DE
;------------------------------------------------------------------

VPT3_400 :;B

        MOVW   AX,RVERCF               ; CAPSTAN SPEED ERROR
        MOVW   DE,RVC_Kvp              ; SPEED PHASE ERROR MIX RATE

        MULW   DE                      ; A(XD)E

        MOV    A,X                     ; 8 BITS SHIFT (SET VALID ONLY 16 BITS)
        MOV    X,D                     ;

;------------------------------------------------------------------
;  Capstan speed adjustment amount + Capstan phase adjustment amount
;------------------------------------------------------------------

        MOVW   DE,RVERCP_Y             ; CAPSTAN PHASE ERROR
        SHLW   DE,2                    ; %%% 3

        ADDW   AX,DE                   ; CAPSTAN SPEED ERROR AMOUNT + CAPSTAN
                                       ; PHASE ERROR
        BNV    $VPT3_511               ;

        MOVW   AX,#7FFFH               ;
        ADDC   X,#0                    ; 7FFFH ← OVERFLOW
        ADDC   A,#0                    ; 8000H ← UNDERFLOW
VPT3_511 :

        XCHW   AX,RVERCMX              ; CAPSTAN SPEED PHASE MIX ERROR
        MOVW   RVERCMX_1,AX            ;

;------------------------------------------------------------------
;  Speed/phase MIX error amount digital filter processing
;------------------------------------------------------------------
;
;   0. Set filter coefficient
;
;     NTSC  SP/EP
;         f1  = 0.45 Hz                □□□
;         f2  = 1.8 Hz                  □ calculation from
;         CFG = 360 Hz                 □□□
;
;         MAL   a = –0.96906992
;         MBL   b = –0.99217674
;         MGL   g =  0.25293372
;
;     NTSC      LP
;         f1  = 0.45 Hz                □□□
;         f2  = 2.5 Hz                  □ calculation from
;         CFG = 270 Hz                 □□□
;
;         MAL   a = –0.94346684
;         MBL   b = –0.98958257
;          MGL    g =  0.18427114
;
;     PAL         SP
;          f1  = 0.42 Hz                □□□
;          f2  = 2.7 Hz                  □ calculation from
;          CFG = 252.51 Hz              □□□
;          MAL    a = –0.93499961
;          MBL    b = –0.98960350
;          MGL    g =  0.15994518
;
;
VPT3_600 :;B

VPT3_610 :
        ;///// Run mode judgment ////////

        MOV    B,NTSC_PAL              ;

        MOV    A,RVFSRV_2              ;
        AND    A,#00000011B            ; READ RUN MODE

        CMP    A,#CVLP                 ; LP?
        BNE    $CPT3_611               ;

        MOV    B,#MODE LP              ;
                                       ; B ← NTSC(0)/PAL(2)/LP(4)
CPT3_611 :
        MOVW   AX,tCMX_fG[B]           ;
        MOVW   B_buf,AX                ;
        MOVW   AX,tCMX_fAG[B]          ;
        MOVW   B_buf+2,AX              ;
        MOVW   AX,tCMX_fB[B]           ;
        MOVW   B_buf+4,AX              ;

        MOV    B,#LOW(B_buf)           ;
        MOV    C,#LOW(RVERCMX)         ;
        MOVW   DE,RVERCMX_bY           ;
        MOVW   AX,RVERCMX_bY+2         ;

        MACSW  2                       ;

        MOVW   RVERCMX_Y,AX            ; Y

        MOVW   DE,B_buf+4              ;

        MULW   DE                      ;
        SHLW   DE,1                    ;
        ROLC   X,1                     ;
        ROLC   A,1                     ;

        MOVW   RVERCMX_bY,DE           ; (–b)•Y
        MOVW   RVERCMX_bY+2,AX         ;
```

```
;-----------------------------------------------------------------------
;  Capstan speed/phase mix (Yn) x gain adjustment
;-----------------------------------------------------------------------
;
;       AX ← AX + DE
;-----------------------------------------------------------------------

VPT3_700:

        MOVW   AX,RVC_Kmp                ;
        MOVW   DE,RVERCMX_Y              ;

        MULW   DE                        ; A(XD)E

        CMPW   AX,#0FF80H                ; ZERO CHECK
        BNC    $VPT3_730                 ;

        CMP    A,#80H                    ;
        BNC    $VPT3_729                 ; UNDERFLOW

        CMPW   AX,#0080H                 ;
        BC     $VPT3_730

        MOVW   AX,#7FFFH                 ; 7FFFH ← OVERFLOW
        BR     VPT3_731                  ;
VPT3_729 :
        MOVW   AX,#8000H                 ; 8000H ← UNDERFLOW
        BR     VPT3_731                  ;

VPT3_730 :
        MOV    A,X                       ; 8-BIT SHIFT (ONLY FOR 16-BIT)
        MOV    X,D                       ;

VPT3_731 :

;-----------------------------------------------------------------------
;  Capstan speed/phase MIX adjustment value + bias value
;-----------------------------------------------------------------------

VPT3_800 :;B

; swBIAS8      EQU 0                     ; BIAS IS LESS THAN 8000H
  swBIAS8      EQU 1                     ; BIAS IS 8000H OR MORE

$_IF(swBIAS8)
;              Add processing at bias (=> 8000H)

        BF     A.7,$VPT3_812             ;

        ADDW   AX,RVCBAS                 ; NEGATIVE NUMBER BEFORE ADDING
        BR     VPT3_820                  ;

VPT3_812 :
        ADDW   AX,RVCBAS                 ; WHEN POSITIVE NUMBER BEFORE ADDING
                                         ; OVERFLOW MAY OCCUR
        BNC    $VPT3_820                 ;

        MOVW   AX,#0FFFFH                ; 0FFFFH ← OVERFLOW
```

```
$ELSE
;              Add processing at bias (=< 7FFFH)

        BF     A.7,$VPT3_812            ;

        ADDW   AX,RVCBAS                ; WHEN NEGATIVE NUMBER BEFORE ADDING
                                        ; UNDERFLOW MAY OCCUR
        BC     $VPT3_820                ;

        MOVW   AX,#0000H                ; 0000H ← UNDERFLOW
        BR     VPT3_820                 ;

VPT3_812 :
        ADDW   AX,RVCBAS                ; POSITIVE NUMBER BEFORE ADDING

$ENDIF

;---------------------------------------------------------------------------------
;                     Capstan speed level judgment at FF/REW
;---------------------------------------------------------------------------------

VPT3_820 :;B
        MOVW   BC,AX                    ; SAVE PWM OUTPUT DATA
        MOV    A,RVSRVCD
        AND    A,#11110000B             ; CLEAR LOW-ORDER 4 BITS
        CMP    A,#CVFFREW               ; FF/REW?
        BNE    $VPT3_830                ; No

        MOV    A,RVERCF                 ; CAPTURE HIGH-ORDER BYTE OF CAPSTAN
                                        ; SPEED ADJUSTMENT
        BT     A.7,$VPT3_823            ; IS CAPSTAN SPEED ADJUSTMENT AMOUNT (–)?
                                        ; No
                                        ; %%% MODIFICATION IS REQUIRED

VPT3_821 :
        CMP    A,#2
        BC     $VPT3_823                ; C-ERR   0H - 1FFH
        CMP    A,#6
        BC     $VPT3_822                ; C-ERR 200H - 5FFH
                                        ; C-ERR 600H - MAX
        MOV    A,#0                     ; LEVEL 0
        BR     $VPT3_824
VPT3_822 :
        MOV    A,#1                     ; LEVEL 1
        BR     $VPT3_824
VPT3_823 :
        MOV    A,#2                     ; LEVEL 2
VPT3_824 :
        MOV    RSFRSPED,A               ; CAPSTAN FF/REW SPEED LEVEL SET
        MOVW   BC,#0FFFFH               ; PWM OUTPUT FULL

VPT3_830 :;B
        MOVW   AX,BC                    ; PWM OUTPUT DATA RETURN
```

```
;--------------------------------------------------------------------------------
;               Capstan PWM suppression control at PLAY → REVIEW
;--------------------------------------------------------------------------------

VPT3_840 :
        MOVW   BC,AX                    ; SAVE PWM OUTPUT DATA
        MOV    A,RVSRVCD
        AND    A,#11110000B             ; CLEAR LOW-ORDER 4 BITS

        CMP    A,#CVFR6HVD              ; 6Hrs PLAY? (AT SPIN OFF RF GEAR)
        BE     $VPT3_841                ; Yes

        CMP    A,#CVRVS                 ; RVS PLAY? (AT REVERSE PLAY)
        BNE    $VPT3_842                ; No
VPT3_841 :
        MOVW   AX,BC                    ;
        CMPW   AX,#0B333H               ; CAPSTAN PWM 0B333H (3.5 V) OR HIGHER?
        BC     $VPT3_842                ;
        MOVW   BC,#0B333H               ; PWM OUTPUT 3.5 V
VPT3_842 :
        MOVW   AX,BC                    ; PWM OUTPUT DATA RETURN

;--------------------------------------------------------------------------------
;                               Output capstan PWM
;--------------------------------------------------------------------------------
VPT3_900 :;B

;v%PWM limitations
        CMPW   AX,#0100H
        BC     $VPT3_901

        CMPW   AX,#0FF00H
        BNH    $VPT3_902

        MOVW   AX,#0FF00H
        BR     VPT3_902

VPT3_901 :
        MOVW   AX,#0100H
VPT3_902 :
;^%PWM limitations
        MOVW   PWM1,AX                  ; SET CAPSTAN PWM DATA

        ;///// SAVE CPT3 ///////

        MOVG   RVCPT3,UUP               ;

;--------------------------------------------------------------------------------
;                        Multiple interruption prohibited
;--------------------------------------------------------------------------------
VPT3_A00 :

        DI

        POP    AX                       ; %RETURN MASK REGISTER
        MOVW   MK1,AX                   ; %SET MASK REGISTER
        POP    AX                        ; RETURN MASK REGISTER

        MOV1   CY,CRMK02                 ; LOAD INTCR02 INTERRUPT MASK FLAG
        MOV1   A.0,CY                    ; SAVE INTCR02 INTERRUPT MASK FLAG

        MOV1   CY,PMK2                   ; %LOAD INTP2 INTERRUPT MASK FLAG
        MOV1   A.7,CY                    ; %SAVE INTP2 INTERRUPT MASK FLAG

        MOVW   MK0,AX                    ; SET MASK REGISTER

        RETI                             ; CAPSTAN FG INTERRUPT PROCESSING END

;%%%ctl v
$       NOLIST
$       SUBTITLE('SRV0.ASM : INTCR13 ROUTINE CTL detection & output interruption')

$       LIST
$       EJECT

VCTL    CSEG   UNIT

;--------------------------------------------------------------------------------
;★★★    INTCR13 CTL detection & output interrupt
;--------------------------------------------------------------------------------

;       VR13_000 : Interrupt initial processing
;       VR13_100 : CTL detection & output interrupt processing

;--------------------------------------------------------------------------------
;  Interrupt initial processing
;--------------------------------------------------------------------------------

VR13_000:;V
        ;/////Register setting /////////

        SEL    RB3                       ; HIGHEST-ORDER INTERRUPT!!

