TX System RISC

32bit RISC Microprocessor
TX39 family

TMPR3904F

Rev.2.0

Jan. 12. 1998

1 INTRODUCTION

1.1 Overview

The TMPR3904F (to be called "TX3904" hereinafter) is a standard micro controller of the 32-bit RISC Microprocessor TX39 family.

The TX3904 uses the TX39 Processor Core as the CPU. The TX39 Processor Core is a RISC CPU core Toshiba developed based on the R3000A architecture of MIPS Group, a division of Silicon Graphics, Inc. ("MIPS").

As micro-controllers that can be embedded, besides the TX39 Processor Core, the TX3904 has built-in peripheral circuits such as memory controllers, DMA controllers, serial ports, and timers/counters.

As for the architecture of the TX39 Processor Core such as the instruction set, please refer to the following document:

32-bit RISC Microprocessor, TX39 Family: Users' Manual

(Toshiba Material Number: 4623-B)

R3000A is a trademark of MIPS Technologies, Inc.

1.2 Notation used in this manual

Mathematical notation

Hexadecimal numbers are expressed as follows (example shown for decimal number 42)
0x2A

A K(kilo) byte is 2^{10} = 1,024 bytes, a M(mega) byte is 2^{20} = 1,024 x 1,024 = 1,048,576 bytes, and a G(giga) byte is 2^{30} = 1,024 x 1,024 x 1,024 = 1,073,741,824 bytes.

Data notation

Byte: eight bits
Halfword: two contiguous bytes (16 bits)

Word: four contiguous bytes (32 bits)

Doubleword: eight contiguous bytes (64 bits)

Signal notation

Low active signals are indicated by an asterisk (*) at the end of the signal name (e.g.: RESET*).
Changing a signal to active level is to "assert" a signal, while changing it to a non-active level is to "de-assert" the signal.

1.3 Kind of accessing by the TX3904

Byte access

An access with a unit of eight bits. The BE* signal shows 0111, 1011, 1101 or 1110 in 32-bit bus mode, and 1101 or 1110 in 16-bit bus mode.

Halfword access

An access with a unit of 16 bits. The BE* signal shows 0011 or 1100 in 32-bit bus mode, and 1100 in 16-bit mode.

Triple bytes access

An access with a unit of 24 bits. The BE* signal shows 0001 or 1000 in 32-bit bus mode. In 16-bit bus mode, one logical access is divided by two physical accesses with BE*=1100/1101 or 1100/1110.

Word access

An access with a unit of 32 bits. The BE* signal shows 0000 in 32-bit bus mode. In 16-bit bus mode, one logical access is divided by two physical accesses with BE*=1100/1100.

1.4 Precautions in the TMPR3904F specification

Little endian

The TMPR3904F doesn't support little endian mode. Don't set a Big bit as little endian mode in the channel control register of DMAC.

Bus error

In the TMPR3904F, if a BUSERR* is asserted during write operation, bus control signals (BSTART*, LAST*, R/W*) malfunction and a BEOW bit in a chip configuration register doesn't indicate bus error. Don't use this function.

Time out error

In the TMPR3904F, if time out error occurs during write operation, an NMI is generated but a BEOW bit in a chip configuration register doesn't indicate bus error.

Halt mode

A store instruction which sets a halt bit in the configuration register of the TX39 processor core must be placed on an address whose low four bit is 0x8. If not, the TMPR3904F may not return to the normal mode correctly. Refer to the section 6.6.2 Halt mode for more detail.

Burst mode access by the DRAMC and ROMC

The bus master issues a burst starting address for the first word, then the DRAMC/ROMC generates the successive address for the second word and after. The DRAMC/ROMC increments address like as 0–4–8–C–10–... if the lower four–bit of burst starting address is 0x0. On the other hand, it decrements address like as C–8–4–0–... if the starting address is 0xC.

SIO external clock

In the TMPR3904F, an external clock SCLKIN can't be used for SIO boudot rate generator.

SIO stop bit

In the TMPR3904F, if two-stop bit is set as a transmitter parameter, the receiver can't detect the next start bit during 1.5 bits time period after detecting a stop bit. Therefore a transmitter as a counter part of the receiver of the TMPR3904F SIO must insert an interval of 1.5 bits or more between every characters. If not, the SIO have little margin to boudot rate variance.

Bus ownership of external bus master requests

The TX39 processor core has four prioritized bus request inputs; HPSREQ, HPGREQ, SREQ, and GREQ. Do not use the HPGREQ and GREQ because the processor core has problem in its bus arbitration circuit. In the TMPR3904F, a bus request from an external bus master is designated by the POBus bit in the chip configuration register. A bus request of on-chip DMAC is designated by the Sreq bit in the channel control register.

2 FEATURES

- Built-in TX39 Processor Core
  Toshiba has uniquely developed this on the basis of the R3000A architecture of the MIPS.
  Instruction cache 4KB/Data cache 1KB
  Built-in debug support unit
- DRAM Controller
  Four-bank x two-channel configuration
  Fast page mode/Hyper page (EDO) mode support
- ROM Controller
  Two-bank x two-channel configuration
  Mask ROM, EPROM, E^2PROM, Flash ROM, SRAM support
  Page mode ROM support
- DMA Controller
  Independent four channels
  Single address mode/Dual address mode
- Interrupt Controller
  Maskable interrupt: internal nine sources, external eight sources
  Non-maskable interrupt
- Timer/Counter
  24-bit up counter three channels
  Watchdog timer mode support
- Serial I/O
  Two-channel UART
- I/O Ports
  Exclusive port: one channel; shared port: two channels
- 16-bit Bus Support

- Power Supply Voltage: 3.3V
- Power Consumption: T.B.D. (3.3V, at 50 MHz operation, typ.)
- Operation Frequency: 33 – 50 MHz
- Package: 208 pin plastic QFP

3 CONFIGURATION

The following is the block diagram of the TX3904:

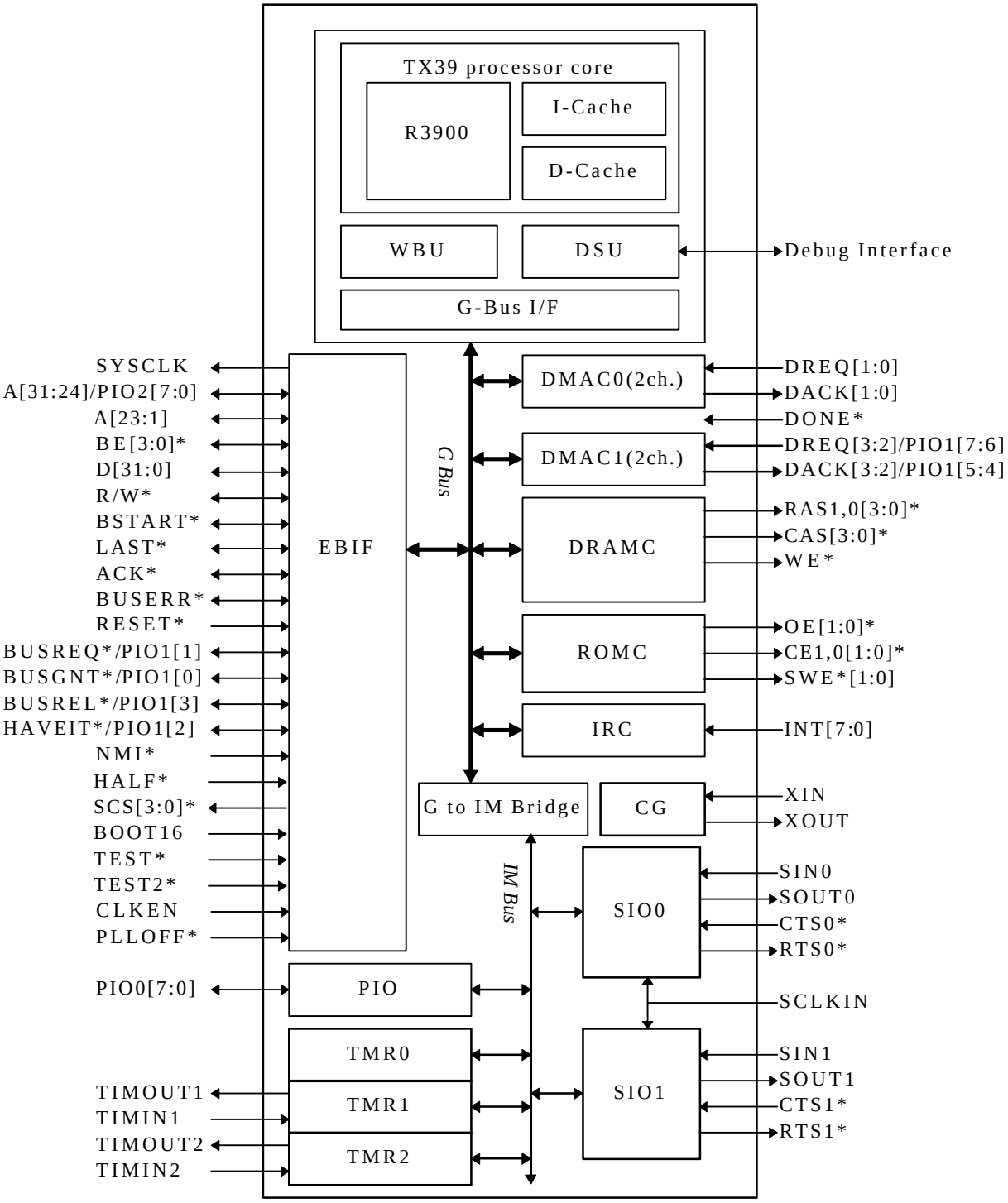


Fig. 3-1 TX3904 Block Diagram

WBU: Write Buffer

Four-store write buffer. It stores data output from the TX39 Processor Core.

DSU: Debug Support Unit

Realizes the debug function of the TX39 Processor Core.

DMAC1/DMAC0: DMA Controllers

Conducts data transfers in place of the TX39 Processor Core.

DRAMC: DRAM Controller

Controls DRAM access.

ROMC: ROM Controller

Controls Mask ROM, EPROM, E^2ROM, FLASH ROM, and SRAM access.

IRC: Interrupt Controller

Arbitrates interrupt requests from inside and outside the TX3904.

EBIF: External Bus Interface

Controls the bus operation of the TX3904.

CG: Clock Generator

Generates the TX3904's internal clock from external crystal oscillators.

SIO1/SIO0: Serial I/O's

UART.

PIO: I/O Port

8-bit port.

TMR2/TMR1/TMR0: Timers/Counters

24-bit up counters. TMR2 supports the watchdog timer function.

