NEC

MOS INTEGRATED CIRCUIT

μPD61051, 61052

MPEG2 AUDIO/VIDEO ENCODER

The μPD61051 and μPD61052 are LSIs of MPEG audio and video encoding, decoding and transcoding.

The μPD61051 has MPEG2 video encoder, MPEG audio encoding DSP, 32-bit RISC CPU, video input/output unit which contains a processing filter and a time base corrector (TBC), and MPEG system layer which contains the multiplexer and de-multiplexer. It combines with 64 M or 128 Mbit SDRAM and it uses. The μPD61052 has a Dolby™ Digital Consumer Encoder in addition to the μPD61051.

The μPD61051, 61052 are the optimal choice for consumer digital video recording replay equipment to process a MPEG.

FEATURES

- Video encode
  - Stream standard: MPEG2 video MP@ML, SP@ML standard, MPEG1 standard
  - Picture size: Horizontal: 720, 704, 544, 480, 352 dots/line
    Vertical: 480, 240, 576, 288 line/frame
  - Single pass variable bit rate (VBR), constant bit rate (CBR) encoding
  - Transcoding: Bit rate conversion, VBR ⇔ CBR
  - Video input/output
    Format: 8-bit Y/Cb/Cr 4:2:2 (ITU-R BT.656)
    Pre analysis: Film detect, scene changing detect, and motion estimation assist
    TBC, VBI data slicer
- Audio encoding
  - Bit length: 16 bits, 20 bits, 24 bits
  - Sampling rate: 32 kHz, 44.1 kHz, 48 kHz
  - MPEG1 audio layer 2 standard based
  - Dolby Digital Consumer Encoder standard based (Only the μPD61052)
  - Elementary stream and PCM audio input/output
- MPEG system processing
  - Multiplex: MPEG2-PS, MPEG2-TS, DVD-Video, and DVD-VR
  - De-multiplex: MPEG2-PS, MPEG2-TS
  - Transcoding: MPEG2 format conversion (MPEG2-TS ⇔ MPEG2-PS)
  - Partial TS generation
- Package: 208-pin fine pitch QFP
- Power supply: 1200 mW (Typ.)
- Power supply voltage: 3.3±0.165 V, 2.5±0.2 V (Internal circuit power)

"Dolby" is a trademark of Dolby Laboratories.
To use the μPD61052, a license from Dolby Laboratories Licensing Corporation is necessary.

Document No. S15082EJ3V0DS00 (3rd edition)
Date Published April 2003 NS CP (K)
Printed in Japan

The mark ★ shows major revised points.

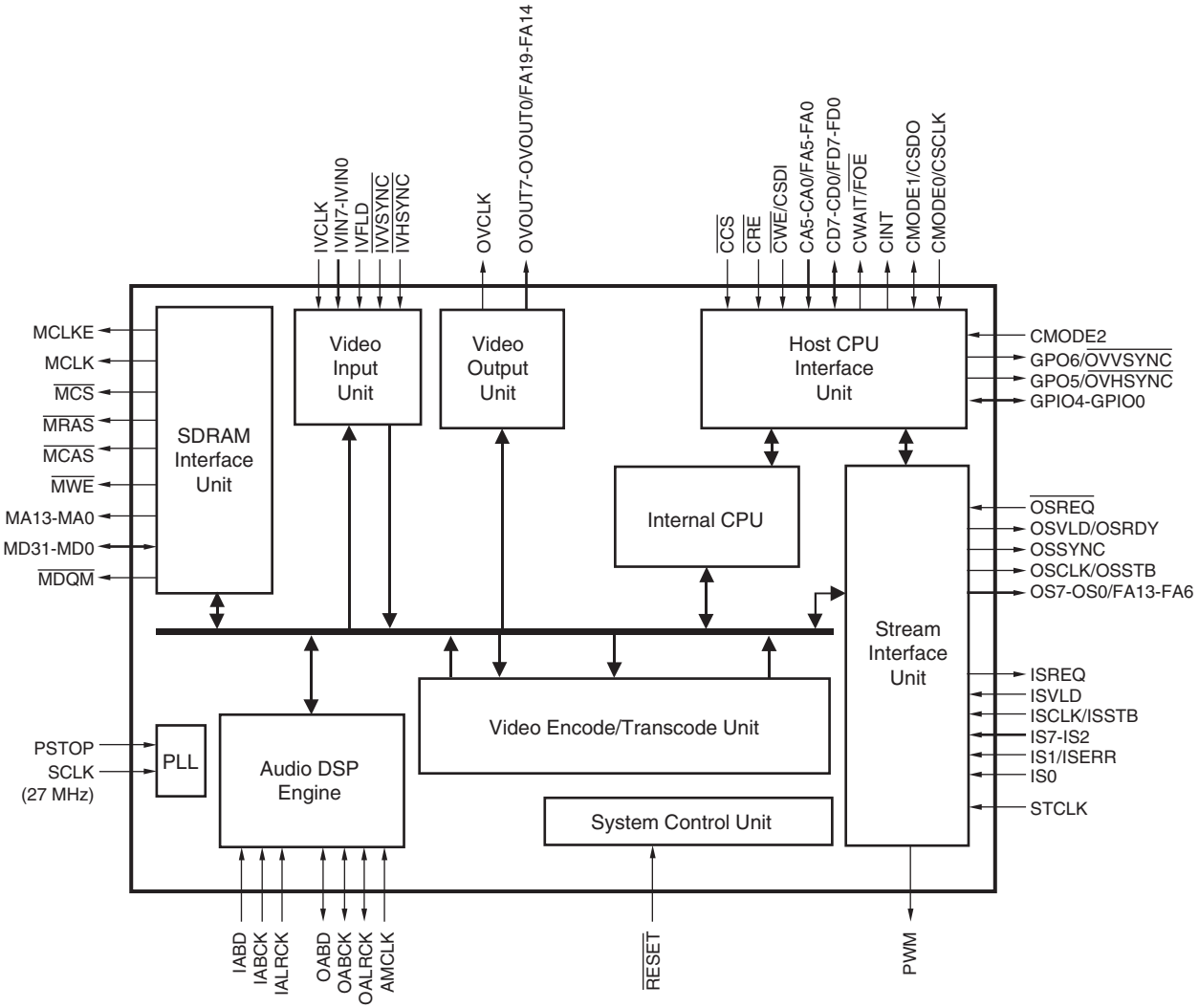
APPLICATION

D-VHS, DVD video recorder, HDD video recorder

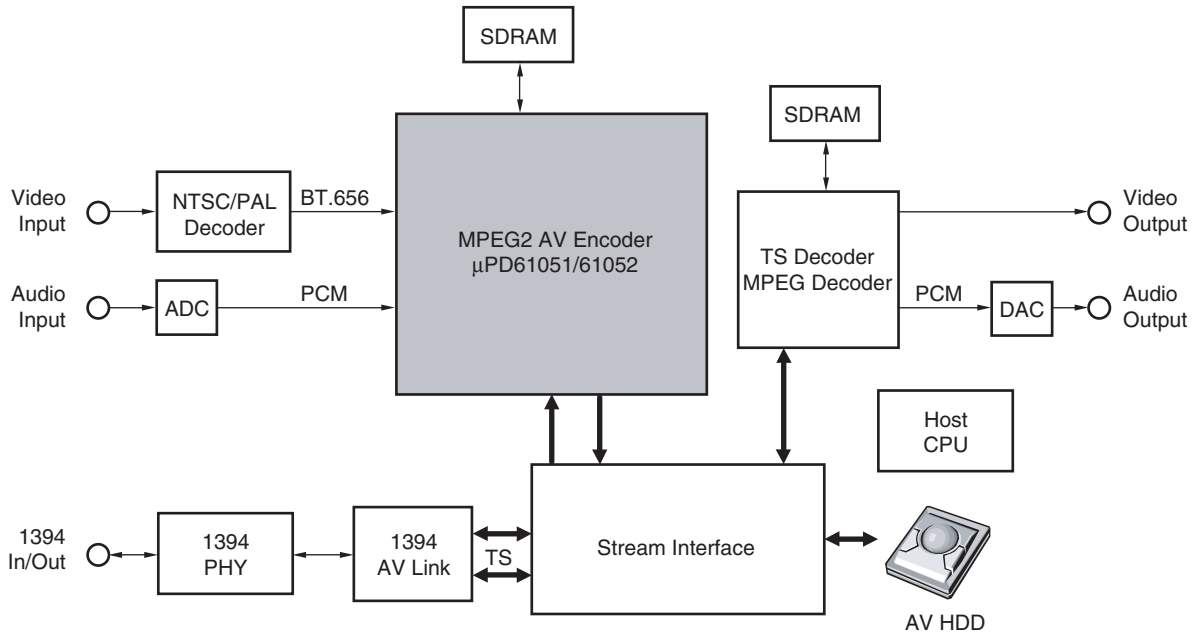
ORDERING INFORMATION

Part Number	Package
μPD61051GD-LML	208-pin plastic QFP (Fine pitch)(28×28)
μPD61052GD-LML	208-pin plastic QFP (Fine pitch)(28×28)

BLOCK DIAGRAM

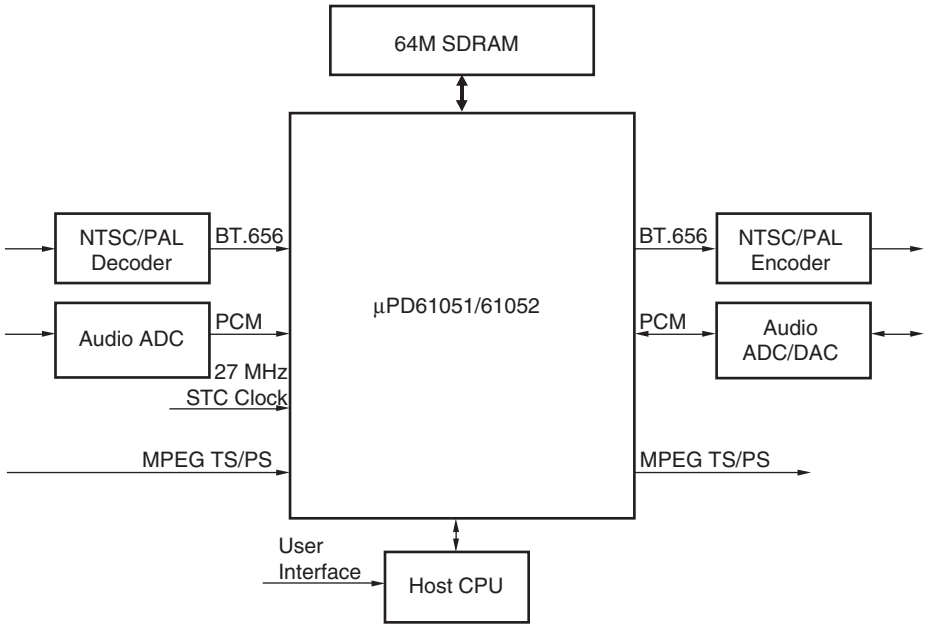


PERIPHERAL CONNECTION

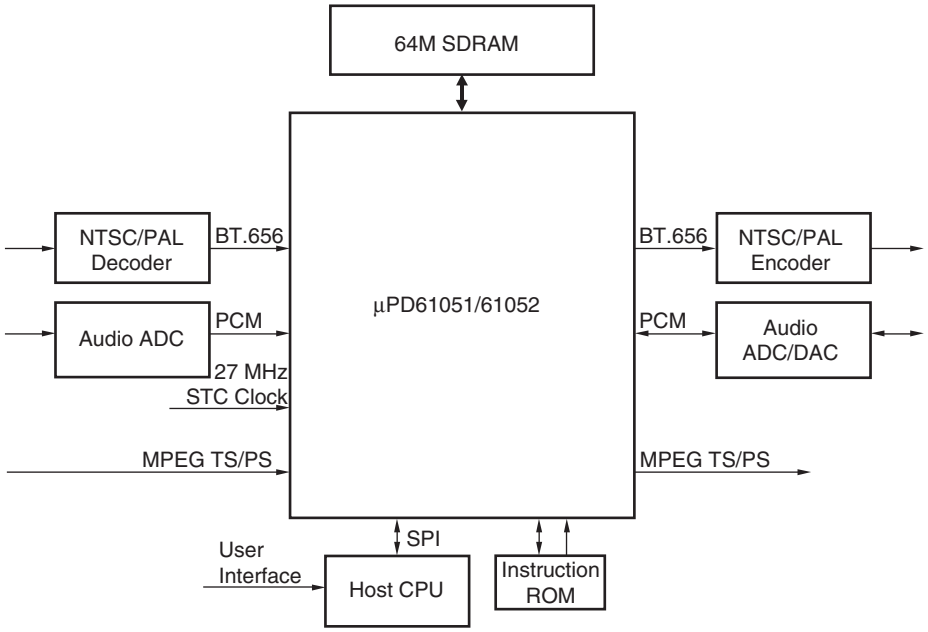


This LSI deals with two kinds of methods to connect a system controller.

Parallel Bus Interface



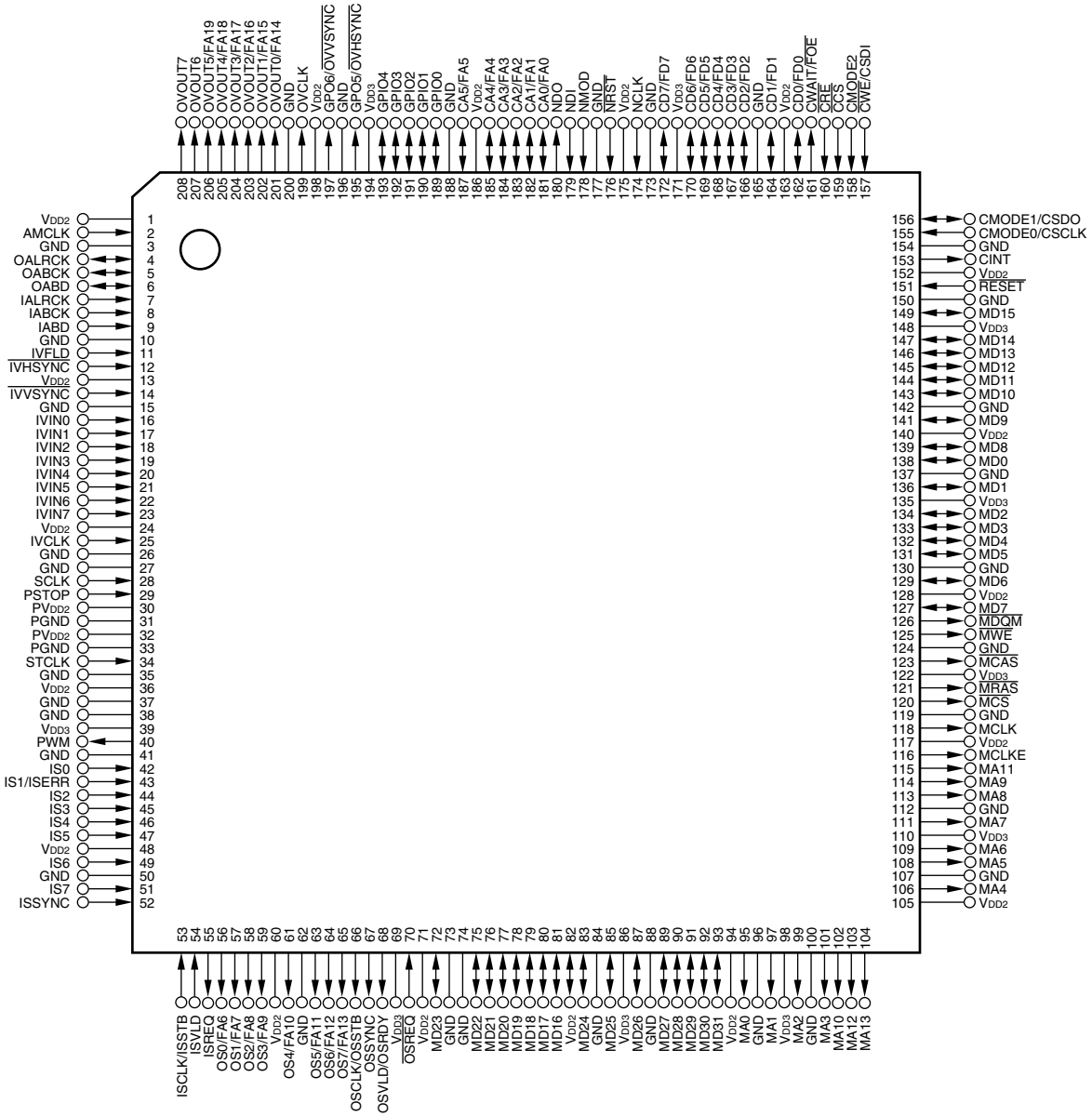
Serial Bus Interface



PIN CONFIGURATION (TOP VIEW)

- 208-pin plastic QFP (Fine pitch)(28×28)

μPD61051GD-LML
μPD61052GD-LML



PIN LIST

AMCLK :Audio Main Clock
CA0/FA0 to CA5/FA5 :Host CPU Address/ Instruction ROM Address
$\overline{CCS}$:Host CPU Chip Select
CD0/FD0 to CD7/FD7 :Host CPU Data/ Instruction ROM Data
CINT :Host CPU Interrupt
CMODE0/CSCLK :Host CPU Mode/ SPI Clock
CMODE1/CSDO :Host CPU Mode/ SPI Data Output
CMODE2 :Host CPU Mode
$\overline{CRE}$:Host CPU Read Enable
CWAIT/$\overline{FOE}$:Host CPU Wait/ Instruction ROM Output Enable
$\overline{CWE}$/CSDI :Host CPU Write Enable/ SPI Data Input
GND :Ground
GPIO0 to GPIO4 :General Purpose IO
GPO5/$\overline{OVHSYNC}$:General Purpose Output/ Output Video Horizontal Sync
GPO6/$\overline{OVVSYNC}$:General Purpose Output/ Output Video Vertical Sync
IABCK :Input Audio Bit Clock
IABD :Input Audio Bit Data
IALRCK :Input Audio LR Clock
IS0, IS2 to IS7 :Input Stream Data
IS1/ISERR :Input Stream Data/ Input Stream Error
ISCLK/ISSTB :Input Stream Data Clock/ Input Stream Data Strobe
ISREQ :Input Stream Data Request
ISSYNC :Input Stream Data Sync
ISVLD :Input Stream Data Valid
IVCLK :Input Video Clock
IVFLD :Input Video Field Index
$\overline{IVHSYNC}$:Input Video Horizontal Sync
IVIN0 to IVIN7 :Input Video Data
$\overline{IVVSYNC}$:Input Video Vertical Sync
MA0 to MA13 :Memory Address
$\overline{MCAS}$:Memory Column Address Strobe
MCLK :Memory Clock
MCLKE :Memory Clock Enable
$\overline{MCS}$:Memory Chip Select
MD0 to MD31 :Memory Data
MDQM :Memory DQ Mask Enable
$\overline{MRAS}$:Memory Row Address Strobe
$\overline{MWE}$:Memory Write Enable
NCLK :N-wire Clock
NDI :N-wire Data Input
NDO :N-wire Data Output
NMOD :N-wire Mode
$\overline{NRST}$:N-wire Reset
OABCK :Output Audio Bit Clock
OABD :Output Audio Bit Data
OALRCK :Output Audio LR Clock
OS0/FA6 to OS7/FA13 :Output Stream Data/ Instruction ROM Address
OSCLK/OSSTB :Output Stream Data Clock/ Output Stream Data Strobe
$\overline{OSREQ}$:Output Stream Data Request
OSSYNC :Output Stream Data Sync
OSVLD/OSRDY :Output Stream Data Valid/ Output Stream Data Ready
OVCLK :Output Video Clock
OVOUT0/FA14 to OVOUT5/FA19 :Output Video Data/ Instruction ROM Address
OVOUT6,OVOUT7 :Output Video Data
PGND :PLL Ground
PSTOP :PLL Stop
PV_{DD2} :PLL 2.5 V Power Supply
PWM :PWM Output
$\overline{RESET}$:Reset
SCLK :System Clock
STCLK :System Time Clock
V_{DD2} :2.5 V Power Supply
V_{DD3} :3.3 V Power Supply

CONTENTS

1. PIN FUNCTION

Sharing pin is bold faced in name and explains the feature shown.

1.1 Video Input Interface

The video input is based on the ITU-R BT.656 format. The horizontal synchronization signal, and the vertical synchronization signal, the field index can be used without using SAV and EAV to provide at ITU-R BT. 656, too.

Name	IO	Pin Number	Function	Active Polarity
IVIN7 to IVIN0	I	23 to 16	Video data	
IVCLK	I	25	Video clock (27 MHz)	↑
$\overline{\text{IVHSYNC}}$	I	12	Horizontal synchronization	L
$\overline{\text{IVVSYNC}}$	I	14	Vertical synchronization	L
IVFLD	I	11	Field index	

1.2 Video Output Interface

The video output is based on the ITU-R BT.656 format. It is able to output horizontal and vertical synchronization signals with SAV/EAV. These synchronization signals are chosen output by the firmware. These ports become GPO until the firmware initializes after hardware reset.

At the time of the odd field, $\overline{\text{OVVSYNC}}$ falls in the 4th clock after falling of $\overline{\text{OVHSYNC}}$.

At the time of the even field, $\overline{\text{OVVSYNC}}$ falls in to the H/2+4th clock the $\overline{\text{OVHSYNC}}$ falling.

Name	IO	Pin Number	Function	Active Polarity
OVOUT7, OVOUT6	O	208, 207	Video data	
OVOUT5 to OVOUT0/ FA19 to FA14	O	206 to 201	Video data	
OVCLK	O	199	Video clock (27 MHz)	↑
GPO5/**$\overline{\text{OVHSYNC}}$**	O	195	Horizontal synchronization	L
GPO6/**$\overline{\text{OVVSYNC}}$**	O	197	Vertical synchronization	L

1.3 Audio Input Interface

Name	IO	Pin Number	Function	Active Polarity
IALRCK	I	7	Left/Right clock	
IABCK	I	8	Bit clock	↑
IABD	I	9	Bit data	

1.4 Audio Input/output Interface

After hardware reset, it becomes input. OALRCK, OABCK and OABD connect with 3.3 V V_{DD} through the 10 kΩ pull up resistance. Firmware controls input/output of those pins.

Name	IO	Pin Number	Function	Active Polarity
OALRCK	IO	4	Left/Right clock	
OABCK	IO	5	Bit clock	↑
OABD	IO	6	Bit data	
AMCLK	I	2	Audio clock	↑

1.5 Stream Input Interface

Stream input corresponds to MPEG TS/PS stream. When slave mode (MPEG2-TS input with using valid signal), data input is possible to select 8 bits parallel data or serial data mode. When serial data mode, data input to IS0.

Active polarity of ISREQ is selected by the port setup register.

Active polarity of ISCLK/ISSTB, ISSYNC ISERR and ISVLD are selected by firmware. These are unsettled after the turning on.

Name	IO	Pin Number	Function	Active Polarity
ISREQ	O	55	Stream data request Only parallel interface, this pin is active. After reset, default is active low.	
ISCLK/**ISSTB**	I	53	Stream data strobe After reset, default is ISCLK.	
ISCLK/ISSTB	I	53	Stream data clock After reset, default is active high edge.	
ISSYNC	I	52	Stream data synchronization After reset, default is active high.	
ISVLD	I	54	Stream data valid After reset, default is active low.	
IS1/**ISERR**	I	43	Stream error After reset, default is active high.	
IS1/ISERR	I	43	Stream data input	
IS7 to IS2, IS0	I	51,49, 47 to 44, 42	Stream data input	

Remark In this table, means of reset are hardware reset by the $\overline{\text{RESET}}$ pin and ALL RESET of the reset register.

1.6 Stream Output Interface

This interface outputs MPEG TS/PS stream. When in master mode (MPEG2-TS output with using valid signal), data output is possible to select 8bits parallel data or serial data mode. In serial mode, data output from OS0.

Active polarity of OSVLD is selected by the port setup register.

Active polarity of OSCLK/OSSTB and OSSYNC are selected by firmware. These are unsettled after the turning on.

Name	IO	Pin Number	Function	Active Polarity
$\overline{\text{OSREQ}}$	I	70	Stream data request in slave mode	L
OSCLK/**OSSTB**	O	66	Stream data strobe After reset, default is active high edge.	
OSCLK/OSSTB	O	66	Stream data clock After reset, default is OSSTB.	
OSSYNC	O	67	Stream data synchronization After reset, default is active high.	
OSVLD/OSRDY	O	68	Stream data valid After reset, default is OSRDY.	
OSVLD/**OSRDY**	O	68	Stream data ready prepared After reset, default is active low.	
OS7 to OS0/ FA13 to FA6	O	65 to 63, 61, 59 to 56	Stream data output	

Remark In this table, means of reset are hardware reset by the $\overline{\text{RESET}}$ pin and ALL RESET of the reset register.

1.7 SDRAM Interface

Name	IO	Pin Number	Function	Active Polarity
MA13 to MA0	O	104, 103, 115, 102, 114, 113, 111, 109, 108, 106, 101, 99, 97, 95	Address of row/column	
MD31 to MD0	IO	93 to 89, 87, 85, 83, 72, 75 to 81, 149, 147 to 143, 141, 139, 127, 129, 131 to 134, 136, 138	Data (Built-in 50 kΩ pull up resistor)	
MCLK	O	118	Clock	↑
MCKE	O	116	Clock enable	H
$\overline{\text{MCS}}$	O	120	Chip selection	L
$\overline{\text{MRAS}}$	O	121	Row address strobe	L
$\overline{\text{MCAS}}$	O	123	Column address strobe	L
$\overline{\text{MWE}}$	O	125	Write enable	L
$\overline{\text{MDQM}}$	O	126	Data input/output mask enable	L

1.8 Host CPU Interface

It chooses a parallel bus connection and a serial bus connection by the setting of CMODE2.

Name	IO	Pin Number	Function	Active Polarity
CMODE2	I	158	Host CPU interface select L: Parallel, H: Serial	

1.8.1 Parallel bus interface

Name	IO	Pin Number	Function	Active Polarity
CA5 to CA0/ FA5 to FA0	I	187, 185 to 181	Address	
CD7 to CD0/ FD7 to FD0	IO	172, 170 to 166, 164, 162	Data	
$\overline{\text{CWE}}$/CSDI	I	157	Write enable	L
$\overline{\text{CRE}}$	I	160	Read enable	L
$\overline{\text{CCS}}$	I	159	Chip selection	L
CINT	O	153	Interrupt	H
CWAIT/$\overline{\text{FOE}}$	O	161	Wait	
CMODE0/CSCLK	I	155	Setting of polarity of CWAIT L: Low wait, H: High wait	
CMODE1/CSDO	I	156	Setting of operation of CWAIT (Built-in 50 kΩ pull up resistor) L: Wait operation.(after ready, pin continues ready) H: Ready operation.(after ready, pin turns to wait)	

1.8.2 Serial bus interface

When connecting a serial bus, it downloads instruction of internal CPU from instruction ROM.

(1) Serial bus interface

Name	IO	Pin Number	Function	Active Polarity
CMODE0/**CSCLK**	I	155	SPI serial interface clock Fix CSCLK to high level during $\overline{\text{CCS}}$ is disable (high level).	↑
$\overline{\text{CWE}}$/**CSDI**	I	157	SPI serial interface data input	
CMODE1/**CSDO**	O	156	SPI serial interface data output (Built-in 50 kΩ pull up resistor)	
$\overline{\text{CCS}}$	I	159	Chip selection	L
CINT	O	153	Interrupt	H

(2) Instruction ROM interface

Name	IO	Pin Number	Function	Active Polarity
CA5 to CA0/ **FA5 to FA0**	O	187, 185 to 181	Address	
OS7 to OS0/ **FA13 to FA6**	O	65 to 63, 61, 59 to 56	Address	
OVOUT5 to OVOUT0/ **FA19 to FA14**	O	206 to 201	Address	
CD7 to CD0/ **FD7 to FD0**	I	172, 170 to 166, 164, 162	Data	
CWAIT/**$\overline{\text{FOE}}$**	O	161	Output enable	L

1.9 Clock, Reset

Name	IO	Pin Number	Function	Active Polarity
SCLK	I	28	System clock	↑
STCLK	I	34	System time clock	↑
PSTOP	I	29	Internal PLL operation control L: Normal, H: Internal PLL stop	H
PWM	O	40	PWM output	
$\overline{\text{RESET}}$	I	151	Reset	L

1.10 N-Wire

IE Port for firmware of Internal CPU evaluation

★ When not connecting an in-circuit emulator, take countermeasures against noise by pulling up the NDI pin to avoid the pin becoming low level.

Name	IO	Pin Number	Function	Active Polarity
★ NMOD	I	178	Pin used when connecting IE Pull up when connecting IE	H
NCLK	I	174	Serial clock	↑
$\overline{\text{NRST}}$	I	176	N-wire reset	L
NDI	I	179	Data input	
NDO	O	180	Data output	

1.11 GPIO

GPIO becomes input after hardware reset by the $\overline{\text{RESET}}$ pin and ALL RESET by the reset register. GPIO connect with 3.3 V V_{DD} through the 10 kΩ pull up resistance.

Name	IO	Pin Number	Function	Active Polarity
GPIO0	IO	189	Firmware use pin	
GPIO1	IO	190	Firmware use pin	
GPIO2	IO	191	Firmware use pin	
GPIO3	IO	192	Firmware use pin	
GPIO4	IO	193	Firmware use pin	
GPO5/$\overline{\text{OVHSYNC}}$	O	195	Firmware use pin	
GPO6/$\overline{\text{OVVSYNC}}$	O	197	Firmware use pin	

1.12 Power Supply

Name	IO	Pin Number	Function	Active Polarity
V_{DD3}	-	39, 69, 86, 98, 110, 122, 135, 148, 171, 194	3.3 V power supply for interface	
V_{DD2}	-	1, 13, 24, 36, 48, 60, 71, 82, 94, 105, 117, 128, 140, 152, 163, 175, 186, 198	2.5 V power supply for the internal circuit	
GND	-	3, 10, 15, 26, 27, 35, 37, 38, 41, 50, 62, 73, 74, 84, 88, 96, 100, 107, 112, 119, 124, 130, 137, 142, 150, 154, 165, 173, 177, 188, 196, 200	GND	
PV_{DD2}	-	30, 32	2.5 V power supply for PLL	
PGND	-	31, 33	GND for PLL	

1.13 Recommended Connections of Unused Pins

Connect unused pins as follows.

Name	IO	Connection
IVIN7 to IVIN0	I	GND
IVCLK	I	GND
$\overline{\text{IVHSYNC}}$	I	GND
$\overline{\text{IVVSYNC}}$	I	GND
IVFLD	I	GND
OVOUT7, OVOUT6	O	Open
OVOUT5 to OVOUT0/FA19 to FA14	O	Open
OVCLK	O	Open
IALRCK	I	GND
IABCK	I	GND
IABD	I	GND
OALRCK	IO	Pull up with 10 kΩ resistor
OABCK	IO	Pull up with 10 kΩ resistor
OABD	IO	Pull up with 10 kΩ resistor
AMCLK	I	GND
ISREQ	O	Open
ISCLK/ISSTB	I	GND
ISSYNC	I	GND
ISVLD	I	GND
IS7 to IS0	I	GND
$\overline{\text{OSREQ}}$	I	GND
OSSYNC	O	Open
CA5 to CA0/FA5 to FA0	IO	Open
CD7 to CD0/FD7 to FD0	IO	Pull up with 10 kΩ resistor
$\overline{\text{CRE}}$	I	GND
CINT	O	Open
CWAIT/$\overline{\text{FOE}}$	O	Open
PWM	O	Open
NMOD	I	Pull up with 4.7 kΩ resistor
NCLK	I	Pull up with 4.7 kΩ resistor
$\overline{\text{NRST}}$	I	Pull down with 50 kΩ resistor
NDI	I	Pull up with 4.7 kΩ resistor
NDO	O	Pull up with 4.7 kΩ resistor
GPIO4 to GPIO0	IO	Pull up with 10 kΩ resistor
GPO5/$\overline{\text{OVHSYNC}}$	O	Open
GPO6/$\overline{\text{OVVSYNC}}$	O	Open

2. FEATURE OVERVIEW

★ The functions and I/O interfaces are set using firmware.
Supported functions differ depending on firmware.

2.1 Video

This LSI can do flexible encoding and decoding by using the firmware control of internal CPU and an exclusive use circuit. NTSC/PAL video format, which is possible of the encoding and the decoding, is as in **Table 2-1**. NTSC/PAL video format of the transcoding is under 720 dots by 480/576 line/frame.

Table 2-1. Video Format

MPEG2	MPEG1	Video format
Yes	No	720 dots by 480/576 line/frame
Yes	No	704 dots by 480/576 line/frame
Yes	No	544 dots by 480/576 line/frame
Yes	No	480 dots by 480/576 line/frame
Yes	No	352 dots by 480/576 line/frame
Yes	Yes	352 dots by 240/288 line/frame

2.1.1 Encoding

It encodes the video that was converted from the 4:2:2 format into the 4:2:0 format in the video input/output unit with MPEG2 standard MP@ML, SP@ML and the MPEG1 standard. It is encoding in variable bit rate (single path VBR encoding) or constant bit rate (CBR). The pre analysis supports high quality picture encoding. Encode supports frame structure.