VR13_100:
;********************************************************************************
; GAIN CONTROL
;********************************************************************************
                                         ; REWRITE GAIN AT PLAY and CUE/REV
        CALL   !GAINADJ                  ; CTL AMP GAIN ADJUST

        SET1   CRMK13                    ; INTCR13 INTERRUPT DISABLE

        RETI
        ;

;***************************
; GAIN ADJUST SUBROUTINE
;***************************
GAINADJ :
        MOV    A,AMPMO
;
        SET1   FLGCLR                    ; CTL FLAG CLEAR
        MOV1   CY,A.3
        XOR1  CY,A.1
        BC    $GAIN_E
;
        MOV   X,CTLM
        AND   X,#00011111B
;
        BF    A.3,$GAIN_UP
        CMP   X,#0                    ;DOWN
        BZ    $GAIN_E
        DEC   X
        BR    GAINSET
GAIN_UP :                             ;UP
        CMP   X,#1FH
        BZ    $GAIN_E
        INC   X
GAINSET :
        MOV   CTLM,X
GAIN_E :
        RET
        ;
;%%%CTL ^

$EJECT

VCPG    CSEG  UNIT

;------------------------------------------------------
;★★★INTCR12 Capstan phase error detection interrupt
;           At Play: Interrupt by PBCTL signal
;------------------------------------------------------

;       VR12_000 : Interrupt initial processing
;       VR12_020 : VISS signal detection processing
;       VR12_200 : Play run mode automatic judgment processing
;       VR12_300 : Capstan phase error calculation
;       VR12_400 : Lag read filter processing
;       VR12_500 : PBCTL signal missing check
;       VR12_600 : Processing after interrupt
;       VR12_T00 : Run mode judgment table

;------------------------------------------------------
; Interruption initial processing
;------------------------------------------------------

VR12_000 :;V
        ;///// Register setting /////////

        SEL   RB2                     ; High-order interrupt

VR12_010 :
        ;///// Highest-order interrupt enable ///

        MOVW  AX,MK0                  ; Save mask register
        PUSH  AX
        MOVW  AX,MK1                  ;%Save mask register
        NOP                           ; %%%
        PUSH  AX                      ;%

        OR MK0L,#11101111B            ;%INTCR00 ENABLE
        OR MK0H,#01111100B            ;%INTP2, INTCR02,
                                        ; INTCR11 ENABLE
        OR    MK1L,#11110111B           ;%INTCR13 ENABLE ;%CTL
        OR    MK1H,#11111110B           ;%INTP3 ENABLE

        EI
  ;
VR12_020 :
        ;/////Macro service ////////

        CMP   RVMCCR12,#00H             ; IS MSC INTERRUPTING WITH “0”?
        BNE   $VR12_032                 ; No

        CMPW  RVB2CR12,#DT_CMP          ; ARE BUFFER 1 AND 2 COMPARISON AREA
                                        ; INFORMATION?
        BE    $VR12_032                 ; Yes

        MOV   RVMCCR12,RVCRAM           ; SET MACRO SERVICE COUNTER VALUE
        SET1  CRISM12                   ; SET INTCR12 MACRO SERVICE INTERRUPT
        BR    VR12_120                  ;%
        ;

VR12_032 :;B
        MOV   RVMCCR12,RVCRAM           ; SET MACRO SERVICE COUNTER VALUE
        SET1  CRISM12                   ; SET INTCR12 MACRO SERVICE INTERRUPT

VR12_040 :
;       /// SEARCH MODE CHECK ///       ;%

        BF    FSCAPON,$VR12_111         ; CAPSTAN ON?
        BF    FNSTENA,$VR12_111         ; SEARCH DETECT DI? (150 msec)
;
        MOV   A,RVSRVCD
        AND   A,#11110000B              ; CLEAR LOW-ORDER 4 BITS
        CMP   A,#CVFFRW6H               ; AT FF/REW START?
        BNE   $VR12_045                 ; No
        BR    !VR12_111                 ; DISABLE VISS!
;
VR12_045 :;B
        CMP   RSNOW,#CSMPLAY            ; DURING PLAY?
        BE    $VR12_050                 ; Yes
                                        ; No
        CMP   RSNEXT,#CSMPLAY           ;
        BNE   $VR12_060                 ;%a No
        BR    VR12_111                  ;%a Yes
        ;

VR12_050 :;B
        BT    FSVM0FRQ,$VR12_111        ; V_MUTE OFF?(”1” PULSE DETECTION?)
        BF    PQVD,$VR12_111            ;
;                                       ; No
VR12_060 :;B
        BT    FSVISSI,$VR12_100         ; INDEX SEARCH MODE?
        BT    FSVISSO,$VR12_100         ; ONCE MORE SEARCH MODE?
        BF    FSVISSME,$VR12_111        ; MARK/ERASE MODE?
                                        ; Yes
VR12_100 :;B
;       /// VISS OK ///                 ;%
        SET1    FSVISSOK                ; SET VISS SIGNAL DETECTION FLAG!!

VR12_111 :;B
;%      /// BUFFER AREA CLEAR ///       ;%
```

```
         MOVW  RVB2CR12,#0FFFFH         ;%BUFFER AREA 1,2 CLEAR
                                        ; (REVERSE/FORWARD ALL 1 CLEAR)

VR12_120 :;B
;        /// COUNTER FLAG CLEAR ///     ;%
         BF    FSVISTR,$VR12_121
         CLR1  FSVISTR                  ; CLEAR VISS SIGNAL DETECTION START FLAG!!
         CLR1  FSVISSOK                 ; CLEAR VISS SIGNAL DETECTION FLAG!!
         MOVW  RVB2CR12,#0FFFFH         ;%BUFFER AREA 1,2 CLEAR
                                        ; (REVERSE/FORWARD ALL 1 CLEAR)

;        /// Set coefficient multiplied by CR30 at Macro Service ///    ;%
VR12_121 :;B
         MOV   A,#PTN_REW               ;% (REVERSE)
         BT    PCAPFWD,$VR12_122        ;% CAPSTAN FORWARD OR REVERSE ?
         MOV   A,#PTN_FF                ;% (FORWARD)
VR12_122 :;B                            ;%
         MOV   RVMKEISU,A               ;%

;%VISS ^
$        EJECT

;%%%CTL v
         ;---------------------------------------------------
         ;-- DETERMINE CTL AMP GAIN SETTING POSITION
         ;---------------------------------------------------

VR12_A000:
         CLR1  INTM1.4                  ;%a (REVERSE) PBCTL:↓ EDGE
         BT    PCAPFWD,$VR12_A00        ;%a CAPSTAN FORWARD OR REVERSE ?
         SET1  INTM1.4                  ;%a (FORWARD) PBCTL: ↑ EDGE

VR12_A00 :;B                            ;
         BT    FSMDCHG,$VR12_A10        ; AT TRANSITION? Yes

         MOV   A,RVSRVCD
         AND   A,#11110000B             ; CLEAR LOW-ORDER 4 BITS
         CMP   A,#CVFFREW               ; FF/REW ?
         BNE   $VR12_A01                ; No

;        /// CR13 COMPARATOR UPDATE (FF/REW) /// ;%

         MOVW  AX,CPT30                 ; LOAD PBCTL CAPTURE DATA
         MOVW  BC,#0133H                ;%(REVERSE) CPT30 x 1.2
                                        ; ...(256 x 1.2)
         BT    PCAPFWD,$VR12_A05        ; CAPSTAN FORWARD OR REVERSE ?
         MOVW  BC,#01CDH                ;%(FORWARD) CPT30 x 1.8
                                        ; ...(256 x 1.8)
VP12_A05 :;B                            ;
         MULUW BC                       ; CPT30 x ***
         MOV   A,X                      ; AX ← XB
         MOV   X,B                      ;
         BR    VR12_A04                 ;
         ;
         ;
;
```

```
;       /// CR13 COMPARATOR UPDATE (PLAY, CUE/REV) ///

VR12_A01 :;B
        MOVW  AX,CPT30                 ; LOAD PBCTL CAPTURE DATA
        MOVW  BC,#4CCDH                ;%(REVERSE) CPT30 x 0.3
                                       ; ...(65536 x 0.3)
        BT    PCAPFWD,$VR12_A02        ; CAPSTAN FORWARD OR REVERSE ?
        MOVW  BC,#0B333H               ;%(FORWARD) CPT30 x 0.7
                                       ; ...(65536 x 0.7)
VR12_A02 :;B
        MULUW BC                       ; CPT30 x ***
VR12_A04 :;B
        ADDW  AX,CR12                  ; (CPT30 x ***) + CR12

        CMPW  AX,CR10                  ;
        BC    $VR12_A03                ;
        SUBW  AX,CR10                  ;

VR12_A03 :
        MOVW  CR13,AX                  ; CR13 UPDATE

        CLR1  CRIF13                   ; CLEAR INTCR13 INTERRUPT REQUEST
        CLR1  CRMK13                   ; ENABLE INTCR13 INTERRUPT
VR12_A10 :;B
;%%%CTL

;-------------------------------------------------
;-- VISS MARK/ERASE
;-------------------------------------------------

        CALL  !SR12_000                ; VISS MARK/ERASE

        ;///// Servo mode judgment ///////

        MOV   A,RVSRVCD
        AND   A,#11110000b             ; CLEAR LOW-ORDER 4 BITS

        BF    FVPHFX,$VR12_200         ; PH FIX ON? No
        BR    VR12_500                 ; Yes (FF/REW)

$       EJECT
;-------------------------------------------------
; PLAY RUN MODE AUTOMATIC JUDGMENT PROCESSING
;-------------------------------------------------
;
;       Run mode is judged by the count number
;       of capstan FG signals after the event divider division
;       that is input into one cycle of PBCTL.
;
;
VR12_200 :;B

        BF    FVCFE05,$VR12_210        ; CAPSTAN SPEED ERROR IS MORE THAN 5%?
        BR    VR12_2B0                 ; Yes

VR12_210 :;B

        MOVG  WHL,#VR12_T00            ; %REFER TO TABLE
        MOV    A,RVCEVFG                ; RUN MODE JUDGEMENT CFG COUNTER
        MOVW   BC,#0900H                ; COUNTER INITIALIZATION

VR12_220 :;B

        CMP    A,[HL]
        BNL    $VR12_230                ; >=

        INC    A                        ; +1 CHECK
        CMP    A,[HL]
        BNE    $VR12_2B0                ; NOT MATCH

VR12_230 :;B

        BE     $VR12_240                ; =

        INCW   HL                       ; SET NEXT DATA
        INC    C                        ; SET PULSE TYPE COUNTER +1

        DBNZ   B,$VR12_220              ; CHECK COMPLETE? No
        BR     $VR12_2B0                ; Yes

VR12_240 :;B

        MOV    A,RVSLPCH                ; JUDGMENT CHATTERING COUNTER
        AND    A,#0FH                   ; READ BACK UP DATA