4 PINS

4.1 Positions of Pins

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	XIN	31	TOUT[3]	61	VDD	91	DACK[0]
2	VDD	32	WE*	62	A[22]	92	DACK[1]
3	LAST*	33	A[1]	63	A[23]	93	DACK[2]
4	R/W*	34	VDD	64	A[24]	94	DACK[3]
5	BE[3]*	35	A[2]	65	A[25]	95	DONE*
6	BE[2]*	36	A[3]	66	A[26]	96	VSS
7	BE[1]*	37	A[4]	67	A[27]	97	VDD
8	BE[0]*	38	VSS	68	VSS	98	DREQ[0]
9	VSS	39	A[5]	69	VDD	99	DREQ[1]
10	VDD	40	A[6]	70	A[28]	100	DREQ[2]
11	RAS0[0]*	41	A[7]	71	A[29]	101	DREQ[3]
12	RAS0[1]*	42	A[8]	72	A[30]	102	HAVEIT*
13	RAS0[2]*	43	VDD	73	A[31]	103	BUSREQ*
14	VDD	44	VSS	74	CE0[0]*	104	VDD
15	VSS	45	A[9]	75	CE0[1]*	105	SCS[3]*
16	RAS0[3]*	46	A[10]	76	CE1[0]*	106	SCS[2]*
17	RAS1[0]*	47	A[11]	77	CE1[1]*	107	SCS[1]*
18	RAS1[1]*	48	A[12]	78	OE1*	108	SCS[0]*
19	VSS	49	A[13]	79	VSS	109	VSS
20	RAS1[2]*	50	A[14]	80	VDD	110	PIO0[0]
21	RAS1[3]*	51	A[15]	81	OE0*	111	PIO0[1]
22	CAS[0]*	52	VDD	82	SWE0*	112	PIO0[2]
23	CAS[1]*	53	VSS	83	SWE1*	113	PIO0[3]
24	VDD	54	A[16]	84	TST02	114	VDD
25	CAS[2]*	55	A[17]	85	TST01	115	PIO0[4]
26	CAS[3]*	56	A[18]	86	BOOT16	116	PIO0[5]
27	TOUT[0]	57	A[19]	87	HALF*	117	PIO0[6]
28	TOUT[1]	58	A[20]	88	TEST2*	118	PIO0[7]
29	VSS	59	A[21]	89	BUSGNT*	119	VSS
30	TOUT[2]	60	VSS	90	BUSREL*	120	TIMOUT2

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
121	TIMOUT1	143	D[24]	165	VDD	187	PCST[2]
122	TIMIN2	144	D[23]	166	D[7]	188	PCST[1]
123	TIMIN1	145	D[22]	167	D[6]	189	PCST[0]
124	SCLK	146	D[21]	168	D[5]	190	DCLK
125	SIN1	147	VDD	169	D[4]	191	SDAO
126	SIN0	148	VSS	170	D[3]	192	DBGE*
127	CTS1*	149	D[20]	171	D[2]	193	SDI*
128	CTS0*	150	D[19]	172	VSS	194	DRESET*
129	VDD	151	D[18]	173	VDD	195	TEST*
130	SOUT1	152	D[17]	174	D[1]	196	RESET*
131	SOUT0	153	D[16]	175	D[0]	197	ACK*
132	RTS1*	154	D[15]	176	NMI*	198	BUSERR*
133	RTS0*	155	D[14]	177	INT[0]	199	BSTART*
134	D[31]	156	VDD	178	INT[1]	200	VSS
135	D[30]	157	VSS	179	INT[2]	201	VDD
136	D[29]	158	D[13]	180	VDD	202	SYSCLK
137	VDD	159	D[12]	181	INT[3]	203	PLLOFF*
138	D[28]	160	D[11]	182	INT[4]	204	CLKEN
139	D[27]	161	D[10]	183	INT[5]	205	VDDP
140	D[26]	162	D[9]	184	INT[6]	206	VSSP
141	D[25]	163	D[8]	185	INT[7]	207	VSS
142	VSS	164	VSS	186	VSS	208	XOUT

4.2 Functions of Pins

Name of Signal	I/O	Function
System Interface		
SYSCLK	O	System Clock Outputs a clock with frequency either equal to or half of that of the TX39 Processor Core.
A[31:1]	I/O	Address bus When the TX3904 is a bus master, it is an output, otherwise input. A[31:24] pins are shared with PIO.
(PIO2[7:0])	I/O	
BE[3:0]*	I/O	Byte Enable Indicates valid data positions on the data bus D[31:0]. When the TX3904 is a bus master, it is an output, otherwise input. BE[3]* : D[31:24] BE[2]* : D[23:16] BE[1]* : D[15:8] BE[0]* : D[7:0]
D[31:0]	I/O	Data bus. D[15:0] is used in 16-bit bus mode.
SCS[3:0]*	O	System Chip Select Asserts when accessing the address range that is set by the internal register.
R/W*	I/O	Read/Write Indicates the bus operation being executed is either read or write. When the TX3904 is a bus master, it is an output, otherwise input. High: Read Low: Write
BSTART*	I/O	Bus Start Asserts during the first clock of the bus operation. When the TX3904 is a bus master, it is an output, otherwise input.
LAST*	I/O	Last Indicates that it is the last of the bus operation. When the TX3904 is a bus master, it is an output, otherwise input.
ACK*	I/O	Acknowledge External circuits inform the TX3904 that the bus operation may be finished. When the TX3904 is a bus master, it is an input, otherwise output.

BUSERR*	I/O	Bus Error Informs of bus errors. When the TX3904 is a bus master, it is an input, otherwise output.
RESET*	I	Reset Initializes the TX3904 by setting the RESET* signal low for 12 SYSCLK or more.
Clock		
XIN	I	Crystal Input Connect a crystal oscillator.
XOUT	O	Crystal Output Connect a crystal oscillator.
PLLOFF*	I	PLL OFF A signal to halt the PLL oscillation of the TX3904 built-in clock generator.
CLKEN	I	Clock Enable A signal to halt the TX3904 internal clock.
External Bus Master Interface		
BUSREQ*	I	Bus Request Changes to low when the external bus master requests for the bus ownership of the TX39 Processor Core. A pin that is shared with the PIO.
(PIO1[1])	I/O	
BUSGNT*	O	Bus Grant Asserted when the TX39 Processor Core informs that it is releasing the bus ownership in response to BUSREQ*. A pin that is shared with the PIO.
(PIO1[0])	I/O	
HAVEIT*	I	Have It Asserted when the external or internal bus master has the bus ownership. A pin that is shared with the PIO.
(PIO1[2])	I/O	
BUSREL*	O	Bus Release Asserted when TX39 processor core requests the external bus master to release the bus ownership. A pin that is shared with the PIO.
(PIO1[3])	I/O	
Interrupt Signals		
NMI*	I	Non Maskable Interrupt Non-maskable interrupt input.
INT[7:0]	I	Interrupt Request External interrupt request signals. Edge triggered.

Memory Interface		
RAS0[3:0]* RAS1[3:0]*	O	Row Address Strobe RAS signals for the DRAM.
CAS[3:0]*	O	Column Address Strobe CAS signals for the DRAM.
WE*	O	Write Enable Write enable signal for the DRAM access.
OE0* OE1*	O	Output Enable Output enable of the ROM.
CE0[1:0]* CE1[1:0]*	O	Chip Enable Chip select of the ROM.
SWE0* SWE1*	O	SRAM Write Enable Write enable signals of the SRAM and the Flash ROM.
DMA Interface		
DREQ[3:0]	I	DMA Request The external I/O device requests a DMA transfer. DREQ[3:2] are pins shared with the PIO.
(PIO1[7:6])	I/O	
DACK[3:0]	O	DMA Acknowledge Acknowledge signals to DMA transfer request through the DREQ. DACK[3:2] are pins shared with the PIO.
(PIO1[5:4])	I/O	
DONE*	I/O	Done Input: LOW is input to terminate data transfer. Output: Notification that transfer ended. It is asserted for one SYSCLK time period when DMA transfer ends.
Timer/Counter		
TIMOUT2 TIMOUT1	O	Timer Output Output signals of the timer.
TIMIN2 TIMIN1	I	Timer Input External signals for the timer's count.
Serial Port		
SIN1 SIN0	I	Serial Input Data input signals of the serial I/O.
SOUT1 SOUT0	O	Serial Output Data output signals of the serial I/O.

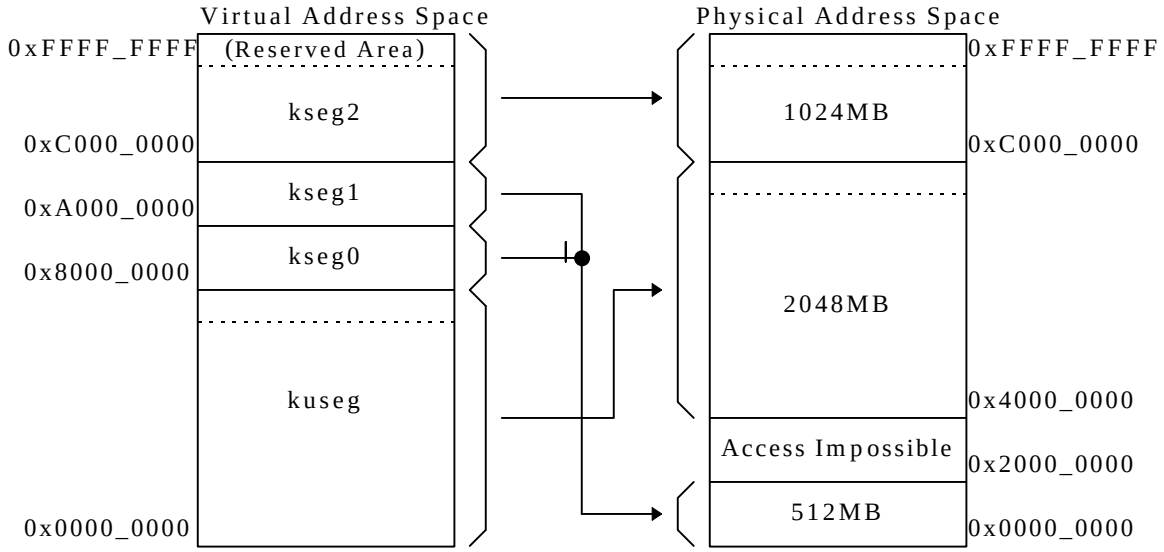
CTS0* CTS1*	I	Serial Port Control Control signals of the serial I/O.
RTS0* RTS1*	O	
SCLKIN	I	Serial Clock Input Clock input of the serial I/O. The TMPR3904F can't use this input.
I/O Port		
PIO0[7:0]	I/O	I/O Port0 A signal for I/O Port0. Input/Output can be set in each bit.
Debug Interface		
PCST[2:0] DCLK SDA0	O	Debug A signal for the external real time debug system. DBGE*, SDI*, and DRESET* are pulled up internally.
DBGE* SDI* DRESET*	I	
Others		
BOOT16	I	Boot 16-bit Fix to either high or low according to the bus width of the boot ROM. High: 16 bits Low: 32 bits
HALF*	I	Half Speed Bus Mode Designates the half speed bus mode. At low, the TX3904 becomes the half speed bus mode so that the frequency of the bus operation becomes a half of the operation frequency of the TX39 Processor Core. Fix to either high or low.
TEST* TEST2*	I	Test A signal for tests. Fix to high.
TOUT[3:0] TST01 TST02	O	Test output A signal for tests. Leave open.

Signals followed by * are active low.
5V tolerant signals are A[31:1], BE[3:0]*, D[31:0], R/W*, BSTART*, LAST*, ACK*, BUSERR*, RESET*, PLLOFF*, CLKEN, BUSREQ*,BUSGNT*, HAVEIT*, BUSREL*, NMI*, INT[7:0], DREQ[3:0], DACK[3:2], DONE*, TIMIN1, TIMIN2, SIN0, SIN1, CTS0*, CTS1*, SCLKIN, PIO0[7:0], BOOT16, HALF*, TEST*, TEST2*

5 ADDRESS MAPS

5.1 Memory Map

The memory space of the TX3904 is 4 Gbytes. The memory map of the TX3904 is managed by the memory management unit (MMU) of the TX39 Processor Core. The following diagram shows the memory map of the TX3904.
As for details of the memory map of the TX39 Processor Core, please refer to:
TX39 Family Users' Manual



Segment	Virtual Address	Physical Address	Cache	Mode
kseg2 (Reserved Area)	0xFFFF_FFFF 0xFF00_0000	0xFFFF_FFFF 0xFF00_0000	Impossible	Kernel
kseg2	0xFEFF_FFFF 0xC000_0000	0xFEFF_FFFF 0xC000_0000	Possible	Kernel
kseg1	0xBFFF_FFFF 0xA000_0000	0x1FFF_FFFF 0x0000_0000	Impossible	Kernel
kseg0	0x9FFF_FFFF 0x8000_0000	0x1FFF_FFFF 0x0000_0000	Possible	Kernel
kuseg	0x7FFF_FFFF 0x7F0000_0000	0xBFFF_FFFF 0xBF00_0000	Impossible	Kernel & User
kuseg	0x7EFF_FFFF 0x0000_0000	0xBEFF_FFFF 0x4000_0000	Possible	Kernel & User

Fig. 5-1 TX3904 Address Map

In the TX3904, the address to which memories (DRAM, ROM, etc.) are to be allocated shall be set up in the memory controller. However, Channel 0 of the ROM is for the boot program; so that the

address is determined at the time of reset.
The high-order 16 MB of kseg2 is a reserved area. The registers of the TX3904 built-in peripheral modules are assigned to this reserved area.