- Using the following, only 64 Mbits SDRAM is needed.
  Encoding with locally decoding and/or time base corrector (TBC)
  PAL encoding
- DVD encoding needs equal to 128 Mbits SDRAM area.
- The motion estimation size
  P picture: ±128 dots (H) by ±64 lines (V)
  B picture: ±96 dots (H) by ±48 lines (V), ±64 dots (H) by ±32 lines (V)
- I/P picture period in MP@ML : $M \leq 3$
- Dual prime estimate, only at the time of M = 1.

2.1.2 Transcoding

It transcodes the stream of MPEG2 standard MP@ML based. It is possible for the bit rate conversion.

2.1.3 Input/output processing

(1) Video input

The video input format is ITU-R BT.656 (8-bit Y/Cb/Cr the 4:2:2 format) and 8-bit Y/Cb/Cr which deals with the 4:2:0 format. The horizontal synchronization signal, the vertical synchronization signal and the field index can be used without using SAV and EAV. In this case, IVFLD can be used by taking with $\overline{\text{IVVSYNC}}$ or it judges a field judgment in the polarity of $\overline{\text{IVHSYNC}}$ behind the falling edge two clock of $\overline{\text{IVVSYNC}}$. It judges that an odd field is 'H' and an even field is 'L'. $\overline{\text{IVVSYNC}}$ and $\overline{\text{IVHSYNC}}$ need the high / low period more than 3 IVCLK. The video-input unit watches over the synchronization signals and detects synchronous error.

(2) Picture size conversion filter

For adapting to the bit rate of the stream, the picture size of the encoding can be changed. In addition, picture size changed with the external filter to the 4:2:0 format can be inputted directly, too.

Table 2-2. Input Video Data Arrangement

Format	Line	Data arrangement
4:2:2	Odd/even lines	Cb0, Y0, Cr0, Y1, Cb1, Y2, Cr1, Y3, Cb2, Y4, Cr2, Y5, …
4:2:0	Odd lines	Cb0, Y0, Cr0, Y1, Cb1, Y2, Cr1, Y3, Cb2, Y4, Cr2, Y5, …
	Even lines	(-), Y0, (-), Y1, (-), Y2, (-), Y3, (-), Y4, (-), Y5, …

(3) Time base corrector (TBC)

It has a frame-type TBC. It is possible to make stable encoding of the channel changing and the nonstandard video signal such as VTR. When using TBC, it needs over 64 Mbits SDRAM. The following video signals can be corrected.

Table 2-3. Correctable Video Signals

	Horizontal Sync	Vertical Sync
NTSC	1626 to 1806 IVCLK/H	246 to 278 H/V
PAL	1628 to 1828 IVCLK/H	294 to 330 H/V

Remark IVCLK: 27 MHz

(4) Noise reduction

Respectively the noise reduction of the luminance signal and the color signal can be set three levels

(5) Slicer

Slicer decodes the luminance signal to the vertical blanking data. It detects VBID, Closed Caption, and Wide Screen Signal. The host CPU can read, and stop encoding and re-write the copy control information in VBID and the Wide Screen Signal, on the host CPU interface.

Table 2-4. Slicer

TV method	VBI data	Detection line
NTSC	VBID	20, 283
	Closed caption	21, 284
PAL	Wide screen signal	23 (336)

(6) Video output

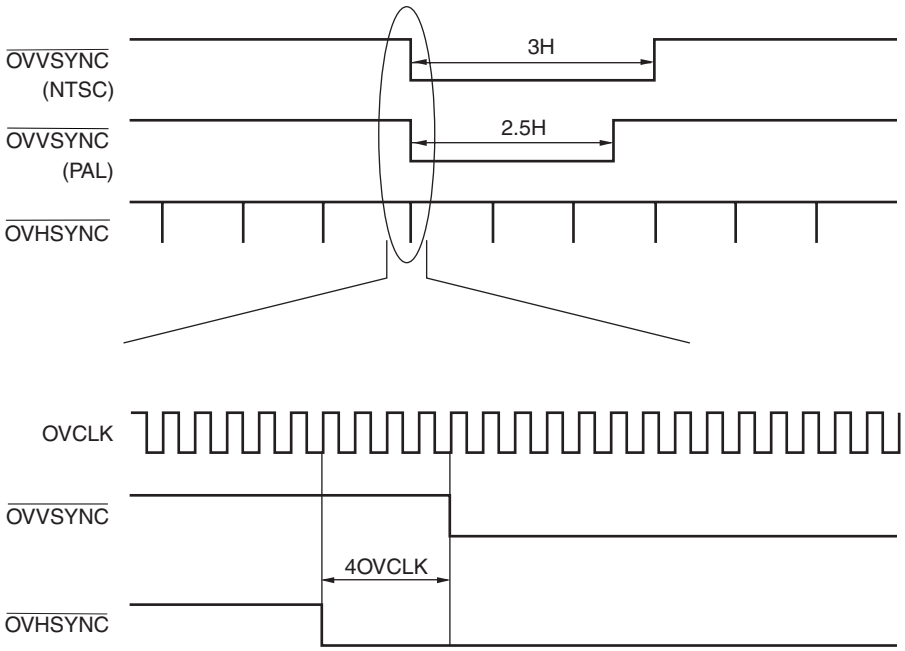
It converts an input video or a local-decoded video into picture size of 720 dots by 480/576 line and outputs with the ITU-R BT.656 format.

Horizontal and vertical synchronization signals are switched from GPO.

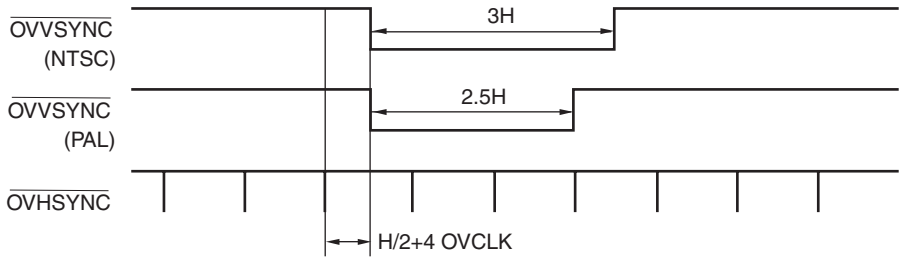
Field detection is easy due to vertical synchronization signal delays 4VCLK since horizontal synchronization signal.

Figure 2-1. Video Output

(a) Odd Field



(b) Even Field



★ 2.2 Audio

This LSI encodes the MPEG audio encoding and transcode with the internal DSP.

2.2.1 Encoding

It encodes MPEG1 audio layer 2 or Dolby Digital Consumer Encoder (only the μPD61052). In addition, it is possible to bypass internal audio encode DSP, when the audio elementary stream is encoded by an external audio encoder are inputted.

2.2.2 Transcoding (DEMUX, MUX)

It is possible to multiplex two de-multiplexed audio streams. It analyzes MPEG1 audio stream, and extracts the information to multiplex and notify to the host CPU.

2.2.3 Input/output processing

Two PCM audio signals can be inputted to the audio input interface and the audio input-output interface. When inputting two audio signals, an audio signal is encoded, and another one bypasses the audio encoding DSP, and transfers to the multiplexer. When inputting an audio elementary stream that has been encoded by the external audio encoder and PCM audio, it can multiplex two audio elementary streams.

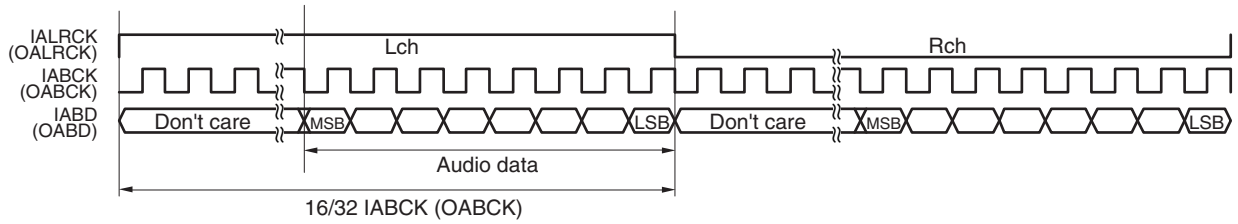
The PCM audio or the audio elementary stream can be outputted from the audio input-output interface. The audio clock (AMCLK) types the clock by which a phase was locked up STC clock (STCLK).

Table 2-4. Audio Input/output

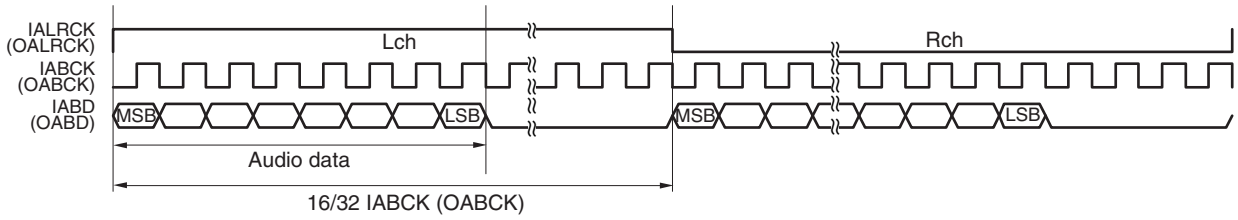
Item	Input/output format
Data length	16 bits, 20 bits, 24 bits
Sampling frequency	32 kHz, 44.1 kHz, 48 kHz
Justification of transfer	MSB first I^2S Compatible/Left justified/Right justified
Format	PCM Audio, IEC60958 based

Figure 2-2. Audio Input

(a) MSB First Right Justified Mode



(b) MSB First Left Justified Mode



(c) I^2S Mode

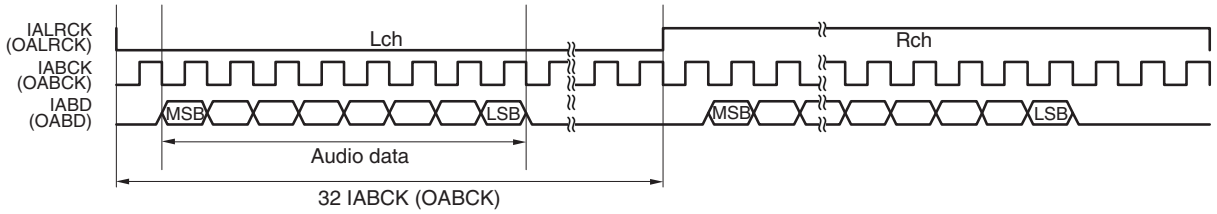
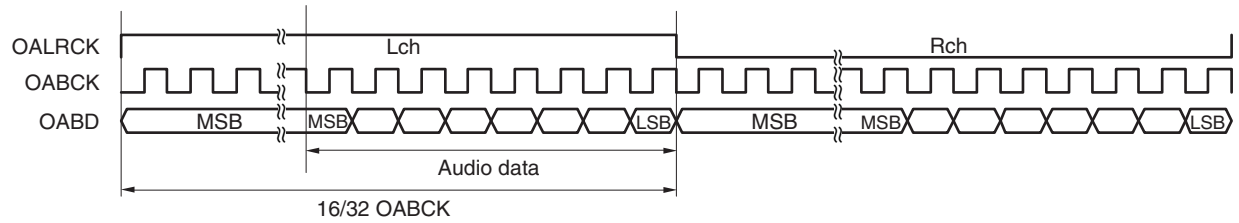
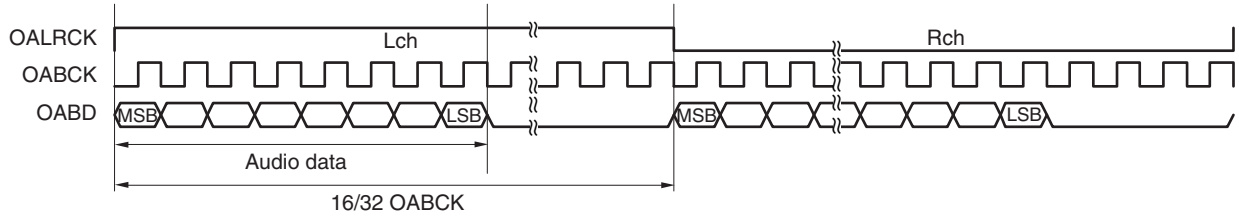


Figure 2-3. Audio Output

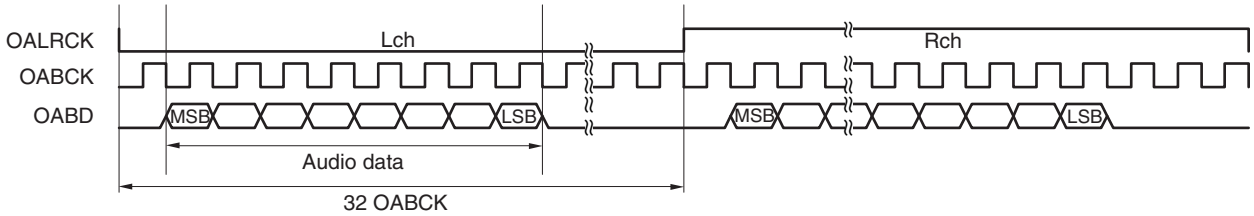
(a) MSB First Right Justified Mode



(b) MSB First Left Justified Mode



(c) I^2S Mode



2.3 MPEG System Processing

This LSI multiplexes and/or de-multiplexes Audio and video streams based on MPEG2-TS/PS and MPEG1. By combining the multiplexer and de-multiplexer, it does the transcode which is accompanied by MPEG2-TS⇔MPEG2 PS conversion.

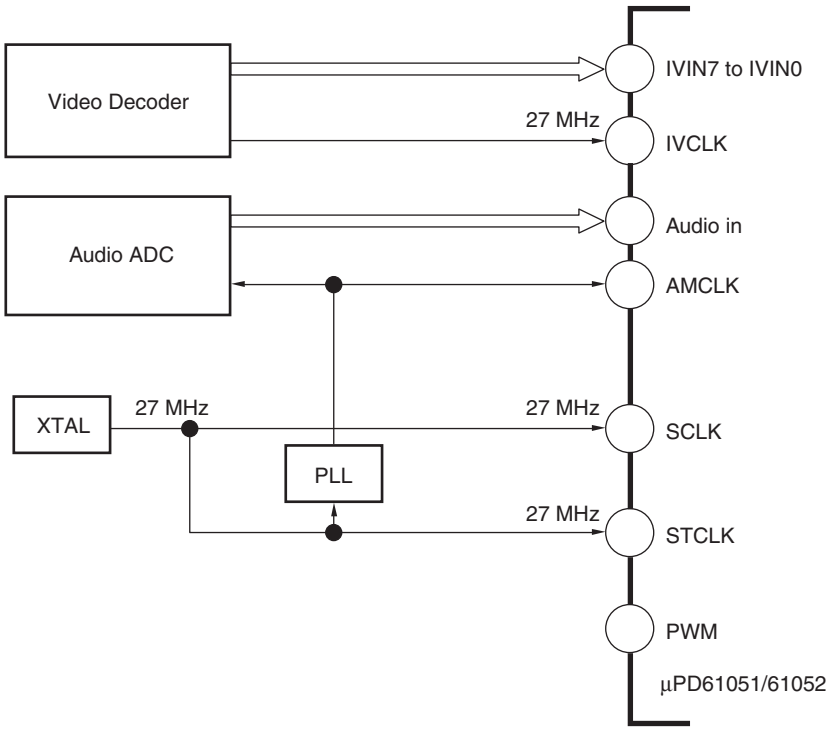
2.3.1 System time clock

(1) Encoding system

When the encoding system operates, it uses the clock input to STCLK that is generated with the 27 MHz oscillator.

Audio master clock is made with 27 MHz of STCLK, and then Audio synchronizes to STC.

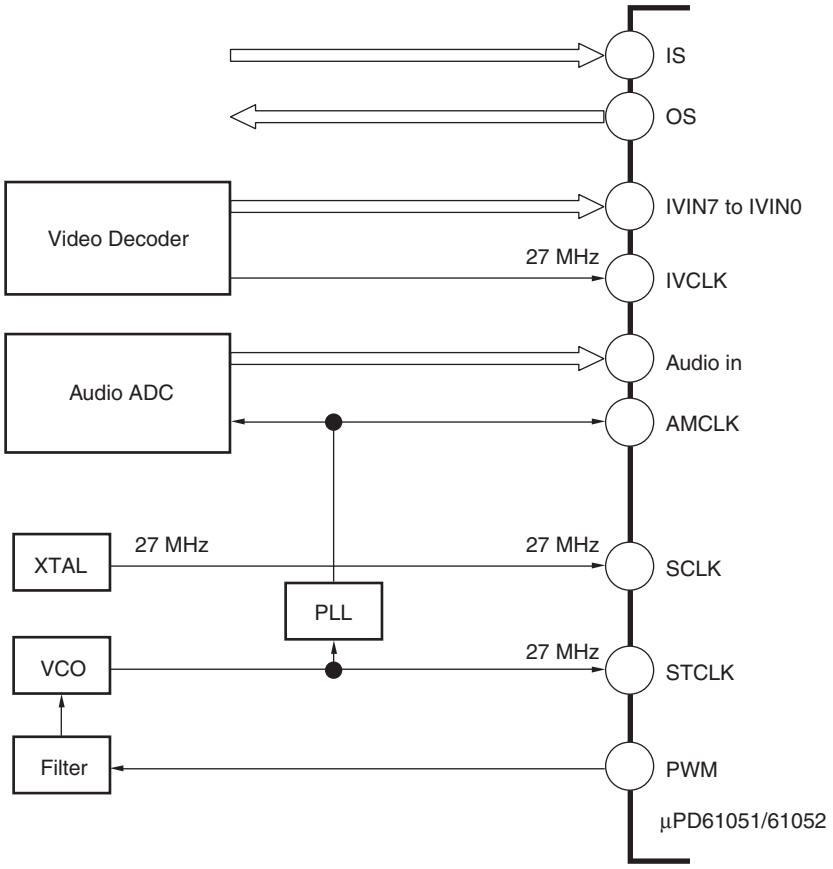
Figure 2-4. System Time Clock Input (Encoding System)



(2) Encoding and Transcoding system

It can output the signal, which generates the pulse wide modulation (PWM) with comparing PCR/SCR of the stream and system time clock value, for making the reference clock of the system.

Figure 2-5. System Time Clock Input (Encoding and Transcoding System)



2.3.2 Multiplex

It stamps SCR, PCR, DTS and PTS after multiplexing streams that are from the video encoder and the audio encoder based on MPEG2-TS/PS.

Partial TS can be made by forming SIT packet from PSI and SI data of base on DVB.

It is possible to multiplex the packet that inputted from the host CPU interface.

2.3.3 De-multiplex

(1) MPEG2-TS

Using the PID filter corresponding to 16 PIDs, It separates MPEG2-TS to one video stream, two audio streams, and two user data streams. Internal CPU extracts section data in PSI and SI of base on DVB.

(2) MPEG2-PS

With the stream ID filter, it separates MPEG2-PS to one video stream, one audio stream, and two user data streams.

(3) VBI data

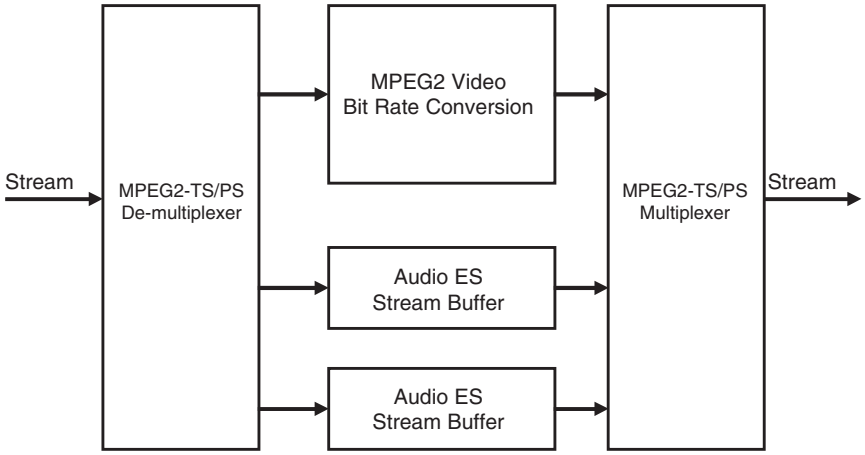
The user data stream, the wide screen signal, the closed caption, VBID and format of the video and the audio can be read from the host CPU interface.

2.3.4 Transcode

The transcode is a combined multiplexer and de-multiplexer. MPEG2-TS/PS separates into a video stream, two audio streams, and two user data streams. The video stream and the audio stream are multiplexed to MPEG2-TS/PS after transcode on the elementary. PCR, SCR, PTS and DTS are corrected when multiplexing.

In the transcode of MPEG2-TS, it can generate partial TS using the data detected by the PID filter and the section filter.

Figure 2-6. Transcode



The change of the MPEG system layer is shown below.

MPEG2-TS ⇒ MPEG2-TS
MPEG2-TS ⇒ MPEG2-PS
MPEG2-PS ⇒ MPEG2-TS
MPEG2-PS ⇒ MPEG2-PS
MPEG1 ⇒ MPEG1

2.4 Stream Interface

When it inputs MPEG2-TS, it is able to connect parallel data or serial data with the μPD61051/61052. When it inputs MPEG2-PS, it should connect parallel data with the μPD61051/61052.

2.4.1 Parallel steam data interface

This LSI connects to external device by the master mode or the slave mode. When parallel interface, the maximum stream input rate is 100 Mbps, the maximum stream output rate is 30 Mbps. The stream of MPEG encoding and transcode is limited to 15 Mbps on MPEG MP@ML.

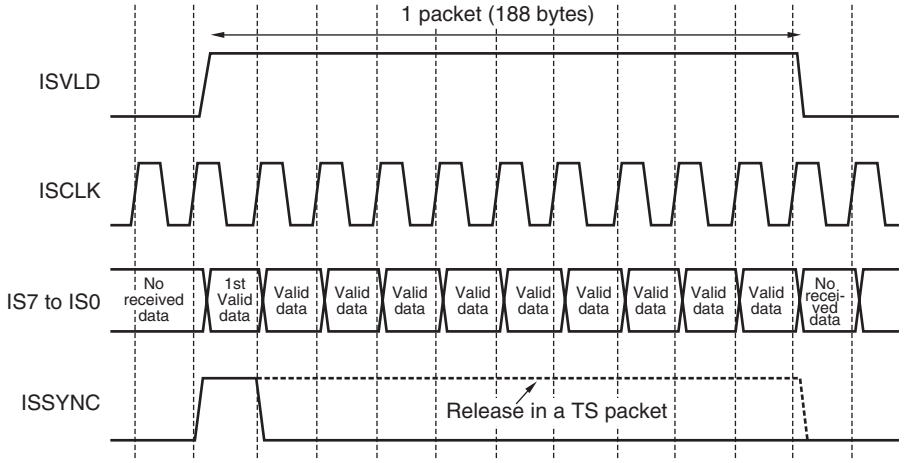
(1) Stream Input

It is possible to receive 4 bytes data after invalid of ISREQ of the stream input.

Remark ISSTB and ISCLK are identical pins.

Figure 2-7. Parallel Stream Receiving Mode (1/2)

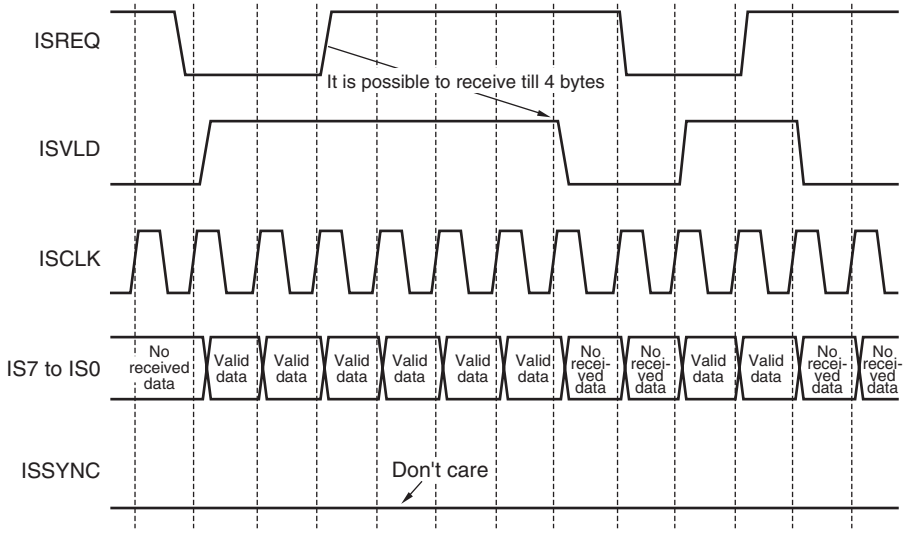
(a) Example for Receiving of MPEG2-TS



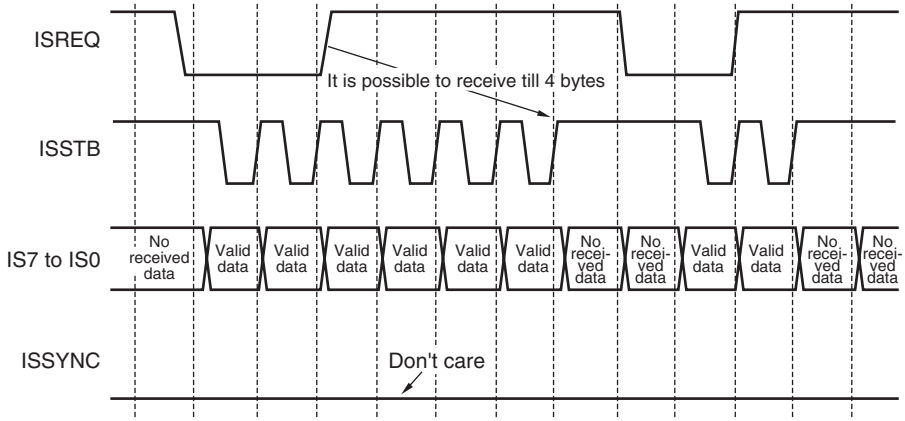
ISCLK shall be under 13.5 MHz.

Figure 2-7. Parallel Stream Receiving Mode (2/2)

(b) Example of Receiving MPEG2-PS, ES with Valid and Clock



(c) Example of Receiving MPEG2-PS, MPEG2-ES with a Strobe



★ **(2) Stream output**

There are three modes: valid operation master mode, strobe operation burst transfer mode, and strobe operation byte transfer mode.

The appropriate transfer mode for the system can be selected by setting the three stream output mode and transfer rate.

Remark OSSTB and OSRDY are the same pins as OSCLK and OSVLD, respectively. Operation can be selected using combinations of OSSTB and OSRDY or OSCLK and OSVLD.

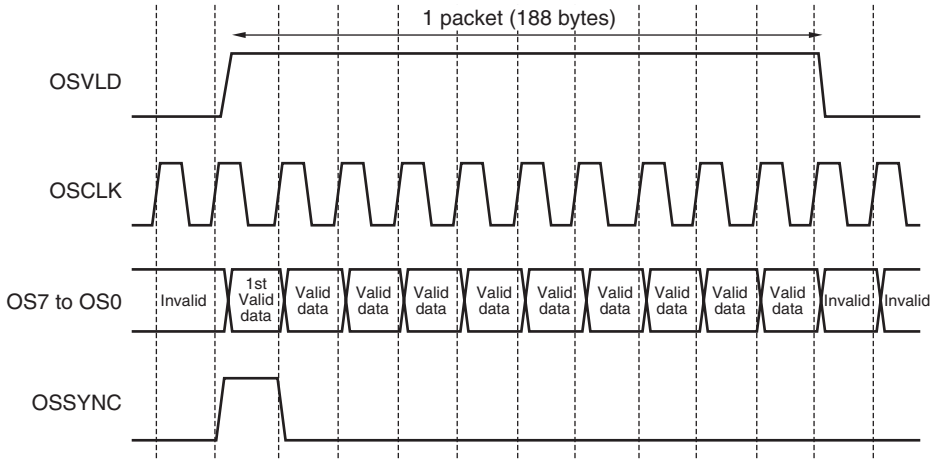
(a) Master Mode Valid

This is the MPEG2-TS dedicated output mode.

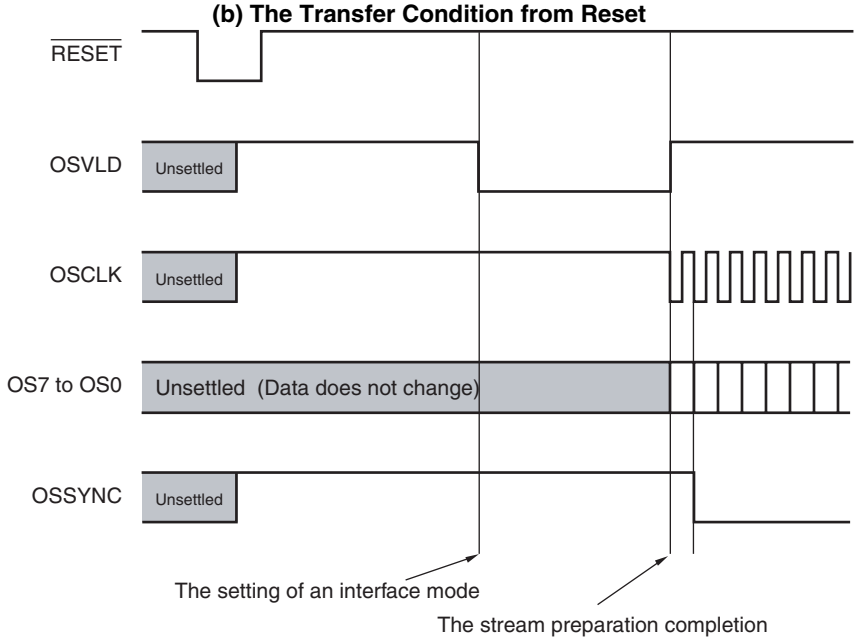
The period of OSCLK can be selected from n times 37 ns (1/27 MHz) (3 ≤ n ≤ 255, n is an integer). If using local decode or input video display, the period is 4 ≤ n ≤ 255 (n is an integer).

Figure 2-8. Parallel Stream Transmission Mode ; Transmission of MPEG2-TS (Packet Length is 188 Bytes)

(a) Master Mode, Valid



(b) The Transfer Condition from Reset



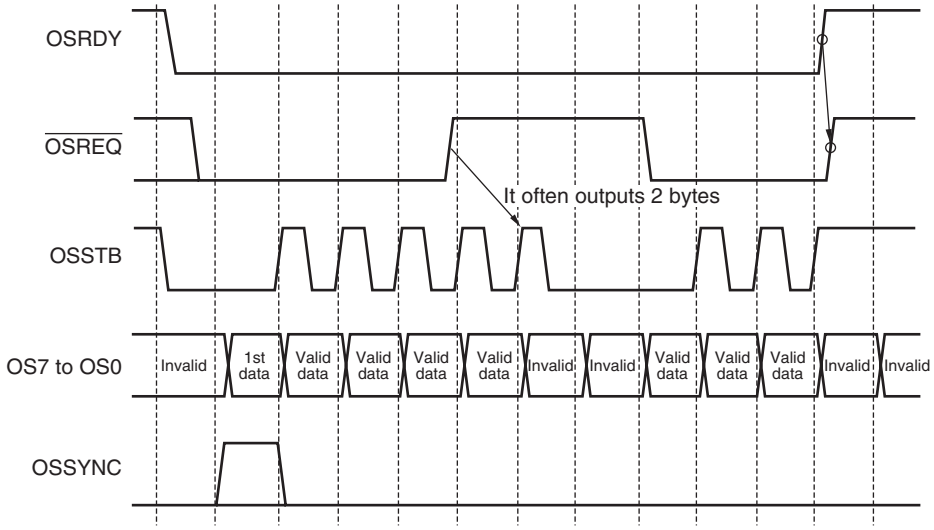
(b) Burst Transfer Mode, Strobe

In burst transfer mode, the period of OSSTB is 4 times 37 ns (1/27 MHz).

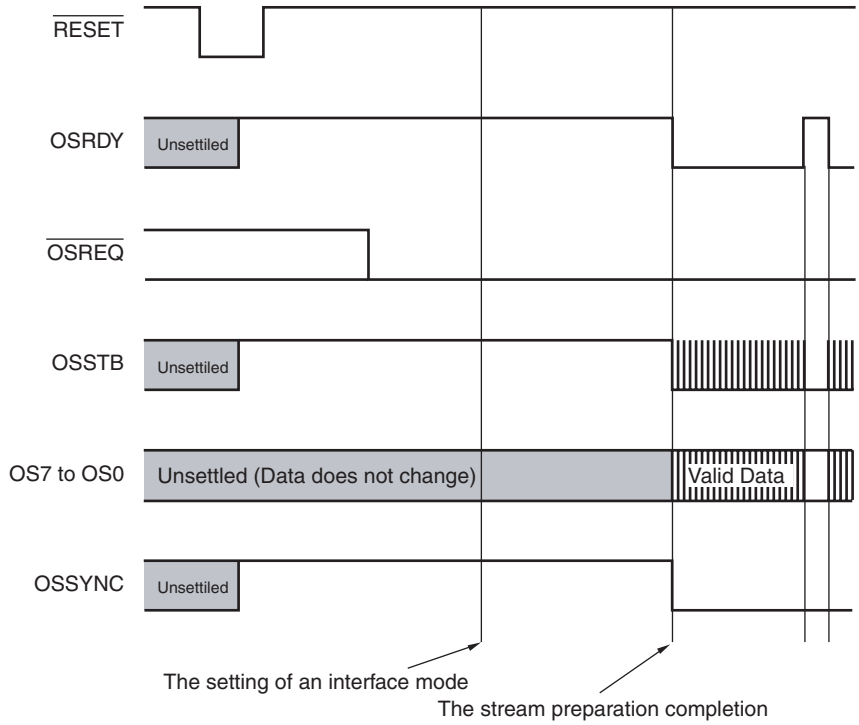
In the stream output after OSREQ cancellation, data up to 2 bytes may be output.