        XCH    A,C
        CMP    A,C                      ; MATCH?
        BE     $VR12_250                ; Yes

        MOV    RVSLPCH,A                ; INITIALIZE
        BR     VR12_2C0

VR12_250 :;B

        ADD    RVSLPCH,#10H             ; JUDGMENT CHATTERING COUNTER
                                        ; H INCREMENT

        CMP    RVSLPCH,#30H             ; CHATTERING ABSORPTION COMPLETE?
        BC     $VR12_2C0                ; No
                                        ; Yes

VR12_260 :
        ;//// Run mode set ///////

        XCH    A,C                      ; STORE PULSE TYPE COUNTER

        MOV    A,RVFSRV_2               ; GET RUN MODE
        AND    A,#3

        ADD    A,A
        MOV    B,A
        MOVW   AX,ttVR12_SPEED[B]
        MOV    B,#0
        ADDW   AX,BC
        MOVW   HL,AX

VR12_262 :
        MOV    A,[HL]
        CMP    A,#03                    ; CHANGE TO PAL MODE
```

```
         BNE   $VR12_264                ; No

         BF    FHIFIM,$VR12_264         ; PAL MODE ? Yes
         BR    VR12_2B0                 ; NO MODE CHANGE

VR12_264 :;J
         MOV   A,RVFSRV_2
         AND   A,#0FCH
         OR    A,[HL]
         XCH   A,RVFSRV_2               ; SET MODE

         XOR   A,RVFSRV_2
         AND   A,#03
         BE    $VR12_2B0                ; NO MODE CHANGE

VR12_270 :

         CALL  !YVTBL_00                ; REFER TO SET SERVO CODE & REFER TO TABLE

VR12_280 :

         CALLF !YSA01_R1                ; 1SEC TIMER START FOR AUTO-TRACKING

         CLR1  FSAEND                   ; ONE AUTO-TRACKING END
                                        ; CLEAR FLAG

         SET1  FSSPDCHG                 ; SET MARK/ERASE RELEASE REQUEST FLAG

VR12_2B0 :;B

         MOV   RVSLPCH,#00H             ; RUN MODE JUDGMENT CHATTERING COUNTER

VR12_2C0 :
         MOV   RVCEVFG,#00H             ; RUN MODE JUDGMENT CFG COUNTER
                                        ; INITIALIZE
         BR    $VR12_300

$        EJECT
;----------------------------------------------------------------------
; Capstan phase error calculation
;----------------------------------------------------------------------
;
;        Value of CR12
;              at PLAY   : Capture value of TM1 by PBCTL
;              at RECORD : Capture value of TM1 by CFG division signal
;
;        E CP = (CR12 value) – N CPL
;
;                        N CPL  : Capstan phase target value
;                        NF     : Internal reference timer value (CR10)
;----------------------------------------------------------------------

VR12_300 :;B
         ;/////Internal reference timer value ///////

         MOVW  DE,CR10                  ; INTERNAL REFERENCE TIMER VALUE
         SHRW  DE,1                     ; SET HALF CYCLE
```

```
VR12_310 :
        ;///// Phase error calculation ////////

        MOVW   AX,CR12                        ; LOAD PHASE CAPTURE DATA
        SUBW   AX,RVCPRF                      ; E CP = NP – N CPL
        MOVW   HL,AX                          ; HL

        BC     $VR12_321                      ; IS PHASE ERROR (-)?

                                              ; WHEN (+),
        SUBW   AX,DE                          ; SUBTRACT HALF CYCLE
        BC     $VR12_322                      ; WHEN WITHOUT CARRY,
                                              ; SUBTRACT HALF CYCLE AGAIN

        SUBW   AX,DE                          ;
        MOVW   HL,AX                          ;

        BR     VR12_323                       ; BECAUSE SIGN IS OPPOSITE,
                                              ; COMPARE WITH MINIMUM VALUE

VR12_321 :                                    ; WHEN (–),
        ADDW   AX,DE                          ; ADD HALF CYCLE
        BC     $VR12_323                      ; WHEN WITHOUT CARRY,
                                              ; ADD HALF CYCLE AGAIN

        ADDW   AX,DE                          ;
        MOVW   HL,AX                          ;

VR12_322 :
        MOV    B,NTSC_PAL                     ; (+) → COMPARE WITH MAXIMUM VALUE
        MOVW   AX,tCP_MAX[B]                  ;
        CMPW   AX,HL                          ;
        BNC    $VR12_325                      ;

        MOVW   HL,AX                          ;

        BR     $VR12_325                      ;

VR12_323 :
        MOV    B,NTSC_PAL                     ; (–) → COMPARE WITH MINIMUM VALUE
        MOVW   AX,tCP_MIN[B]                  ;
        CMPW   AX,HL                          ;
        BNH    $VR12_325                      ;

        MOVW   HL,AX                          ;

;;      BR     $VR12_325                      ;

VR12_325 :

        XCHW   HL,RVERCP                      ; RVERDP ← PHASE ERROR AMOUNT OF THIS TIME
        MOVW   RVERCP_1,HL                    ; RVERDP_1 ← PHASE ERROR AMOUNT OF LAST TIME
```

```
;------------------------------------------------------------------------------------
;  Lag read filter processing
;------------------------------------------------------------------------------------
;
;  0. Set filer coefficient
;
;    NTSC
;
;        f1 = 0.0245 Hz                  ─┐
;        f2 = 0.180  Hz                   |
;        DPG = 30 Hz                   □□┘
;
;        MAL a = –0.96299834
;        MBL b = –0.99488186
;        MGL g =  0.13832185
;
;
;    PAL
;
;        f1 = 0.0245 Hz                  ─┐
;        f2 = 0.190  Hz                   |
;        DPG = 25 Hz                   □□┘
;
;        MAL a = –0.95336134
;        MBL b = –0.99386137
;        MGL g =  0.13162089
;
VR12_400:;B

VR12_410:
        ; *** Clear filter memory when loading ***
        ;     * Because of inputting error information at loading

        CMP   RSNEXT,#CSMLOAD          ; LOADING?
        BNE   $VR12_420                ; No

        MOVW  RVERCP_Y,#0              ;
        MOVW  RVERCP,#0                ;
        MOVW  RVERCP_1,#0              ;
        MOVW  RVERCP_bY,#0             ;
        MOVW  RVERCP_bY+2,#0           ;

VR12_420:;B
        ;/////Run mode judgment ////////

        MOV   B,NTSC_PAL               ;

        MOVW  AX,tCP_fG[B]             ;
        MOVW  B_buf,AX                 ;
        MOVW  AX,tCP_fAG[B]            ;
        MOVW  B_buf+2,AX               ;
        MOVW  AX,tCP_fB[B]             ;
        MOVW  B_buf+4,AX               ;

VR12_426:;B

        MOV   B,#LOW(B_buf)            ;
        MOV   C,#LOW(RVERCP)           ;
        MOVW  DE,RVERCP_bY             ;
        MOVW  AX,RVERCP_bY+2           ;
        MACSW  2                        ;

        MOVW   RVERCP_Y,AX              ; CAPSTAN PHASE ERROR AMOUNT
                                        ; (AFTER FILTER OPERATION)

        MOVW   DE,B_buf+4               ;

        MULW   DE                       ;
        SHLW   DE,1                     ;
        ROLC   X,1                      ;
        ROLC   A,1                      ;

        MOVW   RVERCP_bY,DE             ;
        MOVW   RVERCP_bY+2,AX           ; (–b)•Y


;---------------------------------------------------------------------------------
; PBCTL signal missing check counter initialize
;---------------------------------------------------------------------------------

VR12_500:;B
        MOV    RVCEVFG,#00H             ; Clear CFG counter
        CLR1   FVFLCTL                  ; Reset PBCTL missing flag

;---------------------------------------------------------------------------------
;  Multiple interruption disable
;---------------------------------------------------------------------------------
VR12_600:

        DI

        POP    AX                       ; %RETURN MASK REGISTER

        MOV1   CY,CRMK13                ;% LOAD INTCR13 INTERRUPT MASK FLAG
                                        ;%CTL
        MOV1   X.3,CY                   ;% SAVE INTCR13 INTERRUPT MASK FLAG
                                        ;%CTL

        MOVW   MK1,AX                   ;%SET MASK REGISTER

        POP    AX                       ; RETURN MASK REGISTER

        MOV1   CY,CRMK02                ; LOAD INTCR02 INTERRUPT MASK FLAG
        MOV1   A.0,CY                   ; SAVE INTCR02 INTERRUPT MASK FLAG

        MOV1   CY,PMK2                  ;%LOAD INTP2 INTERRUPT MASK FLAG
        MOV1   A.7,CY                   ;%SAVE INTP2 INTERRUPT MASK FLAG

        MOV1   CY,CRMK11                ; LOAD INTCR11 INTERRUPT MASK FLAG
        MOV1   A.1,CY                   ; SAVE INTCR11 INTERRUPT MASK FLAG

        MOVW   MK0,AX                   ;

        RETI                            ; Vsync OR CR10 match
                                        ; INTERRUPT PROCESSING END
```

```
;--------------------------------------------------------------------------------
;  Run mode judgment table
;--------------------------------------------------------------------------------

        ;//// Number of CFG division ////////

VR12_T00 :
        DB     5,7,9,11,13,16,19,31,37   ; PULSE DATA

        ;//// Run mode conversion table ///

ttVR12_SPEED:
        DW     tVR12_T10_SP
        DW     tVR12_T10_SLP
        DW     tVR12_T10_LP
        DW     tVP12_T10_PAL

VR12_T10 :
tVR12_T10_SP:
        DB     1,2,0,3,0,0,0,0,0         ; SP mode
tVR12_T10_SLP:
        DB     1,1,1,1,1,1,2,3,0         ; SLP mode
tVR12_T10_LP:
        DB     2,1,2,2,2,3,0,2,2         ; LP mode
tVR12_T10_PAL:
        DB     1,2,3,3,0,3,3,3,3         ; SP (PAL) mode

$       EJECT

VCR00   CSEG   UNIT

;--------------------------------------------------------------------------------
;★★★  INTCR00 QUASI Vsync timing setting
;--------------------------------------------------------------------------------

;       VR00_000 : Register setting processing
;       VR00_100 : RFS level check
;       VR00_200 : Quasi Vsync rising edge timing setting
;       VR00_300 : Drum start processing (interrupt enable condition judgment)

;--------------------------------------------------------------------------------
;  Register setting processing
;--------------------------------------------------------------------------------

VR00_000 :;V
        ;/////Register setting /////////

        SEL    RB3                       ; HIGHEST-ORDER INTERRUPT!!