5.2 Register Map

The following is the register map of the TX3904 built-in modules:

Table 5-1 TX3904 Register Map

Address	Module	Register
0xFFFF_F704	PIO2	Data Register
0xFFFF_F700	PIO2	Direction Register
0xFFFF_F604	PIO1	Data Register
0xFFFF_F600	PIO1	Direction Register
0xFFFF_F504	PIO0	Data Register
0xFFFF_F500	PIO0	Direction Register
0xFFFF_F430	SIO1	Receiver FIFO Buffer
0xFFFF_F420	SIO1	Transmitter FIFO Buffer
0xFFFF_F414	SIO1	Baud Rate Control Register
0xFFFF_F410	SIO1	FIFO Control Register
0xFFFF_F40C	SIO1	DMA/Interrupt Status Register
0xFFFF_F408	SIO1	DMA/Interrupt Control Register
0xFFFF_F404	SIO1	Line Status Register
0xFFFF_F400	SIO1	Line Control Register
0xFFFF_F330	SIO0	Receiver FIFO Buffer
0xFFFF_F320	SIO0	Transmitter FIFO Buffer
0xFFFF_F314	SIO0	Baud Rate Control Register
0xFFFF_F310	SIO0	FIFO Control Register
0xFFFF_F30C	SIO0	DMA/Interrupt Status Register
0xFFFF_F308	SIO0	DMA/Interrupt Control Register
0xFFFF_F304	SIO0	Line Status Register
0xFFFF_F300	SIO0	Line Control Register
0xFFFF_F2F0	TMR2	Timer Read Register
0xFFFF_F240	TMR2	Watchdog Timer Mode Register
0xFFFF_F230	TMR2	Pulse Generator Mode Register
0xFFFF_F220	TMR2	Event Counter Mode Register
0xFFFF_F210	TMR2	Interval Timer Mode Register
0xFFFF_F20C	TMR2	Compare Register 2
0xFFFF_F208	TMR2	Compare Register 1
0xFFFF_F204	TMR2	Timer Interrupt Control Register
0xFFFF_F200	TMR2	Timer Control Register
0xFFFF_F1F0	TMR1	Timer Read Register
0xFFFF_F140	TMR1	Watchdog Timer Mode Register

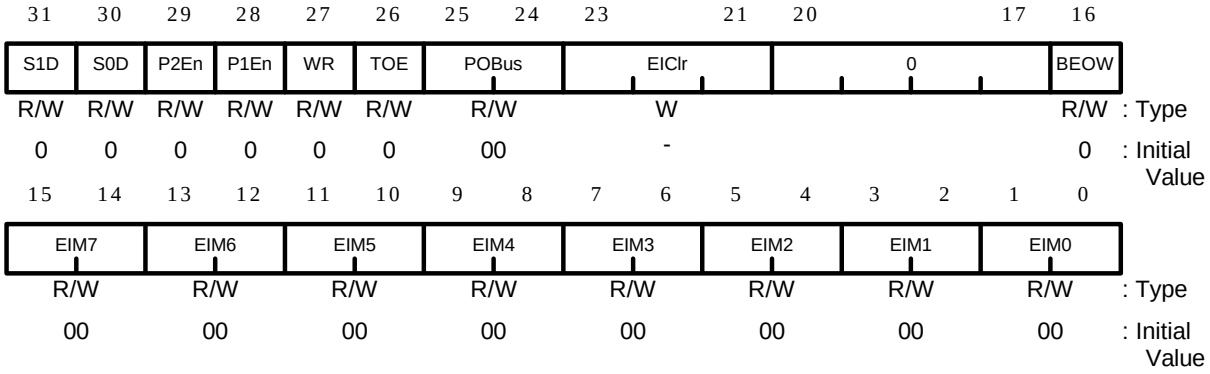
0xFFFF_F130	TMR1	Pulse Generator Mode Register
0xFFFF_F120	TMR1	Event Counter Mode Register
0xFFFF_F110	TMR1	Interval Timer Mode Register
0xFFFF_F10C	TMR1	Compare Register 2
0xFFFF_F108	TMR1	Compare Register 1
0xFFFF_F104	TMR1	Timer Interrupt Control Register
0xFFFF_F100	TMR1	Timer Control Register
0xFFFF_F0F0	TMR0	Timer Read Register
0xFFFF_F040	TMR0	Watchdog Timer Mode Register
0xFFFF_F030	TMR0	Pulse Generator Mode Register
0xFFFF_F020	TMR0	Event Counter Mode Register
0xFFFF_F010	TMR0	Interval Timer Mode Register
0xFFFF_F00C	TMR0	Compare Register 2
0xFFFF_F008	TMR0	Compare Register 1
0xFFFF_F004	TMR0	Timer Interrupt Control Register
0xFFFF_F000	TMR0	Timer Control Register
0xFFFF_E018	EBIF	SCS Wait Register
0xFFFF_E014	EBIF	SCS Mask Register
0xFFFF_E010	EBIF	SCS Address Register
0xFFFF_E000	EBIF	Chip Configuration Register
0xFFFF_C004	IRC	Interrupt Mask Register
0xFFFF_C01C	IRC	Interrupt Level Register 3
0xFFFF_C018	IRC	Interrupt Level Register 2
0xFFFF_C014	IRC	Interrupt Level Register 1
0xFFFF_C010	IRC	Interrupt Level Register 0
0xFFFF_C000	IRC	Interrupt Status Register
0xFFFF_B080	DMA1	DMA Control Register
0xFFFF_B034	DMA1	Next Byte Count Register (ch.3)
0xFFFF_B030	DMA1	Byte Count Register (ch.3)
0xFFFF_B02C	DMA1	Destination Address Register (ch.3)
0xFFFF_B028	DMA1	Source Address Register (ch.3)
0xFFFF_B024	DMA1	Channel Status Register (ch.3)
0xFFFF_B020	DMA1	Channel Control Register (ch.3)
0xFFFF_B014	DMA1	Next Byte Count Register (ch.2)
0xFFFF_B010	DMA1	Byte Count Register (ch.2)
0xFFFF_B00C	DMA1	Destination Address Register (ch.2)
0xFFFF_B008	DMA1	Source Address Register (ch.2)
0xFFFF_B004	DMA1	Channel Status Register (ch.2)
0xFFFF_B000	DMA1	Channel Control Register (ch.2)
0xFFFF_A080	DMA0	DMA Control Register
0xFFFF_A034	DMA0	Next Byte Count Register (ch.1)
0xFFFF_A030	DMA0	Byte Count Register (ch.1)

0xFFFF_A02C	DMA0	Destination Address Register (ch.1)
0xFFFF_A028	DMA0	Source Address Register (ch.1)
0xFFFF_A024	DMA0	Channel Status Register (ch.1)
0xFFFF_A020	DMA0	Channel Control Register (ch.1)
0xFFFF_A014	DMA0	Next Byte Count Register (ch.0)
0xFFFF_A010	DMA0	Byte Count Register (ch.0)
0xFFFF_A00C	DMA0	Destination Address Register (ch.0)
0xFFFF_A008	DMA0	Source Address Register (ch.0)
0xFFFF_A004	DMA0	Channel Status Register (ch.0)
0xFFFF_A000	DMA0	Channel Control Register (ch.0)
0xFFFF_9104	ROMC	Base Address Mask Register 1
0xFFFF_9100	ROMC	Channel Control Register 1
0xFFFF_9004	ROMC	Base Address Mask Register 0
0xFFFF_9000	ROMC	Channel Control Register 0
0xFFFF_8800	DRAMC	Refresh Control Register
0xFFFF_8108	DRAMC	Wait Register 1
0xFFFF_8104	DRAMC	Base Address Mask Register 1
0xFFFF_8100	DRAMC	Channel Control Register 1
0xFFFF_8008	DRAMC	Wait Register 0
0xFFFF_8004	DRAMC	Base Address Mask Register 0
0xFFFF_8000	DRAMC	Channel Control Register 0

For details of each register, please refer to the chapter of each module. As for the SCS wait register, SCS mask register, and SCS address register of the EBIF, please refer to Chapter 7. The chip configuration register of the EBIF will be explained in the next section.

5.3 Chip Configuration Register

The TX3904 has functions and pins that can be set up with software. These set-ups are conducted in the chip configuration register (CConR).



5.3.1 DMA transfer of SIO

The TX3904 built-in DMA controller can, as a transfer device, select the internal SIO as well as devices outside the TX3904.

Bit	Mnemonic	Name of Field	Description
31	S1D	SIO1 DMA connection	SIO1 DMA Enable (Initial value: 0) Selects SIO1 as the I/O device of the DMA's channel 2: i.e., the DMA transfer request signal and the DMA transfer acknowledge signal of Channel 2 will be connected to SIO1 inside the TX3904. The DREQ[2] and DACK[2] that are external pins of the TX3904 will be invalid. 1: Channel 2 of the DMA will be connected to SIO1. 0: Channel 2 of the DMA will be connected to the external I/O device through the DREQ[2] and DACK[2] signals.
30	S0D	SIO0 DMA connection	SIO0 DMA Enable (Initial value: 0) Selects SIO0 as the I/O device of the DMA's channel 1: i.e., the DMA transfer request signal and the DMA transfer acknowledge signal of Channel 1 will be connected to SIO0 inside the TX3904. The DREQ[1] and DACK[1] that are external pins of the TX3904 will be invalid. 1: Channel 1 of the DMA will be connected to SIO0. 0: Channel 1 of the DMA will be connected to the external I/O device through the DREQ[1] and DACK[1] signals.

5.3.2 PIO2 and PIO1

The TX3904 has built-in 3-channel 8-bit I/O ports (PIO2, PIO1, PIO0). Of these, PIO2 and PIO1 share their pins with others. It should be set up whether these pins are used as I/O Port pins or as other functions.

The following table shows the correspondence of the shared pins of PIO2 and PIO1:

Bit	Mnemonic	Name of Field	Description
29	P2En	PIO2 enable	PIO2 Enable (Initial value: 0) 1: Enables PIO2 for usage. 0: PIO2 cannot be used.
28	P1En	PIO1 enable	PIO1 Enable (Initial value: 0) 1: Enables PIO1 for usage. 0: PIO1 cannot be used.

P2En=1	PIO2[7]	PIO2[6]	PIO2[5]	PIO2[4]	PIO2[3]	PIO2[2]	PIO2[1]	PIO2[0]
P2En=0	A[31]	A[30]	A[29]	A[28]	A[27]	A[26]	A[25]	A[24]

P1En=1	PIO1[7]	PIO1[6]	PIO1[5]	PIO1[4]	PIO1[3]	PIO1[2]	PIO1[1]	PIO1[0]
P1En=0	DREQ[3]	DREQ[2]	DACK[3]	DACK[2]	BUSREL*	HAVEIT*	BUSREQ*	BUSGNT*

5.3.3 Error processing

Sets up the watchdog interrupt from the TX3904 built-in timer/counter and bus error detection at write operation.

Bit	Mnemonic	Name of Field	Description
27	WR	Watchdog timer reset	Watch Dog Timer to Reset (Initial value: 0) 1: Connects the watchdog timer interrupt to the TX3904 internal reset signal. 0: Connects the watchdog timer interrupt to the TX3904 internal NMI signal.
26	TOE	Time out error detection	Time Out Error Enable (Initial value: 0) 1: Enable time out error detection. 0: Disable time out error detection.
16	BEOW	Bus error occurred during write operation	Bus Error On Write Indicates that a bus error was generated by a write operation of the TX39 processor core. Notification of anomaly is sent due to a non-maskable interrupt if one of the following occurs: access of nonexistent register, access of nonexistent address (0xFFFF_Dxxx), or BUSERR* signal assert. Simultaneous to this, sets the BEOW bit to 1 and indicates that a bus error was the cause of the non-maskable interrupt. 1: Non-maskable interrupt was generated since a bus error occurred during a write operation of the TX39 processor core.