Figure. 2-9 Parallel Stream Transmission Mode; Transmission of MPEG2-PS, MPEG2-ES

(a) Example for Transmission of MPEG2-PS, MPEG2-ES



(b) The Transfer Condition from Reset

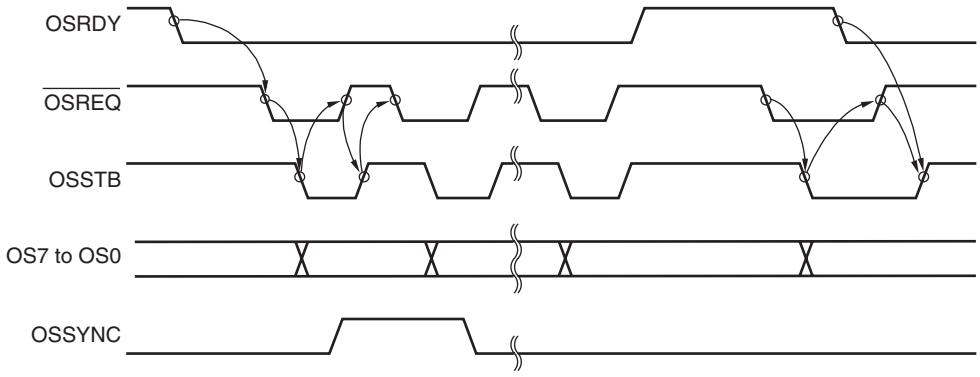


(c) Bytes Transfer Mode, Strobe

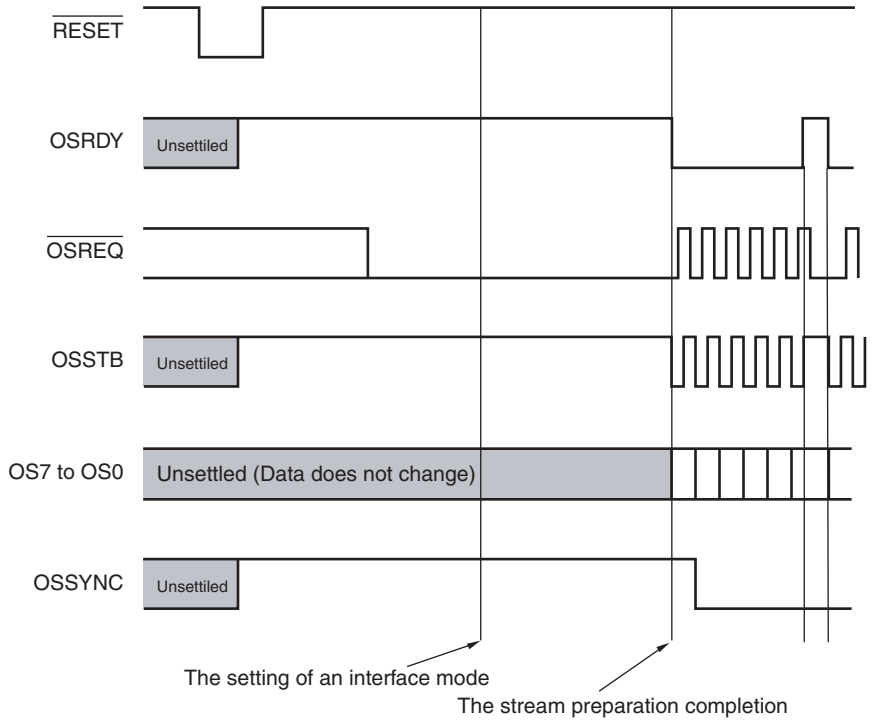
In byte transfer mode, the transfer rate is determined by the handshake of $\overline{\text{OSREQ}}$ and OSSTB.

Figure 2-10. Parallel Stream Transmission Mode (Transmission of MPEG2-PS, MPEG2-ES)

(a) Example for Transmission of Strobe Mode One Byte Transfer



(b) The Transfer Condition from Reset



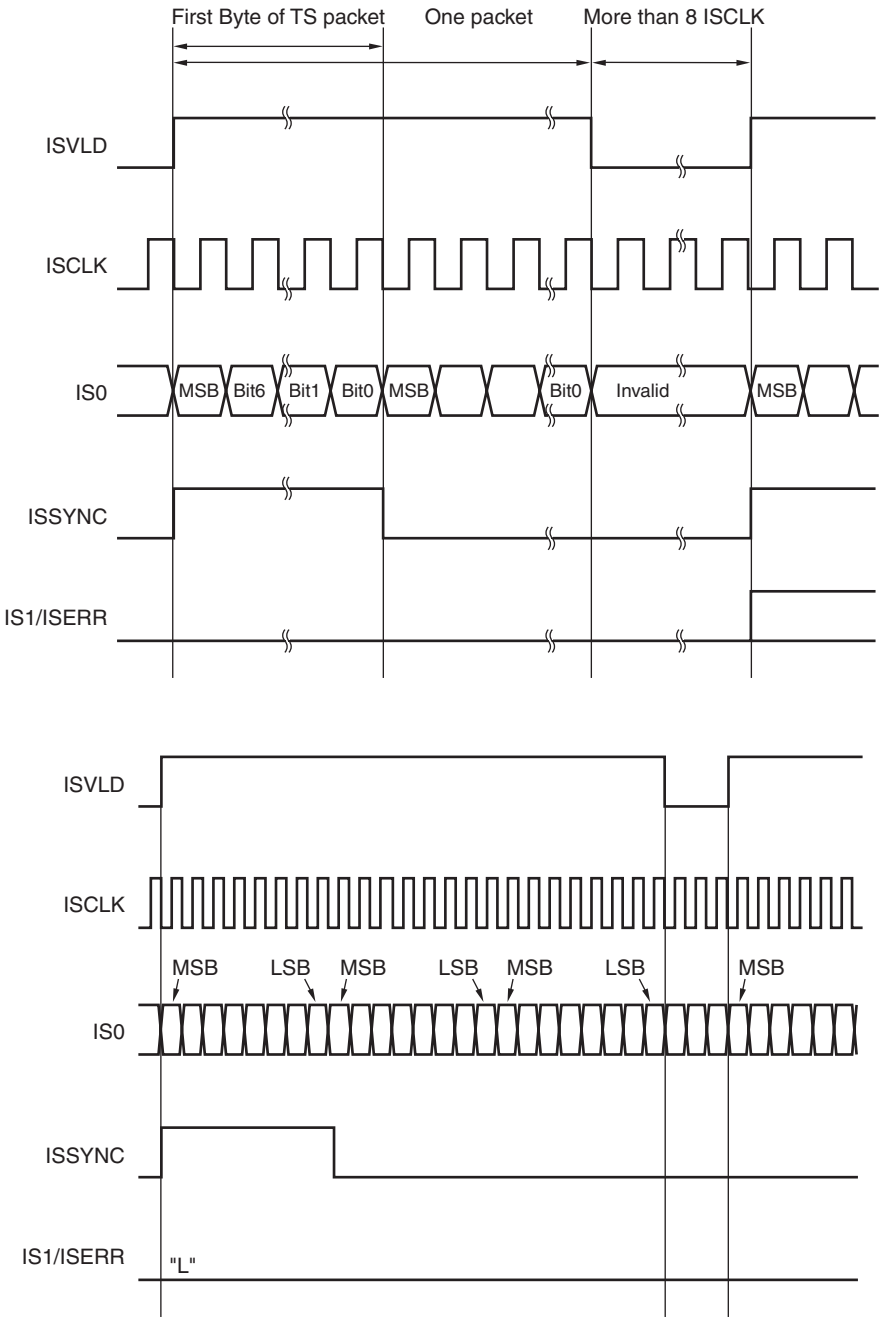
2.4.2 Serial stream data interface

This LSI is able to input a serial stream. Bit rate of serial input is limited less than parallel interface. Serial Stream Interface can transfer only MPEG2-TS stream. Maximum bit rate of stream input is less then 64 Mbps. Bit rate of stream out is 27 Mbps. Additionally, encoding and transcoding bit rate is limited to 15 Mbps on MPEG2 MP@ML.

(1) Stream input

ISCLK is input by less than 64 MHz clock. Data is MSB first. ISSYNC should active while first byte each packet. If packet error occurred, ISERR should active from ISSYNC of the packet. ISVLD should valid while each byte. ISVLD shall invalid while 8 bits between each packets.

Figure 2-11. Serial Stream Input

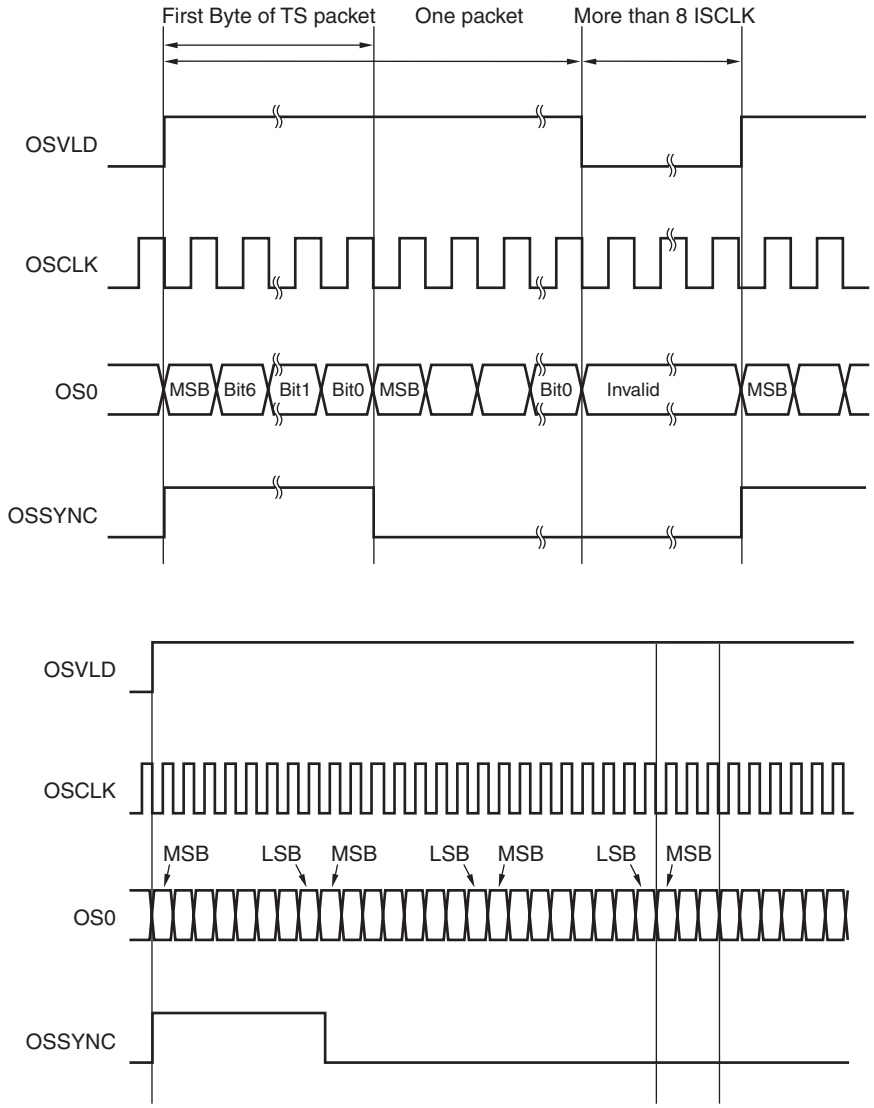


Remark Example for ISVLD, ISSYNC, ISERR active high, ISCLK active high edge

(2) Stream Output

OSCLK is fixed 27 MHz OSSYNC active at first byte in each packet. OSVLD is active of 1 packet continuously. Data is the MSB first outputs. ISSYNC becomes active among 1 byte at the head of the packet.

Figure 2-12. Serial Stream Output

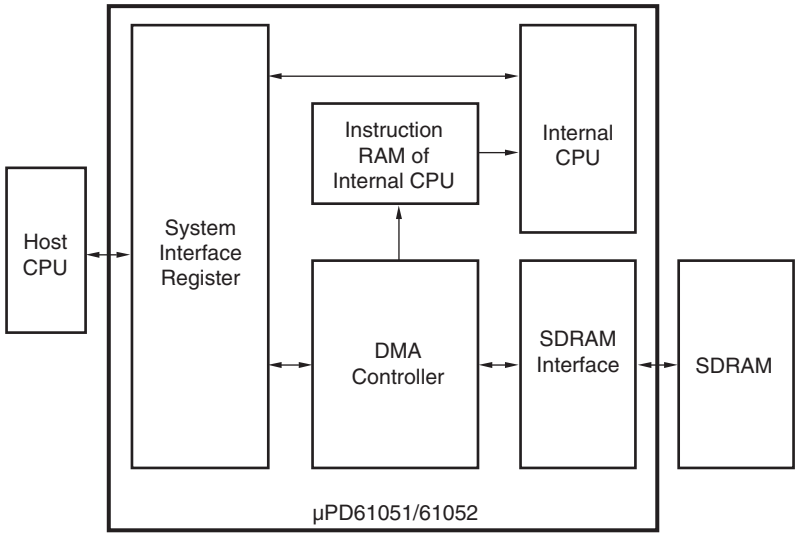


Remark Example for OSVLD, OSSYNC, OSERR active high

2.5 Host CPU Interface

The connection of the host CPU can select the eight bits parallel data interface and serial interface (SPI). Internal CPU sends and receives command status through the System Interface Register, which is in the host CPU interface unit. In addition, to control an internal DMA controller through the system interface register, it loads an instruction for internal CPU to the instruction RAM and the transfer of the large-volume data can be sent to the data area on SDRAM.

Figure 2-13. Host CPU Interface



The following describes loading of internal CPU instruction.

(1) Parallel interface

When parallel interface is selected, host interface has 6-bit address, 8-bit data bus and control ports.
CWAIT is selected with CMODE1 to wait on ready signal mode, CMODE1 selects active polarity of CWAIT.

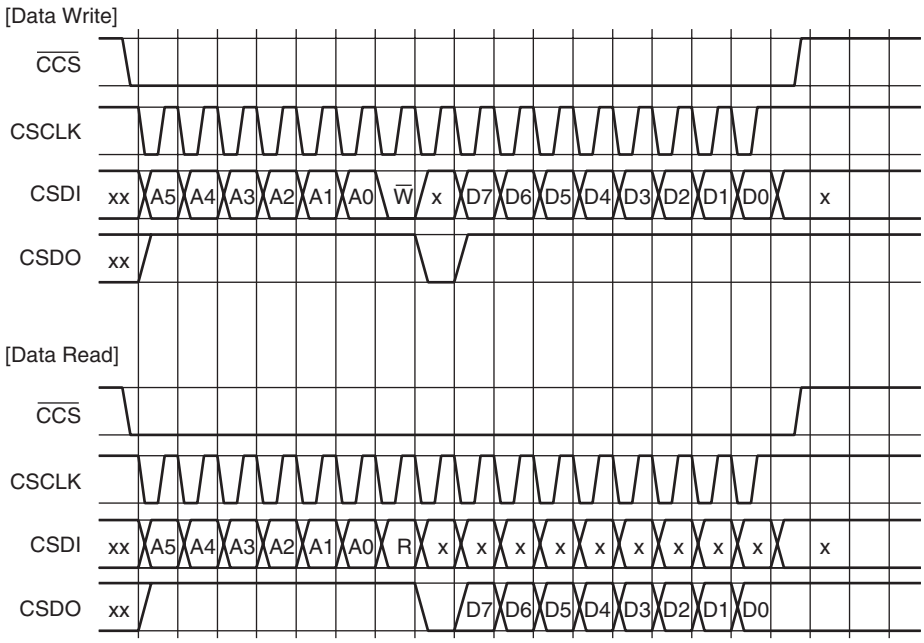
(2) Serial interface

The μPD61051/61052 communicates with the host CPU using the SPI (serial peripheral interface) serial bus. The host CPU becomes a bus master.
The low edge of the chip selection is communication beginning. Its high edge is communication ending.
An address and the reading / writing mode are shown at the first byte after the chip selection becomes low.
It is the MSB first of six bits of addresses, eight bits of data. Fix CSCLK to high level during $\overline{CCS}$ is disabled (high level).

The μPD61051/61052 becomes a master and downloads the instruction of the internal CPU from external ROM.

CSCLK:	The serial clock
CSDI:	The data input
CSDO:	The data output
$\overline{CCS}$:	The chip selection

Figure 2-14. Serial Interface



2.6 SDRAM Interface

External memory is SDRAM. It is possible to use the following.

Table 2-6. Use Memory

Memory	Data bus width	Quantity	Use memory capacity
16 Mbit SDRAM	16 bits	2	32 Mbits
64 Mbit SDRAM	32 bits	1	64 Mbits
64 Mbit SDRAM	16 bits	2	128 Mbits
128 Mbit SDRAM	16 bits	2	128 Mbits
128 Mbit SDRAM	32 bits	1	128 Mbits

The μPD61051/61052 preserves the part of the parameter that is necessary to generate the stream, entry video image, a video stream, an audio stream, a stream header, user data, and the instruction of the firmware at this memory.

This system uses only CAS latency = 3, burst length = 4.

When encode using time base corrector and/or displays local decoding picture, it needs equal to or more than 64 Mbits SDRAM.

When PAL encoding, it needs equal to or more than 64 Mbit SDRAM.

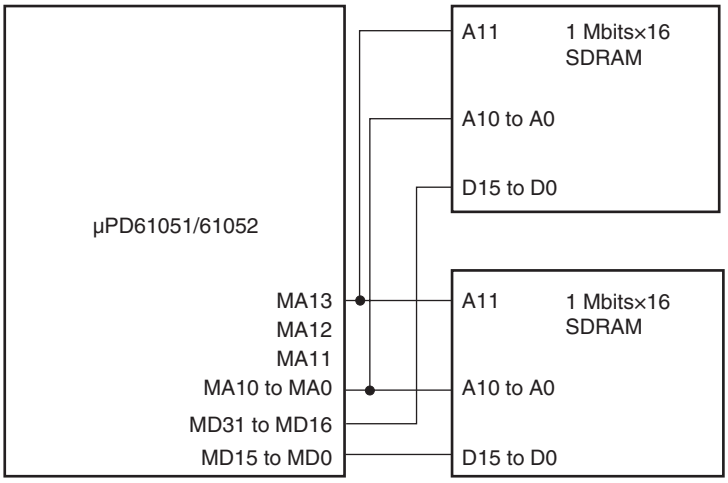
When transcoding, it needs equal to or more than 64 Mbit SDRAM.

2.7 Memory Connection Diagram

Each memory connection is as follows.

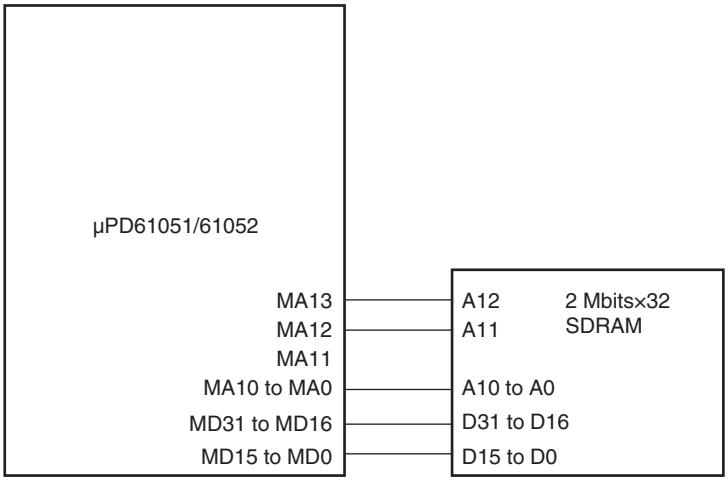
Figure 2-15. Memory Connection Diagram (1/2)

(a) 16 Mbit SDRAM by 2



Bank A: SDRAM address = 0x xxxx xxxx xxxxB
Bank B: SDRAM address = 1x xxxx xxxx xxxxB

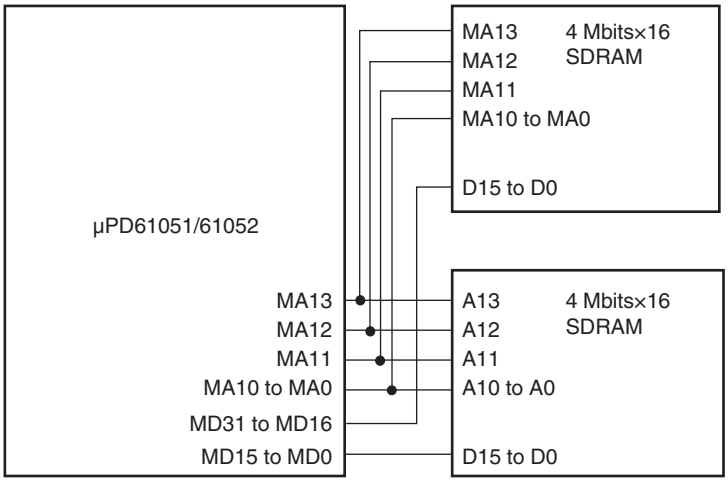
(b) 64 Mbit SDRAM by 1



Bank A: SDRAM address = 00 xxxx xxxx xxxxB
Bank B: SDRAM address = 10 xxxx xxxx xxxxB
Bank C: SDRAM address = 01 xxxx xxxx xxxxB
Bank D: SDRAM address = 11 xxxx xxxx xxxxB

Figure 2-15. Memory Connection Diagram (2/2)

(c) 64 Mbit SDRAM by 2 or 128 Mbit SDRAM by 2



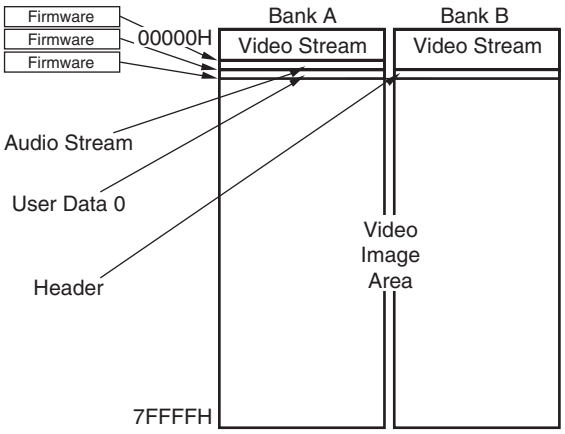
Bank A: SDRAM address = 00 xxxx xxxx xxxxB
Bank B: SDRAM address = 10 xxxx xxxx xxxxB
Bank C: SDRAM address = 01 xxxx xxxx xxxxB
Bank D: SDRAM address = 11 xxxx xxxx xxxxB

2.8 Memory Map

Firmware sets memory map such as video image area and usable work area. Firmware cabinet (temporal buffered area) is the area which firmware does not use. Video Image area size is changed NTSC or PAL. Each area are changed by the firmware.

Figure 2-16. Memory Map (1/2)

(a) 16 Mbit SDRAM by 2



(b) Example for 64 Mbit SDRAM by 1

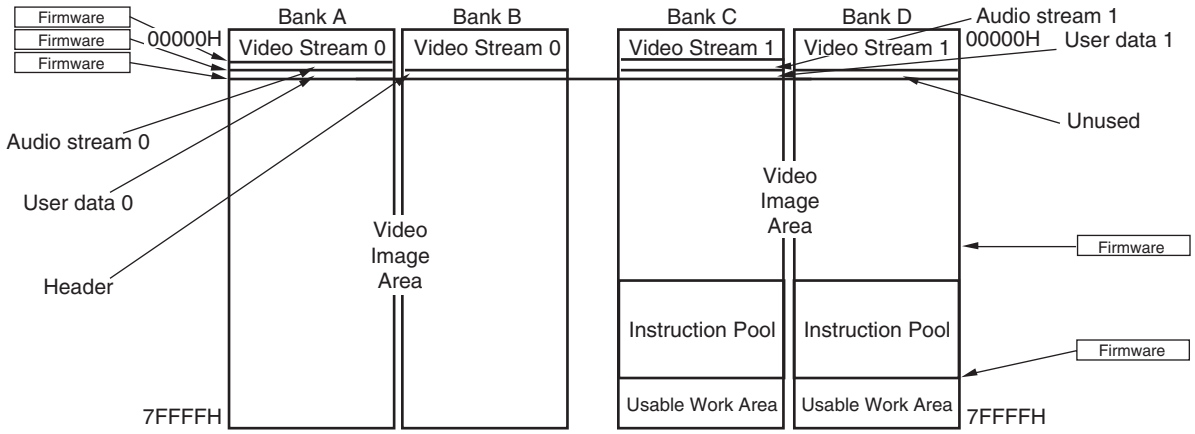
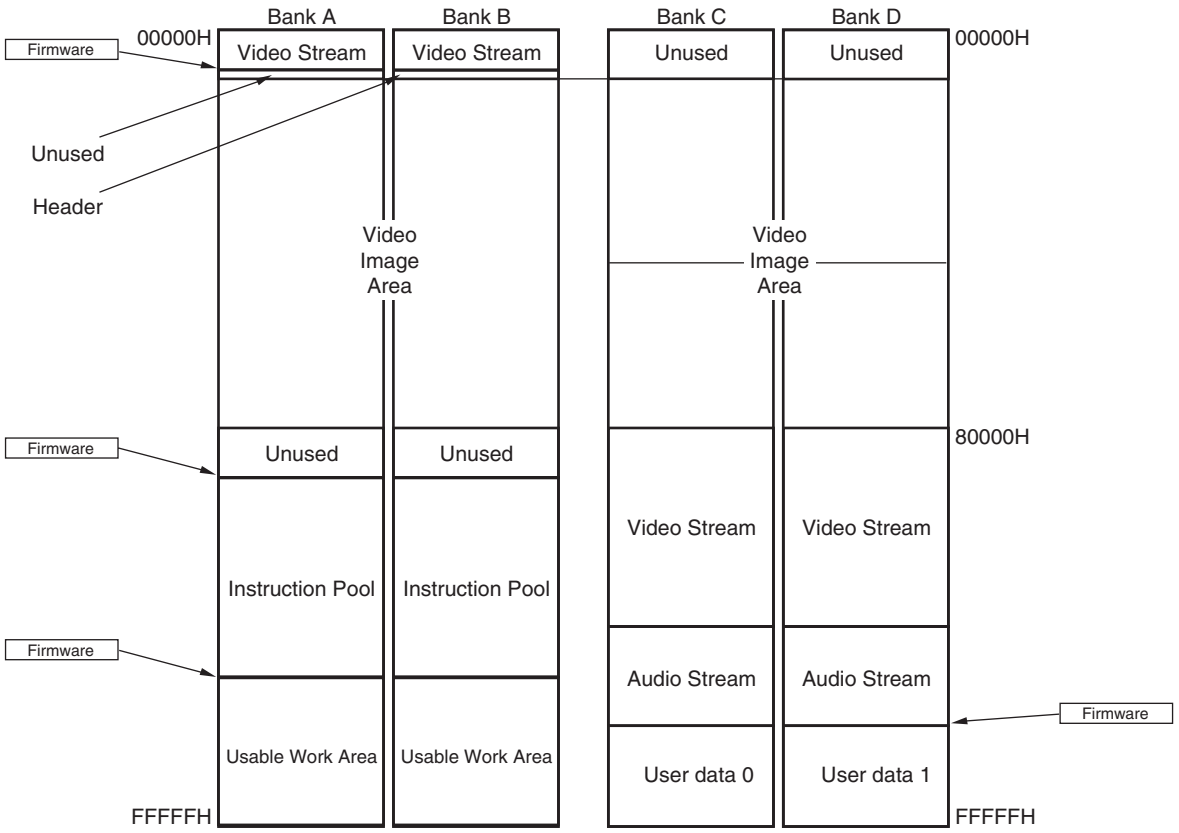


Figure 2-16. Memory Map (2/2)

(c) Example for 64 Mbit SDRAM by 2 or 128 Mbit SDRAM by 2



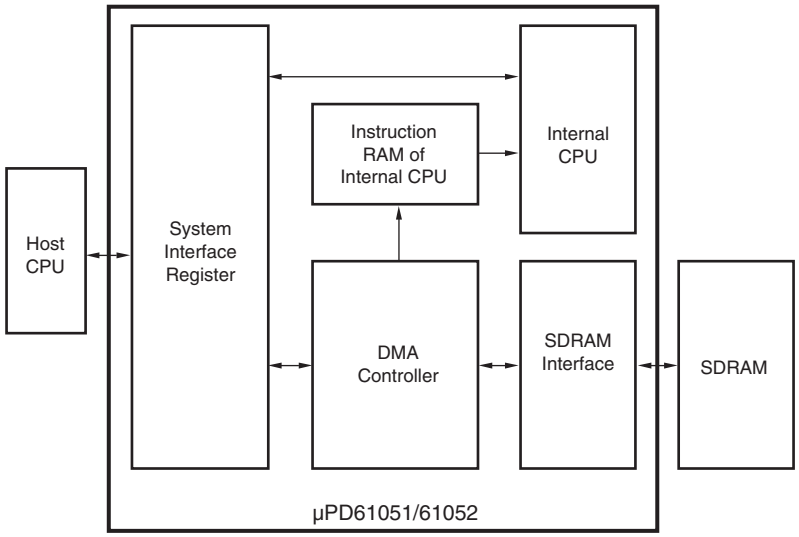
3. SYSTEM INTERFACE REGISTER

This LSI corresponds to the various operation modes in exchange instruction of internal CPU from SDRAM to instruction RAM (iRAM).

This has 64 byte Registers. They are defined to common registers, interrupt registers and interrupt mask registers. When there is access in the same address from both of the internal CPU and the host CPU, the later data is left at the register.

Also, when the writing occurs to the same address at the same time about the common register, the data of the host CPU is left at the register

Figure 3-1. System Interface Register



3.1 Register Mapping (General Mapping)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
00H to 1FH	Defined by firmware								R/W	
20H	SI	SSD	SDI	MSD	MI		SDW	SDR	R/W	Download mode
21H					SA19 to SA16				R/W	Source address
22H	SA15 to SA8								R/W	Source address
23H	SA7 to SA0								R/W	Source address
24H								DA16	R/W	Destination address
25H	DA15 to DA8								R/W	Destination address
26H	DA7 to DA0								R/W	Destination address
27H						TC18 to TC16			R/W	Transfer data count
28H	TC15 to TC8								R/W	Transfer data count
29H	TC7 to TC0								R/W	Transfer data count
2AH								iCPU-INT	R/W	Int. to internal CPU
2BH						DMA-ERR-M	DMA-RDY-M	DMA-DONE-M	R/W	Interrupt mask0
2CH	Defined by firmware								R/W	Interrupt mask1
2DH	Defined by firmware								R/W	Interrupt mask2
2EH	Defined by firmware								R/W	Interrupt mask3
2FH	Defined by firmware								R/W	Interrupt mask4
30H						DMA-ERR	DMA-RDY	DMA-DONE	R/W	Interrupt0
31H	Defined by firmware								R/W	Interrupt1
32H	Defined by firmware								R/W	Interrupt2
33H	Defined by firmware								R/W	Interrupt3
34H	Defined by firmware								R/W	Interrupt4
35H						iROM2 to iROM0			R/W	Mask ROM cycle
36H							ISREQ	OSVLD	R/W	Port setup
37H to 3DH										
3EH							NBR	ALL RESET	R/W	Reset
3FH	TD7 to TD0								R/W	Transfer data

3.2 Register Functions

3.2.1 Common register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W
00H to 1FH	Defined by firmware								R/W

Each firmware defines these registers.

These registers are used to communicate with host CPU and internal CPU.

For the details of the register, refer to the application notebook.

The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes addresses 00H and 01H addresses to 0H. The original value of the other register is unsettled. It keeps a setting value before reset.

3.2.2 Data transfer register

These registers are defined data transfer such as host CPU → SDRAM, SDRAM → host CPU, host CPU → iRAM of internal CPU, SDRAM → iRAM of internal CPU and instruction ROM → iRAM of internal CPU.

The host CPU transfers with SDRAM via had a transfer buffer of 128 bytes on this LSI.

The transfer with the instruction RAM becomes 4 bytes.

★ A transfer error occurs if the transfer mode register, source address register, destination address register, or transfer counter register is changed before releasing the transfer mode register following transfer completion after setting the transfer mode register and starting the transfer. When transferring data as follows: host CPU → instruction RAM of internal CPU, host CPU → SDRAM, SDRAM → instruction RAM of internal CPU, instruction ROM → SDRAM, instruction ROM → instruction RAM of internal CPU, execute a software reset of the internal CPU (address 3EH ← 02H) before transfer and release the reset after transfer.