;--------------------------------------------------------------------------------
;   RFS level check
;--------------------------------------------------------------------------------

VR00_100 :
        BF     ICR.6,$VR00_200           ; CAPTURE BY RFS RISING EDGE? No

        MOVW   AX,CPT1L                  ;
        MOVW   HL,AX                     ;
        MOV    A,CPT1H                  ;
        MOV    W,A                      ;
        MOVG   RVCPT1,WHL               ;

        SET1   FSAFRQ                   ; DOWN EDGE SET FOR AUTO-TRACKING

;---------------------------------------------------------------------------------
;  Set QUASI Vsync rising timing
;---------------------------------------------------------------------------------
; QUASI Vsync outputs during search mode
; (CUE/REV), halt, and V.C mode.
;
 ;      ★ Rising timing
;
;       Fixed value
;       τd = HSW delay amount + 3
;          = (CR00 setting value) + 191 μsec
;          = (CR00 setting value) + 191d                           ;%
;
;       Variable value: CH2 at STILL/FRAME
;       τd = HSW delay amount + 2H to 4H (initial value) to 6H
;          = (CR00 setting value) + 128 to 255 to 382 μsec         ;%
;          = (CR00 setting value) + 128d to 255d to 382d           ;%
;
;       ★ Falling edge timing
;
;       4H = 63.55 x 4 = 254.2 μsec
;          = 254D                                                  ;%
;                                       ∴1H = 63.55 μsec
;
;            * Timing match interrupt → INTCR02
;
;---------------------------------------------------------------------------------
VR00_200 :;B

        BF     FVDOUT,$VR00_300         ; QUASI Vsync OUTPUT MODE? No

        BT     ICR.6,$VR00_230          ; RFS RISING EDGE? No

VR00_210 :

        MOV    A,RVSRVCD
        AND    A,#11110000B             ; CLEAR LOW-ORDER 4 BIT
        CMP    A,#CVSTILL               ; SERVO CODE STILL?

        BNE    $VR00_230                ; No

VR00_220 :
        ;///// Variable value ////////

        MOVW   AX,#00                   ; CLEAR BUFFER

        MOV    A,RVPSVCNT               ; VARIABLE VALUE (00-FEH)
        XCH    A,X

        ADDW   AX,#128D                 ; %BASIC VALUE
        ADDW   AX,CR00                  ; RISING TIMING
        MOVW   CR02,AX                  ; SET FALLING TIMING

        BR     VR00_240
```

```
VR00_230:;B
        ;/////Fixed value ////////

        MOVW  AX,CR00                 ; RISING TIMING
        ADDW  AX,#191D                ;%SET DATA
        MOVW  CR02,AX                 ;

VR00_240:;B

        SET1  P8L.0                   ;%<DATA OUTPUT TO P80 WITH TRIGGER: “1”>

        CLR1  CRMK02                  ; INTCR02 ENABLE
        SET1  FVHQVDT                 ; SET RISING TIMING FLAG

;------------------------------------------------------------------
;  Drum rising processing (Interrupt enable conditions judgment)
;------------------------------------------------------------------
VR00_300:;B

        BF    FSDRMON,$VR00_400       ; DRUM ON? No

        BF    ICR.6,$VR00_400         ; RFS FALLING EDGE INPUT? No

        CLR1  FSEICPT2                ; CLEAR INTCPT2 INTERRUPT ENABLE
                                      ; REQUEST FLAG

;------------------------------------------------------------------
;       V-MUTE release RFS synchronization processing
;------------------------------------------------------------------
VR00_400:;B

        BF    FSVM0FRQ,$VR00_500      ; V-MUTE RELEASE REQUEST? No

        CLR1  FSVM0FRQ                ; CLEAR REQUEST
        CLR1  PQVD                    ; V-MUTE OFF
        CLR1  FPQVD                   ; SET PORT REFRESH FLAG
        SET1  PMC8.0                  ;%P80 PT0 OUTPUT MODE


;------------------------------------------------------------------
;  Reverse brake at CUE → PLAY RFS synchronization processing
;------------------------------------------------------------------
VR00_500:;B

        BF    FSCRRFRQ,$VR00_600      ; REQUEST CAPSTAN REVERSE RFS
                                      ; SYNCHRONIZATION?
        CLR1  FSCRRFRQ

        SET1  PCAPFWD                 ; CAPSTAN MOTOR REVERSE START
        SET1  PCAPF_R
        SET1  FPCAPF_R                ; SET PORT REFRESH FLAG

        MOVG  WHL,#RNSTIM0            ;%
        MOV   A,RVFSRV_2
        AND   A,#00000011B            ; READ RUN MODE

        CMP   A,#CVPAL                ; PAL?
        BE     $VR00_510                ; Yes

        CMP    A,#CVSP                  ; NTSC SP?

        BE     $VR00_510

        CMP    A,#CVSLP                 ; NTSC SLP?
        BE     $VR00_520

        MOV    A,#33H                   ; NTSC LP
        BR     $VR00_530                ; 70 msec TIMER SET

VR00_510 :;B
                                        ; PAL SP, NTSC SP
        MOV    A,#50H                   ; 110 msec TIMER SET
        BR     $VR00_530

VR00_520 :;B
                                        ; NTSC SLP
        MOV    A,#2CH                   ; 60 msec TIMER SET

VR00_530 :;B
        MOV    [HL],A                   ; TIMER START
        CLR1   FNSTENO                  ;

VR00_600 :;B

        RETI                            ; INTCR00 INTERRUPT PROCESSING END

$       EJECT

VCR02   CSEG   UNIT


;--------------------------------------------------------------------------------
;★★★ Set INTCR02 QUASI Vsync timing
;--------------------------------------------------------------------------------

;       VR02_000 : Register set processing
;       VR02_100 : Set quasi Vsync falling timing
;       VR02_200 : INTCR02 interrupt disable processing


;--------------------------------------------------------------------------------
;   Register setting processing
;--------------------------------------------------------------------------------

VR02_000 :;V
        ;///// Register setting /////////

        SEL    RB3                      ; Highest-order interrupt!!


;--------------------------------------------------------------------------------
; Set QUASI Vsync falling edge timing
;--------------------------------------------------------------------------------

VR02_100 :

        BTCLR  FVHQVDT,$VR02_110        ; Rising timing interrupt?
        BR     $VR02_200                ; No
```

```
VR02_110 :;B

        MOVW  AX,CR02                  ; FALLING EDGE TIMING
        ADDW  AX,#254D                 ;%SET DATA
        MOVW  CR02,AX

        CLR1  P8L.0                    ;%<DATA OUTPUT TO P80 WITH TRIGGER:
                                       ; “0” > (Addition)

        BR    $VR02_300

;-------------------------------------------------------------------
;       INTCR02 interrupt disable processing
;-------------------------------------------------------------------
VR02_200 :
        SET1  CRMK02                   ; INTCR02 disable

VR02_300 :
        RETI                           ; INTCR02 interrupt end
$       EJECT


;-------------------------------------------------------------------
;★★★                  SERVO DATA TABLE
;-------------------------------------------------------------------
;
; • Table description
;
;
;       DB  TMC0  (Timer 0 control register setting value)
;       DB  CPTM  (Capture mode register setting value)
;       DB  INTM1 (External capture input mode register setting value)
;
;      [SP/LP/SLP/PAL]
;
;       DB  Value of EDVC
;       DB  CR12 Macro service counter
;       DW  REF30Hz                    (CR10)
;       DW  Drum speed target value    (CPT2H)
;       DB  Drum speed target value    (CPT2L)
;       DW  Drum bias adding value
;       DW  Capstan speed target value (CPT3)
;       DW  Capstan bias adding value
;       DW  Capstan gain adjustment value
;
;-------------------------------------------------------------------

        ;------------------------------
        ;     PLAY
        ;------------------------------
SDT_PLAY :
        DB    10001001B                ; TMC0   COUNT:EN      TM1:CLR
                                       ; TM0:CLR
        DB    00110000B                ;%CPTM   CPT0-TRG:TM1=CR10
                                       ; CR12-TRG:CTI11,CPT1:↑↓ EDGE
        DB    00010001B                ;%INTM1  PBCTL:AN_AMP,
                                       ; PBCTL:↑ EDGE CFG:↑ EDGE
                                       ; <-(01010001B)
        ;/////SP /////////////////

SDT_PLS0:
        DB    03D                        ; EDVC COUNT
SDT_PLS1:
        DB    01H                        ; MACRO COUNT
SDT_PLS2:
        DW    8256H                      ; CR10
SDT_PLS3:
        DG    2B72H                      ; CPT2
SDT_PLS5:
        DW    66F0H                      ; DRUM BIAS
SDT_PLS6:
        DG    56CEH                      ; CPT3
SDT_PLS7:
        DW    85E0H                      ; CAPSTAN BIAS
SDT_PLS8:
        DW    600H                       ; LOOP GAIN
SDT_PLS9:
        DW    433H                       ; CAPSTAN KV/KP
SDT_PLSA:
        DB    17H                        ; CTL AMP GAIN

        ;/////LP /////////////////

SDT_PLL0:
        DB    02D                        ; EDVC COUNT
SDT_PLL1:
        DB    01H                        ; MACRO COUNT
SDT_PLL2:
        DW    8256H                      ; CR10
SDT_PLL3:
        DG    2B72H                      ; CPT2
SDT_PLL5:
        DW    66FFH                      ; DRUM BIAS
SDT_PLL6:
        DG    73BDH                      ; CPT3
SDT_PLL7:
        DW    8595H                      ; CAPSTAN BIAS
SDT_PLL8:
        DW    0540H                      ; LOOP GAIN
SDT_PLL9:
        DW    0159H                      ; CAPSTAN KV/KP
SDT_PLLA:
        DB    1DH                        ;%% CTL AMP GAIN

        ;/////SLP /////////////////

SDT_PLE0:
        DB    01D                        ; EDVC COUNT
SDT_PLE1:
        DB    01H                        ; MACRO COUNT
SDT_PLE2:
        DW    8256H                      ; CR10
SDT_PLE3:
        DG    2B72H                      ; CPT2
SDT_PLE5:
        DW    66FFH                      ; DRUM BIAS
SDT_PLE6:
        DG    56CEH                      ; CPT3
```

```
SDT_PLE7:
        DW    86DFH                     ; CAPSTAN BIAS
SDT_PLE8:
        DW    0420H                     ; LOOP GAIN
SDT_PLE9:
        DW    0119H                     ; CAPSTAN KV/KP
SDT_PLEA:
        DB    1DH                       ;%% CTL AMP GAIN

        ;///// PAL //////////////

SDT_PLP0:
        DB    03D                       ; EDVC COUNT
SDT_PLP1:
        DB    01H                       ; MACRO COUNT
SDT_PLP2:
        DW    9C40H                     ; CR10
SDT_PLP3:
        DG    3415H                     ; CPT2
SDT_PLP5:
        DW    66FFH                     ; DRUM BIAS
SDT_PLP6:
        DG    7BC1H                     ; CPT3
SDT_PLP7:
        DW    863FH                     ; CAPSTAN BIAS
SDT_PLP8:
        DW    0600H                     ; LOOP GAIN
SDT_PLP9:
        DW    0166H                     ; CAPSTAN KV/KP
SDT_PLPA:
        DB    1DH                       ;%% CTL AMP GAIN