5.3.4 Connection of external bus master

The TX3904 releases the bus ownership in response to the bus ownership request from the external master. At that time, it prioritizes the bus ownership to be given to the external bus master.

Bit	Mnemonic	Name of Field	Description
25:24	POBus	Bus ownership set-up	Priority of External Bus Request (Initial value: 00) Prioritizes the bus ownership of external bus master requests. The order of priority, from the highest one, is HPSERQ, HPGREQ, SREG, and GREQ. At HPSREQ and SREQ, the snoop function of the TX39 Processor Core works. The priority of the TX3904 built-in DMAC is either SREQ or GREQ. 11: HPSREQ (High Priority Snoop Request) 10: HPGREQ (High Priority General Request) 01: SREQ (Snoop Request) 00: GREQ (General Request) HPGREQ and GREQ are not able to be used.

5.3.5 INT[7:0] active status clear

Clears external interrupt requests when edge mode is set up. Use the Store Byte instruction in order to write into this field. Because when the interrupt mode explained in the next section is re-written, the edge mode interrupt suspended is clear.

Bit	Mnemonic	Name of Field	Description
23:21	EICIr	External interrupt clear	External Interrupt Clear Clears external interrupt requests by INT[7:0]. If this field is read out, the value read is 000. 111: Clears interrupt requests of INT[7]. 110: Clears interrupt requests of INT[6]. 101: Clears interrupt requests of INT[5]. 100: Clears interrupt requests of INT[4]. 011: Clears interrupt requests of INT[3]. 010: Clears interrupt requests of INT[2]. 001: Clears interrupt requests of INT[1]. 000: Clears interrupt requests of INT[0].

5.3.6 INT[7:0] active status set-up

Sets up the active status of the external interrupt request signal INT[7:0]. One from among the falling edge, the rising edge, the high level, and the low level of the INT[7:0] signal will be valid as an external interrupt signal. According to the setup active status, the TX3904 requests for interrupt(s) of the built-in interrupt controller. When an edge mode is set into this field, the previously suspended edge mode interrupt request is clear.

Bit	Mnemonic	Name of Field	Description
15:14	EIM7	Interrupt mode 7	External Interrupt Mode 7 Sets up the active status of the external interrupt signal INT[7]. 11: Acknowledges the rising edge as an interrupt request. 10: Acknowledges the falling edge as an interrupt request. 01: Acknowledges the high level as an interrupt request. 00: Acknowledges the low level as an interrupt request.
13:12	EIM6	Interrupt mode 6	External Interrupt Mode 6
11:10	EIM5	Interrupt mode 5	External Interrupt Mode 5
9:8	EIM4	Interrupt mode 4	External Interrupt Mode 4
7:6	EIM3	Interrupt mode 3	External Interrupt Mode 3
5:4	EIM2	Interrupt mode 2	External Interrupt Mode 2
3:2	EIM1	Interrupt mode 1	External Interrupt Mode 1
1:0	EIM0	Interrupt mode 0	External Interrupt Mode 0

6 CLOCKS

6.1 Clock Generator

The TX3904 has a built-in x4 frequency PLL clock generator. Please connect a crystal oscillator with a frequency of a quarter of the internal system clock (frequency of the TX39 Processor Core's input clock).

6.2 Operation Modes of TX3904

The following describes the operation modes by the TX3904 clock control. The TX3904 has the normal, doze, halt, and RF modes.

6.2.1 Normal mode

Both the TX39 Processor Core and the peripheral mega cells operate at the maximum frequencies.

Blocks that operate at the same frequency as the TX39 Processor Core:

DRAMC, ROMC, DMAC, IRC

Blocks that operate at the frequency that is a half of the TX39 Processor Core:

TMR, SIO, PIO

6.2.2 Halt mode

The halt mode is a mode to lower the power consumption by halting operations by halting the clocks inside the TX39 Processor Core.
By setting the halt bit of the Config register of the TX39 Processor Core, it shifts to the halt mode.
When having entered into the halt mode, the TX39 mega cell core halts the operation while maintaining the pipeline status. Bus release requests from GREQ* and HPGREQ* can be replied to; but those from the SREQ* and HPSREQ* are not replied to. The write buffer does not halt: Therefore, when there are remaining data in the write buffer in the halt mode, the write operation continues until the buffer becomes empty. The SYSCLK does not halt.
The halt mode is released when the halt bit is cleared to 0 by asserting the interrupts by on-chip peripherals (internal interrupt), INT[7:0], NMI*, or RESET* signal. When the TX3904 is recovered from halt mode by the above causes, the corresponding exception handler is executed.
The peripheral blocks halt partial functions by halting the clocks.

Halt mode is not usable in half speed bus mode.

The halt bit must be set to 1 with an example flow shown below in order to make the TX3904 halt mode. An exception must be caused when the TX3904 is recovered form halt mode. An interrupt must not be masked.

```
address          instruction              comment
0x????_????    mfc0 r25, r3             # read Config register
0x????_???c    sync                     #
0x????_???0    ori  r25, r25, 0x100     # set HALT bit
0x????_???4    j    label               #
0x????_???8    mtc0 r25, r3             # write Config register
0x????_???c  label:
```

A store instruction which sets a halt bit in the configuration register of the TX39 processor core must be placed on an address whose low four bit is 0x8. If not, the TMPR3904F may not return to the normal mode correctly.

6.2.3 Doze mode

The doze mode is a mode to lower power consumption by partially halting the TX39 Processor Core's operations. The difference from the halt mode is that bus release requests from outside can be accepted because the doze mode halts some of the clocks inside the Processor Core. The peripheral blocks continue normal operations.
By setting the doze bit of the Config register of the TX39 Processor Core, it shifts to the doze mode.
When having entered into the doze mode, the TX39 mega cell core halts operations while maintaining the pipeline status. The write buffer does not halt. Therefore, if there are remaining data in the write buffer in the doze mode, the write operation continues until the buffer becomes empty. The SYSCLK does not halt, either.
The doze mode is recovered from when the doze bit is cleared to 0 by asserting the interrupts by on-chip peripherals (internal interrupt), INT[7:0], NMI*, or RESET* signal. The value in the IntMask field of the status register is not affected by the recovery from the doze mode. If recovered by the RESET* signal, NMI* signal, or non-masked (internal interrupt) and INT[7:0] signal, the corresponding exception handler is executed. If recovered by the masked (internal interrupt) and INT[7:0] signal, execution resumes from the instruction that follows the instruction that was being executed when shifted to the doze mode.

6.2.4 RF (Reduced Frequency) mode

The frequency of the clock is controlled by operating the clock generator by setting up the RF field of the Config register of the TX39 Processor Core. When an instruction to change the RF field is executed while the bus ownership is being released, the clock's frequency is changed without waiting for the bus ownership to be returned. Also when the RF

field is changed during a bus operation, the clock is changed without waiting for the completion of the bus operation.
The RF mode lowers the power consumption by supplying clocks that correspond the RF field for the peripheral macro cell(s) and the mega cell core in the parts that are not affected by frequency changes.

6.3 Status Shifting

The following diagram shows shifts of the modes.

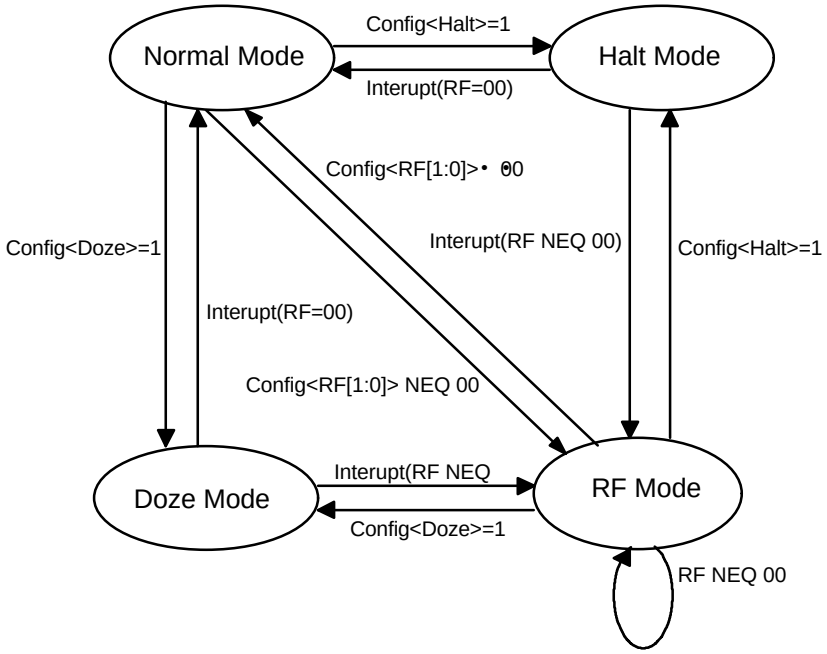


Fig. 6-1 Shifts of Modes

6.4 Operations of each block in the each modes

Table 6-1 Operations of each block in the each modes

Block	Function	Operation in Doze Mode	Operation in Halt Mode	Clock change in RF Mode
TX39	Snoop function	Operation	Halts	Change
	Interrupt function Others	Operation Halts	Operation Halts	
DRAMC	Refresh function	Operation	Operation*1	No change
	Others		Halts	Change
ROMC	All functions	Operation	Halts	Change
DMAC	All functions	Operation	Halts	Change
IRC	All functions	Operation	Operation	Change
SIO	For transmitter/receiver clock generation	Operation	Operation	No change
	Others			Change
TMR	Counter operation	Operation	Operation	No change
	Others			Change
PIO	All functions	Operation	Halts	Change

Note *1: If a DRAM with the self-refresh function is used, the refresh counter can be halted.

7 BUS OPERATIONS

This chapter explains the bus operations of the TX3904: Operations for which the TX3904 built-in memory controllers are not used.

7.1 Basic Bus Operations

There are three kinds in the TX3904 bus operations--the single read operation, the burst read operation, and the single write operation.

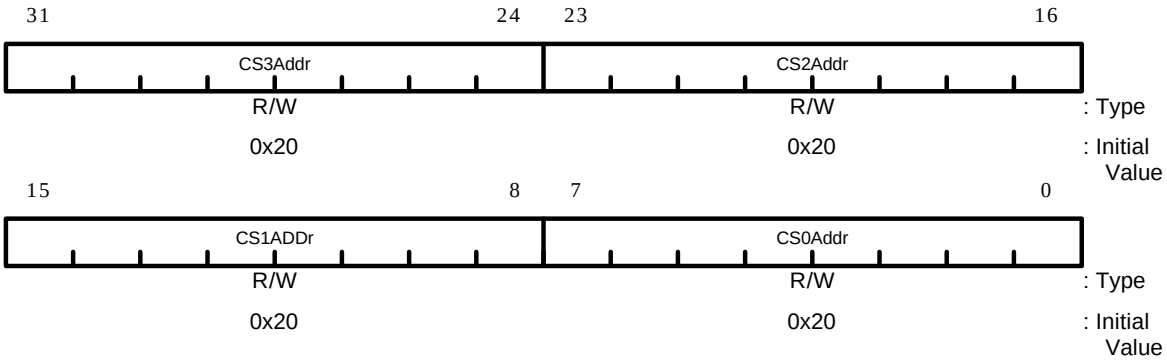
7.1.1 System chip select

In the TX3904, there are four system chip select signals (SCS[3:0]*). These signals are a low-active select signal made by decoding the high-order bit of the address that the TX39 Processor Core or the built-in DMAC outputs. The SCS[3:0]* is set up in three registers of the EBIF.