(1) Data transfer register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
20H	SI	SSD	SDI	MSD	MI		SDW	SDR	R/W	Download mode

Bit	Field	Function	Initial value
7	SI	Host CPU→instruction RAM of internal CPU 0: Releasing of transfer, 1: Transfer Note	0
6	SSD	Host CPU→SDRAM 0: Releasing of transfer, 1: Transfer Note	0
5	SDI	SDRAM→instruction RAM of internal CPU 0: Releasing of transfer, 1: Transfer Note	0
4	MSD	Instruction ROM→SDRAM 0: Releasing of transfer, 1: Transfer Note	0
3	MI	Instruction ROM→instruction RAM of internal CPU 0: Releasing of transfer, 1: Transfer Note	0
2		Reserved (set only 0)	0
1	SDW	Host CPU→SDRAM 0: Releasing of transfer,1: Transfer	0
0	SDR	SDRAM→host CPU 0: Releasing of transfer, 1: Transfer	0

Note Set internal CPU reset (with Register 3EH←02H)

More than one bit cannot be set to 1 at the same time. It becomes a transfer error when writing at the transfer mode register while transferring. When canceling a transfer while transferring, it stops a transfer. At this time, the data in the transfer buffer becomes invalid. The transfer of SDR with once is to a maximum of 128 bytes. If host CPU stops the transfer, host CPU should operate transfer error handling.

(2) Source address register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
21H					SA19 to SA16				R/W	Source address
22H	SA15 to SA8								R/W	Source address
23H	SA7 to SA0								R/W	Source address

It sets the address of the data to transfer. It becomes effective in case of transfer from SDRAM or instruction ROM. Until it releases a transfer mode after setting a transfer mode register, it isn't possible to change. The transfer error occurs when rewriting this register before releasing a transfer mode. The relation with the address of SDRAM, external instruction ROM is shown in **Figure 3-2** and **3-3**. The addressing of SDRAM becomes a 32 address by 4-word unit (128 bytes).

The relation with the SDRAM bank and address is shown in **Table 3-1**.

Figure 3-2. Relation of Source Address and SDRAM Address

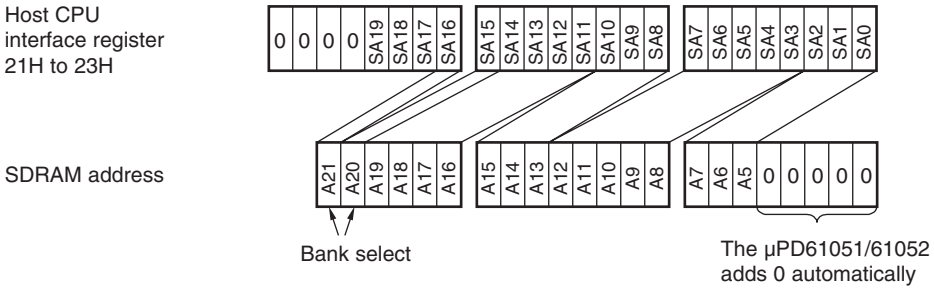


Figure 3-3. Relation of Source Address and External Instruction ROM Address

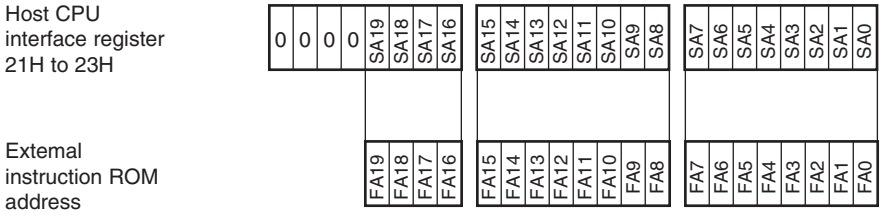


Table 3-1. Relation of SDRAM Bank and Address

Memory	Bank A	Bank B	Bank C	Bank D
16 Mbit SDRAM by 2	000000H to 07FFFFH	200000H to 27FFFFH	-	-
16 Mbit SDRAM by 1	000000H to 07FFFFH	200000H to 27FFFFH	100000H to 17FFFFH	300000H to 37FFFFH
64 Mbit SDRAM by 2 128 Mbit SDRAM by 1 128 Mbit SDRAM by 2	000000H to 0FFFFFH	200000H to 2FFFFFH	100000H to 1FFFFFH	300000H to 3FFFFFH
128 Mbit SDRAM by 1	000000H to 0FFFFFH	200000H to 2FFFFFH	100000H to 1FFFFFH	300000H to 3FFFFFH

(3) Destination address register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
24H								DA16	R/W	Destination address
25H	DA15 to DA8								R/W	Destination address
26H	DA7 to DA0								R/W	Destination address

It sets Destination address. It becomes effective in case of transfer to SDRAM or instruction RAM of internal CPU. It isn't possible to change until it cancels a transfer mode after setting a transfer mode register. It becomes a transfer error when rewriting before canceling a transfer mode. The relation of the address of SDRAM and instruction RAM of internal CPU is as in **Figure 3-4** and **3-5**. The addressing of SDRAM becomes a 32 address by 4-word unit (128 bytes).

Figure 3-4. Relation of Destination Address and SDRAM Address

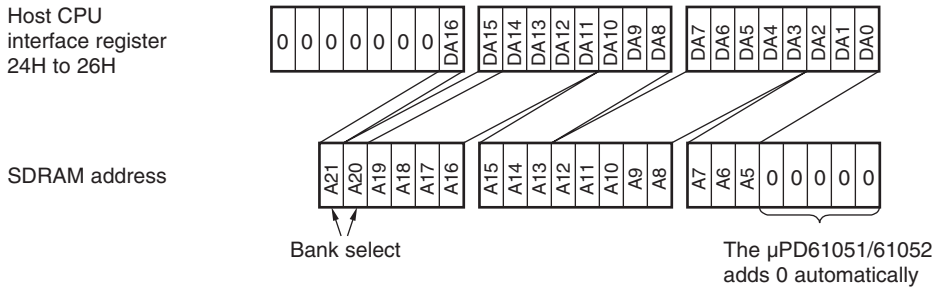
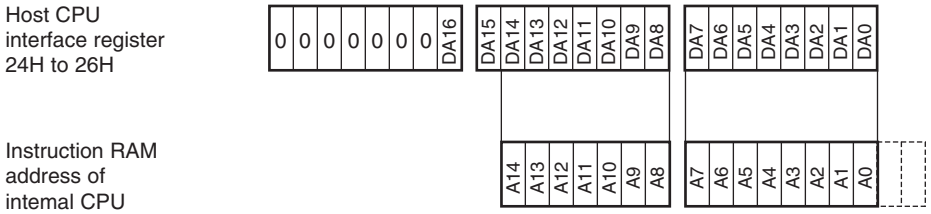


Figure 3-5. Relation of Destination Address and Instruction ROM Address of Internal CPU



(4) Transfer data counter register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
27H						TC18 to TC16			R/W	Transfer data count
28H	TC15 to TC8								R/W	Transfer data count
29H	TC7 to TC0								R/W	Transfer data count

It sets the transfer data number of the bytes.

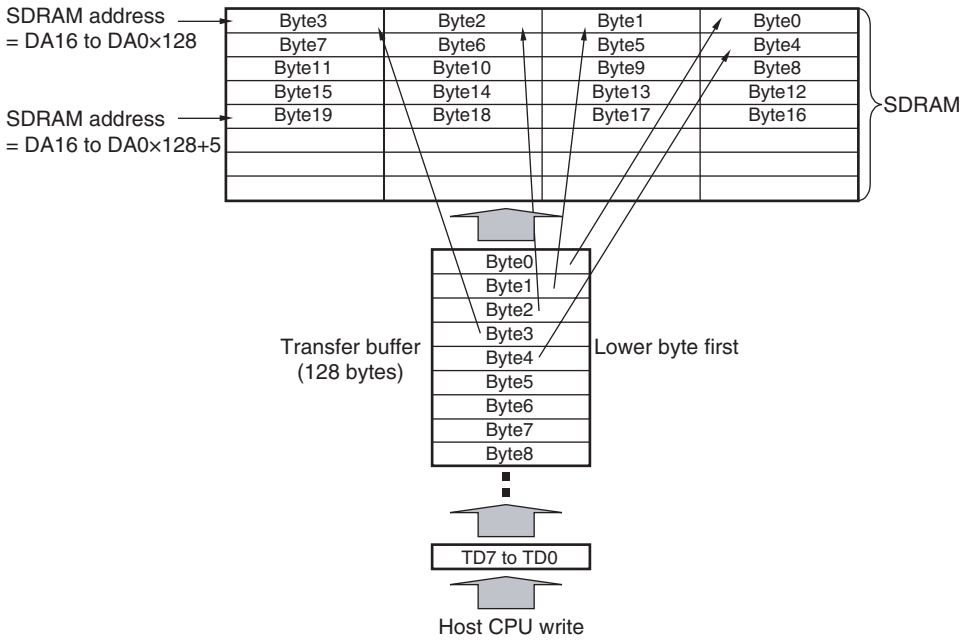
In case of transfer between host CPU and SDRAM, it sets the number of the transfer bytes by 4 bytes unit. In case of transfer from instructions ROM, SDRAM host CPU to the instruction RAM of internal CPU, it sets the number of the transfer bytes /4 by the 4 byte unit.

(5) Transfer data register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
3FH	TD7 to TD0								R/W	Transfer data

This register is transfer data window.

Figure 3-6. SDRAM Write



SDRAM read

<1> Interrupt mask

Host CPU sets mask bit to interrupt mask register (2CH to 2FH) for the interrupt that needs a data transfer.

<2> Set source address

Host CPU sets the address of SDRAM to the source address register (21H to 23H) of the μPD61051/61052.

<3> Set the number (equal to or less than 128 bytes) of the data to read by 4 bytes unit

Host CPU sets the data number of the bytes to the transfer data counter register (27H to 29H) of the μPD61051/61052.

<4> Set the transfer of SDRAM → host CPU.

Host CPU sets 01H to the transfer mode register (20H) of the μPD61051/61052.

<5> CINT interrupt (Interrupt pin)

<6> Confirms that the interrupt factor and clear interrupt factor

Host CPU confirms that the interrupt register 0 (30H) of the μPD61051/61052 becomes 02H or 01H and clears writing a same value of the interrupt register 0 (30H) to the interrupt register 0 (30H) of the μPD61051/61052.

<7> Data read

Host CPU reads data from the number of times with the set number of bytes, the transfer data register (3FH) of the μPD61051/61052.

<8> CINT interrupt (Interrupt pin)

<9> Confirm the interrupt factor

Host CPU confirms that the interrupt register 0 (30H) of the μPD61051/61052 becomes 01H. (It clears a writing interrupt factor in 01H at the interrupt register 0 (30H) register of the μPD61051/61052.)

<10> Release of SDRAM → host CPU mode

Host CPU clears a writing interrupt factor in 01H at the interrupt register 0 (30H) register of the μPD61051/61052 after setting 00H to the transfer mode register (20H) of the μPD61051/61052.

<11> Release of interrupt mask

It releases the limitation on interrupt which set by <1>.

SDRAM write

<1> Interrupt mask

Host CPU sets mask bit to interrupt mask register (2CH to 2FH) for the interrupt that needs a data transfer.

<2> Set destination address

Host CPU sets the address of SDRAM to the destination address register (24H to 26H) of the μPD61051/61052.

<3> Set the number of the data to write by a 4 byte unit

Host CPU sets the data number of the bytes by 4 bytes unit to the transfer data counter register (27H to 29H) of the μPD61051/61052.

<4> Set the transfer of host CPU → SDRAM

Host CPU sets 02H to the transfer mode register (20H) of the μPD61051/61052.

<5> Data write

Host CPU writes data to the transfer data register (3FH) of the μPD61051/61052 at times with more few 128 bytes or transfer data count register setting value.

<6> CINT interrupt (Interrupt pin)

<7> Confirm the interrupt factor

When the number of the transfer data is less then 128 bytes, host CPU confirms that the interrupt register 0 (30H) of the μPD61051/61052 becomes 01H, and go to <9>.

<8> Confirm that next data transfer prepare completed

Host CPU confirms that the interrupt register 0 (30H) of the μPD61051/61052 becomes 02H or 01H and clears a writing sane value of the interrupt register 0 (30H) to the interrupt register 0 (30H) of the μPD61051/61052. Return to <5> and next data write.

<9> Release of SDRAM → host CPU

Host CPU clears a writing interrupt factor in 01H at the interrupt register 0 (30H) register of the μPD61051/61052 after setting 00H to the transfer mode register (20H) of the μPD61051/61052.

<10> Release of interrupt mask

It releases the limitation on interrupt which is set by <1>.

<11> In the case of an interrupt to internal CPU, it is necessary

Host CPU sets a data bank number and the number of the bytes to the address that defined with the firmware. It sets 01H to the 2AH address of the μPD61051/61052 and it notifies an interrupt to internal CPU.

3.2.3 Internal CPU interrupt register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
2AH								iCPU-INT	R/W	Int. to internal CPU

Host CPU set interrupt to internal CPU. Internal CPU clears this bit after interrupt operation.

The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.

3.2.4 Interrupt mask register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
2BH						DMA-ERR-M	DMA-RDY-M	DMA-DONE-M	R/W	Interrupt mask0
2CH	Defined by firmware								R/W	Interrupt mask1
2DH	Defined by firmware								R/W	Interrupt mask2
2EH	Defined by firmware								R/W	Interrupt mask3
2FH	Defined by firmware								R/W	Interrupt mask4

These registers are interrupt masks for next interrupt. Interrupt mask can be set bit by bit. When setting an interrupt mask, CINT does not become high even if the interrupt register becomes 1.

The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.

3.2.5 Download interrupt register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
30H						DMA-ERR	DMA-RDY	DMA-DONE	R/W	Interrupt0

It is set for 1 when the interrupt factor occurs.

The interrupt bit clears when host CPU writes to this register after the interrupt processing.

The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.

★ Clear processing continues until interrupt registers is cleared.

Bit	Field	Function	Initial value
7 to 3		Reserved (set 0)	
2	DMA-ERR	Data transfer error 0: Normal, 1: Error	0
1	DMA-RDY	Data transfer prepared 0: Normal, 1: Transfer	0
0	DMA-DONE	Data transfer ended 0: Normal, 1: Transfer ended	0

It outputs DMA-RDY or DMA-DONE every 128-byte transfer. DMA-DONE is output when the transfer ends.

3.2.6 Interrupt register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
31H	Defined by firmware								R/W	Interrupt1
32H	Defined by firmware								R/W	Interrupt2
33H	Defined by firmware								R/W	Interrupt3
34H	Defined by firmware								R/W	Interrupt4

It is set for 1 when the interrupt factor occurs.
The interrupt bit clears when host CPU writes 1 in the bit of the interrupt after the interrupt processing.
When the other interrupt (which isn't masked) is set to 1 when clearing a interrupt, CINT becomes high 1 μs later.
The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.
★ Clear processing continues until interrupt registers is cleared.

Address	Bit	Field	Function	Initial value
31H to 34H	7 to 0		Firmware define 0: Normal, 1: Interrupt	0H

3.2.7 Reset register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
3EH							NBR	ALL RESET	R/W	Reset

When the host CPU sets 1 to ALL RESET, it resets the inside and it returns to 0 automatically.
The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.

Bit	Field	Function	Initial value
7 to 2		Reserved (Set 0)	
1	NBR	Internal CPU reset 0: Normal, 1: Reset	0
0	ALL RESET	Same hardware reset 0: Normal, 1: Reset	0

3.2.8 ROM access cycle register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
35H						iROM2 to iROM0			R/W	Mask ROM cycle

It specifies the access cycle of the instruction ROM of internal CPU when connecting host CPU interface with the serial bus. The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 7H.

Bit	Field	Function	Initial value
7 to 3		Reserved (Set 0)	
2 to 0	iROM2 to iROM0	Access cycle of instruction ROM 0: Reserved, 1 to 7: (Setting value+2) by 24.6 MHz	7H

3.2.9 Port setup register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	
36H							ISREQ	OSVLD	R/W	Port setup

This register sets the active polarity of ISREQ and OSVLD. The reset of the $\overline{\text{RESET}}$ pin or ALL RESET of the reset register initializes this address to 0H.

Bit	Field	Function	Initial value
7 to 2		Reserved (Set 0)	
1	ISREQ	Active polarity of ISREQ 0: Low active of request, 1: High active of request	0
0	OSVLD	Active polarity of OSVLD/OSRDY 0: Low active of valid/ready 1: High active of valid/ready	0

4. SYSTEM INTERFACE PROCEDURE

The host CPU transfers the firmware of each operation mode to the instruction RAM of the internal CPU and works it.

This LSI stores up firmware in SDRAM. Host CPU sets to load the firmware of each operation mode in the instruction RAM of internal CPU from SDRAM.

When using a parallel bus interface for the host CPU interface, the host CPU sets a data transfer register after hardware reset and transfers the initialization program of SDRAM to instruction RAM of internal CPU and executing. Host CPU writes firmware to SDRAM.

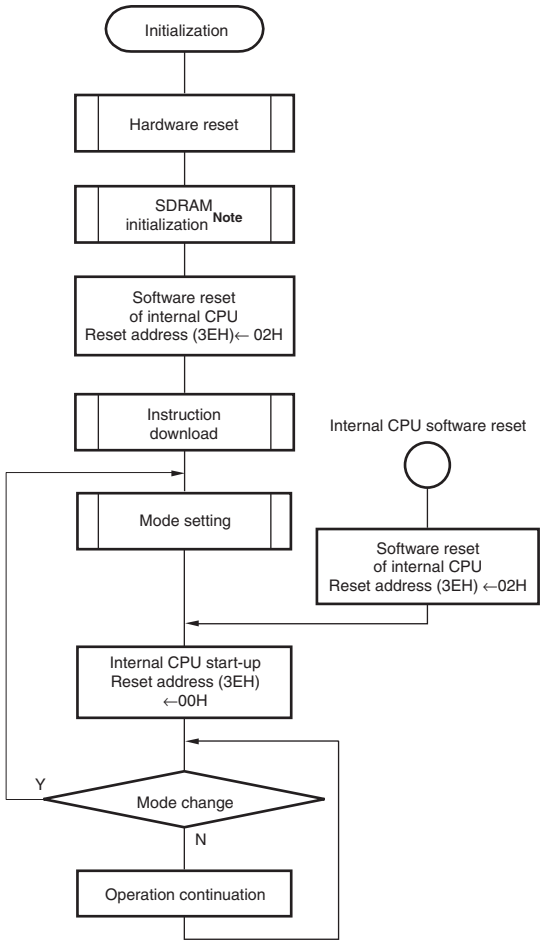
When using a serial bus interface for the host CPU interface, the host CPU sets a data transfer register after hardware ware reset and transfers the initialization program of SDRAM to instruction RAM of internal CPU from external instruction ROM and executing. Host CPU loads firmware in SDRAM from instruction ROM outside.

It stores the firmware of the encoding and the transcode to SDRAM from ROM in case of start-up of the system, and then it can do the changing of a feature at short time by the high-speed transfer of SDRAM.

The host CPU sets the mode of the terminal of the μPD61051/61052 and the access cycle of ROM to the system interface register after hardware reset and sets the transfer of the instruction of the internal CPU after SDRAM is initialized.

4.1 Outline

An overview from the reset of the hardware to the setting of an operation mode is shown.



Note This is not necessary in case that the SDRAM initialization firmware is not separated.

4.2 Firmware Download

The host CPU downloads the firmware at the instruction RAM for the internal CPU.

When a host CPU is connected with the serial bus, the firmware can be downloaded from the external ROM for the download processing to speed up. In addition, it stores more than one piece of firmware in the instruction pool area of SDRAM and it can be replaced depending on the need, too.

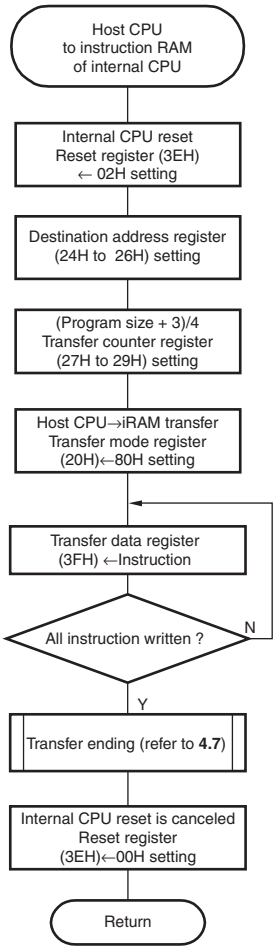
When transferring to the instruction RAM of the internal CPU, the transfer counter register setting value (number of the transfer bytes / 4) is (program size +3)/ 4.

4.2.1 Host CPU to instruction RAM of internal CPU

Host CPU transmits the firmware to instruction RAM of the internal CPU.

When transferring data continuously, transfer during resetting an internal CPU, If reset of internal CPU is can canceled on the way, the internal CPU sometime malfunction.

★

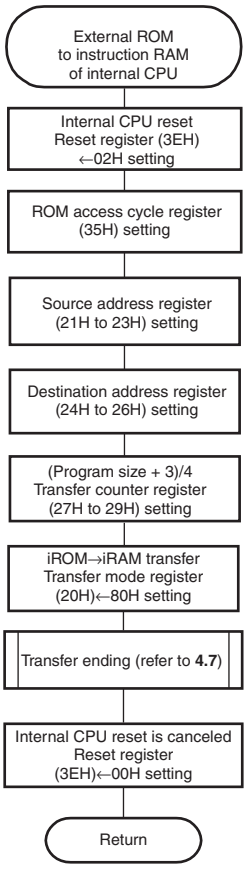


4.2.2 External ROM to instruction RAM of internal CPU

When the host CPU is a serial bus type, CPU transmits the instruction of a mode from external ROM to instruction RAM of Internal CPU.

When transferring data continuously, transfer during resetting an internal CPU, If reset of internal CPU is can canceled on the way, the internal CPU sometime malfunction.

★

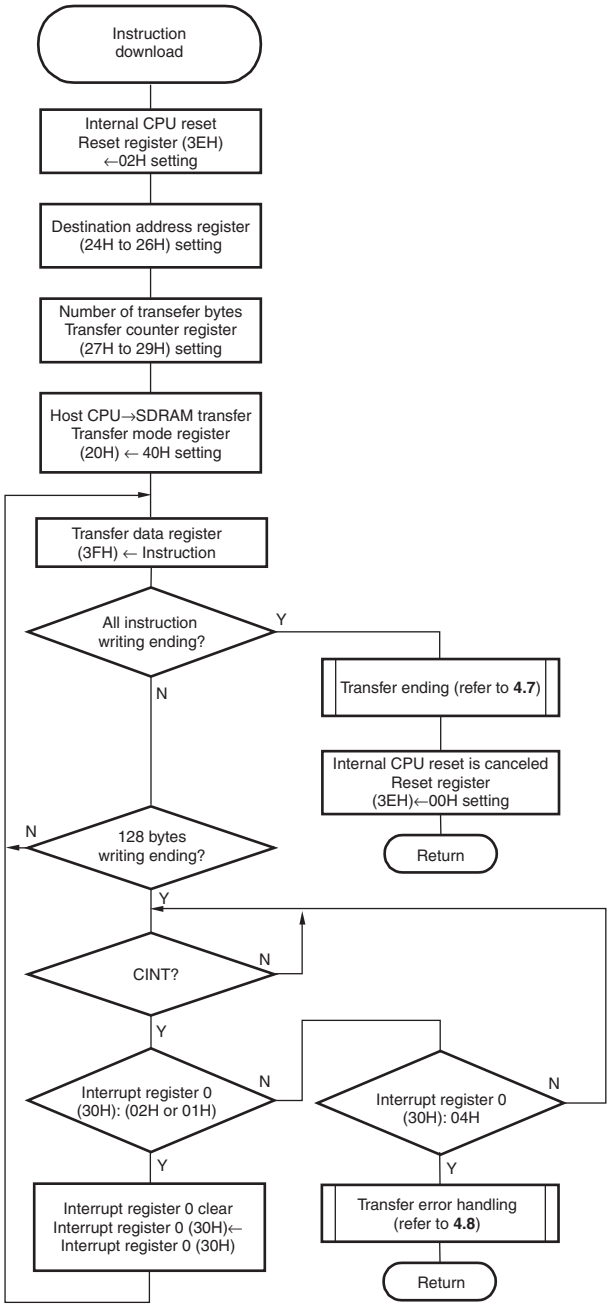


4.2.3 Host CPU to SDRAM

The host CPU can store firmware in the instruction pool area of SDRAM for the internal CPU. It stores more than one piece of firmware and it can be replaced depending on the need, too.

When transferring data continuously, transfer during resetting an internal CPU, If reset of internal CPU is can canceled on the way, the internal CPU sometime malfunction. The number of the transfer bytes is a 4-byte unit.

★



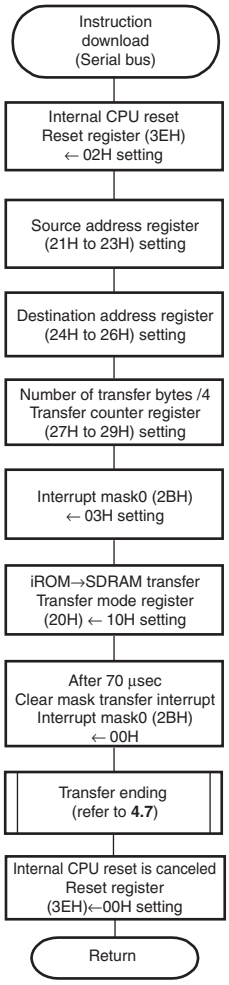
4.2.4 External ROM to SDRAM

The firmware for the internal CPU can be stored in the firmware cabinet of SDRAM from the external ROM. It stores more than one piece of firmware beforehand and it can be replaced according to need, too.

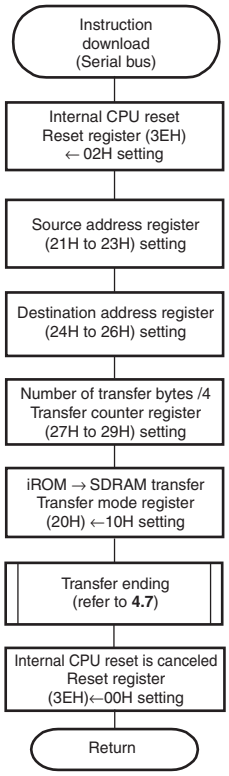
When transferring data continuously, transfer during resetting an internal CPU, If reset of internal CPU is can canceled on the way, the internal CPU sometime malfunction. When transferring data below the 1k-byte, transfer, dividing every 128 bytes. The number of the transfer bytes is a 4-byte unit.

★

(a) Transfer over 1 Kbytes

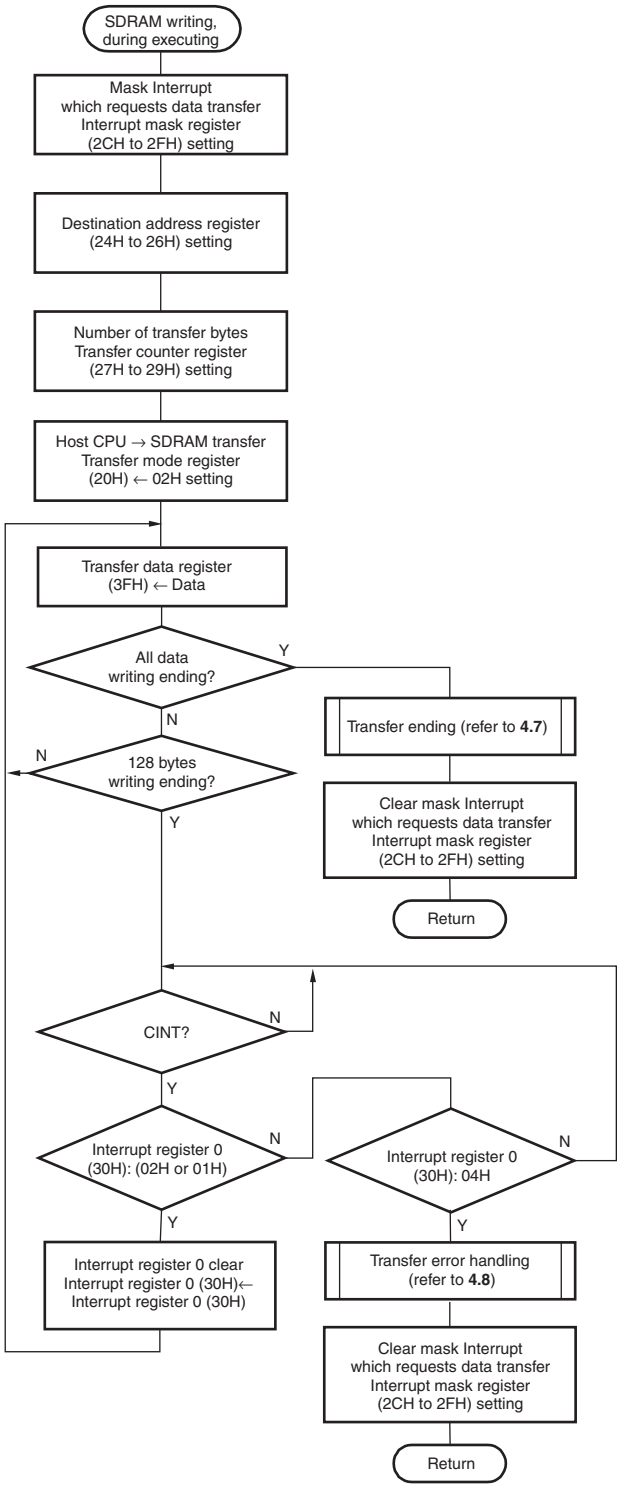


(b) Transfer below 128 bytes



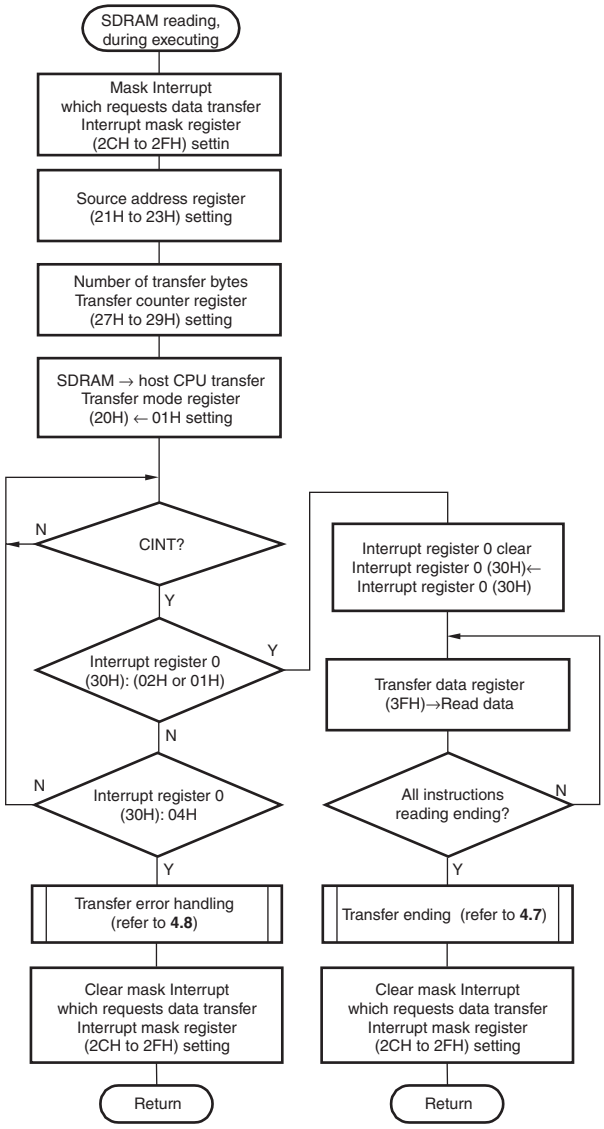
4.3 SDRAM Write during Executing

While encoding, the host CPU can transfer parameters to the internal CPU through SDRAM. The number of the transfer bytes is a 4-byte unit.



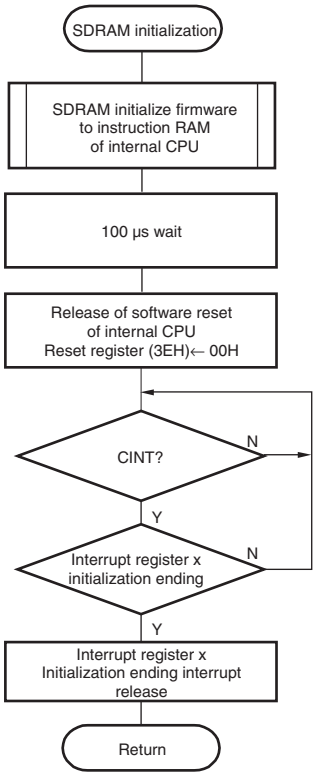
4.4 SDRAM Read during Executing

While encoding, the host CPU reads parameters of usable work area of SDRAM. The maximum data of the reading once is 128 bytes. When reading is equal to or more than 128 byte data, execute reading processing repeatedly. The number of the transfer bytes is a 4 bytes unit.