        ;------------------------------
        ;     RVS PLAY                  ; SP/LP/SLP/PAL PLAY
        ;------------------------------
SDT_RVS :
        DB    10001001B                 ; TMC0 COUNT:EN  TM1:CLR TM0:CLR
        DB    00110000B                 ;%CPTM CPT0-TRG:TM1=CR10
                                        ; CR12-TRG:CTI11,CPT1:↑↓EDGE
        DB    00010001B                 ;%INTM1 PBCTL:AN_AMP,PBCTL:↑ EDGE
                                        ; CFG:↑ EDGE <-(01010001B)

        ;///// SP //////////////

SDT_RPS0:
        DB    03D                       ; EDVC COUNT
SDT_RPS1:
        DB    01H                       ; MACRO COUNT
SDT_RPS2:
        DW    83D3H                     ; CR10
SDT_RPS3:
        DG    2BF1H                     ; CPT2
SDT_RPS5:
        DW    66FFH                     ; DRUM BIAS
SDT_RPS6:
        DG    56CEH                     ; CPT3
SDT_RPS7:
        DW    8678H                     ; CAPSTAN BIAS
SDT_RPS8:
        DW    0600H                     ; LOOP GAIN
```

```
SDT_RPS9 :
        DW     0233H                         ; CAPSTAN KV/KP
SDT_RPSA :
        DB     17H                           ;%% CTL AMP GAIN

        ;///// LP /////////////////

SDT_RPL0 :
        DB     02D                           ; EDVC COUNT
SDT_RPL1 :
        DB     01H                           ; MACRO COUNT
SDT_RPL2 :
        DW     8314H                         ; CR10
SDT_RPL3:
        DG     2BB1H                         ; CPT2
SDT_RPL5 :
        DW     66FFH                         ; DRUM BIAS
SDT_RPL6 :
        DG     73BDH                         ; CPT3
SDT_RPL7 :
        DW     8595H                         ; CAPSTAN BIAS
SDT_RPL8 :
        DW     0540H                         ; LOOP GAIN
SDT_RPL9 :
        DW     0159H                         ; CAPSTAN KV/KP
SDT_RPLA :
        DB     1DH                           ;%% CTL AMP GAIN

        ;///// SLP /////////////////

SDT_RPE0 :
        DB     01D                           ; EDVC COUNT
SDT_RPE1 :
        DB     01H                           ; MACRO COUNT
SDT_RPE2 :
        DW     82D5H                         ; CR10
SDT_RPE3 :
        DG     2B9CH                         ; CPT2
SDT_RPE5 :
        DW     66FFH                         ; DRUM BIAS
SDT_RPE6 :
        DG     56CEH                         ; CPT3
SDT_RPE7 :
        DW     86DFH                         ; CAPSTAN BIAS
SDT_RPE8 :
        DW     0420H                         ; LOOP GAIN
SDT_RPE9 :
        DW     0119H                         ; CAPSTAN KV/KP
SDT_RPEA :
        DB     1DH                           ;%% CTL AMP GAIN

        ;///// PAL /////////////////

SDT_RPP0 :
        DB     03D                           ; EDVC COUNT
SDT_RPP1 :
        DB     01H                           ; MACRO COUNT
SDT_RPP2 :
        DW     9DC0H                         ; CR10
SDT_RPP3 :
        DG     3495H                    ; CPT2
SDT_RPP5 :
        DW     66FFH                    ; DRUM BIAS
SDT_RPP6 :
        DG     7BC1H                    ; CPT3
SDT_RPP7 :
        DW     863FH                    ; CAPSTAN BIAS
SDT_RPP8 :
        DW     0600H                    ; LOOP GAIN
SDT_RPP9 :
        DW     0166H                    ; CAPSTAN KV/KP
SDT_RPPA :
        DB     1DH                      ;%% CTL AMP GAIN


        ;------------------------------
        ;      FF/REW (2H)              ; NTSC PLAY SP
        ;------------------------------
SDT_FR2H :
        DB     10001001B                ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                        ;CTI11,CPT1;↑↓ EDGE
        DB     00010001B                ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                        ; CFG:↑EDGE <-(01010001B)

        ;/////SP /////////////////

SDT_2HS0:
        DB     03D                      ; EDVC COUNT
SDT_2HS1 :
        DB     01H                      ; MACRO COUNT
SDT_2HS2 :
        DW     8256H                    ; CR10
SDT_2HS3 :
        DG     2B72H                    ; CPT2
SDT_2HS5 :
        DW     66FFH                    ; DRUM BIAS
SDT_2HS6 :
        DG     56CEH                    ; CPT3
SDT_2HS7 :
        DW     8678H                    ; CAPSTAN BIAS
SDT_2HS8 :
        DW     0600H                    ; LOOP GAIN
SDT_2HS9 :
        DW     0233H                    ; CAPSTAN KV/KP
SDT_2HSA :
        DB     0EH                ;%%CTL AMP GAIN

        ;///// LP /////////////////

SDT_2HL0 :
        DB     03D                      ; EDVC COUNT
SDT_2HL1 :
        DB     01H                      ; MACRO COUNT
SDT_2HL2 :
        DW     8256H                    ; CR10
SDT_2HL3 :
        DG     2B72H                    ; CPT2
SDT_2HL5 :
        DW     66FFH                    ; DRUM BIAS
```

```
SDT_2HL6 :
         DG    56CEH                    ; CPT3
SDT_2HL7 :
         DW    8678H                    ; CAPSTAN BIAS
SDT_2HL8 :
         DW    0600H                    ; LOOP GAIN
SDT_2HL9 :
         DW    0233H                    ; CAPSTAN KV/KP
SDT_2HLA :
         DB    0EH                      ;%% CTL AMP GAIN

         ;/////SLP /////////////////

SDT_2HE0:
         DB    03D                      ; EDVC COUNT
SDT_2HE1 :
         DB    01H                      ; MACRO COUNT
SDT_2HE2 :
         DW    8256H                    ; CR10
SDT_2HE3 :
         DG    2B72H                    ; CPT2
SDT_2HE5 :
         DW    66FFH                    ; DRUM BIAS
SDT_2HE6 :
         DG    56CEH                    ; CPT3
SDT_2H1E7 :
         DW    8678H                    ; CAPSTAN BIAS
SDT_2HE8 :
         DW    0600H                    ; LOOP GAIN
SDT_2HE9 :
         DW    0233H                    ; CAPSTAN KV/KP
SDT_2HEA :
         DB    0EH                      ;%% CTL AMP GAIN

         ;/////PAL /////////////////

SDT_2HP0 :
         DB    03D                      ; EDVC COUNT
SDT_2HP1 :
         DB    01H                      ; MACRO COUNT
SDT_2HP2 :
         DW    9C40H                    ; CR10
SDT_2HP3 :
         DG    3415H                    ; CPT2
SDT_2HP5 :
         DW    66FFH                    ; DRUM BIAS
SDT_2HP6 :
         DG    7BC1H                    ; CPT3
SDT_2HP7 :
         DW    8678H                    ; CAPSTAN BIAS
SDT_2HP8 :
         DW    0600H                    ; LOOP GAIN
SDT_2HP9 :
         DW    0233H                    ; CAPSTAN KV/KP
SDT_2HPA :
         DB    0EH                      ;%% CTL AMP GAIN
```

```
        ;------------------------------
        ;      FF/REW (6H)             ; CAPSTAN INITIAL SPEED
        ;------------------------------
SDT_FR6H:
        DB     10001001B               ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B               ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                       ;CTI11,CPT1:↑↓EDGE
        DB     00010001B               ;%INTM1 PBCTL:AN_AMP, PBCTL:↑EDGE
                                       ; CFG:↑ EDGE <-(01010001B)

        ;///// SP ////////////////

SDT_6HS0:
        DB     01D                     ; EDVC COUNT
SDT_6HS1:
        DB     01H                     ; MACRO COUNT
SDT_6HS2:
        DW     8256H                   ; CR10
SDT_6HS3:
        DG     2B72H                   ; CPT2
SDT_6HS5:
        DW     66FFH                   ; DRUM BIAS
SDT_6HS6:
        DG     56CEH                   ; CPT3
SDT_6HS7:
        DW     86DFH                   ; CAPSTAN BIAS
SDT_6HS8:
        DW     00C0H                   ; LOOP GAIN
SDT_6HS9:
        DW     0119H                   ; CAPSTAN KV/KP
SDT_6HSA:
        DB     0EH                     ;%% CTL AMP GAIN

        ;///// LP ////////////////

SDT_6HL0:
        DB     01D                     ; EDVC count
SDT_6HL1:
        DB     01H                     ; MACRO count
SDT_6HL2:
        DW     8256H                   ; CR10
SDT_6HL3:
        DG     2B72H                   ; CPT2
SDT_6HL5:
        DW     66FFH                   ; DRUM BIAS
SDT_6HL6:
        DG     56CEH                   ; CPT3
SDT_6HL7:
        DW     86DFH                   ; CAPSTAN BIAS
SDT_6HL8:
        DW     00C0H                   ; LOOP GAIN
SDT_6HL9:
        DW     0119H                   ; CAPSTAN KV/KP
SDT_6HLA:
        DB     0EH                     ;%% CTL AMP GAIN

        ;///// SLP ////////////////
SDT_6HE0:
        DB     01D                     ; EDVC COUNT
```

```
SDT_6HE1 :
        DB      01H                             ; MACRO COUNT
SDT_6HE2 :
        DW      8256H                           ; CR10
SDT_6HE3 :
        DG      2B72H                           ; CPT2
SDT_6HE5 :
        DW      66FFH                           ; DRUM BIAS
SDT_6HE6 :
        DG      56CEH                           ; CPT3
SDT_6HE7 :
        DW      86DFH                           ; CAPSTAN BIAS
SDT_6HE8 :
        DW      00C0H                           ; LOOP GAIN
SDT_6HE9 :
        DW      0119H                           ; CAPSTAN KV/KP
SDT_6HEA :
        DB      0EH                             ;%% CTL AMP GAIN

        ;///// PAL /////////////////

SDT_6HP0 :
        DB      01D                             ; EDVC COUNT
SDT_6HP1 :
        DB      01H                             ; MACRO COUNT
SDT_6HP2 :
        DW      9C40H                           ; CR10
SDT_6HP3 :
        DG      3415H                           ; CPT2
SDT_6HP5 :
        DW      66FFH                           ; DRUM BIAS
SDT_6HP6 :
        DG      56CEH                           ; CPT3
SDT_6HP7 :
        DW      86DFH                           ; CAPSTAN BIAS
SDT_6HP8 :
        DW      00C0H                           ; LOOP GAIN
SDT_6HP9 :
        DW      0119H                           ; CAPSTAN KV/KP
SDT_6HPA :
        DB      0EH                             ;%% CTL AMP GAIN

        ;------------------------------
        ;       FF/REW (6H) VD OUT      ; CAPSTAN FG 90 PULSES DRIVE
        ;------------------------------
SDT_6HVD :
        DB      10001001B                       ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB      00110000B                       ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                                ; CTI11,CPT1:↑↓EDGE
        DB      00010001B                       ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                                ; CFG:↑EDGE <-(01010001B)