Table 7-1 Register Map for System Chip Select

Address	Module	Register
0xFFFF_E018	EBIF	SCS Wait Register
0xFFFF_E014	EBIF	SCS Mask Register
0xFFFF_E010	EBIF	SCS Address Register

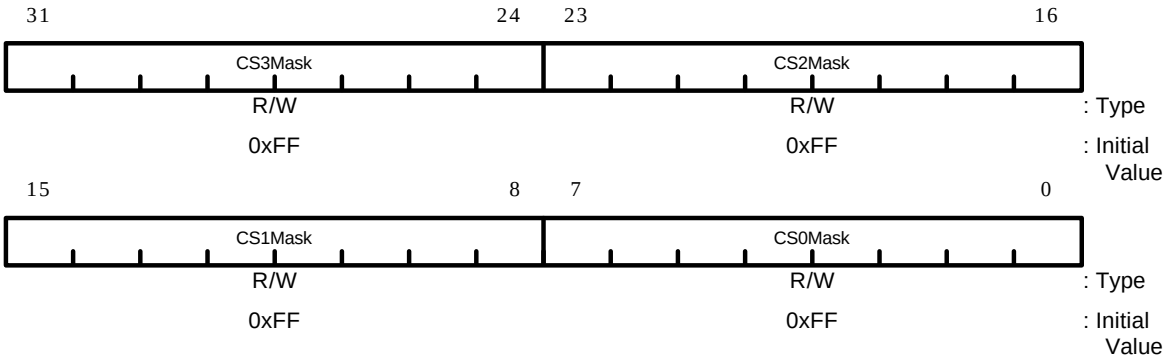
SCS Address register



Bit	Mnemonic	Name of Field	Description
31:24	CS3Addr	SCS3 address	SCS3* Address (initial value: 0x20) Asserts a CS3 signal when the high-order 8 bits (A[31:24]) of the address (physical address) matches the CS3Addr. The efficient range of the CS3Addr shall be set up in the CS3Mask field of the SCS Mask register.
23:16	CS2Addr	SCS2 address	SCS2* Address
15:8	CS1Addr	SCS1 address	SCS1* Address
7:0	CS0Addr	SCS0 address	SCS0* Address

Fig. 7-1 SCS Address Register

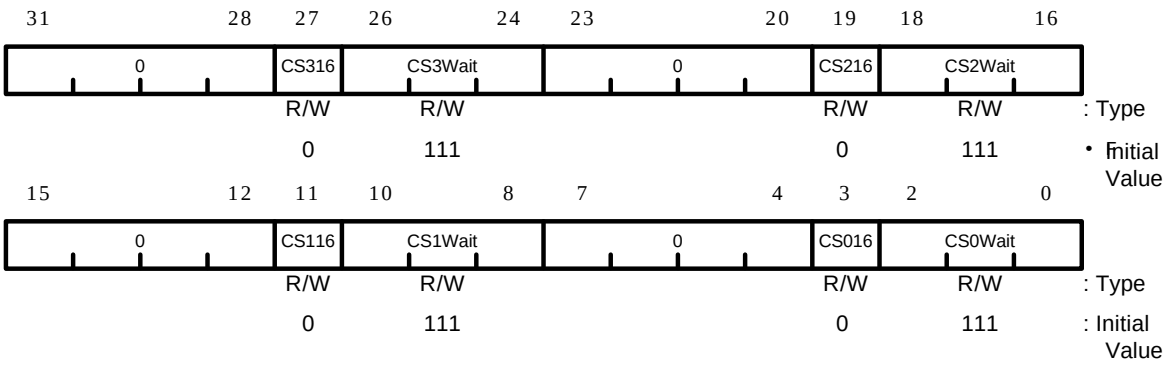
SCS Mask register



Bit	Mnemonic	Name of Field	Description
31:24	CS3Mask	SCS3 mask	SCS3* Mask (initial value: 0xFF) Specifies the valid bit of the address comparison by the CS3Addr field of the SCS Address register. 1: Bit of the corresponding CS3Addr field is compared. 0: Bit of the corresponding CS3Addr field is not compared.
23:16	CS2Mask	SCS2 mask	SCS2* Mask
15:8	CS1Mask	SCS1 mask	SCS1* Mask
7:0	CS0Mask	SCS0 mask	SCS0* Mask

Fig. 7-2 SCS Mask Register

SCS Wait register

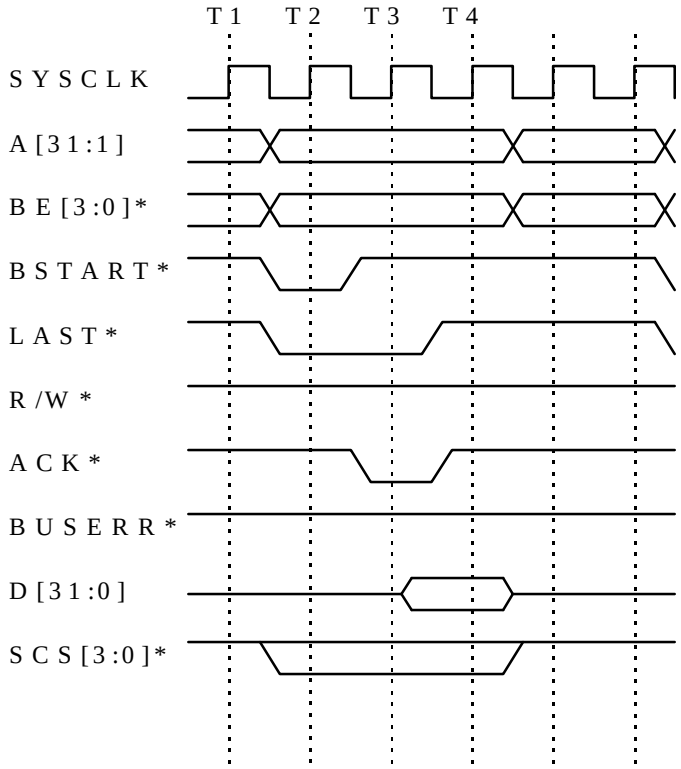


Bit	Mnemonic	Name of Field	Description
27	CS316	CS3 16 bits	CS3 16 bits Sets up the data width of the SCS[3]* area. 0: The data width of the area SCS[3] is 32 bits. 1: The data width of the area SCS[3] is 16 bits.
26:24	CS3Wait	CS3 wait	CS3 Wait Sets up the number of waits for the SCS[3]* signal. Executes the bus operation with the set-up number of waits. If it is set up to 111, drive the ACK* signal from outside. 000: 0 Wait 001: 1 Wait 010: 2 Waits 011: 3 Waits 100: 4 Waits 101: 5 Waits 110: 6 Waits 111: External ACK* Input
19	CS216	CS2 16 bits	CS2 16 bits
18:16	CS2Wait	CS2 wait	CS2 Wait
11	CS116	CS1 16 bits	CS1 16 bits
10:8	CS1Wait	CS1 wait	CS1 Wait
3	CS016	CS0 16 bit1	CS0 16 bits
2:0	CS0Wait	CS0 wait	CS0 Wait

Fig. 7-3 SCS Wait Register

7.1.2 Single read operation

The single read operation is a bus operation to read data of 4 bytes or less. The following diagram shows the operation of the shortest case (no wait) that has no wait cycle:



T1 Outputs the value(s) valid to A[31:1] and BE[3:0]*. At the same time, asserts BSTART* and LAST*. Asserts SCS[3:0]* also, if the address is in the SCSn area. The R/W* becomes high because it is a read operation.

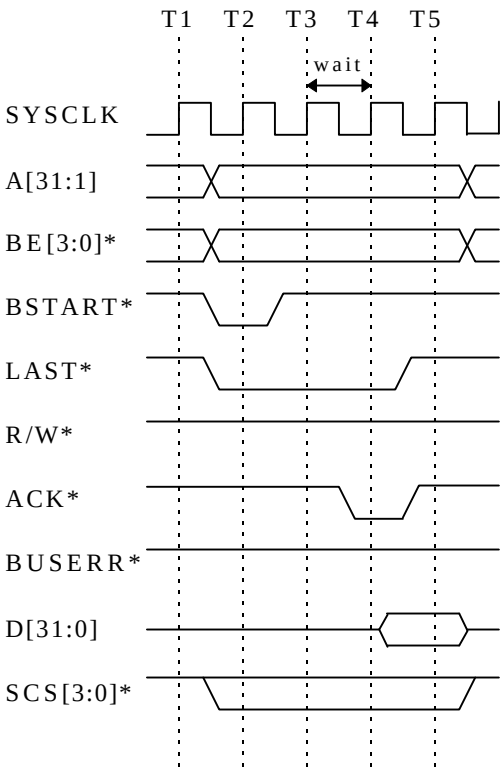
T2 Deasserts the BSTART*.

T3 Deasserts the LAST* because it has acknowledged that the ACK* is low. When a wait is designated in the SCS wait register, the input of the ACK* signal is not necessary. Please input the ACK* signal only when having designated to the external ACK* input.

T4 Takes in the value of D[31:0].

Wait cycles can be entered by not asserting the ACK* signal.

The following diagram shows the operation in the case of one wait:



T1 Outputs the value(s) valid to A[31:1] and BE[3:0]*. At the same time, asserts BSTART* and LAST*. Asserts SCS[3:0]* also, if the address is in the SCSn area. The R/W* becomes high because it is a read operation.

T2 Deasserts the BSTART*.

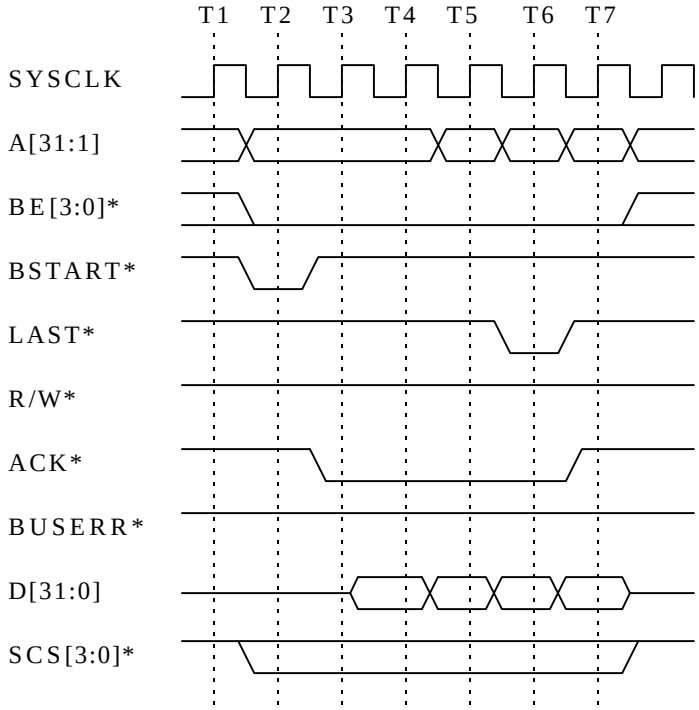
T3 Becomes the wait cycle because the ACK* is high.

T4 Deasserts the LAST* because it has acknowledged that the ACK* is low. When a wait is designated in the SCS wait register, the input of the ACK* signal is not necessary. Please input the ACK* signal only when having designated to the external ACK* input.

T5 Takes in the value of D[31:0].

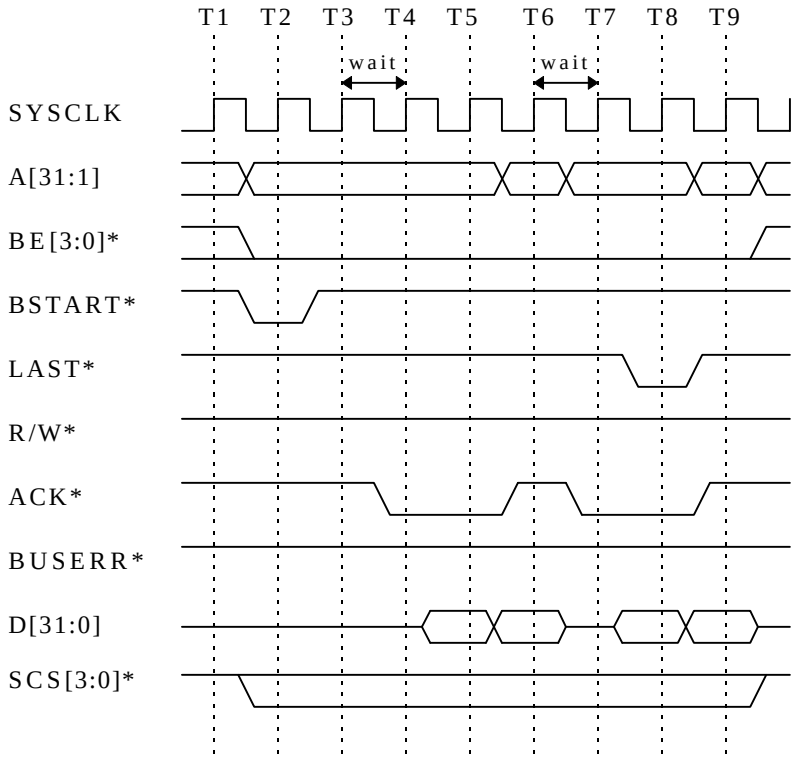
7.1.3 Burst read operation

The burst read operation is a bus operation to conduct refills of multiple words of the cache at a high speed. If instructions and data to be read in by the cache are in the memory that is managed by the TX3904 built-in memory controller, it is a operation to be discussed in the chapters of memory controllers (“DRAM Controller” and “ROM Controller”).
The following diagram shows the timing of the burst read operation without a wait cycle (of no wait). It is the case where the cache refill size is 4 words.