4.5 SDRAM Initialization

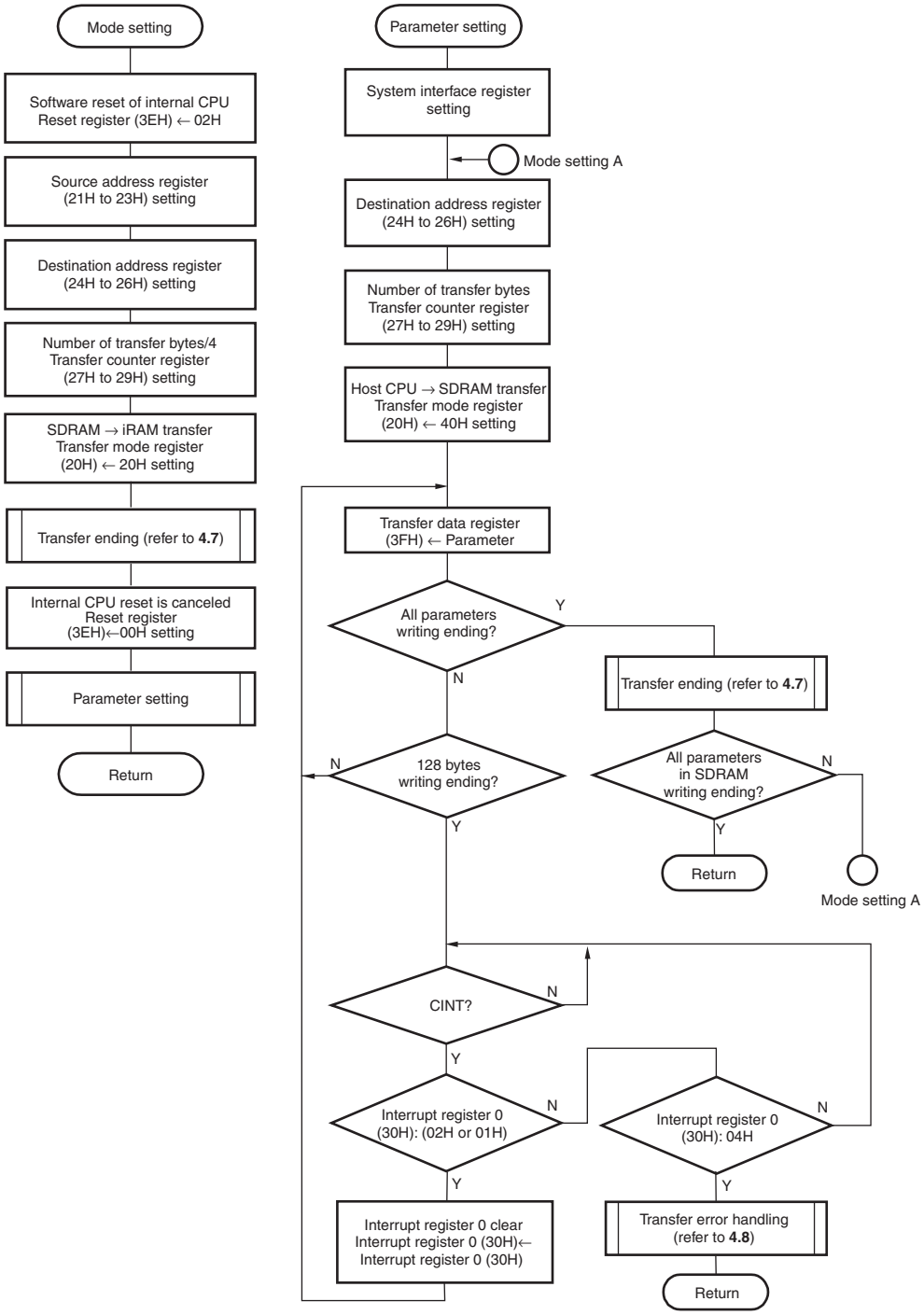
The host CPU transfers the firmware which makes SDRAM a standby condition to the instruction RAM of the internal CPU and executes it.



4.6 Operation Mode Setting by Changing Firmware

When changing a mode, host CPU transfers the instruction of each mode from SDRAM to the instruction RAM of the internal CPU and restarts.

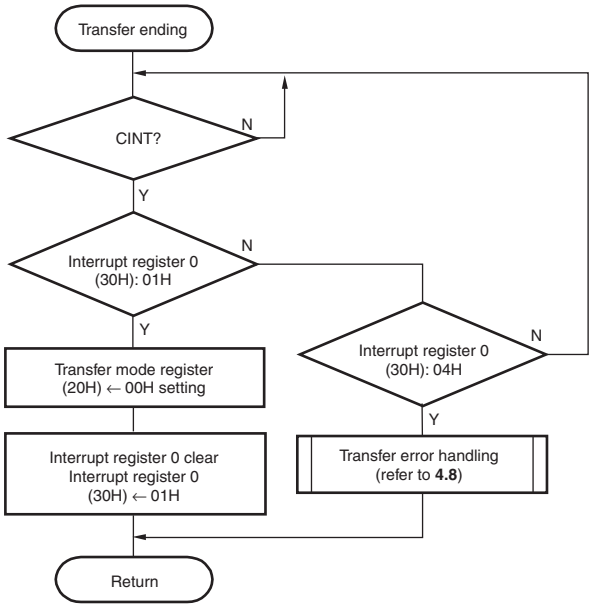
★



4.7 Transfer Ending

The host CPU confirms a transfer error when the instruction or data transfer ends.

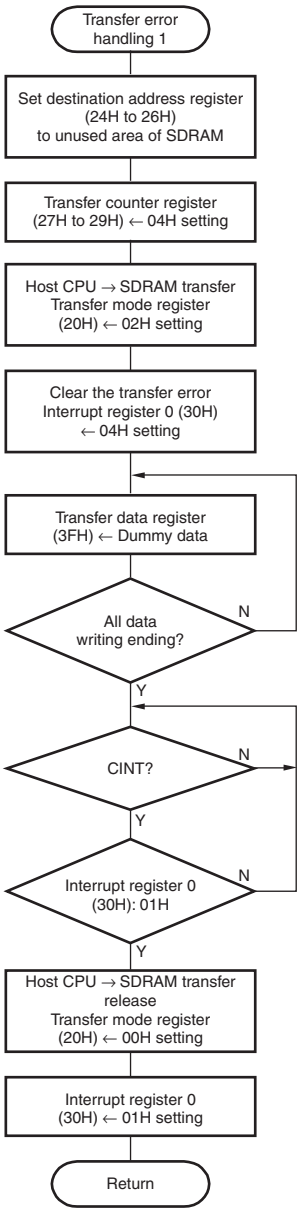
The host CPU clears transfer mode and interrupt registers.



4.8 Transfer Error Handling

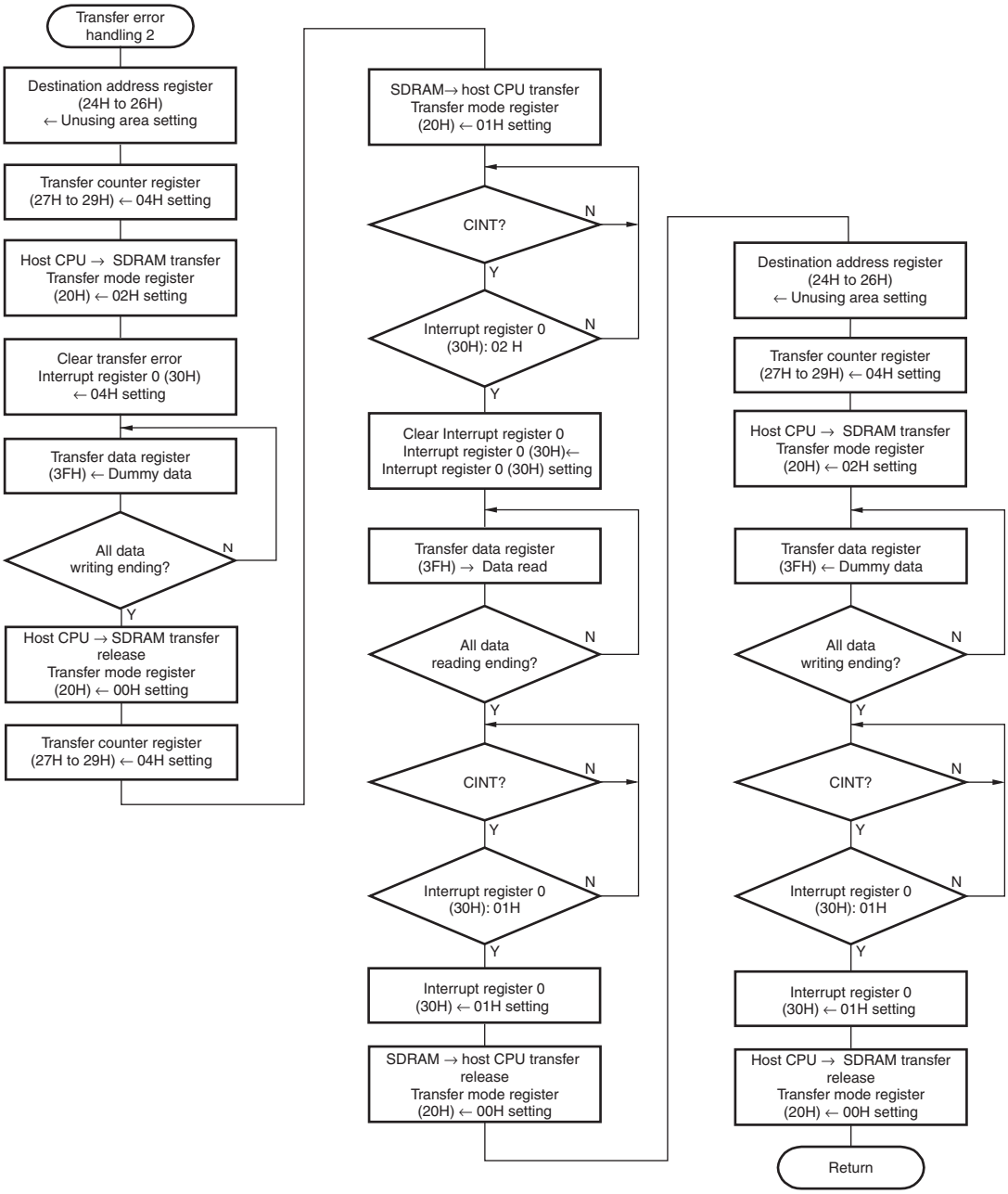
4.8.1 Transfer error handling 1

It is the error handling of DMA-ERR which occurs when interrupting the transfers (the host CPU → the instruction RAM of internal CPU transfer, the host CPU → SDRAM transfer (SSD, SDW), the external ROM → SDRAM transfer and the external ROM → the instruction RAM of internal CPU transfer)



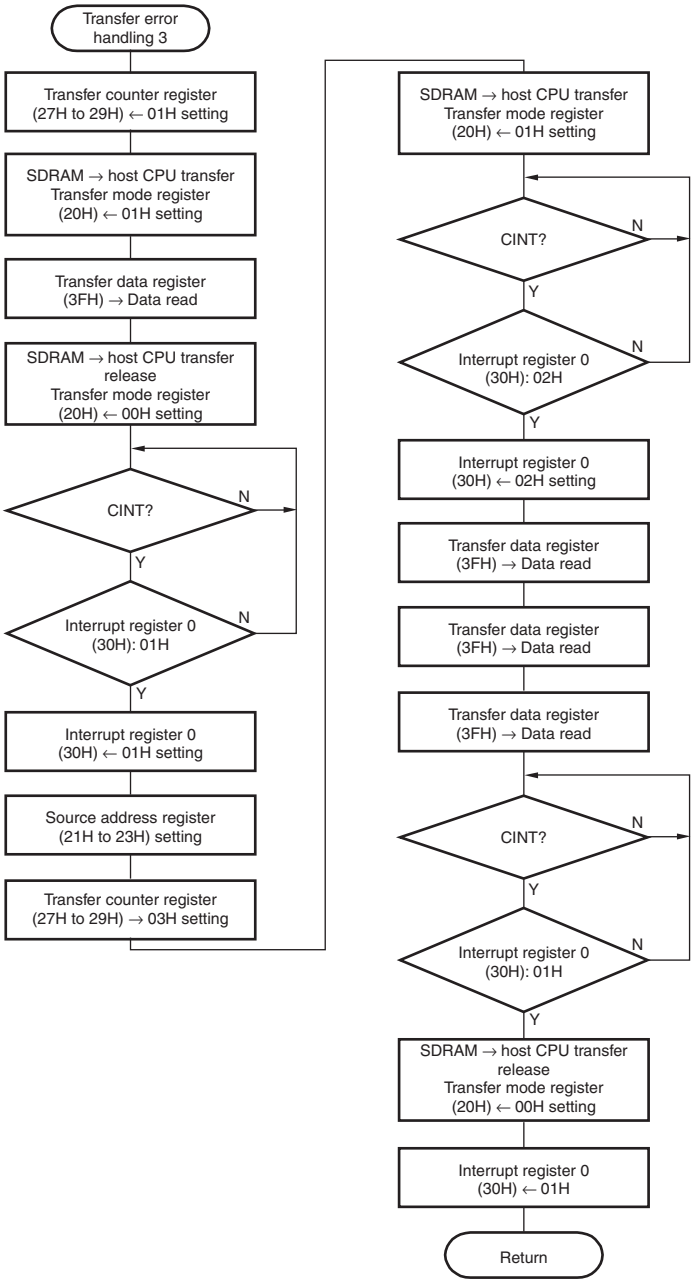
4.8.2 Transfer error handling 2

This is a error handling of DMA-ERR which occurs when interrupting the transfers (SDRAM read during executing and SDRAM → instruction RAM of internal CPU transfer)



4.8.3 Transfer error handling 3

It is the error handling of DMA-ERR which occurs when transfer operation in case of host CPU serial connection with SPI.

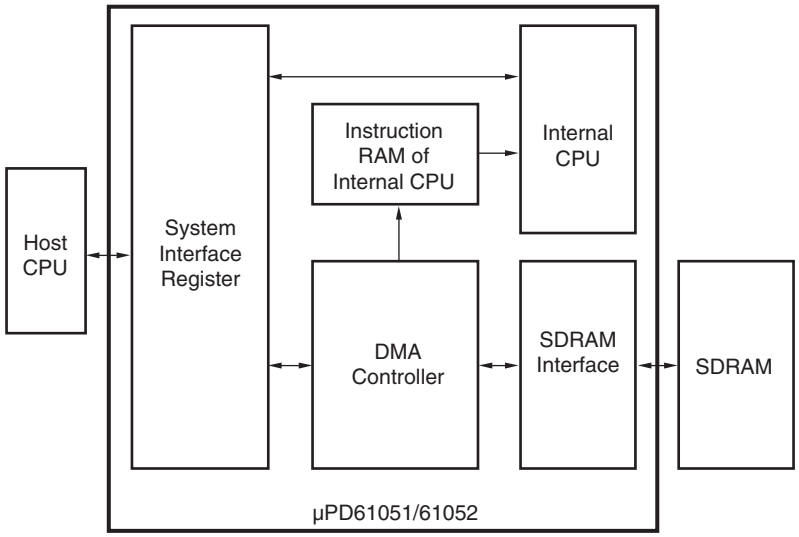


5. EXAMPLE FOR COMMON REGISTER USAGE

The μPD61051, 61052 operates while the “command code register” is in “start”. When “command code register” becomes “start”, internal CPU reads parameter registers, then starts the operation. Additionally, internal register sets “status register”. Register map for system interface register is defined by firmware.

With each application, parameter registers are changed by the firmware.

Figure 5-1. Host Interface Register



5.1 Register Map Example

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00H						COMCODE		
01H						ESTS		
02H to 1FH	Parameters (Defined by each firmware)							
20H	SI	SSD	SDI	MSD	MI		SDW	SDR
21H					SA19 to SA16			
22H	SA15 to SA8							
23H	SA7 to SA0							
24H								DA16
25H	DA15 to DA8							
26H	DA7 to DA0							
27H						TC18 to TC16		
28H	TC15 to TC8							
29H	TC7 to TC0							
2AH								iCPU-INT
2BH						DMA-ERR-M	DMA-RDY-M	DMA-DONE-M
2CH to 2FH	Interrupt Mask (Defined by each firmware)							
30H						DMA-ERR	DMA-RDY	DMA-DONE
31H to 34H	Interrupt (Defined by each firmware)							
35H						iROM2 to iROM0		
36H							ISREQ	OSVLD
37H to 3DH								
3EH							NBR	ALL RESET
3FH	TD7 to TD0							

: Reserved

5.2 Example of the Common Register Which A Firmware Defines

5.2.1 COMCODE: Command code register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00H						COMCODE		

The host CPU can change the state of operation to the command code register. The μPD61051/61052 accepts commands to operate in three states as shown in the table below.

Command	Code
Standby / Stop	001
Start	011
Reserved	Others

The command which it is possible to set depend on the internal state.

In case of the command whose state transfer is possible, the state transfers according to the command.

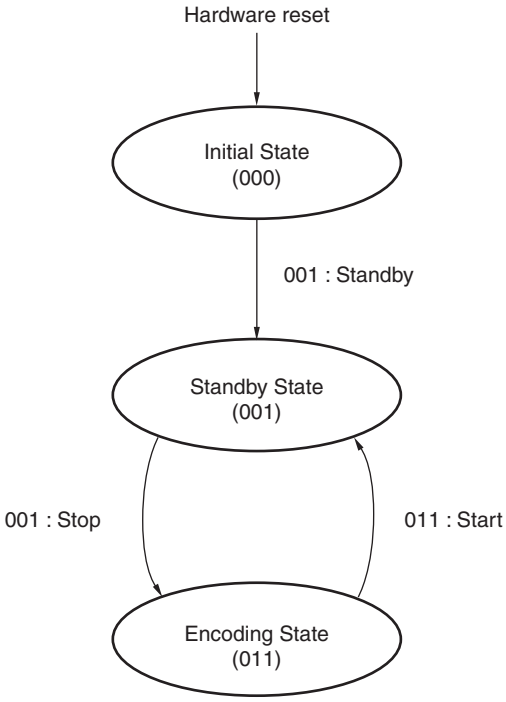
5.2.2 ESTS: Status register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
01H						ESTS		

This register shows processing state, when command is illegal, the state doesn't transfer.

ESTS	Code
Initial State	000
Standby State	001
Encoding State	011

Figure 5-2. Command Status Transition



Valid Command in Initial State: Standby
Valid Command in Standby State: Start
Valid Command in Operation State: Stop

6. ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Rating	Unit
Supply Voltage	V_{DD3}	V_{DD3}, vs GND	4.6	V
	V_{DD2}	V_{DD2}, vs GND PV_{DD2}, vs PGND	3.6	V
Input Voltage	V_{IN}	Vs GND3	−0.5 to +4.6	V
Output Voltage	V_{OUT}	Vs GND3	−0.5 to +4.6	V
Output Current	I_{OUT}		20	mA
Permissible Loss	P_D		2	W
Operating Ambient Temperature	T_A		0 to +70	°C
Storage Temperature	T_{stg}		−55 to +125	°C

Caution If any of the parameters exceeds the absolute maximum ratings, even momentarily, the quality of the product may be impaired. The absolute maximum ratings are values that may physically damage the products. Be sure to use the products within the ratings.

DC Characteristics (T_A = 0 to +70°C, V_{DD3} = 3.3±0.165 V, V_{DD2} = 2.5±0.2 V)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Supply voltage	V_{DD3}	V_{DD3}, vs GND	3.135	3.3	3.465	V
	V_{DD2}	V_{DD2}, vs GND PV_{DD2}, vs PGND	2.3	2.5	2.7	V
High-level input voltage	V_{IH}		2.2		V_{DD3}+0.5	V
Low-level input voltage	V_{IL}	SCLK	−0.5		+0.6	V
		Except SCLK	−0.5		+0.7	V
High-level output voltage	V_{OH}		2.4			V
Low-level output voltage	V_{OL}				0.4	V
Input leakage current	I_{LI}	Except MD31 to MD0 and CMODE1			±10	μA
Operating current	I_{DD3}	3.3 V power supply			70	mA
	I_{PDD}	2.5 V PLL power supply			15	mA
	I_{DD2}	Internal logic power supply of 2.5 V			510	mA

Pin Capacitance (T_A = 25°C)

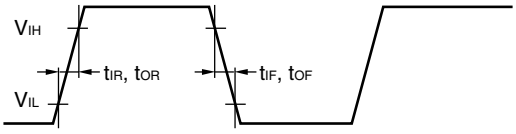
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input capacitance	C_I				20	pF
Output capacitance	C_O				20	pF
I/O capacitance	C_{IO}				20	pF

AC Characteristics (T_A = 0 to +70°C, V_{DD3} = 3.3±0.165 V, V_{DD2} = 2.5±0.2 V, C_L = 15 pF, t_R = t_F = 1 ns)

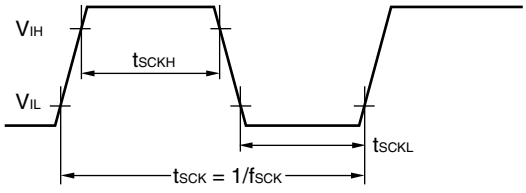
(1) System

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLK frequency	f_{SCK}			27.0		MHz
SCLK high-level width	t_{SCKH}	Duty 40:60	13.2			ns
SCLK low-level width	t_{SCKL}	Duty 40:60	13.2			ns
PSTOP release time1	t_{STP1}	Vs V_{DD3}	1			μs
PSTOP release time2	t_{STP2}	Vs V_{DD2}	1			μs
PSTOP release time3	t_{STP3}	Vs PV_{DD2}	1			μs
PSTOP release time4	t_{STP4}	Vs SCLK	1			μs
PSTOP pulse width	t_{WSTP}		1			μs
$\overline{\text{RESET}}$ release time	t_{RES}	Vs falling edge of PSTOP	100			μs
Video input reset time	t_{IVRES}	After stable IVCLK	600			ns
Audio reset time	t_{AURES}	After stable AMCLK	600			ns
STC reset time	t_{STRES}	After stable STCLK	600			ns
Reset pulse width	t_{RESW}	After stable all clock	600			ns
★ Input rising time	t_{IR}	Vs AMCLK, STCLK, SCLK, ISCLK			3	ns
		Vs IVCLK			5	ns
★ Input falling time	t_{IF}	Vs AMCLK, STCLK, SCLK, ISCLK			3	ns
		Vs IVCLK			5	ns
Output rising time	t_{OR}				3	ns
Output falling time	t_{OF}				3	ns

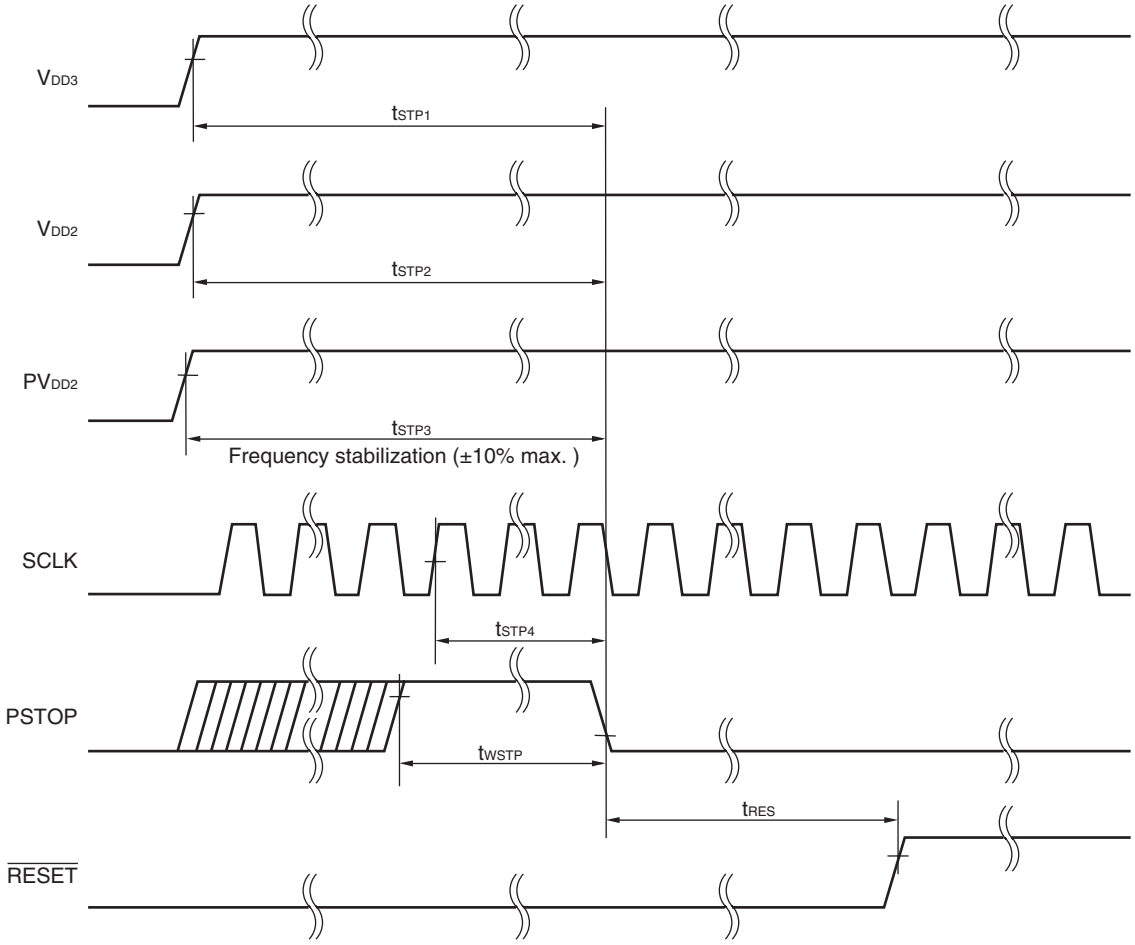
High level, low level



Clock input

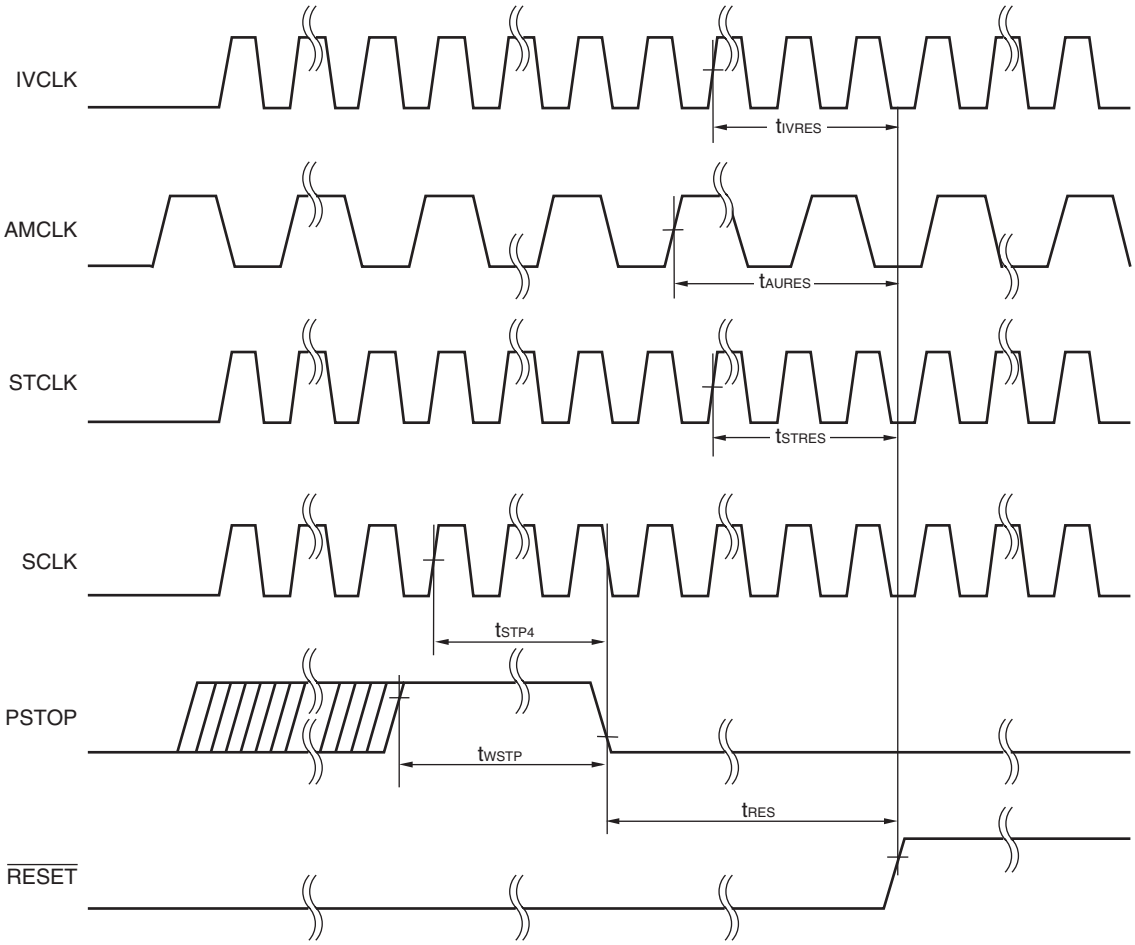


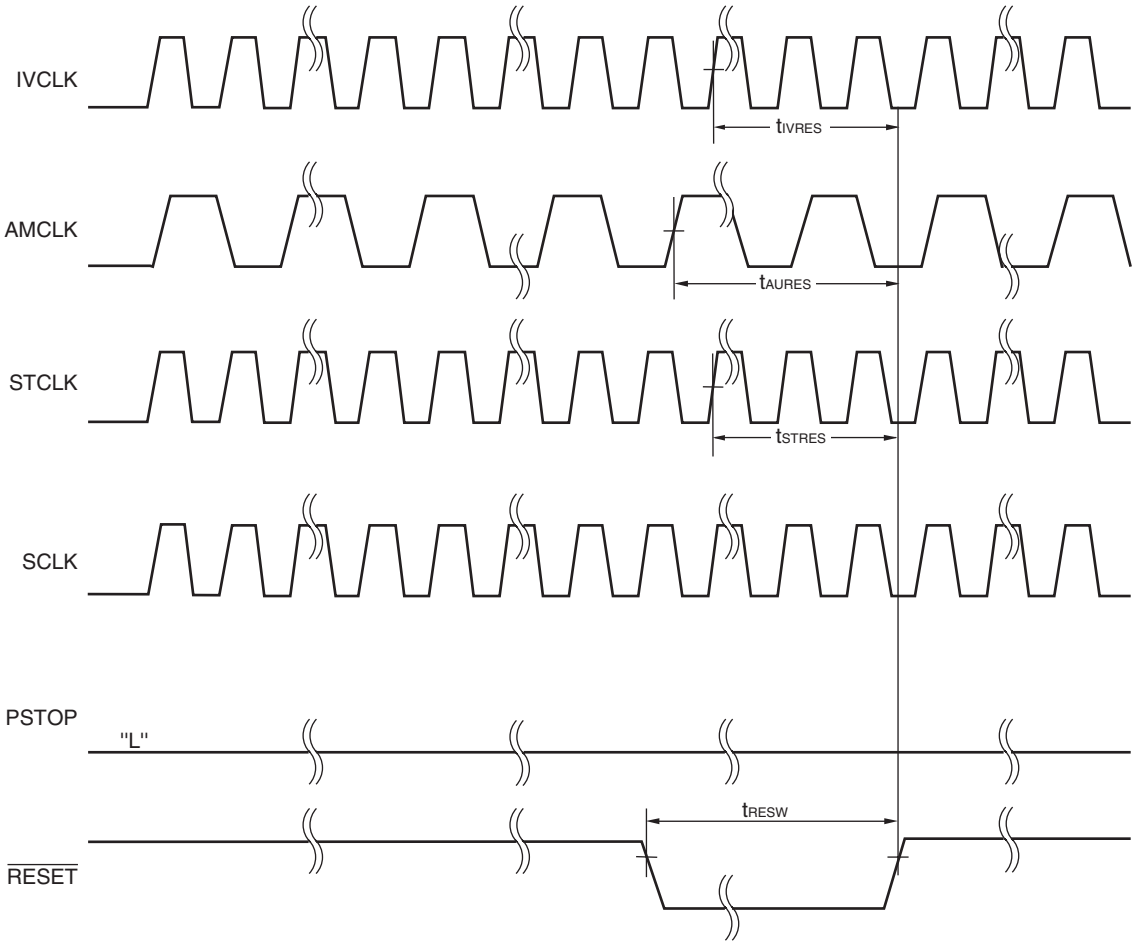
Reset input



★ **Caution Notes on power on/off**

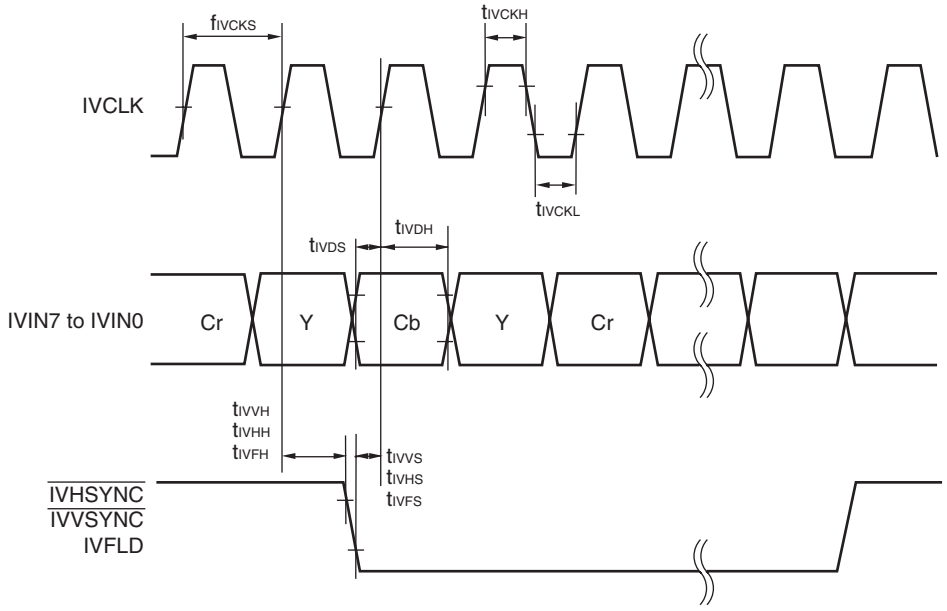
- **Apply power to V_{DD3}, and V_{DD2} and PV_{DD2} at the same time.**
- **If it is difficult to apply the power to these pins at the same time, apply the power to V_{DD2} and PV_{DD2} first.**
- **Cut the power of V_{DD3}, and V_{DD2} and PV_{DD2} at the same time.**
- **If it is difficult to cut the power of these pins at the same time, cut the power of V_{DD2} and PV_{DD2} last.**