        ;///// SP /////////////////

SDT_6VS0 :
        DB      01D                             ; EDVC COUNT
SDT_6VS1 :
        DB      01H                             ; MACRO COUNT
SDT_6VS2 :
        DW      8315H                           ; CR10
```

```
SDT_6VS3 :
        DG    2BB1H                    ; CPT2
SDT_6VS5 :
        DW    66FFH                    ; DRUM BIAS
SDT_6VS6 :
        DG    56CEH                    ; CPT3
SDT_6VS7 :
        DW    86DFH                    ; CAPSTAN BIAS
SDT_6VS8 :
        DW    0420H                    ; LOOP GAIN
SDT_6VS9 :
        DW    0119H                    ; CAPSTAN KV/KP
SDT_6VSA :
        DB    0EH                      ;%% CTL AMP GAIN

        ;///// LP /////////////////

SDT_6VL0 :
        DB    01D                      ; EDVC COUNT
SDT_6VL1 :
        DB    01H                      ; MACRO COUNT
SDT_6VL2 :
        DW    82B5H                    ; CR10
SDT_6VL3 :
        DG    2B91h                    ; CPT2
SDT_6VL5 :
        DW    66FFH                    ; DRUM BIAS
SDT_6VL6 :
        DG    56CEH                    ; CPT3
SDT_6VL7 :
        DW    86DFH                    ; CAPSTAN BIAS
SDT_6VL8 :
        DW    0420H                    ; LOOP GAIN
SDT_6VL9 :
        DW    0119H                    ; CAPSTAN KV/KP
SDT_6VLA :
        DB    0EH                      ;%% CTL AMP GAIN

        ;///// SLP /////////////////

SDT_6VE0 :
        DB    01D                      ; EDVC COUNT
SDT_6VE1 :
        DB    01H                      ; MACRO COUNT
SDT_6VE2 :
        DW    8295H                    ; CR10
SDT_6VE3 :
        DG    2B91H                    ; CPT2
SDT_6VE5 :
        DW    66FFH                    ; DRUM BIAS
SDT_6VE6 :
        DG    56CEH                    ; CPT3
SDT_6VE7 :
        DW    86DFH                    ; CAPSTAN BIAS
SDT_6VE8 :
        DW    0420H                    ; LOOP GAIN
SDT_6VE9 :
        DW    0119H                    ; CAPSTAN KV/KP
SDT_6VEA :
        DB    0EH                      ;%% CTL AMP GAIN
        ;///// PAL /////////////////

SDT_6VP0 :
        DB     01D                       ; EDVC COUNT
SDT_6VP1 :
        DB     01H                       ; MACRO COUNT
SDT_6VP2 :
        DW     9D00H                     ; CR10
SDT_6VP3 :
        DG     3455H                     ; CPT2
SDT_6VP5 :
        DW     66FFH                     ; DRUM BIAS
SDT_6VP6 :
        DG     56CEH                     ; CPT3
SDT_6VP7 :
        DW     86DFH                     ; CAPSTAN BIAS
SDT_6VP8 :
        DW     0420H                     ; LOOP GAIN
SDT_6VP9 :
        DW     0119H                     ; CAPSTAN KV/KP
SDT_6VPA :
        DB     0EH                       ;%% CTL AMP GAIN

        ;------------------------------
        ;      FF/REW (2H*3)             ; NTSC SLP CUE/REV
        ;------------------------------
SDT_FRX3 :
        DB     10001001B                 ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                 ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                         ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                 ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                         ; CFG:↑EDGE <-(01010001B)

        ;///// SP /////////////////

SDT_X3S0:
        DB     03D*3                     ; EDVC COUNT
SDT_X3S1 :
        DB     09D                       ; MACRO COUNT
SDT_X3S2 :
        DW     8315H                     ; CR10
SDT_X3S3 :
        DG     2BB1H                     ; CPT2
SDT_X3S5 :
        DW     66FFH                     ; DRUM BIAS
SDT_X3S6 :
        DG     56CEH                     ; CPT3
SDT_X3S7 :
        DW     8678H                     ; CAPSTAN BIAS
SDT_X3S8 :
        DW     0600H                     ; LOOP GAIN
SDT_X3S9 :
        DW     034CH                     ; CAPSTAN KV/KP
SDT_X3SA :
        DB     0EH                       ;%% CTL AMP GAIN
        ;///// LP /////////////////

SDT_X3L0:
        DB     03D*3                     ; EDVC count
SDT_X3L1 :
        DB     09D                       ; MACRO count
SDT_X3L2 :
        DW     82B5H                     ; CR10
SDT_X3L3 :
        DG     2B91H                     ; CPT2
SDT_X3L5 :
        DW     66FFH                     ; DRUM BIAS
SDT_X3L6 :
        DG     56CEH                     ; CPT3
SDT_X3L7 :
        DW     8595H                     ; CAPSTAN BIAS
SDT_X3L8 :
        DW     0600H                     ; LOOP GAIN
SDT_X3L9 :
        DW     034CH                     ; CAPSTAN KV/KP
SDT_X3LA :
        DB     0EH                       ;%% CTL AMP GAIN

        ;///// SLP /////////////////

SDT_X3E0 :
        DB     03D*3                     ; EDVC COUNT
SDT_X3E1 :
        DB     09D                       ; MACRO COUNT
SDT_X3E2 :
        DW     8295H                     ; CR10
SDT_X3E3 :
        DG     2BB7H                     ; CPT2
SDT_X3E5 :
        DW     66FFH                     ; DRUM BIAS
SDT_X3E6 :
        DG     56CEH                     ; CPT3
SDT_X3E7 :
        DW     86DFH                     ; CAPSTAN BIAS
SDT_X3E8 :
        DW     0600H                     ; LOOP GAIN
SDT_X3E9 :
        DW     034CH                     ; CAPSTAN KV/KP
SDT_X3EA :
        DB     0EH                       ;%%CTL AMP GAIN

        ;///// PAL /////////////////

SDT_X3P0 :
        DB     03D*3                     ; EDVC COUNT
SDT_X3P1 :
        DB     09D                       ; MACRO COUNT
SDT_X3P2 :
        DW     9D00H                     ; CR10
SDT_X3P3 :
        DG     3455H                     ; CPT2
SDT_X3P5 :
        DW     66FFH                     ; DRUM BIAS
SDT_X3P6 :
        DG     56CEH                     ; CPT3
SDT_X3P7 :
        DW     863FH                     ; CAPSTAN BIAS
```

```
SDT_X3P8 :
        DW     0600H                    ; LOOP GAIN
SDT_X3P9 :
        DW     034CH                    ; CAPSTAN KV/KP
SDT_X3PA :
        DB     0EH                      ;%% CTL AMP GAIN

        ;------------------------------
        ;      FF/REW                   ; NTSC SP PLAY
        ;------------------------------
SDT_FFRW :
        DB     10001001B                ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                        ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                        ; CFG:↑EDGE <-(01010001B)

        ;///// SP /////////////////

SDT_FRS0 :
        DB     40D                      ; EDVC COUNT
SDT_FRS1 :
        DB     08D                      ; MACRO COUNT
SDT_FRS2 :
        DW     8256H                    ; CR10
SDT_FRS3 :
        DG     2B72H                    ; CPT2
SDT_FRS5 :
        DW     66FFH                    ; DRUM BIAS
SDT_FRS6 :
        DG     56CEH                    ; CPT3
SDT_FRS7 :
        DW     8678H                    ; CAPSTAN BIAS
SDT_FRS8 :
        DW     0600H                    ; LOOP GAIN
SDT_FRS9 :
        DW     0233H                    ; CAPSTAN KV/KP
SDT_FRSA :
        DB     0EhH                     ;%% CTL AMP GAIN

        ;///// LP /////////////////

SDT_FRL0:
        DB     40D                      ; EDVC COUNT
SDT_FRL1 :
        DB     08D                      ; MACRO COUNT
SDT_FRL2 :
        DW     8256H                    ; CR10
SDT_FRL3 :
        DG     2B72H                    ; CPT2
SDT_FRL5 :
        DW     66FFH                    ; DRUM BIAS
SDT_FRL6 :
        DG     56CEH                    ; CPT3
SDT_FRL7 :
        DW     8595H                    ; CAPSTAN BIAS
SDT_FRL8 :
        DW     0600H                    ; LOOP GAIN
SDT_FRL9 :
        DW     0233H                    ; CAPSTAN KV/KP
```

```
SDT_FRLA:
        DB     0EH                       ;%% CTL AMP GAIN

        ;///// SLP /////////////////

SDT_FRE0:
        DB     40D                       ; EDVC COUNT
SDT_FRE1:
        DB     08D                       ; MACRO COUNT
SDT_FRE2:
        DW     8256H                     ; CR10
SDT_FRE3:
        DG     2B72H                     ; CPT2
SDT_FRE5:
        DW     66FFH                     ; DRUM BIAS
SDT_FRE6:
        DG     56CEH                     ; CPT3
SDT_FRE7:
        DW     86DFH                     ; CAPSTAN BIAS
SDT_FRE8:
        DW     0600H                     ; LOOP GAIN
SDT_FRE9:
        DW     0233H                     ; CAPSTAN KV/KP
SDT_FREA:
        DB     0EH                       ;%% CTL AMP GAIN

        ;///// PAL /////////////////

SDT_FRP0:
        DB     40D                       ; EDVC COUNT
SDT_FRP1:
        DB     08D                       ; MACRO COUNT
SDT_FRP2:
        DW     9C40H                     ; CR10
SDT_FRP3:
        DG     3415H                     ; CPT2H
SDT_FRP5:
        DW     66FFH                     ; DRUM BIAS
SDT_FRP6:
        DG     7BC1H                     ; CPT3
SDT_FRP7:
        DW     863H                      ; CAPSTAN BIAS
SDT_FRP8:
        DW     0600H                     ; LOOP GAIN
SDT_FRP9:
        DW     0233H                     ; CAPSTAN KV/KP
SDT_FRPA:
        DB     0EH                       ;%% CTL AMP GAIN

        ;------------------------------
        ;           CUE
        ;------------------------------
SDT__CUE:
        DB     10001001B                 ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                 ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                         ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                 ;%INTM1 PBCTL:AN_AMP,PBCTL:↑ EDGE
                                         ; CFG:↑EDGE <-(01010001B)
        ;/////SP ////////////////

SDT_CUS0:
        DB    15D                        ; EDVC COUNT
SDT_CUS1:
        DB    05D                        ; MACRO COUNT
SDT_CUS2:
        DW    7F5CH                      ; CR10
SDT_CUS3:
        DG    2A74H                      ; CPT2
SDT_CUS5:
        DW    66FFH                      ; DRUM BIAS
SDT_CUS6:
        DG    54E8H                      ; CPT3
SDT_CUS7:
        DW    8678H                      ; CAPSTAN BIAS
SDT_CUS8:
        DW    0600H                      ; LOOP GAIN
SDT_CUS9:
        DW    0433H                      ; CAPSTAN KV/KP
SDT_CUSA:
        DB    11H                        ;%% CTL AMP GAIN