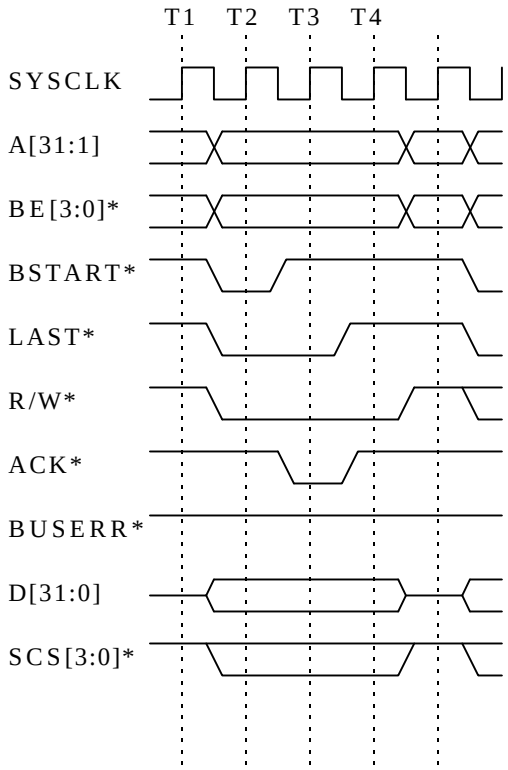
T1 Outputs the value valid to A[31:1]. The BE[3:0]* is always low. Asserts BSTART*. Does not assert LAST*. This point that it does not assert the LAST* is the difference from the single read operation. Asserts the SCS[3:0]* when the address is in the SCSn area. The R/W* becomes high because it is a read operation.
T2 Deasserts the BSTART*.
T3 Reads in data at the timing of T3 because the ACK* is low.
T4 Reads in the first datum from D[31:0]. Reads in the datum at the timing of T4 because the ACK* is low. Increments the value of A[31:1].
T5 Reads in the second datum from D[31:0]. Reads in the datum at the timing of T5 because the ACK* is low. Increments the value of A[31:1]. Also, asserts LAST*.
T6 Reads in the third datum from D[31:0]. Reads in the datum at the timing of T6 because the ACK* is low. Increments the value of A[31:1]. Deasserts the LAST*.
T7 Reads in the fourth (last) datum from D[31:0].

In the case of a burst read operation also, wait cycles can be entered by not asserting the ACK* signal.
The following diagram shows the operation in the case where a wait cycle is entered while the first and third data are being read. If SCS is used, the number of waits cannot be changed at each datum.



7.1.4 Single write operation

The only write operation that the TX39 Processor Core supports is the single write operation.
The following diagram shows the timing of the single write operation without a wait cycle (of no wait):



T1 Outputs the value(s) valid to A[31:1] and BE[3:0]*. At the same time, asserts BSTART* and LAST*. Asserts SCS[3:0]* also, if the address is in the SCSn area. The R/W* becomes low because it is a write operation. Outputs the value valid to D[31:0].

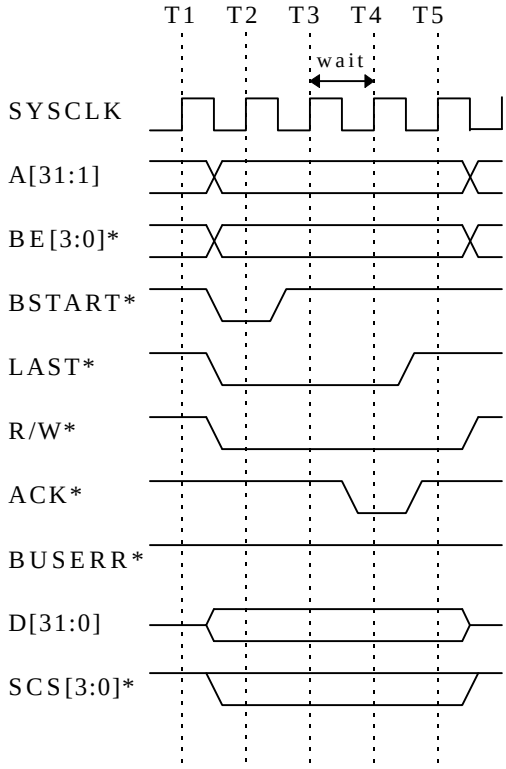
T2 Deasserts the BSTART*.

T3 Deasserts the LAST* because it has acknowledged that the ACK* is low. When a wait is designated in the SCS wait register, the input of the ACK* signal is not necessary. Please input the ACK* signal only when having designated to the external ACK* input.

T4 Stops the drive of D[31:0].

Wait cycles can be inserted by not asserting the ACK* signal.

The following diagram shows the operation in the case where a wait cycle has been inserted (one wait).



T1 Outputs the value(s) valid to A[31:1] and BE[3:0]*. At the same time, asserts BSTART* and LAST*. Asserts SCS[3:0]* also, if the address is in the SCSn area. The R/W* becomes low because it is a write operation. Outputs the value valid to D[31:0].

T2 Deasserts the BSTART*.

T3 Becomes the wait cycle because the ACK* is high.

T4 Deasserts the LAST* because it has acknowledged that the ACK* is low. When a wait is designated in the SCS wait register, the input of the ACK* signal is not necessary. Please input the ACK* signal only when having designated to the external ACK* input.

T5 Stops the drive of D[31:0].

7.2 Bus Error

7.2.1 BUSERR* signal

The TX3904 employs the BUSERR* input signal. The peripheral circuits can report that problems occurred during a bus operation by asserting a BUSERR* signal. The following operation occurs with respect to the BUSERR* signal.

* **During read operation of the TX39 processor core**

  The TX3904 immediately suspends the read operation. The TX39 processor core generates a bus error exception.

* **During write operation of the TX39 processor core**

  The TX3904 immediately suspends the write operation. The TX39 processor core generates a nonmaskable interrupt exception. Also, "1" is set to the BEOW bit of the CConR register.

* **During bus operation of the TX3904 on-chip DMA controller**

  The DMA controller immediately suspends transfer operation, then abnormally ends the channel operation. The DMA controller generates an interrupt if abnormal end interrupts are not prohibited.

7.2.2 Absence register access

Access to an address which does not actually exist in a register is processed as a bus error in each module register area inside the TX3904. Also, a bus error is generated when the external bus master attempts to access the inside of the TX3904 (0xFFxx_xxxx).

A bus error exception is generated during read operation by the TX39 processor core. A nonmaskable interrupt exception is generated during a write operation by the TX39 processor core.

7.2.3 Time-out error

If there is no response within 256 SYSCLK after the bus operation started, the R3904 can generate a bus error exception as a time-out error. To use the time-out error, set 1 to the TOE bit of the CConR. A time-out error occurs also in a bus operation by the external bus master.

A time-out error occurs when the ACK* signal (external or internal) is not asserted even after 256 SYSCLK has passed from the SYSCLK at which the BSTART* was asserted. Also, in the burst read operation, a time-out error occurs when the number of SYSCLK between the ACK* signals reaches 256.

The following operation occurs with respect to the time out error.

* **During read operation of the TX39 processor core**

  The TX39 processor core generates a bus error exception.

* **During write operation of the TX39 processor core**

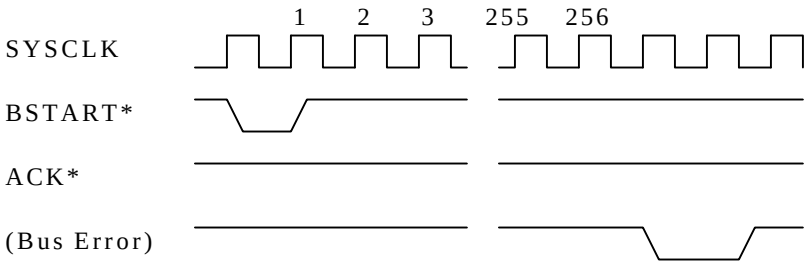
  The TX39 processor core generates a non-maskable interrupt exception. Also, "1" is set to the BEOW bit of the CConR register.

* **During bus operation of the TX3904 on-chip DMA controller**

The DMA controller immediately suspends transfer operation, then abnormally ends the channel operation.

* **During bus operation of the external bus master**

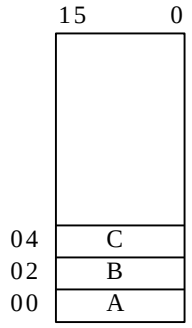
The TX3904 asserts BUSERR* signal.



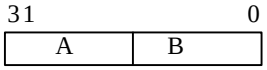
7.3 16-bit Bus Mode

The TX3904 can control the data bus as 16-bit width bus in the memory area of the RAMC, the memory area of the ROMC, and the area of the SCSn (16-bit bus mode). The 16-bit width should be designated in the built-in register of each module as for the memory areas controlled by the RAMC or the ROMC, and in the built-in register of the EBIF as for the area of the SCSn.
In the 16-bit bus mode, a 32-bit operation of the TX39 Processor Core shall be executed in two 16-bit bus operations.
The following diagram shows the relationship between the memory and the data bus/register:

Memory Image



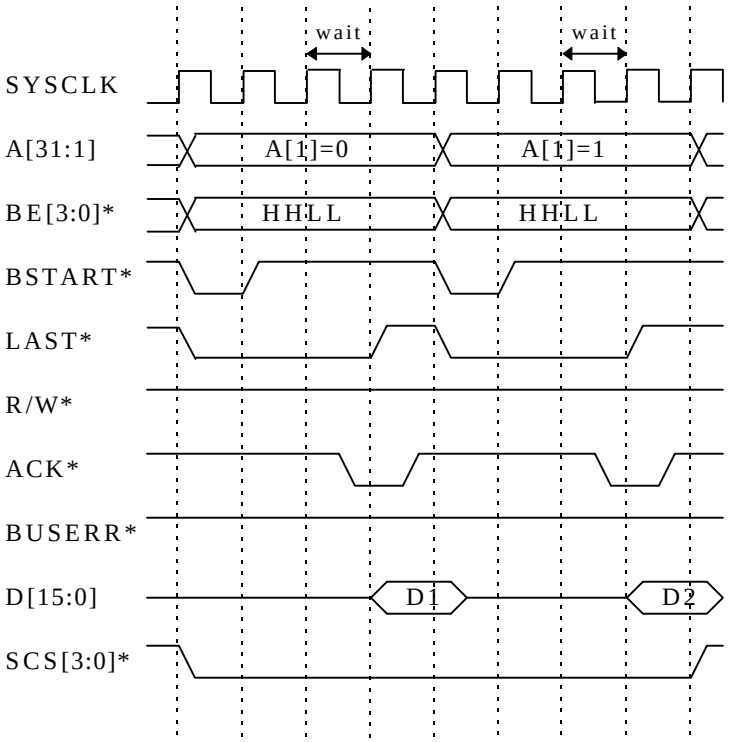
Register/Data Bus Image



If the data to be handled can fit in the upper or lower half word during a 16-bit bus mode, the operation is executed as one bus operation, and NOT two.
The data bus uses D[15:0].