IVCLK
tIVRES
AMCLK
tAURES
STCLK
tSTRES
SCLK
tSTP4
PSTOP
tWSTP
tRES
RESET


IVCLK
tIVRES
AMCLK
tAURES
STCLK
tSTRES
SCLK
PSTOP
"L"
tRESW
RESET

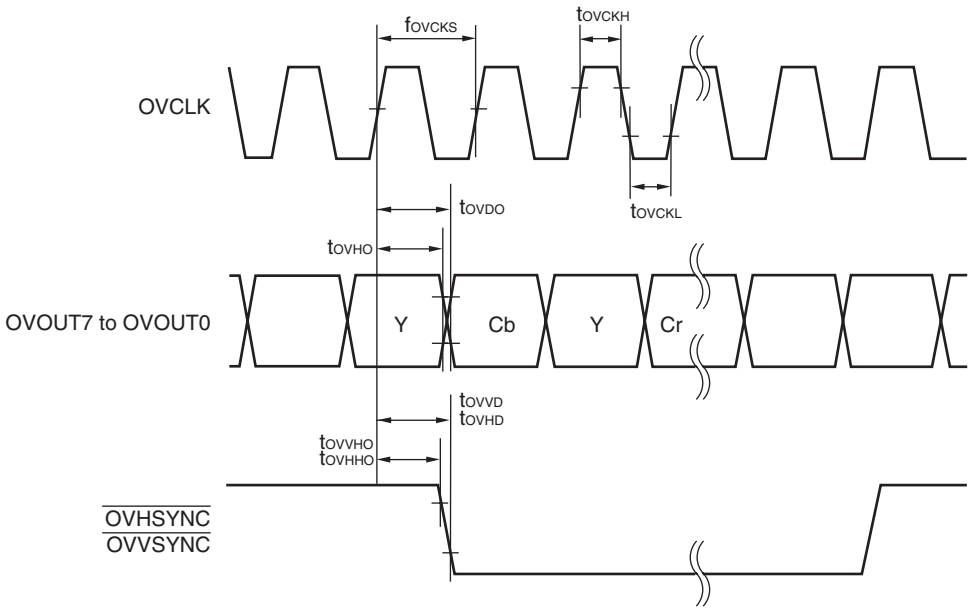
(2) Video input interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
IVCLK frequency	f_{IVCKS}			27		MHz
IVCLK high-level width	t_{VCKH}		10			ns
IVCLK low-level width	t_{VCKL}		10			ns
IVIN7 to IVIN0 setup time	t_{IVDS}	Vs rising edge of IVCLK	5			ns
IVIN7 to IVIN0 hold time	t_{IVDH}	Vs rising edge of IVCLK	4			ns
$\overline{\text{IVVSYNC}}$-input setup time	t_{IVVS}	Vs rising edge of IVCLK	5			ns
$\overline{\text{IVVSYNC}}$-input hold time	t_{IVVH}	Vs rising edge of IVCLK	4			ns
$\overline{\text{IVHSYNC}}$-input setup time	t_{IVHS}	Vs rising edge of IVCLK	5			ns
$\overline{\text{IVHSYNC}}$-input hold time	t_{IVHH}	Vs rising edge of IVCLK	4			ns
IVFLD-input setup time	t_{IVFS}	Vs rising edge of IVCLK	5			ns
IVFLD-input hold time	t_{IVFH}	Vs rising edge of IVCLK	4			ns



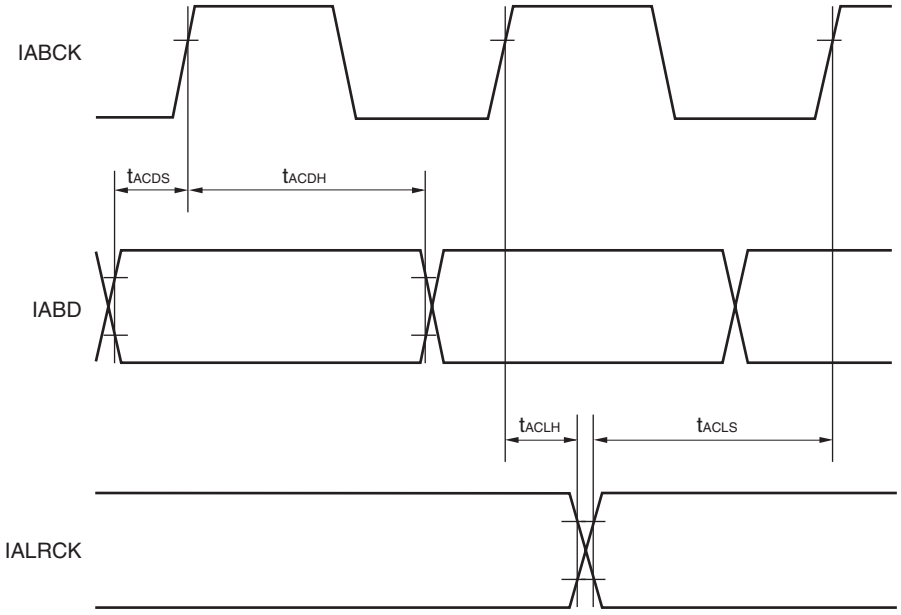
(3) Video output interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
OVCLK frequency	f_{OVCKS}			27		MHz
OVCLK high-level width	t_{OVCKH}		8			ns
OVCLK low-level width	t_{OVCKL}		8			ns
OVOUT7 to OVOUT0 hold time	t_{OVHO}	Vs rising edge of OVCLK	7			ns
OVOUT7 to OVOUT0 delay time	t_{OVDO}	Vs rising edge of OVCLK			28	ns
OVVSYNC hold time	t_{OVVHO}	Vs rising edge of OVCLK	7			ns
OVVSYNC delay time	t_{OVVD}	Vs rising edge of OVCLK			28	ns
OVHSYNC hold time	t_{OVHHO}	Vs rising edge of OVCLK	7			ns
OVHSYNC delay time	t_{OVHD}	Vs rising edge of OVCLK			28	ns



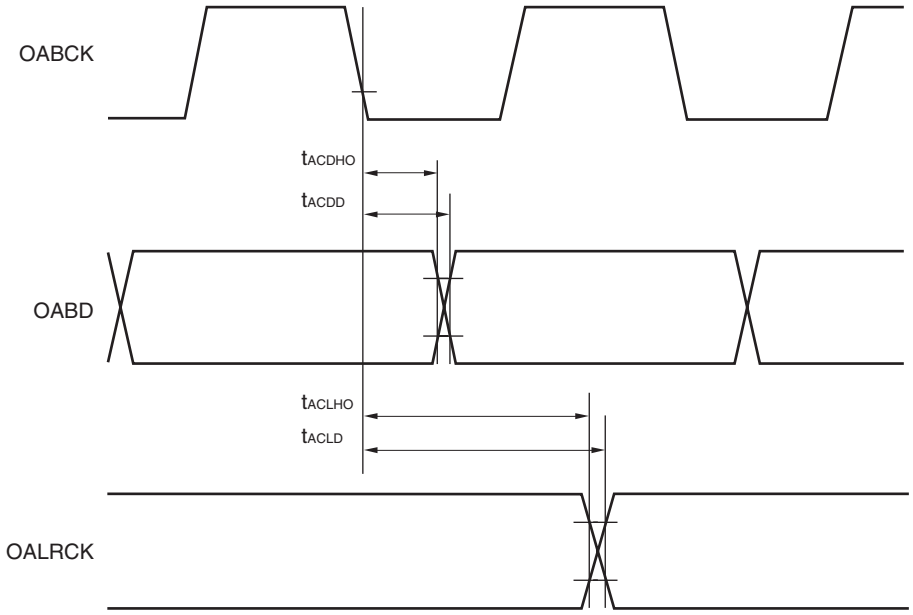
(4) Audio input interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Bit data-in setup time	t_{ACDS}	Vs IABCK	37			ns
Bit data-in hold time	t_{ACDH}	Vs IABCK	37			ns
LRCK-in setup time	t_{ACLS}	Vs IABCK	100			ns
LRCK-in hold time	t_{ACLH}	Vs IABCK	37			ns



(5) Audio output interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Bit data-out hold time	t_{ACDHO}	Vs OABCK	–5			ns
Bit data-out delay time	t_{ACDD}	Vs OABCK			25	ns
LRCK-out hold time	t_{ACLHO}	Vs OABCK	–5			ns
LRCK-out delay	t_{ACLD}	Vs OABCK			25	ns
BCK-out duty ratio	d_{BCK}			50		%
AMCLK duty ratio	d_{AMCLK}			50		%
AMCLK frequency	f_{AMCLK}				18.432	MHz



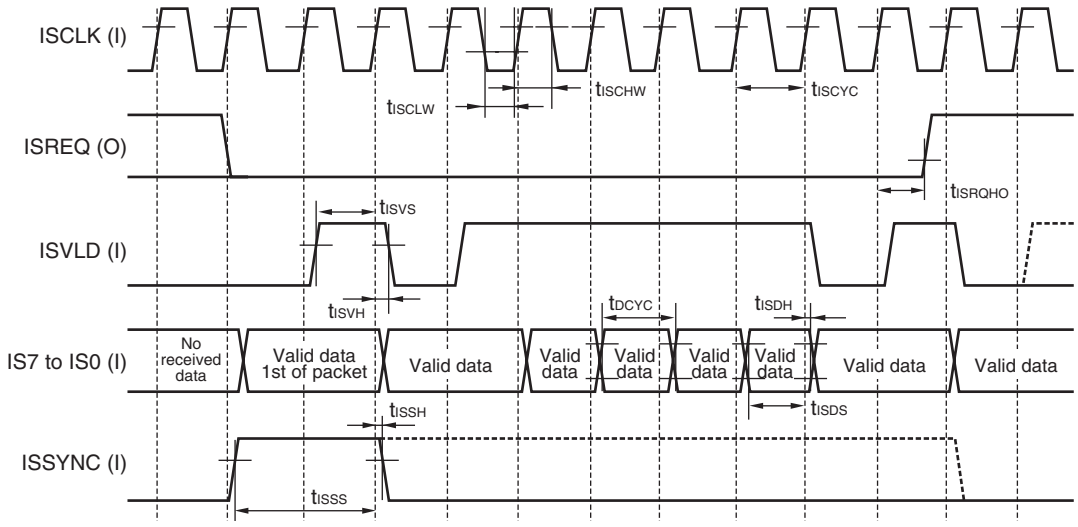
(6) Stream input interface

(a) Parallel stream input

Valid mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
ISCLK cycle	t_{ISCcyc}		80			ns
ISCLK low-level width	t_{ISCLW}		37			ns
ISCLK high-level width	t_{ISCHW}		37			ns
ISREQ output hold time	t_{ISRQHO}	Vs active edge of ISCLK	0			ns
ISVLD setup time	t_{ISVS}	Vs active edge of ISCLK	7			ns
ISVLD hold time	t_{ISVH}	Vs active edge of ISCLK	3			ns
ISSYNC setup time	t_{ISSS}	Vs active edge of ISCLK	7			ns
ISSYNC hold time	t_{ISSH}	Vs active edge of ISCLK	3			ns
IS7 to IS0 setup time	t_{ISDS}	Vs active edge of ISCLK	7			ns
IS7 to IS0 hold time	t_{ISDH}	Vs active edge of ISCLK	3			ns
Data cycle time	t_{DCYC}		80			ns

Remark ISREQ is effective only when it works by the master mode. ISREQ becomes invalid asynchronously to ISCLK. ISREQ output delay time doesn't prescribe to ISCLK.

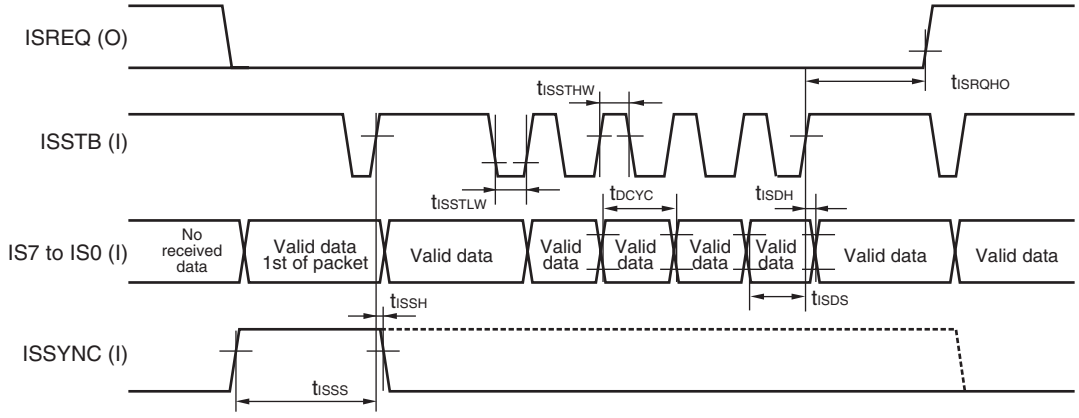


Remark ISSYNC is active high, SREQ is active high and ISCLK is active high edge.

Strobe mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
ISSTB low-level width	t_{ISSTLW}		37			ns
ISSTB high-level width	t_{ISSTHW}		37			ns
ISREQ output hold time	t_{ISRQHO}	Vs active edge of ISSTB	0			ns
ISSYNC setup time	t_{ISSS}	Vs active edge of ISSTB	7			ns
ISSYNC hold time	t_{ISSH}	Vs active edge of ISSTB	3			ns
IS7 to IS0 setup time	t_{ISDS}	Vs active edge of ISSTB	7			ns
IS7 to IS0 hold time	t_{ISDH}	Vs active edge of ISSTB	3			ns
Data cycle time	t_{DCYC}		80			ns

Remark ISREQ becomes invalid asynchronously to ISSTB. ISREQ output delay time doesn't prescribe to ISSTB.

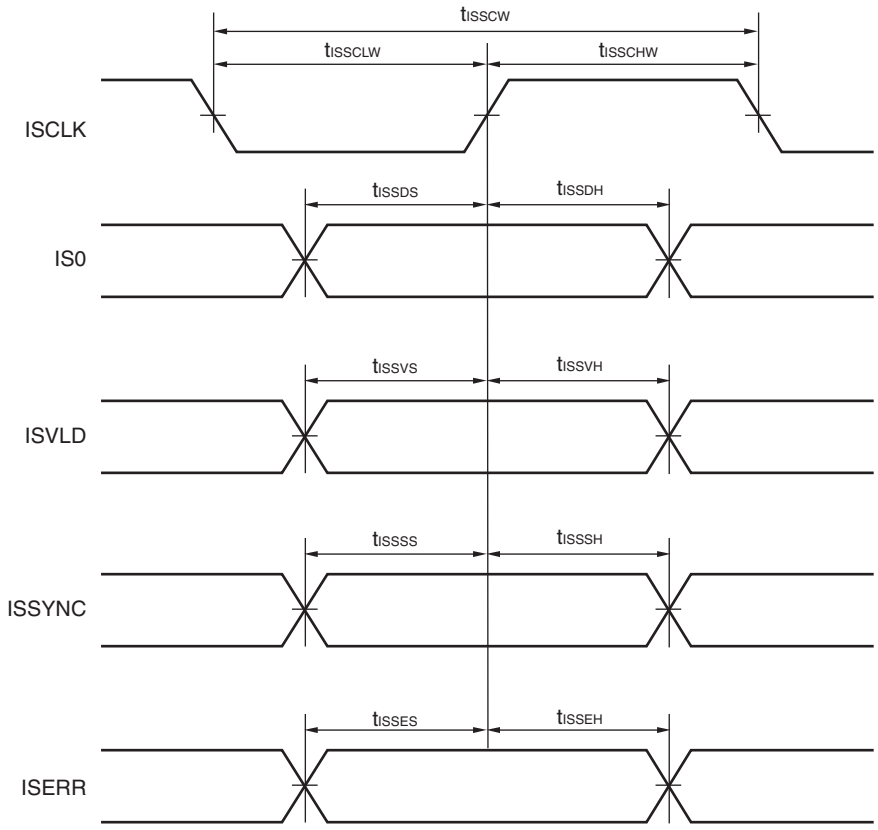


Remark ISSYNC is active high, ISREQ is active low and ISSTB is active high edge.

(b) Serial stream input

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
ISCLK period	t_{ISSCW}		15.6			ns
ISCLK low-level width	t_{ISSCLW}		5.0			ns
ISCLK high-level width	t_{ISSCHW}		5.0			ns
ISVLD setup time	t_{ISSVS}	Vs active edge of ISCLK	2.5			ns
ISVLD hold time	t_{ISSVH}	Vs active edge of ISCLK	2.5			ns
ISSYNC setup time	t_{ISSSS}	Vs active edge of ISCLK	2.5			ns
ISSYNC hold time	t_{ISSSH}	Vs active edge of ISCLK	2.5			ns
ISERR setup time	t_{ISSES}	Vs active edge of ISCLK	2.5			ns
ISERR hold time	t_{ISSEH}	Vs active edge of ISCLK	2.5			ns
IS0 setup time	t_{ISSDS}	Vs active edge of ISCLK	2.5			ns
IS0 hold time	t_{ISSDH}	Vs active edge of ISCLK	2.5			ns

Remark Setup and hold time provide to the activist edge of ISCLK.



Remark ISCLK is active high edge.

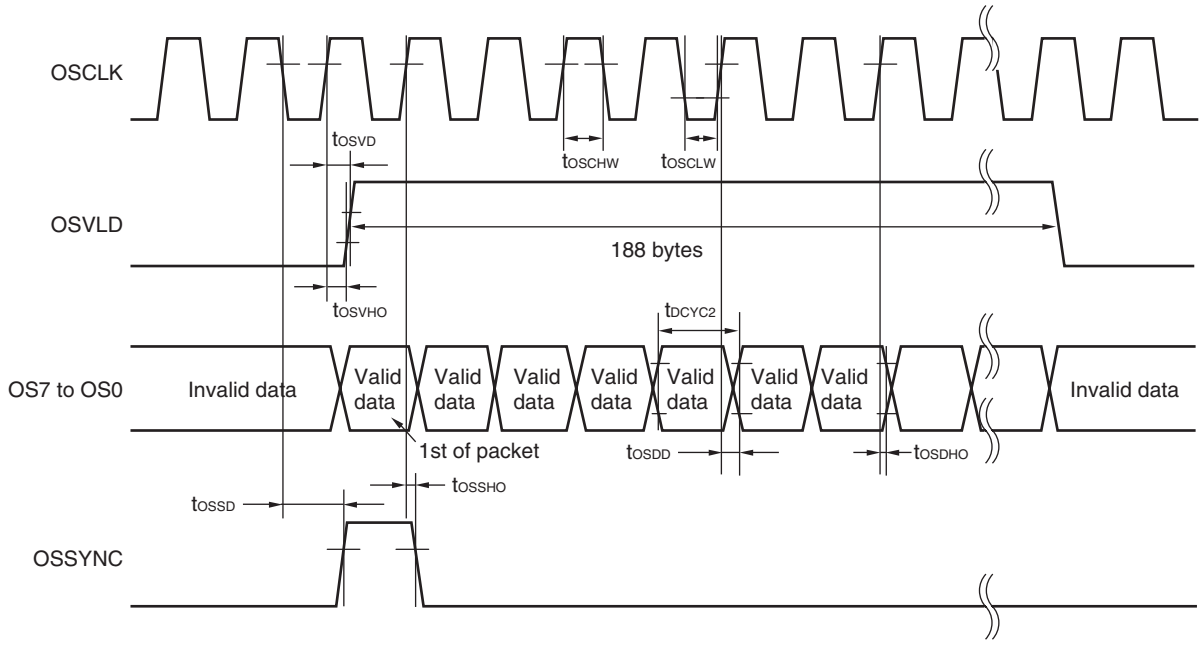
(7) Stream output interface

(a) Parallel stream data output

Valid and master mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
OSCLK low-level width	t_{OSCLW}	Active rising edge	30			ns
		Active falling edge	70			ns
OSCLK high-level width	t_{OSCHW}	Active rising edge	70			ns
		Active falling edge	30			ns
OSVLD hold time	t_{OSVHO}	Vs active edge of OSCLK	30			ns
OSVLD delay time	t_{OSVD}	Vs non active edge of OSCLK	–5		+5	ns
OSSYNC hold time	t_{OSSHO}	Vs active edge of OSCLK	30			ns
OSSYNC delay time	t_{OSSD}	Vs non active edge of OSCLK	–5		+5	ns
OS7 to OS0 hold time	t_{OSDHO}	Vs active edge of OSCLK	30			ns
OS7 to OS0 delay time	t_{OSDD}	Vs non active edge of OSCLK	–5		+5	ns
Data cycle time	t_{DCYC2}		105			ns

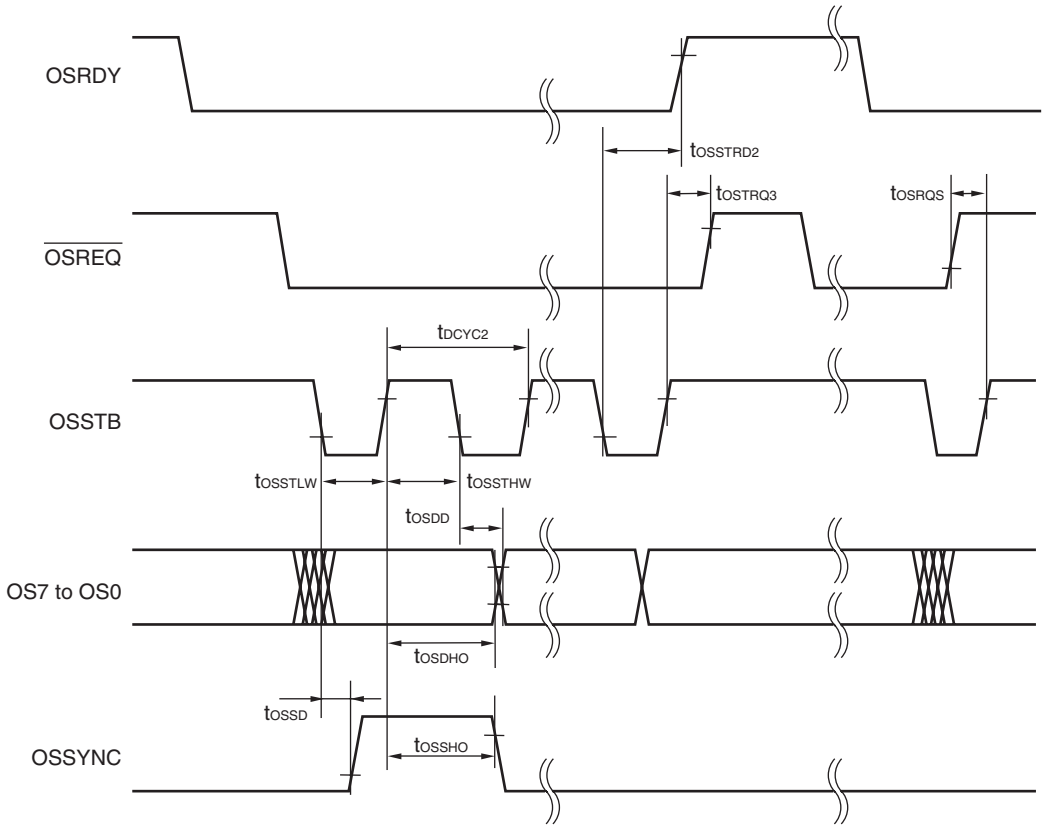
Remark OSVLD is active high, OSSYNC is active high and OSCLK is active high edge.



Strobe and burst mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
OSSTB low-level width	t_{OSSTLW}	Active rising edge	30			ns
		Active falling edge	70			ns
OSSTB high-level width	t_{OSSTHW}	Active rising edge	70			ns
		Active falling edge	30			ns
$\overline{\text{OSREQ}}$ setup time	t_{OSRQS}	Vs active edge of OSSTB			1	STCLK
$\overline{\text{OSREQ}}$ inactive delay time	t_{OSTRQ3}	Vs active edge of OSSTB			1	STCLK
OSRDY delay time	$t_{OSSTRD2}$	Vs non active edge of OSSTB			150	ns
OSSYNC hold time	t_{OSSHO}	Vs active edge of OSSTB	70			ns
OSSYNC delay time	t_{OSSD}	Vs non active edge of OSSTB	–5		+5	ns
OS7 to OS0 hold time	t_{OSDHO}	Vs active edge of OSSTB	70			ns
OS7 to OS0 delay time	t_{OSDD}	Vs non active edge of OSSTB	–5		+5	ns
Data cycle time	t_{DCYC2}		105			ns

Remark In the slave mode, it controls data output with $\overline{\text{OSREQ}}$ and in the master mode, it doesn't control in the output by $\overline{\text{OSREQ}}$.

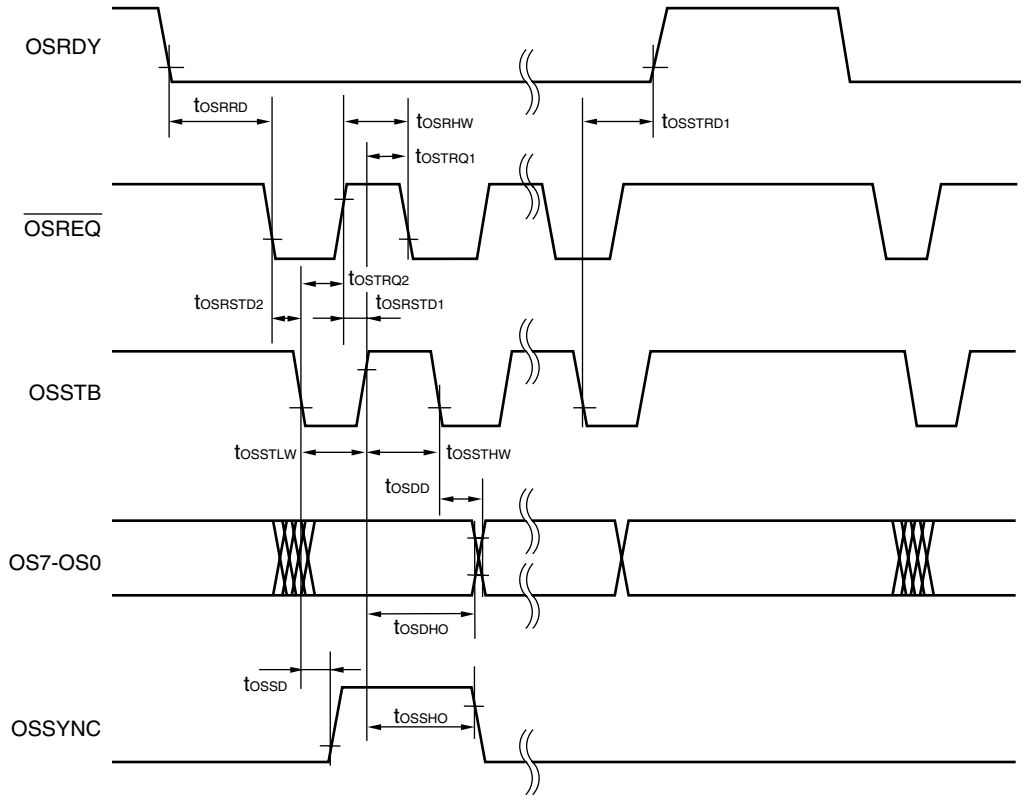


Remark OSSYNC is active high, OSRDY is active low and OSSTB is active high edge.

Strobe and byte mode

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{\text{OSREQ}}$ high-level time	t_{OSRHW}		2			STCLK
OSSTB high-level width	t_{OSSTHW}	Active rising edge	100			ns
		Active falling edge	70			ns
OSSTB low-level width	t_{OSSTLW}	Active rising edge	70			ns
		Active falling edge	100			ns
$\overline{\text{OSREQ}}$ hold time	t_{OSRRD}	Vs active edge of OSRDY	0			ns
$\overline{\text{OSREQ}}$ hold time	t_{OSTRQ1}	Vs active edge of OSSTB	0			ns
	t_{OSTRQ2}	Vs non active edge of OSSTB	0			ns
OSSTB delay time	$t_{OSRSTD1}$	Vs active edge of $\overline{\text{OSREQ}}$	2		3	STCLK
	$t_{OSRSTD2}$	Vs non active edge of $\overline{\text{OSREQ}}$	3			STCLK
OSRDY delay time	$t_{OSSTRD1}$	Vs non active edge of OSSTB			3	STCLK
OSSYNC-out delay time	t_{OSSD}	Vs non active edge of OSSTB	–5		+5	ns
OSSYNC-out hold time	t_{OSSHO}	Vs active edge of OSSTB	70			ns
OS7 to OS0 out delay time	t_{OSDD}	Vs non active edge of OSSTB	–5		+5	ns
OS7 to OS0 out hold time	t_{OSDHO}	Vs active edge of OSSTB	70			ns

★



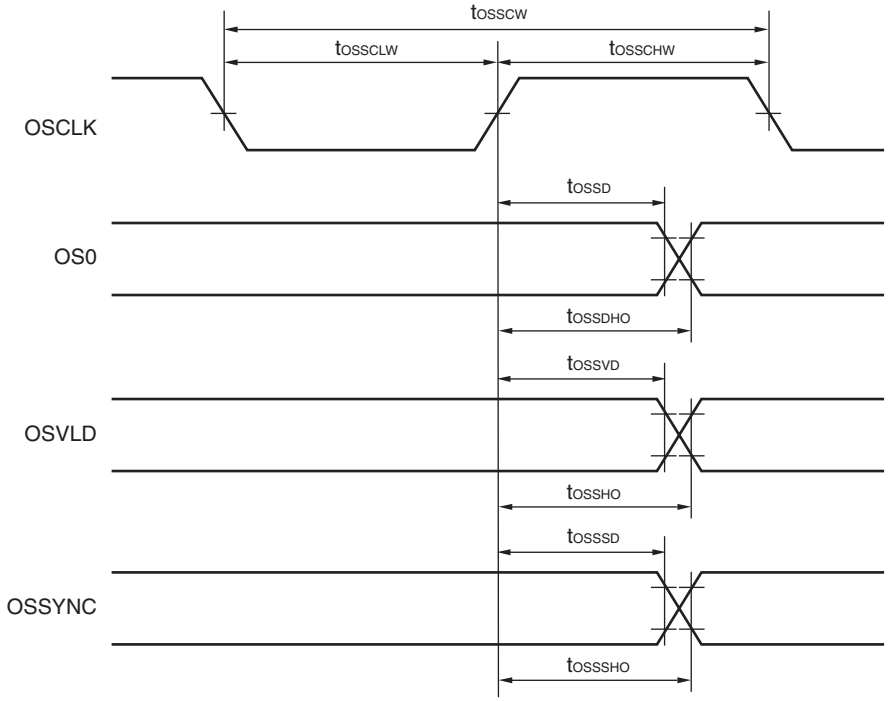
Remark OSSYNC is active high, OSRDY is active low and OSSTB is active high edge.

(b) Serial stream data output

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
OSCLK period	t_{OSSCW}			37		ns
OSCLK low-level width	t_{OSSCLW}		10			ns
OSCLK high-level width	t_{OSSCHW}		10			ns
OS0 delay time	t_{OSSDD}	Vs active edge of OSCLK			27	ns
OS0 hold time	t_{OSSDHO}	Vs active edge of OSCLK	5.0			ns
OSVLD delay time	t_{OSSVD}	Vs active edge of OSCLK			27	ns
OSVLD hold time	t_{OSSVHO}	Vs active edge of OSCLK	5.0			ns
OSSYNC delay time	t_{OSSSD}	Vs active edge of OSCLK			27	ns
OSSYNC hold time	t_{OSSSHO}	Vs active edge of OSCLK	5.0			ns

Remarks 1. Active edge of OSCLK is able to change according to the following circuit.