        ;/////LP ////////////////

SDT_CUL0:
        DB    09D*2                      ; EDVC COUNT
SDT_CUL1:
        DB    09D                        ; MACRO COUNT
SDT_CUL2:
        DW    7F5EH                      ; CR10
SDT_CUL3:
        DG    2A74H                      ; CPT2
SDT_CUL5:
        DW    66FFH                      ; DRUM BIAS
SDT_CUL6:
        DG    54E9H                      ; CPT3
SDT_CUL7:
        DW    8595H                      ; CAPSTAN BIAS
SDT_CUL8:
        DW    0600H                      ; LOOP GAIN
SDT_CUL9:
        DW    034CH                      ; CAPSTAN KV/KP
SDT_CULA:
        DB    12H                        ;%% CTL AMP GAIN

        ;/////SLP ////////////////

SDT_CUE0:
        DB    09D                        ; EDVC COUNT
SDT_CUE1:
        DB    09D                        ; MACRO COUNT
SDT_CUE2:
        DW    805CH                      ; CR10
SDT_CUE3:
        DG    2AC9H                      ; CPT2
SDT_CUE5:
        DW    66FFH                      ; DRUM BIAS
SDT_CUE6:
        DG    5592H                      ; CPT3
```

```
SDT_CUE7 :
        DW     86DFH                      ; CAPSTAN BIAS
SDT_CUE8 :
        DW     0600H                      ; LOOP GAIN
SDT_CUE9 :
        DW     034CH                      ; CAPSTAN KV/KP
SDT_CUEA :
        DB     12H                        ;%% CTL AMP GAIN

        ;///// PAL /////////////////

SDT_CUP0 :
        DB     21D                        ; EDVC COUNT
SDT_CUP1 :
        DB     07D                        ; MACRO COUNT
SDT_CUP2 :
        DW     97C0H                      ; CR10
SDT_CUP3 :
        DG     3295H                      ; CPT2
SDT_CUP5 :
        DW     66FFH                      ; DRUM BIAS
SDT_CUP6 :
        DG     652AH                      ; CPT3
SDT_CUP7 :
        DW     863FH                      ; CAPSTAN BIAS
SDT_CUP8 :
        DW     0600H                      ; LOOP GAIN
SDT_CUP9 :
        DW     034CH                      ; CAPSTAN KV/KP
SDT_CUPA :
        DB     12H                        ;%% CTL AMP GAIN

        ;-----------------------------
        ;      CUE → PLAY
        ;-----------------------------
SDT_CUPL :
        DB     10001001B                  ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                  ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                          ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                  ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                          ; CFG:↑EDGE <-(01010001B)

        ;///// SP /////////////////

SDT_CPS0 :
        DB     15D                        ; EDVC COUNT
SDT_CPS1 :
        DB     05D                        ; MACRO COUNT
SDT_CPS2 :
        DW     80D9H                      ; CR10
SDT_CPS3 :
        DG     2AF3H                      ; CPT2
SDT_CPS5 :
        DW     66FFH                      ; DRUM BIAS
SDT_CPS6 :
        DG     54E8H                      ; CPT3
SDT_CPS7 :
        DW     8678H                      ; CAPSTAN BIAS
SDT_CPS8 :
        DW     0600H                      ; LOOP GAIN
```

```
SDT_CPS9 :
        DW    0433H                       ; CAPSTAN KV/KP
SDT_CPSA :
        DB    17H                         ;%% CTL AMP GAIN

        ;///// LP /////////////////

SDT_CPL0 :
        DB    09D*2                       ; EDVC COUNT
SDT_CPL1 :
        DB    09D                         ; MACRO COUNT
SDT_CPL2 :
        DW    80D9H                       ; CR10
SDT_CPL3 :
        DG    2AF3H                       ; CPT2
SDT_CPL5 :
        DW    66FFH                       ; DRUM BIAS
SDT_CPL6 :
        DG    7137H                       ; CPT3
SDT_CPL7 :
        DW    8595H                       ; CAPSTAN BIAS
SDT_CPL8 :
        DW    0600H                       ; LOOP GAIN
SDT_CPL9 :
        DW    034CH                       ; CAPSTAN KV/KP
SDT_CPLA :
        DB    1DH                         ;%% CTL AMP GAIN

        ;///// SLP /////////////////

SDT_CPE0 :
        DB    09D                         ; EDVC COUNT
SDT_CPE1 :
        DB    09D                         ; MACRO COUNT
SDT_CPE2 :
        DW    8159H                       ; CR10
SDT_CPE3 :
        DG    2B1DH                       ; CPT2
SDT_CPE5 :
        DW    66FFH                       ; DRUM BIAS
SDT_CPE6 :
        DG    55E7H                       ; CPT3
SDT_CPE7 :
        DW    86DFH                       ; CAPSTAN BIAS
SDT_CPE8 :
        DW    0600H                       ; LOOP GAIN
SDT_CPE9 :
        DW    034CH                       ; CAPSTAN KV/KP
SDT_CPEA :
        DB    1DH                         ;%% CTL AMP GAIN

        ;///// PAL /////////////////

SDT_CPP0 :
        DB    21D                         ; EDVC COUNT
SDT_CPP1 :
        DB    07D                         ; MACRO COUNT
SDT_CPP2 :
        DW    9940H                       ; CR10
SDT_CPP3 :
        DG     3315H                    ; CPT2
SDT_CPP5 :
        DW     66FFH                    ; DRUM BIAS
SDT_CPP6 :
        DG     652AH                    ; CPT3
SDT_CPP7 :
        DW     863FH                    ; CAPSTAN BIAS
SDT_CPP8 :
        DW     0600H                    ; LOOP GAIN
SDT_CPP9 :
        DW     034CH                    ; CAPSTAN KV/KP
SDT_CPPA :
        DB     1DH                      ;%% CTL AMP GAIN

        ;-----------------------------
        ;      REVIEW
        ;-----------------------------
SDT__REV :
        DB     10001001B                ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                        ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                        ; CFG:↑EDGE <-(01010001B)

        ;/////SP /////////////////

SDT_RVS0 :
        DB     15D                      ; EDVC COUNT
SDT_RVS1 :
        DB     05D                      ; MACRO COUNT
SDT_RVS2 :
        DW     86CEH                    ; CR10
SDT_RVS3 :
        DG     2CEFH                    ; CPT2
SDT_RVS5 :
        DW     66FFH                    ; DRUM BIAS
SDT_RVS6 :
        DG     59DFH                    ; CPT3
SDT_RVS7 :
        DW     8678H                    ; CAPSTAN BIAS
SDT_RVS8 :
        DW     0600H                    ; LOOP GAIN
SDT_RVS9 :
        DW     0433H                    ; CAPSTAN KV/KP
SDT_RVSA :
        DB     11H                      ;%% CTL AMP GAIN

        ;/////LP /////////////////

SDT_RVL0 :
        DB     09D*2                    ; EDVC COUNT
SDT_RVL1 :
        DB     09D                      ; MACRO COUNT
SDT_RVL2 :
        DW     860DH                    ; CR10
SDT_RVL3 :
        DG     2CAFH                    ; CPT2
SDT_RVL5 :
        DW     66FFH                    ; DRUM BIAS
```

```
SDT_RVL6 :
        DG    7728H                     ; CPT3
SDT_RVL7 :
        DW    8595H                     ; CAPSTAN BIAS
SDT_RVL8 :
        DW    0600H                     ; LOOP GAIN
SDT_RVL9 :
        DW    034CH                     ; CAPSTAN KV/KP
SDT_RVLA :
        DB    12H                       ;%% CTL AMP GAIN

        ;///// SLP ///////////////

SDT_RVE0 :
        DB    09D                       ; EDVC COUNT
SDT_RVE1 :
        DB    09D                       ; MACRO COUNT
SDT_RVE2 :
        DW    84CFH                     ; CR10
SDT_RVE3 :
        DG    2C45H                     ; CPT2
SDT_RVE5 :
        DW    66FFH                     ; DRUM BIAS
SDT_RVE6 :
        DG    588AH                     ; CPT3
SDT_RVE7 :
        DW    86DFH                     ; CAPSTAN BIAS
SDT_RVE8 :
        DW    0600H                     ; LOOP GAIN
SDT_RVE9 :
        DW    034CH                     ; CAPSTAN KV/KP
SDT_RVEA :
        DB    12H                       ;%% CTL AMP GAIN

        ;///// PAL ///////////////

SDT_RVP0 :
        DB    21D                       ; EDVC COUNT
SDT_RVP1 :
        DB    07D                       ; MACRO COUNT
SDT_RVP2 :
        DW    0A240H                    ; CR10
SDT_RVP3 :
        DG    3615H                     ; CPT2
SDT_RVP5 :
        DW    66FFH                     ; DRUM BIAS
SDT_RVP6 :
        DG    6C2AH                     ; CPT3
SDT_RVP7 :
        DW    863FH                     ; CAPSTAN BIAS
SDT_RVP8 :
        DW    0600H                     ; LOOP GAIN
SDT_RVP9 :
        DW    034CH                     ; CAPSTAN KV/KP
SDT_RVPA :
        DB    12H                       ;%% CTL AMP GAIN
        ;------------------------------
        ;      STILL                    ; SP/LP/SLP/PAL PLAY
        ;------------------------------
SDT_STIL :
        DB     10001001B                ; TMC0 COUNT:EN TM1:CLR TM0:CLR
        DB     00110000B                ;%CPTM CPT0-TRG:TM1=CR10 CR12-TRG:
                                        ; CTI11,CPT1:↑↓EDGE
        DB     00010001B                ;%INTM1 PBCTL:AN_AMP,PBCTL:↑EDGE
                                        ; CFG:↑EDGE <-(01010001B)

        ;///// SP ////////////////

SDT_STS0 :
        DB     03D                      ; EDVC COUNT
SDT_STS1 :
        DB     01H                      ; MACRO COUNT
SDT_STS2 :
        DW     8315H                    ; CR10
SDT_STS3 :
        DG     2BB1H                    ; CPT2
SDT_STS5 :
        DW     66FFH                    ; DRUM BIAS
SDT_STS6 :
        DG     56CEH                    ; CPT3
SDT_STS7 :
        DW     8678H                    ; CAPSTAN BIAS
SDT_STS8 :
        DW     0600H                    ; LOOP GAIN
SDT_STS9 :
        DW     0233H                    ; CAPSTAN KV/KP
SDT_STSA :
        DB     17H                      ;%% CTL AMP GAIN

        ;///// LP ////////////////

SDT_STL0 :
        DB     02D                      ; EDVC COUNT
SDT_STL1 :
        DB     01H                      ; MACRO COUNT
SDT_STL2 :
        DW     8265H                    ; CR10
SDT_STL3 :
        DG     2B91H                    ; CPT2
SDT_STL5 :
        DW     66FFH                    ; DRUM BIAS
SDT_STL6 :
        DG     73BDH                    ; CPT3
SDT_STL7 :
        DW     8595H                    ; CAPSTAN BIAS
SDT_STL8 :
        DW     0540H                    ; LOOP GAIN
SDT_STL9 :
        DW     0159H                    ; CAPSTAN KV/KP
SDT_STLA :
        DB     1DH                      ;%% CTL AMP GAIN