The TMPR3904F doesn't support the little endian but supports only the big endian.

As an example, the following diagram shows the word access in the case where the SCSn area is 16-bit bus (one wait):



7.4 Half Speed Bus Mode

To simplify the design of the peripheral system, the TX3904 supports the half speed bus mode that executes the bus operation at half of the operation frequency of the TX39 Processor Core. For example, when a crystal oscillator of 12.5 MHz is connected to the TX3904, the TX39 Processor Core operates at 50 MHz and usually (the full speed bus mode) the bus operation is also at 50 MHz. In the half speed bus mode, the bus operation becomes 25 MHz. The frequency of the SYSCLK also changes.

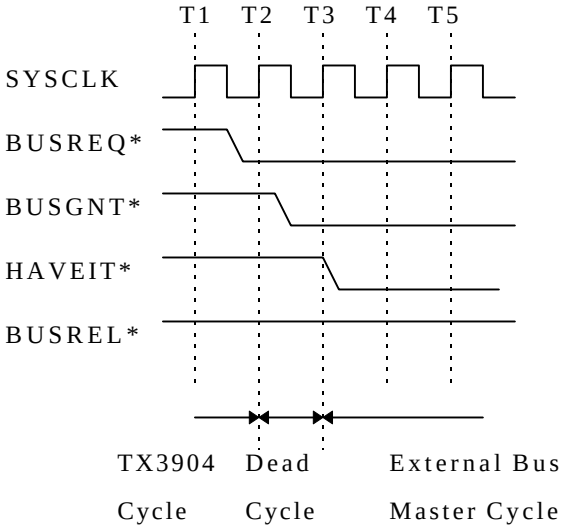
It is to be set up with the HALF* signal whether to make it operate in the half speed bus mode or not. When the HALF* signal is low, it is the half speed bus mode, and at high, it is the full speed bus mode. Please fix the HALF* signal to high or low to use.

7.5 Bus Arbitration

The TX3904 can connect a bus master externally. The arbitration of the bus ownership with an external bus master is conducted through four signals of BUSREQ*, BUSGNT*, BUSREL*, and HAVEIT*. The external bus master is to be daisy-chain connected in the downstream of the TX3904 built-in DMAC (See "10.2.1. TX3904 Internal Connection").

7.5.1 Bus ownership granted

The external bus master requests for the bus ownership of the TX3904 by asserting the BUSREQ* signal. When the TX3904 gives the bus ownership to the external bus master, it asserts the BUSGNT* signal.
When the external bus master has acquired the bus ownership, it has to assert the HAVEIT* signal.
The BUSREQ* and HAVEIT* signals must be asserted until releasing the bus ownership.
The following diagram shows the timing of the bus ownership being granted to an external bus master:



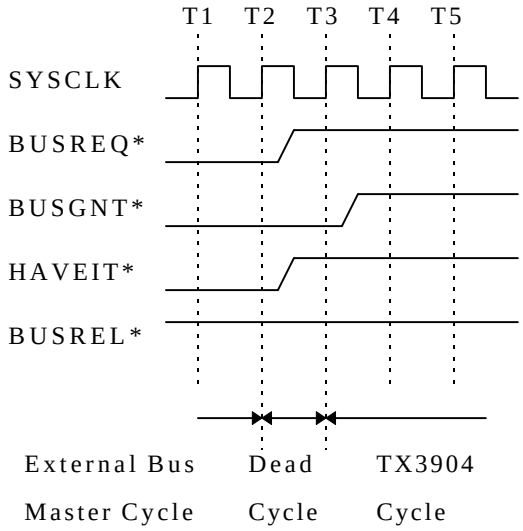
T1 The BUSREQ* is high.
T2 Having acknowledged that the BUSREQ* is low, the TX3904 asserts BUSGNT*. The timing of asserting the BUSGNT* varies by the status of the TX3904.
T3 Having acknowledged that the BUSGNT* is low, the external bus master is granted the bus ownership and asserts HAVEIT*. The bus operation of the external bus master starts the bus operation.

7.5.2 Release of bus ownership

The external bus master releases the bus ownership for two reasons:
Because the bus ownership is no longer necessary
Because of the BUSREL*.

(1) Release of the bus ownership because it is no longer necessary

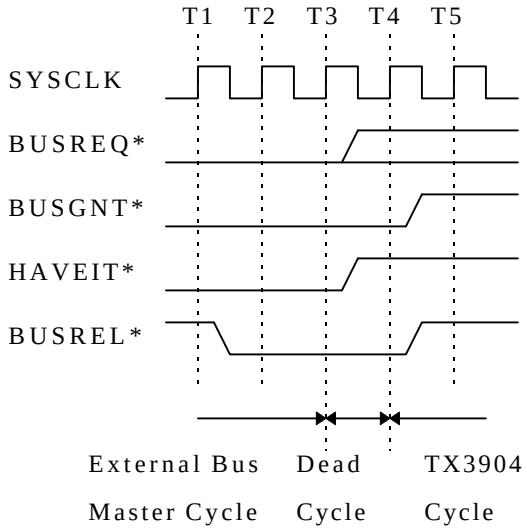
When the bus ownership that has been kept after having been granted becomes unnecessary, the external bus master deasserts the BUSREQ* and HAVEIT* to return the bus ownership to the TX3904.
The following diagram shows the timing of releasing the bus ownership when it is no longer necessary:



T1 The external bus master has the bus ownership.
T2 The external bus master deasserts the BUSREQ* and HAVEIT* because the bus ownership is no longer necessary.
T3 Having acknowledged that the BUSREQ* is high, the TX3904 deasserts the BUSGNT*.

(2) Release by BUSREL*

When the external bus master has the bus ownership, the TX3904 in some cases requests the release of the bus ownership by asserting the BURSEL*. The cases where the TX3904 requests the release of the bus ownership are:
A cache miss has occurred (in the case that snoop function is no in use)
The write buffer of the TX39 Processor Core has become full
The built-in DMAC that has a higher priority than the external bus master has requested for the bus ownership (When the bus ownership of the external DMAC is the GREQ and the bus ownership of the internal bus master is SREQ).
For the release of the bus ownership by BUSREL*, BUSREQ* does not have to be de-asserted. Also, the release request of the bus ownership may be left unanswered.
The following diagram shows the timing of the bus ownership release by BUSREL*:



T1 The TX3904 asserts BUSREL*.
T2 The external bus master acknowledges that the BUSREL* is low.
T3 The external bus master deasserts the HAVEIT* at the timing when the bus ownership can be released (such as a break of the bus operation). Then, the BUSREQ* can be deasserted or left not-deasserted.
T4 Having acknowledged that the HAVEIT* is high, the TX3904 deasserts the BUSGNT* and BUSREL*.

7.5.3 Kinds of bus ownership

There are four kinds in the bus ownership that an external bus master requests:

HPSREQ (High Priority Snoop Request)
HPGREQ (High Priority General Request)
SREQ (Snoop Request)
GREQ (General Request)
Which bus ownership of these should be used is to be set up in the POBus field of the CConR of the EBIF.
These four vary in priority and functions.
The priority is, in order from the highest, HPSREQ, HPGREQ, SREQ, and GREQ. Of these, the TX3904 built-in DMAC has only SREQ and GREQ. Because the external bus master is daisy-chain connected in the downstream of the built-in DMAC, if SREQ or GREQ is used, the priority is lower than the built-in DMAC. If HPSREQ or HPGREQ is used, the priority is definitely higher than the built-in DMAC.
In the case of different priority due to the different kinds of bus ownership, if the bus master with higher priority requests the bus ownership when another bus master with lower priority has the bus ownership, a forceful transit of the bus master occurs through the BUSREL*. In the case of different priority by the daisy-chain connection (when the bus ownership is the same), a forceful transit of the bus master does not occur; and the bus master with higher priority (in upstream) cannot be granted the bus ownership until the bus master with lower priority (in downstream) has no more use of the bus ownership to release it.
The difference in the function is whether or not there is a snoop function request. With HPSREQ and SREQ, the snoop function of the TX39 Processor Core works. With HPGREQ and GREQ, the snoop function does not work.
The snoop function is explained in the next section.

7.5.4 Snoop function

The TX39 Processor Core has a snoop function. It is a function to maintain consistency between the data cache of the TX39 Processor Core and the data of an external memory.
When the snoop function of the TX39 Processor Core works, the TX39 Processor Core watches the bus operation of the bus master; and when the address in the write operation matches the address inside the data cache, it makes the corresponding data inside the data cache invalid. By doing so, it prevents the contents being different from the data cache when the contents of the external memory are rewritten by the bus master.
The TX3904 built-in DMAC and the external bus master can select whether or not to use the snoop function of the TX39 Processor Core.
If the snoop function is used, the consistency between the data cache and the data in the external memory can be maintained. In this case, when the bus master has a bus ownership, the pipeline shall be stalled if the TX39 Processor Core tries to access the data cache or when an instruction cache miss occurs.
If the snoop function is not used, the consistency between the data cache and the data in the external memory cannot be guaranteed. The TX39 processor core issues the bus ownership release request then refills the cache memory. However, when the bus master does not reply to that request, the pipeline shall be stalled because the TX39 Processor Core cannot execute the bus operation.

7.6 Interrupts

In the TX3904, there are eight interrupt signals (INT[7:0]) and one non-maskable interrupt signal (NMI*).

7.6.1 INT[7:0]

The INT[7:0] is a signal to request for an interrupt of the TX3904. It is used when the external circuit requests an interrupt. The active status of INT[7:0] is set up in the CConR of the EBIF. There are four kinds in the active status--the rising edge, the falling edge, the low level, and the high level.
The interrupt request by the INT[7:0] signal is arbitrated in the IRC and informed to the TX39 Processor Core. When it is informed to the TX39 Processor Core, an interrupt exception occurs. The priority (interrupt level) of the INT[7:1] interrupt can be set up in the register of the IRC. Interrupt requests by INT[0] shall be informed to the TX39 Processor Core without going via the IRC.
When the active status of the INT[7:0] is set up at the low level or the high level, the interrupt request must be cleared at the origin of the interrupt request. When the active status is set up at the rising edge or the falling edge, clear the interrupt request in the EIClr field of the CConR.

7.6.2 NMI*

The NMI* is a non-maskable interrupt request signal. This interrupt cannot be masked. When the NMI* is asserted, the TX39 Processor Core generates a non-maskable interrupt exception.
When the TX3904 built-in timer is operating as a watchdog timer and when the WR bit of the CConR is 0, a non-maskable interrupt is generated when a watchdog timer interrupt occurs.

7.7 Reset

By keeping the RESET* signal at low successively for 12 SYSCLK or more, the TX3904 shall be initialized. All the values in the TX3904 internal registers become the initial values. The TX39 Processor Core generates a reset exception.
When the TX3904 built-in timer is operating as a watchdog timer and when the WR bit of the CConR is 1, the TX3904 shall be reset when a watchdog timer interrupt occurs.

8 DRAM CONTROLLER (DRAMC)

8.1 Features

The signals and timings that will be necessary to control the DRAM are generated.

(1) Two-channel support

Up to four banks can be structured on a channel.

(2) Independent timing set-up is possible for each channel

Set-up by the bus master (CPU, internal DMAC, external bus master) is possible

Set-up of the number of waits: 0-5 waits

The set-up of the number of waits in the page mode: 0-3 waits

The number of penalty cycles at the page hit miss

(3) Independent size set-up is possible for each channel

1/2/4/8/16/32/64MB

The size of the bank must be a quarter of the channel size.

(4) Support of 32/16-bit static bus sizing

The static bus sizing to 32/16-bit bus is possible. The bus width of the DRAM is designated at the register set-up after reset.

(5) The fast page mode is supported

(6) The hyper page mode (EDO) DRAM is supported.

(7) CBR Refresh and CBR Self Refresh

In the normal operation, the CAS before RAS refresh is conducted. In the halt mode, the CAS before RAS self refresh can be conducted. (Only when a DRAM with self-refresh function is used.)