2. Period of the OSCLK is provided by STCLK.



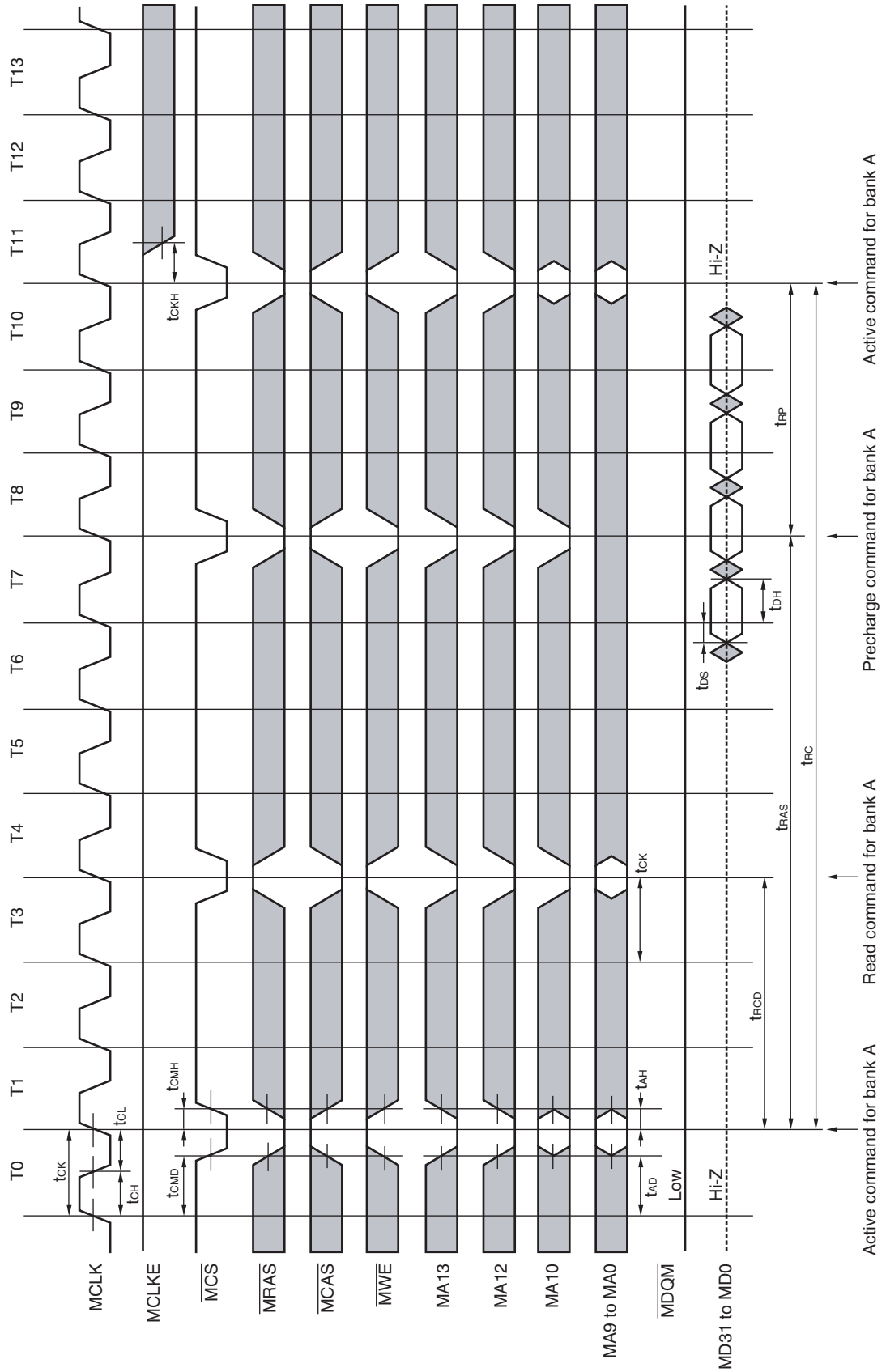
Remark OSCLK is active high edge.

(8) SDRAM interface

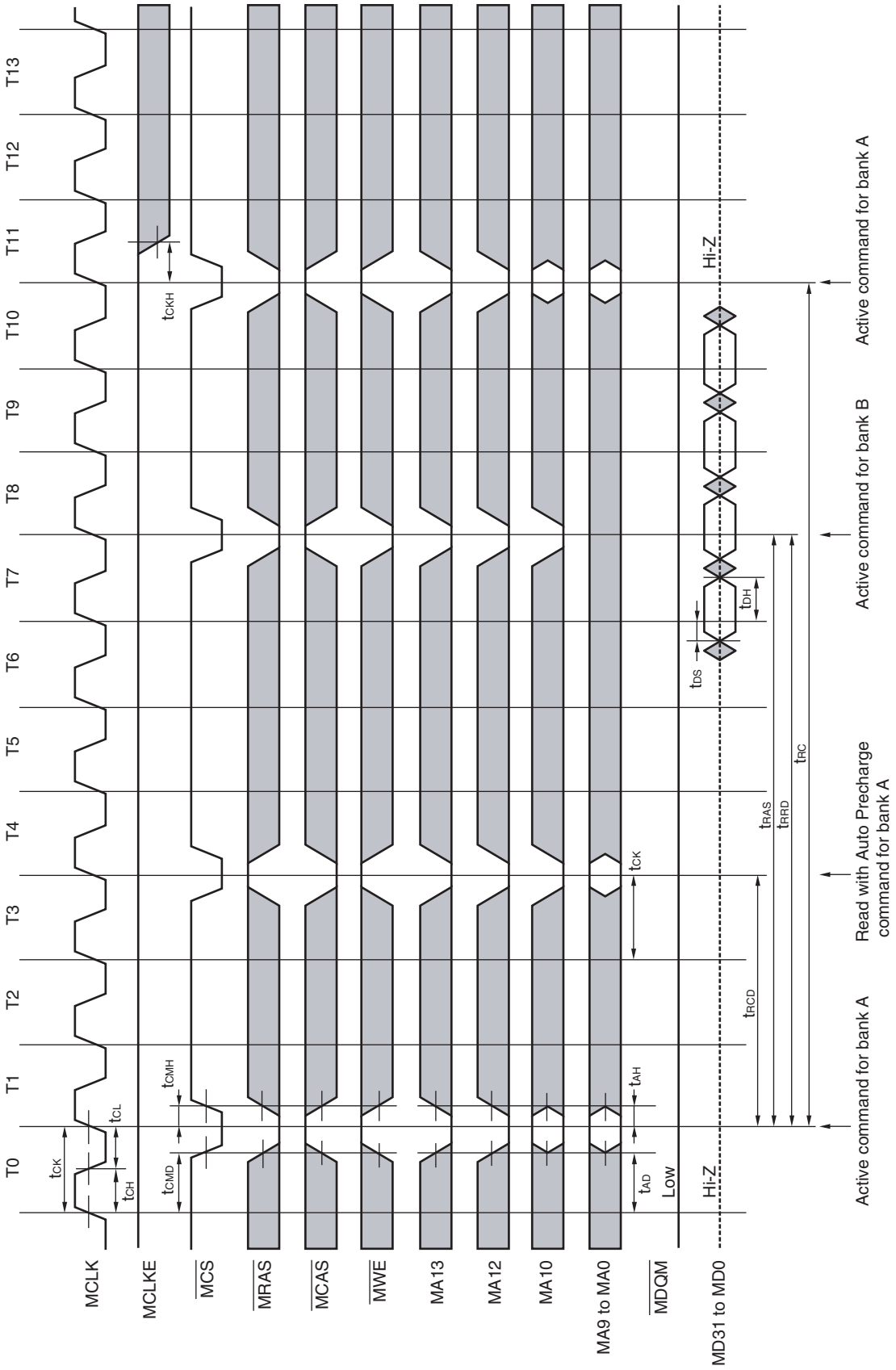
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
MCLK cycle time	t_{CK}			12.3		ns
MCLK high-level width	t_{CH}		3.5			ns
MCLK low-level width	t_{CL}		3.5			ns
MD31 to MD0-out hold time	t_{OH}	Vs MCLK	1.5			ns
MD31 to MD0-out delay time	t_{OD}	Vs MCLK			9	ns
MD31 to MD0 low-Z output time	t_{LZ}	Vs MCLK	0			ns
MD31 to MD0 high-Z output time	t_{HZ}	Vs MCLK			9	ns
MD31 to MD0-in setup time	t_{DS}	Vs MCLK	6			ns
MD31 to MD0-in hold time	t_{DH}	Vs MCLK	2			ns
MA13 to MA0 delay time	t_{AD}	Vs MCLK			9	ns
MA13 to MA0 hold time	t_{AH}	Vs MCLK	1.5			ns
MCLKE delay time	t_{CKS}	Vs MCLK			9	ns
MCLKE hold time	t_{CKH}	Vs MCLK	1.5			ns
Command delay time	t_{CMD}	Vs MCLK			9	ns
Command hold time	t_{CMH}	Vs MCLK	1.5			ns
ACT → REF/ACT command period	t_{RC}		12			MCLK
REF → REF/ACT command period	t_{RC1}		12			MCLK
ACT → PRE command period	t_{RAS}		12			MCLK
PRE → ACT command period	t_{RP}		12			MCLK
ACT → R/W command delay time	t_{RCD}		3			MCLK
ACT (0) → ACT (1) command period	t_{RRD}		4			MCLK
Data-in to PRE command period	t_{DPL}		2			MCLK
Data-in to ACT (REF) command period (Auto pre-charge)	t_{DAL}		6			MCLK
Mode register set cycle period	t_{RSC}		2			MCLK
Refresh Time (4096 refresh cycle)	t_{REF}				50	ms

Remark REF: Refresh, ACT: Active, PRE: Pre-charge

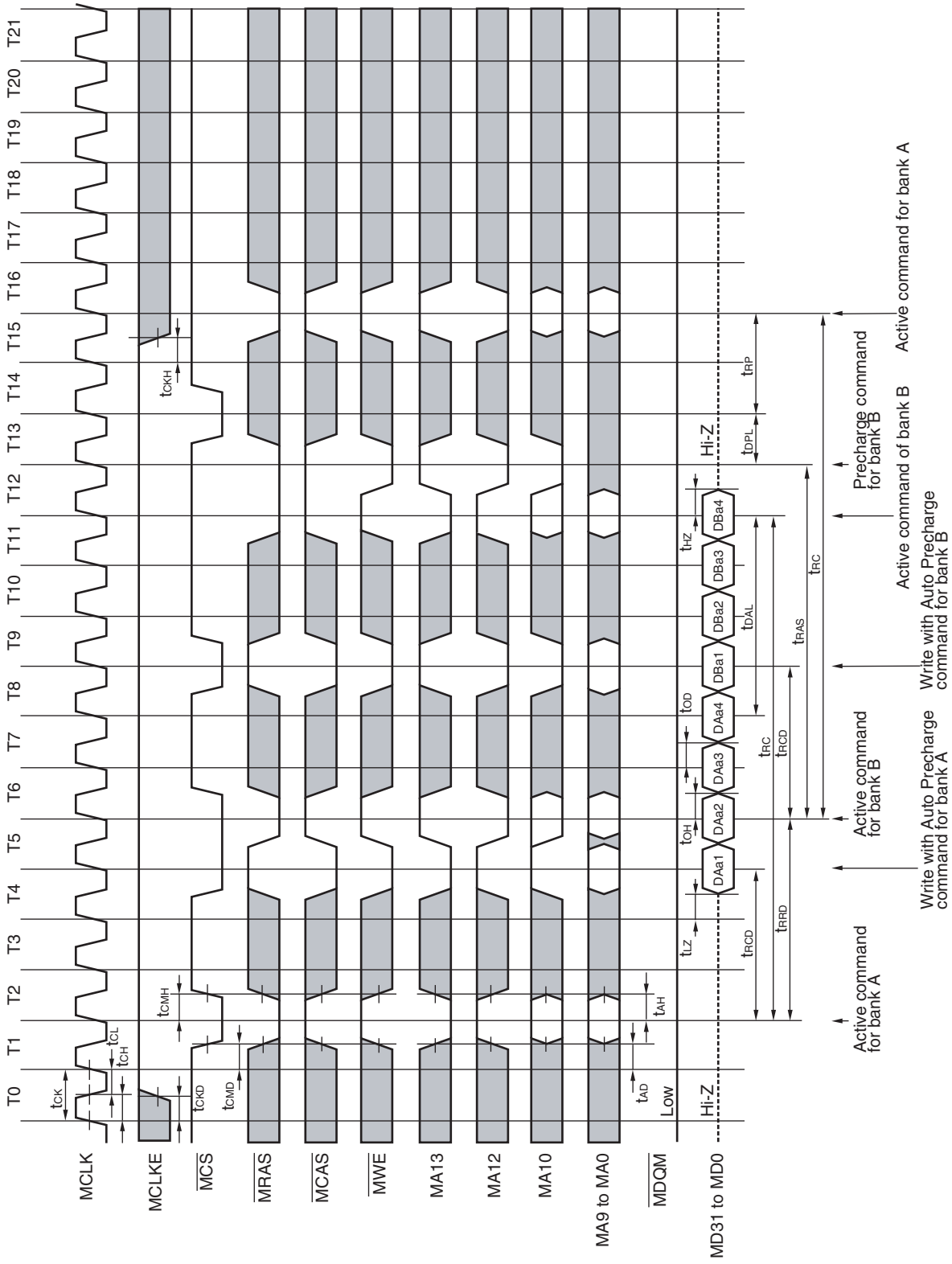
Read timing (Manual pre-charge, burst length = 4, CAS latency = 3)



Read timing (Auto pre-charge, burst length = 4, CAS latency = 3)



Write timing (Burst length = 4, CAS latency = 3)



(9) Host CPU interface

(a) Parallel bus interface: Wait mode

(1/2)

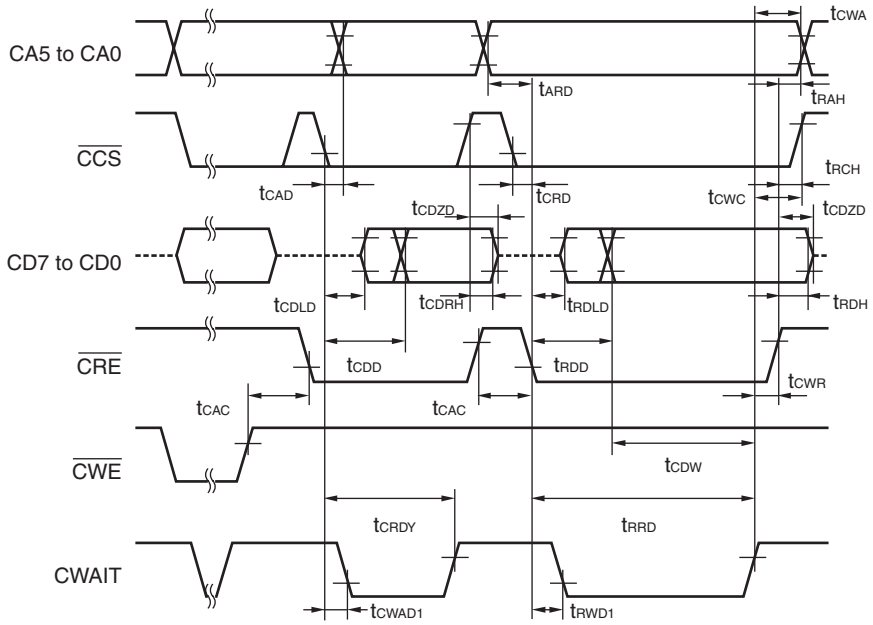
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CCS}\downarrow$ → CA5 to CA0 delay time	t_{CAD}	Vs falling edge of $\overline{CCS}$ Do not care	-	-	-	ns
$\overline{CCS}\downarrow$ → CWAIT delay time	t_{CWAD1}	Vs falling edge of $\overline{CCS}$ $\overline{CCS}$ later than $\overline{CRE}$/$\overline{CWE}$			15	ns
$\overline{CCS}\downarrow$→ CWAIT release time	t_{CRDY}	Vs falling edge of $\overline{CCS}$ $\overline{CCS}$ later than $\overline{CRE}$/$\overline{CWE}$			175	ns
CA5 to CA0 → $\overline{CRE}\downarrow$ delay time	t_{ARD}	Vs CA5 to CA0	–20			ns
$\overline{CCS}\downarrow$ → $\overline{CRE}\downarrow$ delay time	t_{CRD}	Vs falling edge of $\overline{CCS}$	–20			ns
$\overline{CRE}\downarrow$ → CWAIT delay time	t_{RWD1}	Vs falling edge of $\overline{CRE}$			15	ns
$\overline{CRE}\downarrow$ → CWAIT release time	t_{RRD}	Vs falling edge of $\overline{CRE}$			175	ns
$\overline{CCS}\downarrow$ → CD7 to CD0 low-Z time	t_{CDLD}	Vs falling edge of $\overline{CCS}$ Data not fixed	0			ns
$\overline{CRE}\downarrow$ → CD7 to CD0 low-Z time	t_{RDLD}	Vs falling edge of $\overline{CRE}$ Data not fixed	0			ns
$\overline{CCS}\downarrow$→ CD7 to CD0 delay time	t_{CDD}	Vs falling edge of $\overline{CCS}$ Data fixed			150	ns
$\overline{CRE}\downarrow$ → CD7 to CD0 delay time	t_{RDD}	Vs falling edge of $\overline{CRE}$ Data fixed			150	ns
$\overline{CRE}\uparrow$ → CD7 to CD0 hold time	t_{RDH}	Vs rising edge of $\overline{CRE}$ Earlier than rising edge of $\overline{CCS}$	0			ns
$\overline{CRE}\uparrow$ → CA5 to CA0 hold time	t_{RAH}	Vs rising edge of $\overline{CRE}$	–27			ns
$\overline{CRE}\uparrow$ → $\overline{CCS}\uparrow$ hold time	t_{RCH}	Vs rising edge of $\overline{CRE}$	–27			ns
$\overline{CCS}\uparrow$ → CD7 to CD0 hold time	t_{CDRH}	Vs rising edge of $\overline{CCS}$ Earlier than rising edge of $\overline{CRE}$	0			ns
CD7 to CD0 → CWAIT release time	t_{CDW}	Vs CD7 to CD0 fixed	10			ns
CD7 to CD0 Hi-Z delay time	t_{CDZD}	Vs rising edge of $\overline{CRE}$ or $\overline{CCS}$			12	ns
CA5 to CA0 → $\overline{CWE}\downarrow$ delay time	t_{AWD}	Vs CA5 to CA0	–28			ns
$\overline{CCS}\downarrow$ → $\overline{CWE}\downarrow$ delay time	t_{CWD}	Vs falling edge of $\overline{CCS}$	–20			ns
$\overline{CWE}\downarrow$ → CWAIT delay time	t_{WWD1}	Vs falling edge of $\overline{CWE}$			15	ns
$\overline{CWE}\downarrow$ → CWAIT release time	t_{WRD}	Vs falling edge of $\overline{CWE}$			150	ns
$\overline{CWE}\downarrow$ → CD7 to CD0 delay time	t_{WDD}	Vs falling edge of $\overline{CWE}$ Until data fixed			30	ns
$\overline{CWE}\uparrow$ → CD7 to CD0 hold time	t_{WDH}	Vs rising edge of $\overline{CWE}$	–7			ns

(2/2)

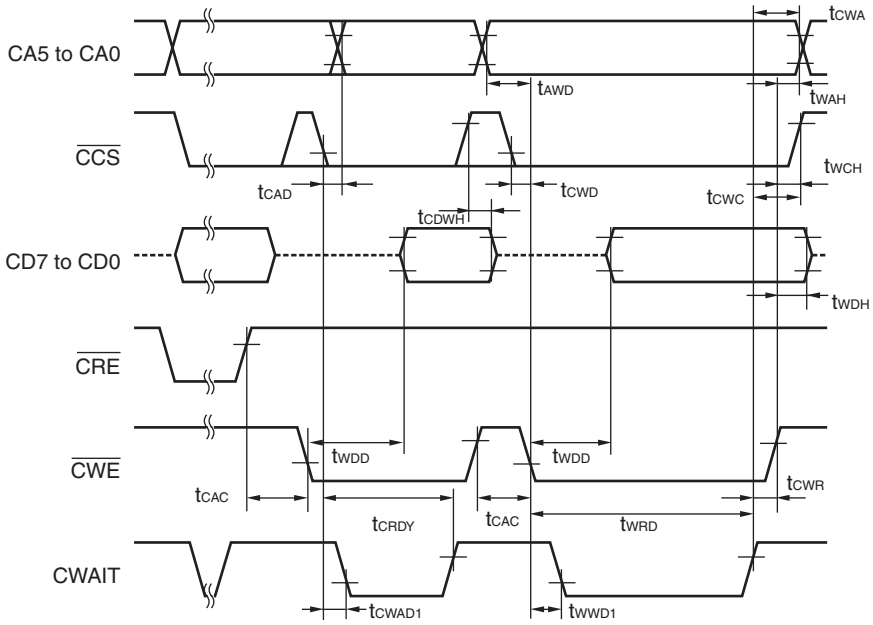
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CWE}\uparrow \rightarrow$ CA5 to CA0 hold time	t_{WAH}	Vs rising edge of $\overline{CWE}$	−27			ns
$\overline{CWE}\uparrow \rightarrow \overline{CCS}\uparrow$ hold time	t_{WCH}	Vs rising edge of $\overline{CWE}$	−27			ns
$\overline{CCS}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{CDWH}	Vs rising edge of $\overline{CCS}$	0			ns
$\overline{CCS}\uparrow \rightarrow$ CWAIT release time	t_{CWAD2}	Vs rising edge of $\overline{CCS}$	0		15	ns
CWAIT release $\rightarrow \overline{CWE}/\overline{CRE}$ hold time	t_{CWR}	Vs CWAIT release	0			ns
CWAIT release $\rightarrow$ CD5 to CD0 hold time	t_{CWA}	Vs CWAIT release	0			ns
CWAIT release $\rightarrow \overline{CSS}\uparrow$ hold time	t_{CWC}	Vs CWAIT release	0			ns
$\overline{CRE}/\overline{CWE}$ recovery time	t_{CAC}		25			ns
Access cycle after other device	t_{CCYC}		200			ns

Remark If $\overline{CCS}$ change to "H" in wait cycle, it cancels CWAIT. In access time, don't make $\overline{CCS}$ "H" until wait released.

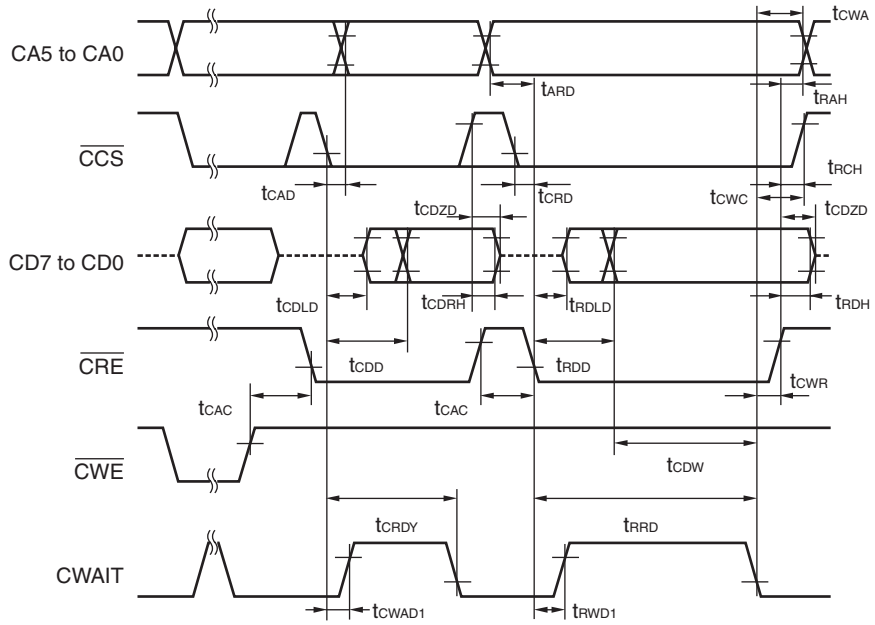
Wait mode (Wait active low, read cycle)



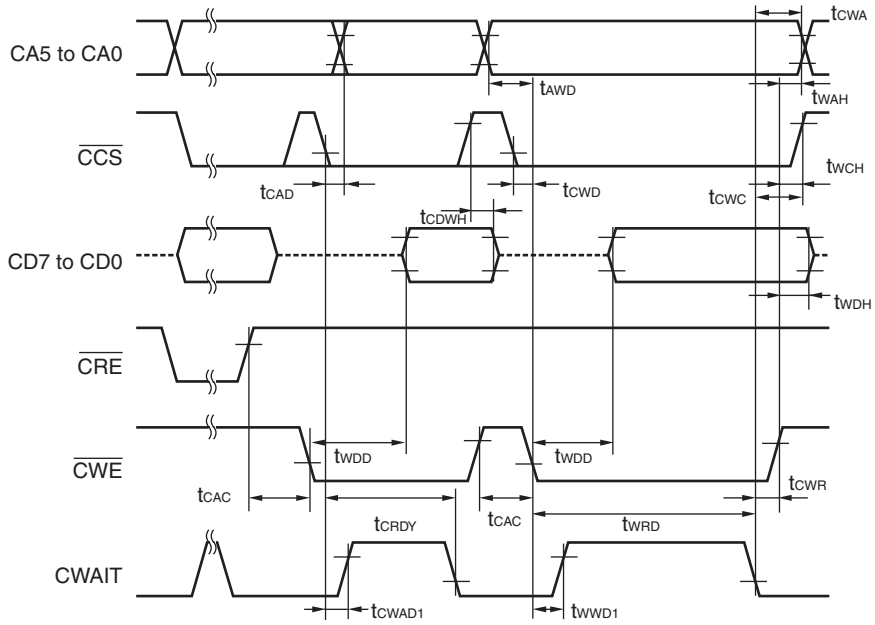
Wait mode (Wait active low, write cycle)

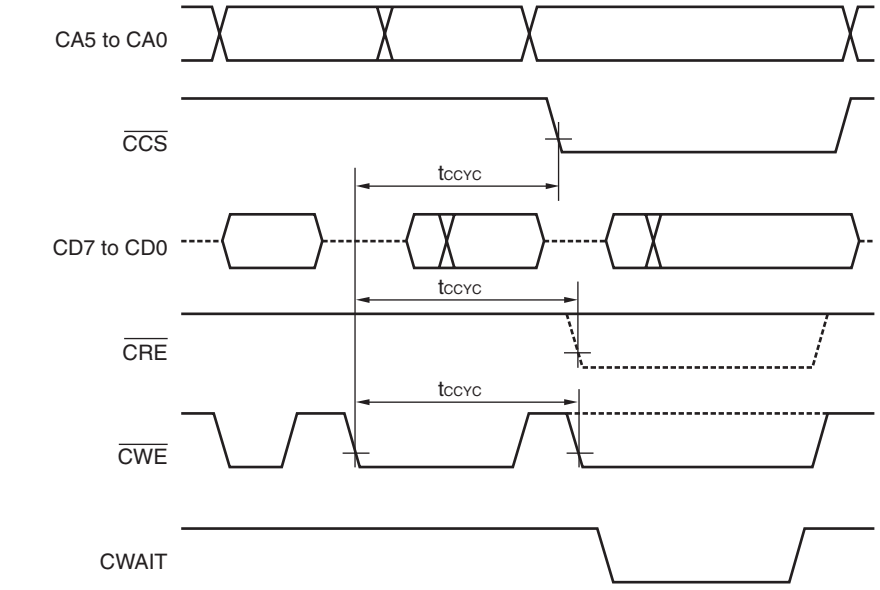


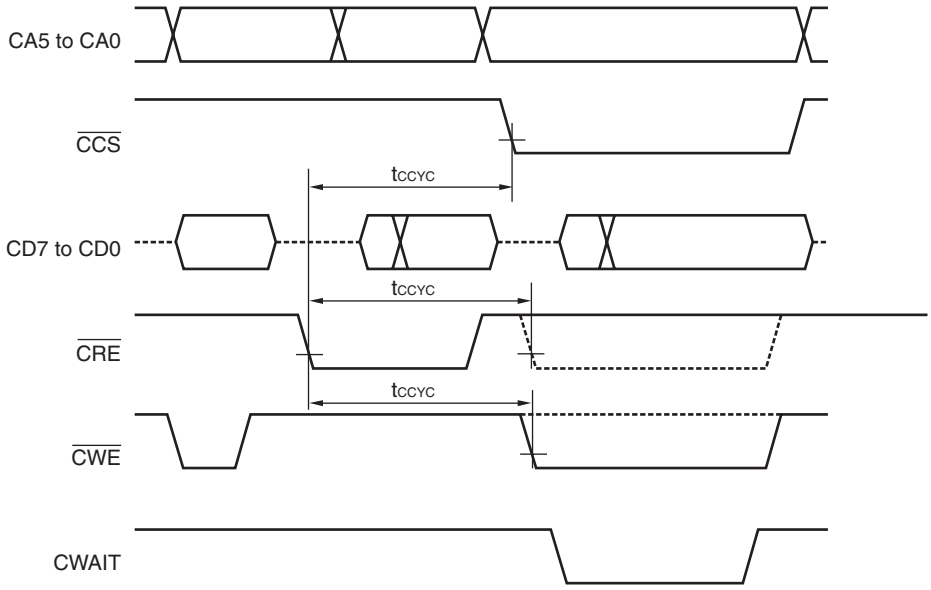
Wait mode (Wait active high, read cycle)



Wait mode (Wait active high, write cycle)




CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE
CWAIT


CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE
CWAIT

(b) Parallel bus interface: Ready mode

(1/2)

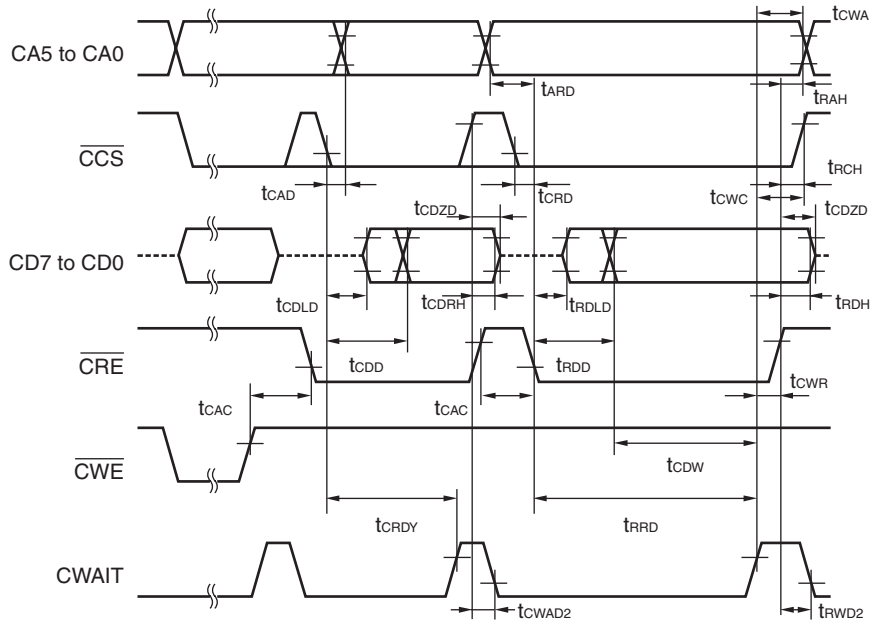
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CCS}\downarrow \rightarrow$ CA5 to CA0 delay time	t_{CAD}	Vs falling edge of $\overline{CCS}$ Do not care	-	-	-	ns
$\overline{CCS}\downarrow \rightarrow$ CWAIT delay time	t_{CWAD1}	Vs falling edge of $\overline{CCS}$ $\overline{CCS}$ later than $\overline{CRE}$/$\overline{CWE}$			15	
$\overline{CCS}\downarrow \rightarrow$ CWAIT ready time	t_{CRDY}	Vs falling edge of $\overline{CCS}$ $\overline{CCS}$ later than $\overline{CRE}$/$\overline{CWE}$			175	ns
CA5 to CA0 $\rightarrow \overline{CRE}\downarrow$ delay time	t_{ARD}	Vs CA5 to CA0	−20			ns
$\overline{CCS}\downarrow \rightarrow \overline{CRE}\downarrow$ delay time	t_{CRD}	Vs falling edge of $\overline{CCS}$	−20			ns
$\overline{CRE}\downarrow \rightarrow$ CWAIT ready time	t_{RRD}	Vs falling edge of $\overline{CRE}$			175	ns
$\overline{CCS}\downarrow \rightarrow$ CD7 to CD0 low-Z time	t_{CDLD}	Vs falling edge of $\overline{CCS}$ Data not fixed	0			ns
$\overline{CRE}\downarrow \rightarrow$ CD7 to CD0 low-Z time	t_{RDLD}	Vs falling edge of $\overline{CRE}$ Data not fixed	0			ns
$\overline{CCS}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{CDD}	Vs falling edge of $\overline{CCS}$ Data fixed			150	ns
$\overline{CRE}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{RDD}	Vs falling edge of $\overline{CRE}$ Data fixed			150	ns
$\overline{CRE}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{RDH}	Vs rising edge of $\overline{CRE}$ Earlier than rising edge of $\overline{CCS}$	0			ns
$\overline{CRE}\uparrow \rightarrow$ CA5 to CA0 hold time	t_{RAH}	Vs rising edge of $\overline{CRE}$	−27			ns
$\overline{CRE}\uparrow \rightarrow \overline{CCS}\uparrow$ hold time	t_{RCH}	Vs rising edge of $\overline{CRE}$	−27			ns
$\overline{CCS}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{CDRH}	Vs rising edge of $\overline{CCS}$ Earlier than rising edge of $\overline{CRE}$	0			ns
CD7 to CD0 $\rightarrow$ CWAIT ready time	t_{CDW}	Vs CD7 to CD0 fixed	10			ns
CD7 to CD0 high-Z delay time	t_{CDZD}	Vs rising edge of $\overline{CRE}$ or $\overline{CCS}$			12	ns
CA5 to CA0 $\rightarrow \overline{CWE}\downarrow$ delay time	t_{AWD}	Vs CA5 to CA0	−28			ns
$\overline{CCS}\downarrow \rightarrow \overline{CWE}\downarrow$ delay time	t_{CWD}	Vs falling edge of $\overline{CCS}$	−20			ns
$\overline{CWE}\downarrow \rightarrow$ CWAIT ready time	t_{WRD}	Vs falling edge of $\overline{CWE}$			150	ns
$\overline{CWE}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{WDD}	Vs falling edge of $\overline{CWE}$ Until data fixed			30	ns
$\overline{CWE}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{WDH}	Vs rising edge of $\overline{CWE}$	−7			ns
$\overline{CWE}\uparrow \rightarrow$ CA5 to CA0 hold time	t_{WAH}	Vs rising edge of $\overline{CWE}$	−27			ns
$\overline{CWE}\uparrow \rightarrow \overline{CCS}\uparrow$ hold time	t_{WCH}	Vs rising edge of $\overline{CWE}$	−27			ns
$\overline{CCS}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{CDWH}	Vs rising edge of $\overline{CCS}$	0			ns
$\overline{CRE}\uparrow \rightarrow$ CWAIT release time	t_{RWD2}	Vs rising edge of $\overline{CRE}$	0		15	ns