        ;///// SLP ////////////////

SDT_STE0 :
        DB     01D                      ; EDVC COUNT
```

```
SDT_STE1 :
        DB     01H                         ; MACRO COUNT
SDT_STE2 :
        DW     8295H                       ; CR10
SDT_STE3 :
        DG     2BB7H                       ; CPT2
SDT_STE5 :
        DW     66FFH                       ; DRUM BIAS
SDT_STE6 :
        DG     56CEH                       ; CPT3
SDT_STE7 :
        DW     86DFH                       ; CAPSTAN BIAS
SDT_STE8 :
        DW     0420H                       ; LOOP GAIN
SDT_STE9 :
        DW     0119H                       ; CAPSTAN KV/KP
SDT_STEA :
        DB     1DH                         ;%% CTL AMP GAIN

        ;///// PAL ///////////////

SDT_STP0 :
        DB     03D                         ; EDVC COUNT
SDT_STP1 :
        DB     01H                         ; MACRO COUNT
SDT_STP2 :
        DW     9D00H                       ; CR10
SDT_STP3 :
        DG     3455H                       ; CPT2
SDT_STP5 :
        DW     66FFH                       ; DRUM BIAS
SDT_STP6 :
        DG     7BC1H                       ; CPT3
SDT_STP7 :
        DW     863FH                       ; CAPSTAN BIAS
SDT_STP8 :
        DW     0600H                       ; LOOP GAIN
SDT_STP9 :
        DW     0166H                       ; CAPSTAN KV/KP
SDT_STPA :
        DB     1DH                         ;%% CTL AMP GAIN

$       EJECT

SERVOSUB       CSEG   FIXED

;--------------------------------------------------------------------------------
;★★★    SERVO DATA Setting SUB
;--------------------------------------------------------------------------------
;
;
;       • TMC0 (Timer 0 Control register)
;       • CPTM (Capture mode register)
;       • INTM1 (External capture input mode register)
;       • EDVC (Event divider control register)
;       • CR12 Macro service counter
;       • REF30Hz (CR10)
;       • Drum speed target value (CPT2H)
;       • Drum speed target value (CPT2L)
;        • Drum bias adding value
;        • Capstan speed target value (CPT3)
;        • Capstan bias adding value
;        • Loop gain
;        • Clear filter
;        • Clear capstan error amount
;
;--------------------------------------------------------------------------------

YVTBL_00 :;C
        ;/////Interrupt control /////////

        DI                              ; TO BE REFERRED AT SERVO RELATION CONTROL

YVTBL_10 :
        ;///// Address setting /////////

        MOV    A,RVSRVCD
        AND    A,#0F0H                  ; SERVO CODE LOW-ORDER MASK

        MOVG   WHL,#SDT_PLAY            ;%SET PLAY

        CMP    A,#CVFFRW2Hrs            ; FF/REW(2Hrs)?
        BNE    $YVTBL_11                ; No
        MOVG   WHL,#SDT_FR2Hrs          ;% Yes

YVTBL_11 :;B
        CMP    A,#CVFFRW6Hrs            ; FF/REW(6Hrs)?
        BNE    $YVTBL_12                ; No
        MOVG   WHL,#SDT_FR6Hrs          :% Yes

YVTBL_12 :;B
        CMP    A,#CVFFRWX3              ; FF/REW(2Hrs*3)?
        BNE    $YVTBL_13                ; No
        MOVG   WHL,#SDT_FRX3            ;% Yes

YVTBL_13 :;B
        CMP    A,#CVFFREW               ; FF/REW?
        BNE    $YVTBL_14                ; No
        MOVG   WHL,#SDT_FFRW            ;% Yes

YVTBL_14 :;B
        CMP    A,#CVCUE                 ; CUE?
        BNE    $YVTBL_15                ; No
        MOVG   WHL,#SDT__CUE            ;% Yes

YVTBL_15 :;B
        CMP    A,#CVREV                 ; REVIEW?
        BNE    $YVTBL_16                ; No
        MOVG   WHL,#SDT__REV            ;% Yes

YVTBL_16 :;B
        CMP    A,#CVSTILL               ; STILL?
        BNE    $YVTBL_17                ; No

        MOVG   WHL,#SDT_STIL            ;% Yes

YVTBL_17 :;B
        CMP    A,#CVRVS                 ; RVS PLAY?
        BNE    $YVTBL_18                ; No
```

```
         MOVG  WHL_#SDT_RVS             ;% Yes

YVTBL_18 :;B
         CMP   A,#CVCUPL                : CUE → PLAY?
         BNE   $YVTBL_19                ; No PLAY

         MOVG  WHL,#SDT_CUPL            ;% Yes

YVTBL_19 :;B
         CMP   A,#CVFR6HVD              ; 6Hrs PLAY VD OUT ?
         BNE   $YVTBL_20                ; No PLAY

         MOVG  WHL,#SDT_6HVD            ;% Yes

YVTBL_20 :
         ;/////TMC0 Read /////

         MOV   A,[HL+]
         MOV   TMC0,A

YVTBL_30 :
         ;/////CPTM Read /////

         MOV   A,[HL+]
         MOV   CPTM,A

YVTBL_40 :
         ;/////INTM1 Read /////

         MOV   A,[HL+]
         MOV   INTM1,A

YVTBL_50 :
         ;/////Address adjustment /////////

         MOV   A,RVFSRV_2               ;
         AND   A,#00000011B             ; Read running mode

         MOVW  BC,#00H                  ; Set address adding value (SP)

         CMP   A,#CVLP                  ; LP?
         BNE   $YVTBL_51                ; No
         MOVW  BC,#SDT_PLL0-SDT_PLS0    ;% Yes (Number of table byte x 1)

YVTBL_51 :;B
         CMP   A,#CVSLP                 ; SLP?
         BNE   $YVTBL_52                ; No
         MOVW  BC,#SDT_PLE0-SDT_PLS0    ;% Yes (Number of table byte x 2)

YVTBL_52 :;B
         CMP   A,#CVPAL                 ; PAL?
         BNE   $YVTBL_53                ; No
         MOVW  BC,#SDT_PLP0-SDT_PLS0    ;% Yes (Number of table byte x 3)

YVTBL_53 :;B
         MOVW  DE,BC                    ;% Address adding
         MOV   T,#0                     ;%
         ADDG  WHL,TDE                  ;%
```

```
YVTBL_60 :;B
        ;///// EDVC /////////////

        MOV    A,[HL+]
        DEC    A                        ; ;ADD%%
        MOV    EDVC,A

YVTBL_70 :
        ;///// CR12 /////////////

        MOV    A,[HL+]
        MOV    RVMCCR12,A
        MOV    RVCRAM,A

YVTBL_80 :
        ;///// CR10 /////////////

        MOVW   AX,[HL+]                 ; WRITE INTO CR10 IS PERFORMED IN
                                        ; INTCR10 ROUTINE.
        MOVW   RVBCR10,AX               ; (TM1 MAY OVERFLOW DEPENDING ON WRITE
                                        ; TIMING!)
        CALL   !YPGADCHG                ; SET PG VALUE


YVTBL_90 :
        ;///// CPT2 ///////////

        MOVG   TDE,WHL
        MOVG   WHL,[TDE+]
        MOVG   RVDFRF,WHL
        MOVG   WHL,TDE

YVTBL_B0 :
        ;///// D-BIAS /////////

        MOVW   AX,[HL+]
        MOVW   RVDBAS,AX

YVTBL_C0 :
        ;///// CPT3 /////////////

        MOVG   TDE,WHL
        MOVG   WHL,[TDE+]
        MOVG   RVCFRF,WHL
        MOVG   WHL,TDE

YVTBL_D0 :
        ;///// C-BIAS /////////

        MOVW   AX,[HL+]
        MOVW   RVCBAS,AX

YVTBL_E0 :
        ;///// C-GAIN ADJUSTMENT /////

        MOVW   AX,[HL+]
        MOVW   RVC_Kmp,AX
```

```
YVTBL_F0 :
         ;///// C-Kv/Kp ///////

         MOVW  AX,[HL+]
         MOVW  RVC_Kvp,AX               ;

YVTBL_F1 :
         ;/////PB_CTL gain ///////

         MOV   A,[HL]                   ;
         MOV   CTLM,A                   ;

YVTBL_G0 :                       =
         ;/////Clear filter /////

                                        ; CAPSTAN SPEED/PHASE
         MOVW  RVERCMX,#0               ;
         MOVW  RVERCMX_1,#0             ;
         MOVW  RVERCMX_bY,#0            ;
         MOVW  RVERCMX_bY+2,#0          ;

         MOVW  RVERCP,#0                ; CAPSTAN PHASE FILTER
         MOVW  RVERCP_1,#0              ;
         MOVW  RVERCP_bY,#0             ;
         MOVW  RVERCP_bY+2,#0           ;

YVTBL_H0 :
         ;/////Clear C-Error ///////

         MOVW  RVERCP_Y,#0              ; CAPSTAN PHASE ERROR
         MOVW  RVERCMX_Y+2,#0           ; CAPSTAN SPEED ERROR

YVTBL_I0 :
         ;/////Interrupt control /////////

         EI
         RET

         END
```

[MEMO]

APPENDIX REVISION HISTORY

The history of revisions hitherto made is shown as follows.

Edition	Revisions	Chapter
Second	Addition of μPD784915A	Throughout
	Addition of related document number	Introduction
	The following figures are changed. • Figure 3-4 Use of Event Counter (EC) is corrected. • Figure 3-5 Event Counter (EC) Operation Timing is corrected. • Figure 3-6 Use of Timer 0 is corrected. • Figure 3-7 Head Switching Signal (V-HSW) Timing (PTO00) is corrected. • Figure 3-19 Timer 1 Peripheral Circuit is corrected. • Figure 3-27 Use of Timer for Drum Phase Control (for recording) is corrected. • Figure 3-28 Drum Phase Control Timing (for recording) is corrected.	CHAPTER 3 EXAMPLES OF STATIONARY TYPE VCR SERVO CONTROL
Third	The μPD784928, 784928Y Subseries and the μPD784915B, 784916B are added.	Throughout
	Document numbers of related documents are added or corrected.	Introduction
	CHAPTER 1 OUTLINE OF NEC VCR SERVO MICROCONTROLLER PRODUCTS is added.	CHAPTER 1 OUTLINE OF NEC VCR SERVO MICROCONTROLLER PRODUCTS
	Table 2-1 Differences among μPD784915 Subseries Products is added.	CHAPTER 2 OUTLINE OF μPD784915 SUBSERIES
	CHAPTER 3 OUTLINE OF μPD784928, 784928Y SUBSERIES is added.	CHAPTER 3 OUTLINE OF μPD784928, 784928Y SUBSERIES

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: ______________________________

Document number: ______________________ Page number: __________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