(2/2)

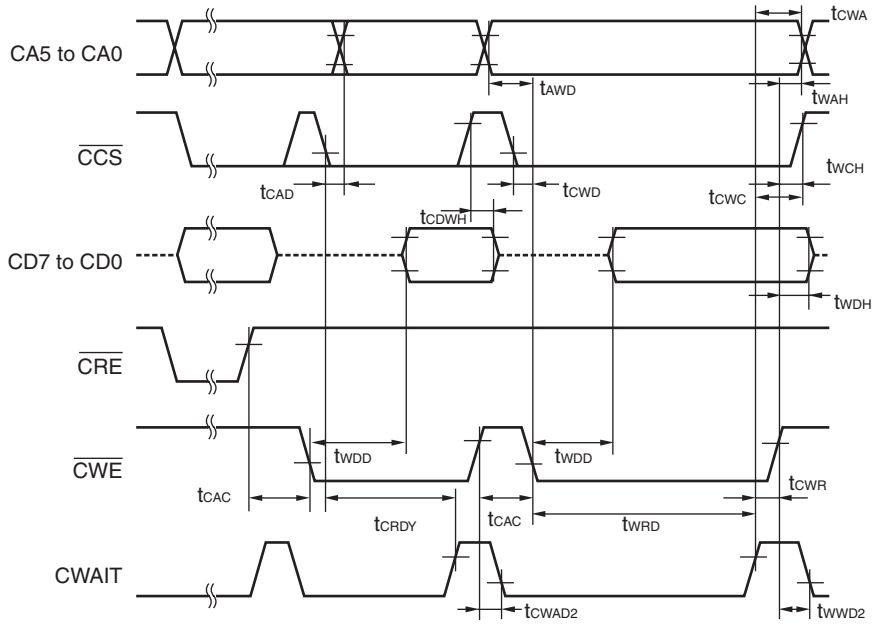
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CWE}\uparrow \rightarrow$ CWAIT release time	t_{WWD2}	Vs rising edge of $\overline{CWE}$	0		15	ns
$\overline{CCS}\uparrow \rightarrow$ CWAIT release time	t_{CWAD2}	Vs rising edge of $\overline{CCS}$	0		15	ns
CWAIT ready $\rightarrow \overline{CWE}/\overline{CRE}$ hold time	t_{CWR}	Vs CWAIT ready	0			ns
CWAIT ready $\rightarrow$ CA5 to CA0 hold time	t_{CWA}	Vs CWAIT ready	0			ns
CWAIT ready $\rightarrow \overline{CCS}\uparrow$ hold time	t_{CWC}	Vs CWAIT ready	0			ns
$\overline{CRE}/\overline{CWE}$ recovery time	t_{CAC}		25			ns
Access cycle after other device	t_{CCYC}		200			ns

Remark If $\overline{CCS}$ change to "H" in wait cycle, it cancels CWAIT. In access time, don't make $\overline{CCS}$ "H" until wait becomes ready.

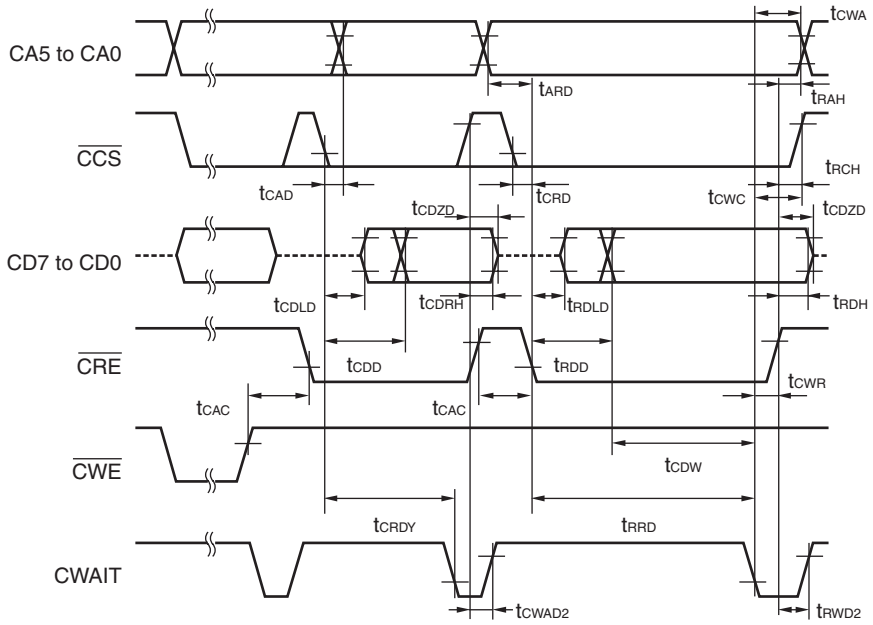
Ready mode (Ready active high, read cycle)



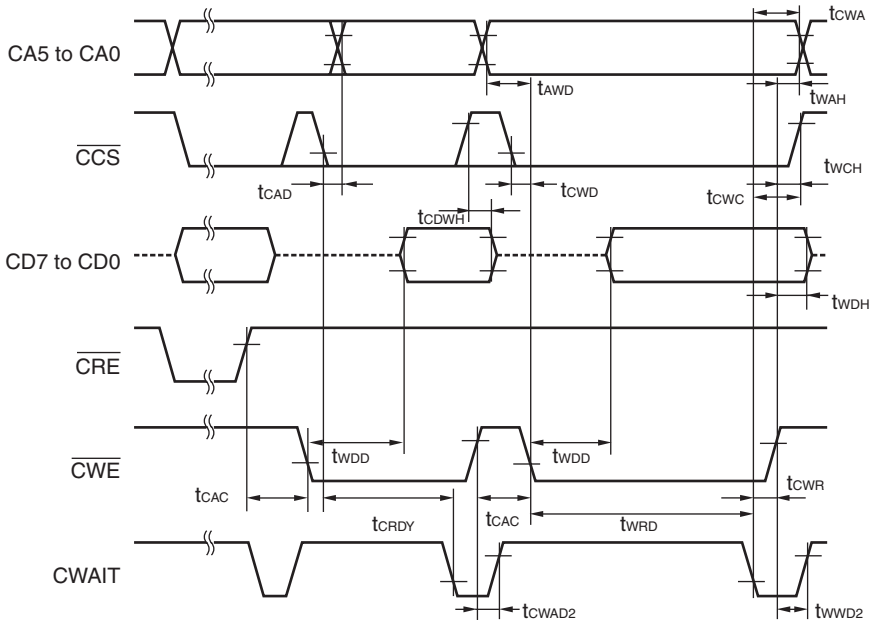
Ready mode (Ready active high, write cycle)

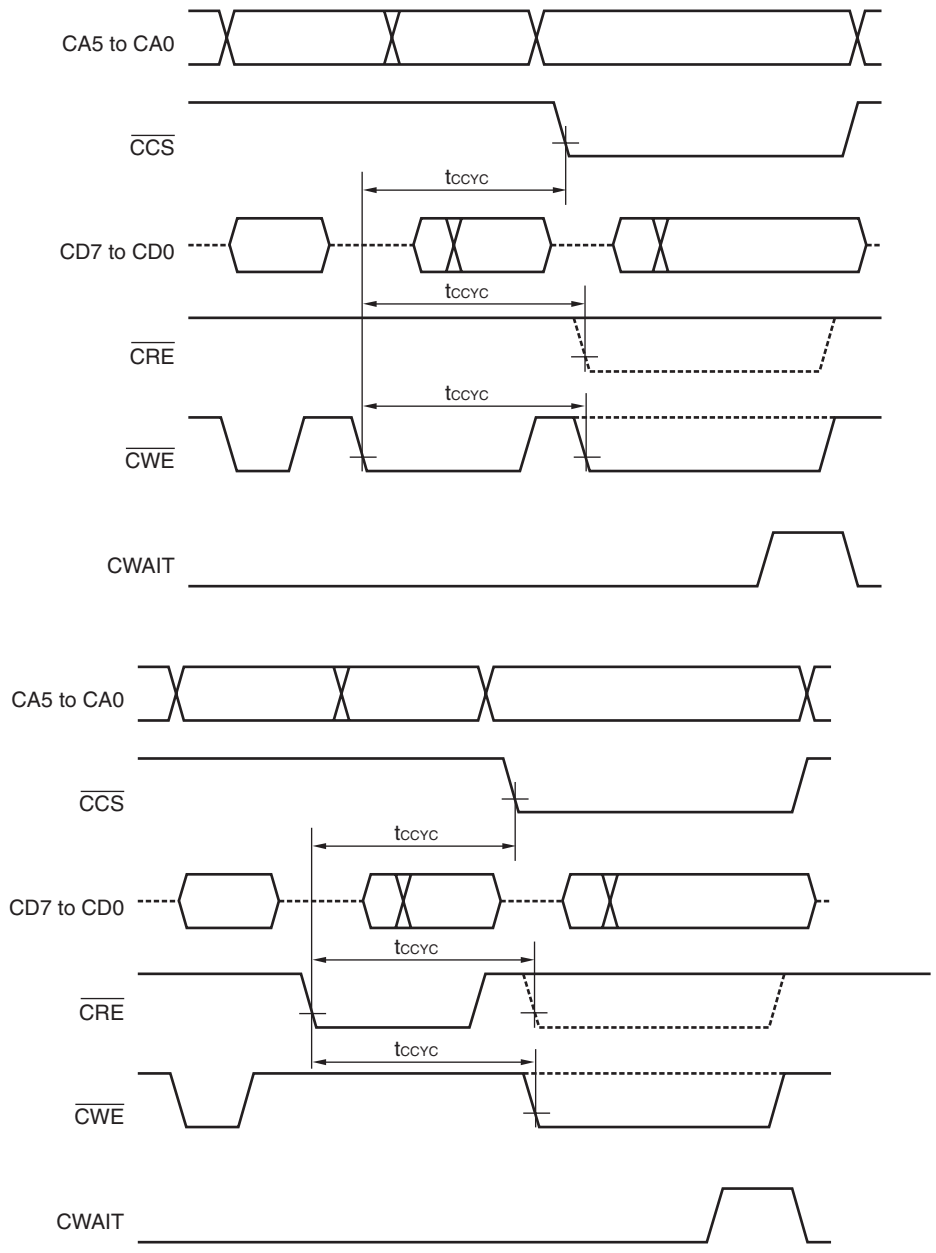


Ready mode (Ready active low, read cycle)



Ready mode (Ready active low, write cycle)

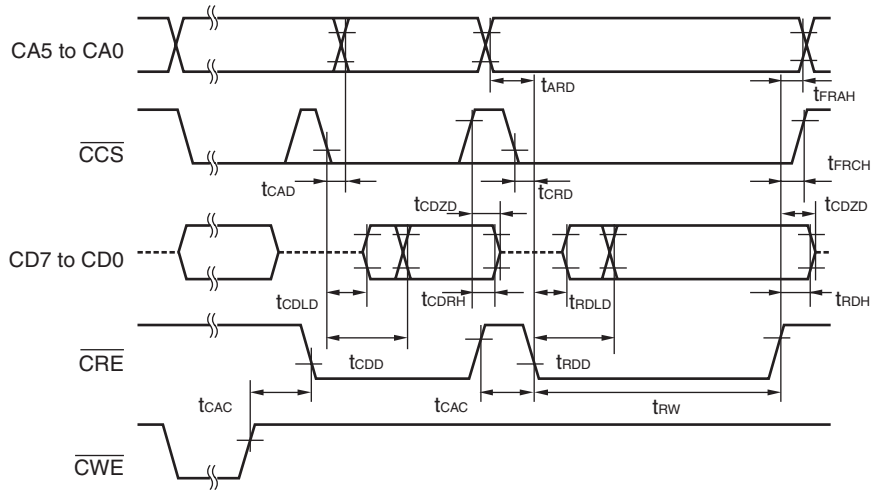



CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE
CWAIT
CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE
CWAIT

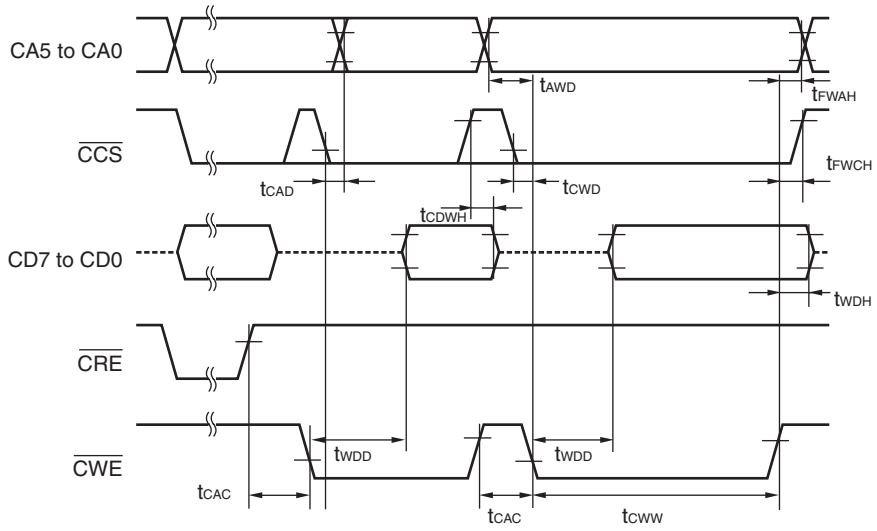
(c) Parallel bus interface: Fixed wait mode

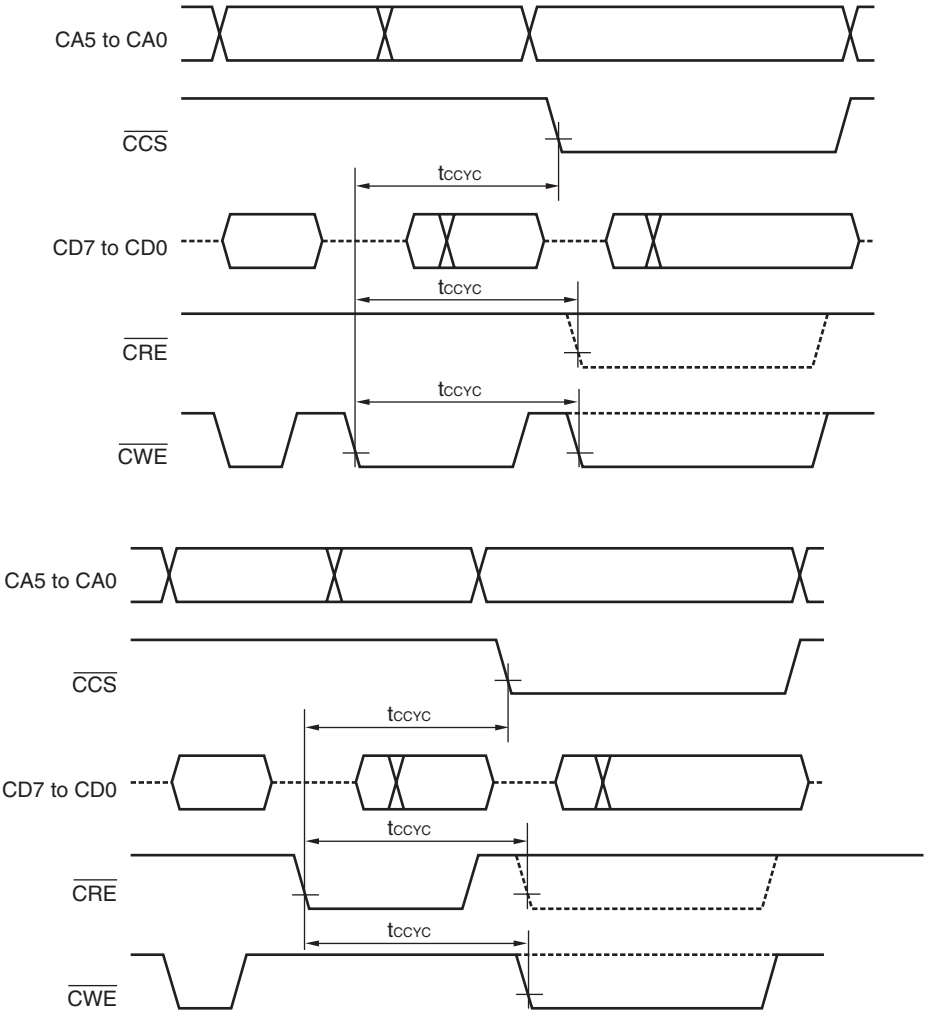
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CCS}\downarrow \rightarrow$ CA5 to CA0 delay time	t_{CAD}	Vs falling edge of $\overline{CCS}$ Do not care	-	-	-	ns
$\overline{CRE}$ pulse width	t_{RW}		175			ns
CA5 to CA0 $\rightarrow \overline{CRE}\downarrow$ delay time	t_{ARD}	Vs CA5 to CA0	–20			ns
$\overline{CCS}\downarrow \rightarrow \overline{CRE}\downarrow$delay time	t_{CRD}	Vs falling edge of $\overline{CCS}$	–20			ns
$\overline{CCS}\downarrow \rightarrow$ CD7 to CD0 low-Z time	t_{CDLD}	Vs falling edge of $\overline{CCS}$ Data not fixed	0			ns
$\overline{CRE}\downarrow \rightarrow$ CD7 to CD0 low-Z time	t_{RDLD}	Vs falling edge of $\overline{CRE}$ Data not fixed	0			ns
$\overline{CCS}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{CDD}	Vs falling edge of $\overline{CCS}$ Data fixed			150	ns
$\overline{CRE}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{RDD}	Vs falling edge of $\overline{CRE}$ Data fixed			150	ns
$\overline{CRE}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{RDH}	Vs rising edge of $\overline{CRE}$ Earlier than rising edge of $\overline{CCS}$	0			ns
$\overline{CRE}\uparrow \rightarrow$ CA5 to CA0 hold time	t_{FRAH}	Vs rising edge of $\overline{CRE}$	–27			ns
$\overline{CRE}\uparrow \rightarrow \overline{CCS}\uparrow$ hold time	t_{FRCH}	Vs rising edge of $\overline{CRE}$	–27			ns
$\overline{CCS}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{CDRH}	Vs rising edge of $\overline{CCS}$	0			ns
CD7 to CD0 high-Z delay time	t_{CDZD}	Vs rising edge of $\overline{CRE}$ or $\overline{CCS}$			12	ns
$\overline{CWE}$ pulse width	t_{WW}		150			ns
CA5 to CA0 $\rightarrow \overline{CWE}\downarrow$ delay time	t_{AWD}	Vs CA5 to CA0	–28			ns
$\overline{CCS}\downarrow \rightarrow \overline{CWE}\downarrow$ delay time	t_{CWD}	Vs falling edge of $\overline{CCS}$	–20			ns
$\overline{CWE}\downarrow \rightarrow$ CD7 to CD0 delay time	t_{WDD}	Vs falling edge of $\overline{CWE}$ Until data fixed			30	ns
$\overline{CWE}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{WDH}	Vs rising edge of $\overline{CWE}$	–7			ns
$\overline{CWE}\uparrow \rightarrow$ CA5 to CA0 hold time	t_{FWAH}	Vs rising edge of $\overline{CWE}$	–27			ns
$\overline{CWE}\uparrow \rightarrow \overline{CCS}\uparrow$ hold time	t_{FWCH}	Vs rising edge of $\overline{CWE}$	–27			ns
$\overline{CCS}\uparrow \rightarrow$ CD7 to CD0 hold time	t_{CDWH}	Vs rising edge of $\overline{CCS}$	0			ns
$\overline{CRE}/\overline{CWE}$ recovery time	t_{CAC}		25			ns
Access cycle after other device	t_{CCYC}		200			ns

Fixed wait mode (Read cycle)



Fixed wait mode (Write cycle)

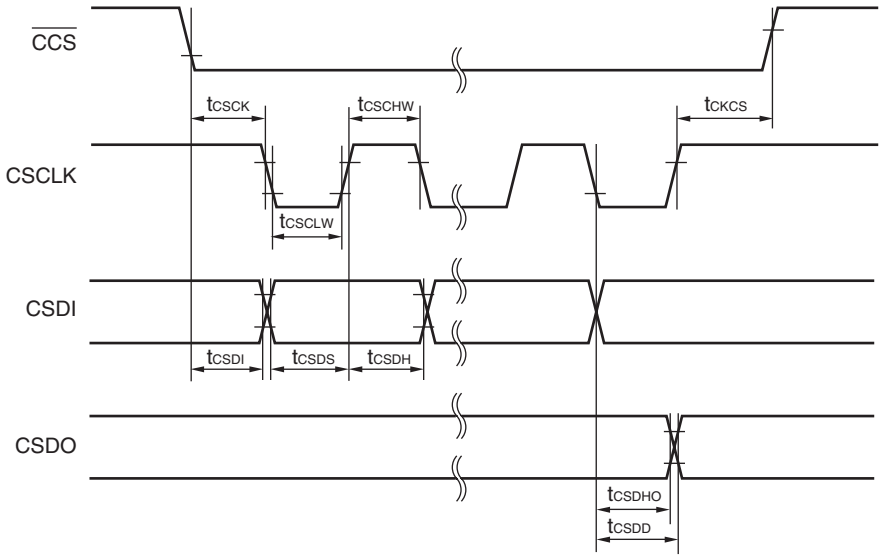


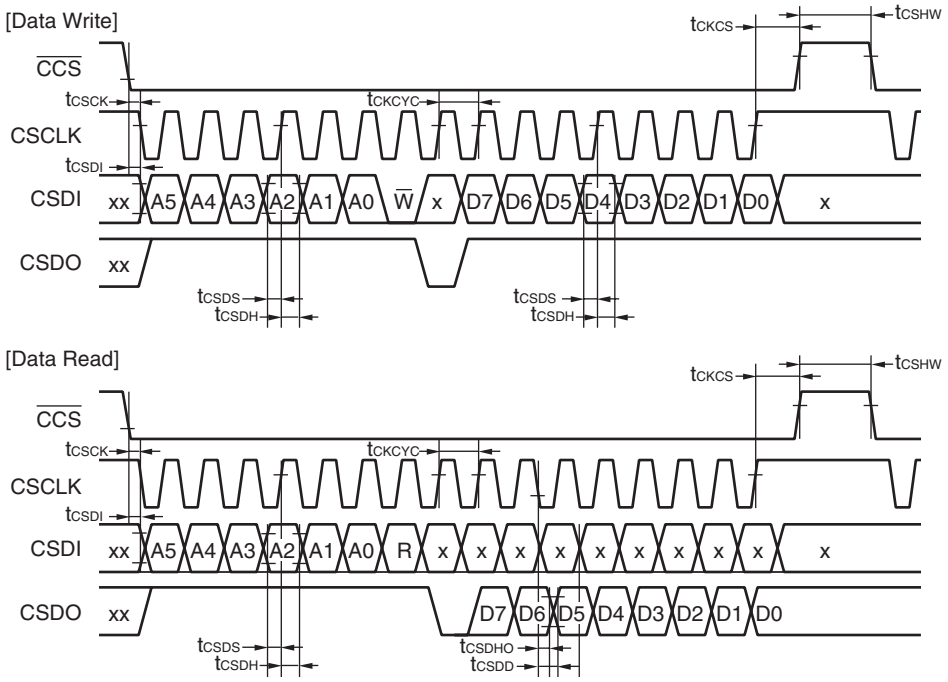

CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE
CA5 to CA0
CCS
tCCYC
CD7 to CD0
tCCYC
CRE
tCCYC
CWE

(10) Serial bus interface

(a) Serial bus interface

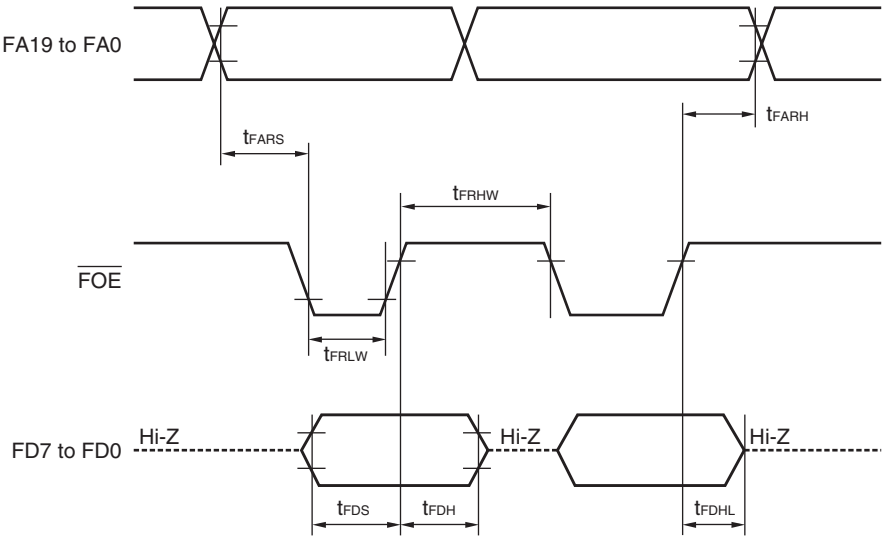
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{CCS}$ → CSCLK delay time	t_{CSCK}	Vs falling edge of $\overline{CCS}$	10			ns
$\overline{CCS}$ → CSDI delay time	t_{CSDI}	Vs falling edge of $\overline{CCS}$	10			ns
CSDI setup time	t_{CSDS}	Vs rising edge of CSCLK	10			ns
CSDI hold time	t_{CSDH}	Vs rising edge of CSCLK	10			ns
CSDO hold time	t_{CSDHO}	Vs falling edge of CSCLK	0			ns
CSDO delay time	t_{CSDD}	Vs falling edge of CSCLK			15	ns
CSCLK → $\overline{CCS}$ hold time	t_{CCKS}	Vs rising edge of CSCLK	75			ns
$\overline{CCS}$ high-level width	t_{CSHW}		125			ns
CSCLK cycle time	t_{CKCYC}		100			ns
CSCLK high-level width	t_{CSCHW}		40			ns
CSCLK high-level width	t_{CSCLW}		40			ns





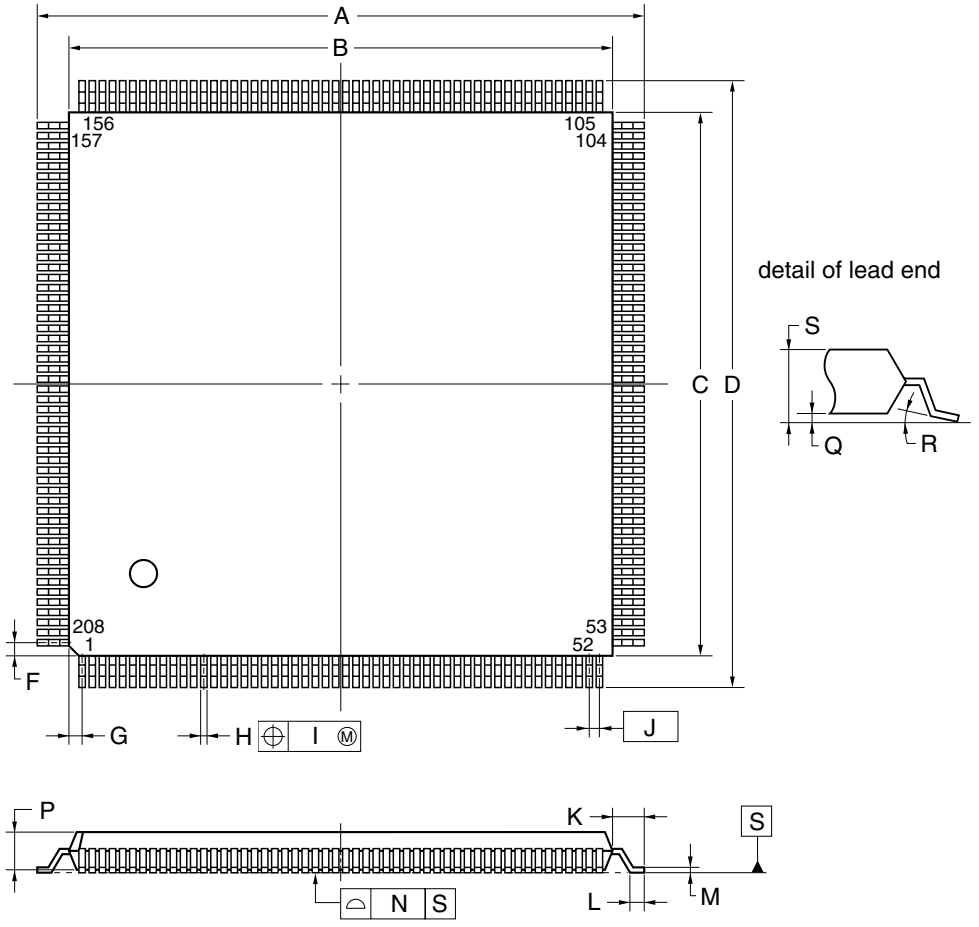
(b) Instruction ROM interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Address setup time	t_{FARS}	Vs falling edge of $\overline{FOE}$	0			ns
Address hold time	t_{FARH}	Vs rising edge of $\overline{FOE}$	5			ns
$\overline{FOE}$ low-level width	t_{FRLW}		70		225	ns
$\overline{FOE}$ high-level width	t_{FRHW}		24			ns
Data setup time	t_{FDS}	Vs rising edge of $\overline{FOE}$	25			ns
Data hold time	t_{FDH}	Vs rising edge of $\overline{FOE}$	0			ns
Data high-Z output time	t_{FDHL}	Vs rising edge of $\overline{FOE}$			60	ns



7. PACKAGE DRAWING

208-PIN PLASTIC QFP (FINE PITCH) (28x28)



NOTE

Each lead centerline is located within 0.10 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	30.6±0.2
B	28.0±0.2
C	28.0±0.2
D	30.6±0.2
F	1.25
G	1.25
H	$0.22^{+0.05}_{-0.04}$
I	0.10
J	0.5 (T.P.)
K	1.3±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	3.2±0.1
Q	0.4±0.1
R	5°±5°
S	3.8 MAX.

P208GD-50-LML,MML,SML,WML-7

8. RECOMMENDED SOLDERING CONDITIONS

The μPD61051, 61052 should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

★ Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 8-1. Surface-Mounted Soldering Conditions

μPD61051GD-LML: 208-pin plastic QFP (Fine pitch)(28×28)
μPD61052GD-LML: 208-pin plastic QFP (Fine pitch)(28×28)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C Time: 30 sec. max. (at 210°C or higher) Count: Three times or fewer Exposure limit: 7 days Note (After that, prebake at 125°C for 20 hours)	IR35-207-3
VPS	Package peak temperature: 215°C Time: 40 sec. max. (at 200°C or higher) Count: Three times or fewer Exposure limit: 7 days Note (After that, prebake at 125°C for 20 hours)	VP15-207-3
Partial heating	Pin temperature: 300°C max. Time: 3 sec. max. (per pin row)	—

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use two or more soldering methods in combination (except for partial heating method).

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.

[MEMO]

Dolby is a trademark of Dolby Laboratories.

- **The information in this document is current as of April, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product 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": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).