8.2 Block Diagrams

The following Figure 8-1 shows the connection of the DRAMC inside the TX3904 and Figure 8-2 shows the internal blocks:

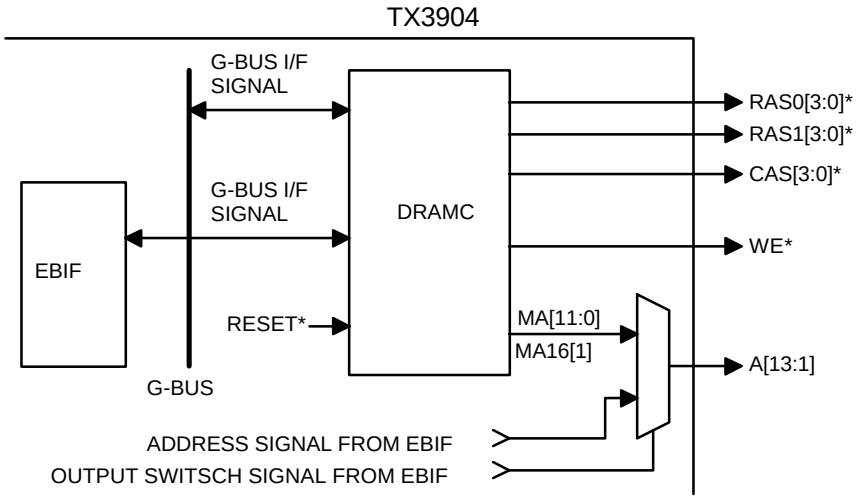


Fig. 8-1 DRAMC Connection Inside TX3904

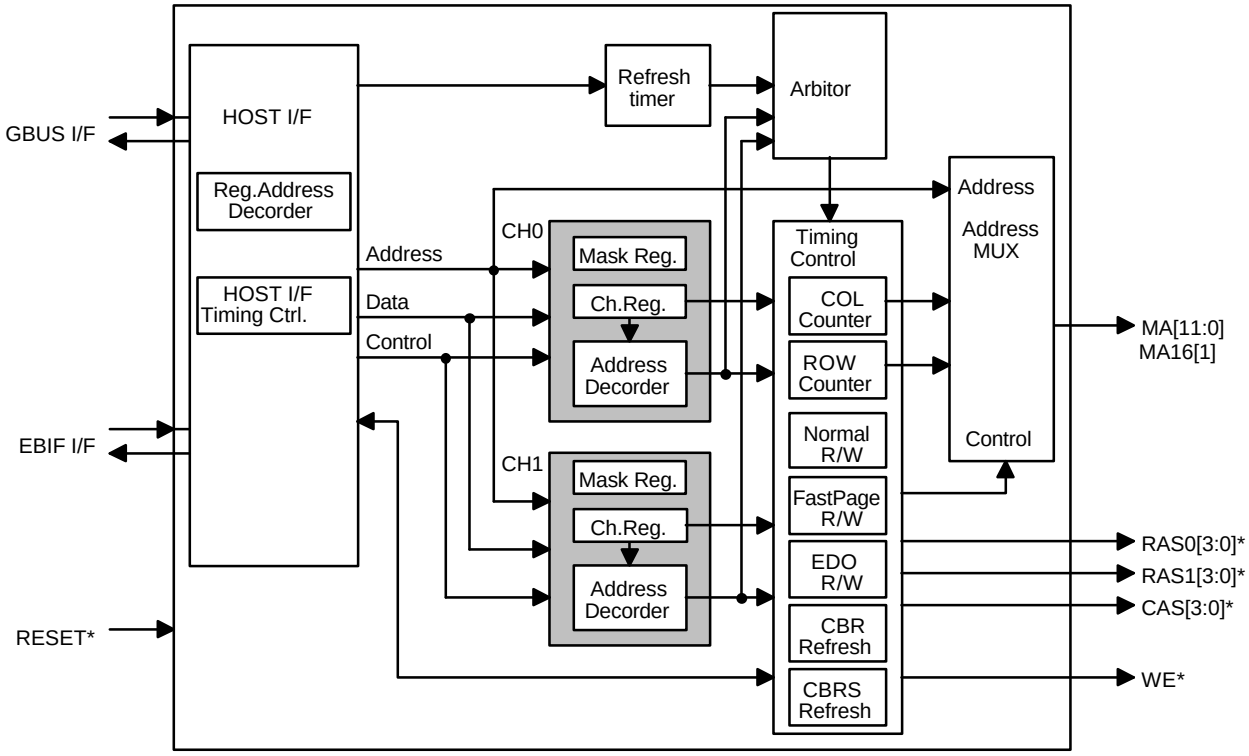


Fig. 8-2 DRAMC Block Diagram

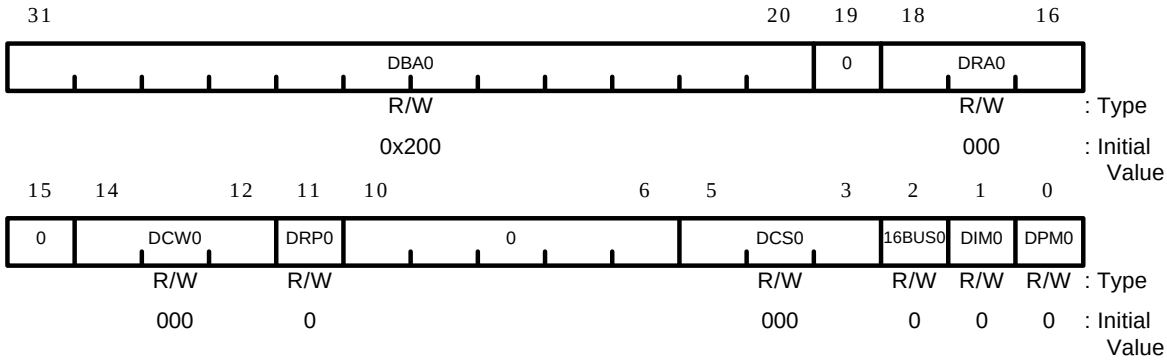
8.3 Registers

Table 8-1 DRAMC Registers

Address	Register Symbol	Register Name
0xFFFF_8000	DCCR0	Channel Control Register 0
0x FFFF_8004	DBMR0	Base Address Mask Register 0
0x FFFF_8008	DWR0	Wait Register 0
0x FFFF_8100	DCCR1	Channel Control Register 1
0x FFFF_8104	DBMR1	Base Address Mask Register 1
0x FFFF_8108	DWR1	Wait Register 1
0x FFFF_8800	DREFC	Refresh Control Register

8.4 Explanations of Registers

8.4.1 Channel control register 0 (DCCR0)



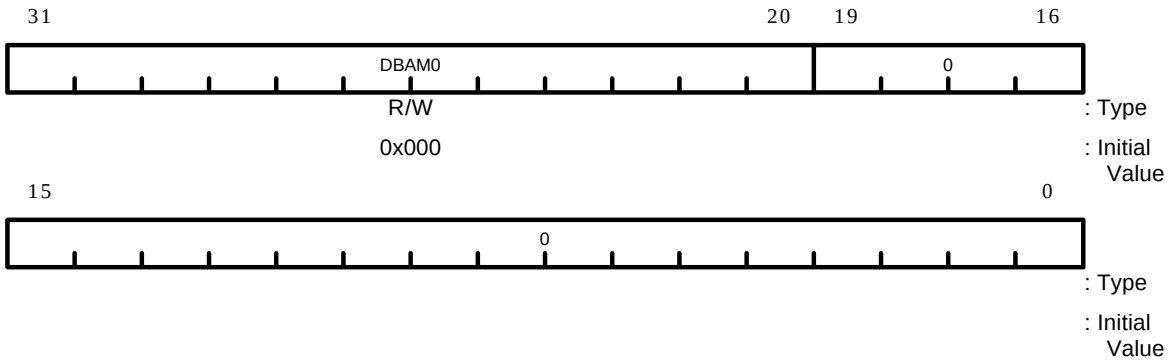
Bit	Mnemonic	Name of Field	Description
31:20	DBA0	DRAM channel 0 base address	DRAM Control Base Address on Channel 0 Designates the base address of Channel 0 using physical address.
18:16	DRA0	DRAM channel 0 row address size	DRAM Control Row Address on Channel 0 Designates the row direction size of the DRAM to be connected to each bank on Channel 0. 000: 512 rows 011: 4096 rows 001: 1024 rows 1**: Reserved 010: 2048 rows
14:12	DCW0	DRAM channel 0 column address size	DRAM Control Column Word on Channel 0 Designates the column direction size of the DRAM to be connected to each bank on Channel 0. 000: 128 words 10*: 2048 words 001: 256 words 11*: Reserved 010: 512 words 011: 1024 words
11	DRP0	DRAM channel 0 RAS precharge time	DRAM RAS Precharge Wait Time on Channel 0 RAS precharge for the following numbers of cycles is performed when a page hit miss occurs while in the page mode, and when recovering from a refresh cycle. 0 : 3 cycles 1 : 2 cycle

Fig. 8-3 DRAM Control Register Channel 0 (1/2)

Bit	Mnemonic	Name of Field	Description
5:3	DCS0	DRAM channel 0 size	DRAM Control Channel Size on Channel 0 Designates the total memory size to be assigned to all of four banks on Channel 0. 000: 1 Mbytes 100: 16 Mbytes 001: 2 Mbytes 101: 32 Mbytes 010: 3 Mbytes 11*: Reserved 011: 8 Mbytes
2	16BUS0	DRAM channel 0 bus size	DRAM Control 16-bit Width Bus Size on Channel 0 Designates the bus width of the memory to be connected to Channel 0. 1: 16-bit width bus size 0: 32-bit width bus size
1	DIM0	Reserved	This bit is reserved. Do not set to 1.
0	DPM0	DRAM channel 0 page mode select	DRAM Control Page Mode on Channel 0 Designates the page mode of the DRAM connected to Channel 0. 1: Extended data output (EDO) mode 0: Fast page mode

Fig. 8-4 DRAM Control Register Channel 0 (2/2)

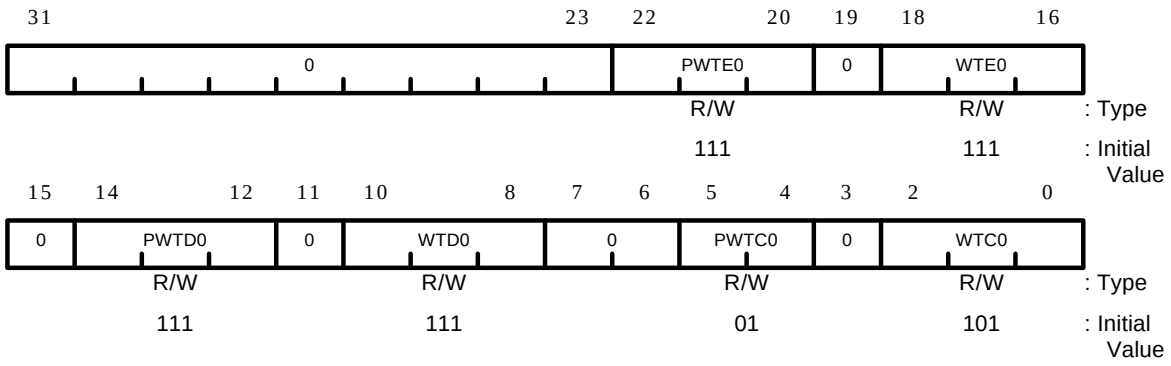
8.4.2 Base address mask register 0 (DBMR0)



Bit	Mnemonic	Name of Field	Description
31:20	DBAM0	DRAM channel 0 base address mask	DRAM Control Base Address Mask on Channel 0 Specifies the valid bit of the address comparison by the DBA0 field of the channel control register. 1: Bit of the corresponding DBA0 field is not compared. 0: Bit of the corresponding DBA0 field is compared.

Fig. 8-5 DRAM Channel 0 Base Address Mask Register

8.4.3 Wait register 0 (DWR0)



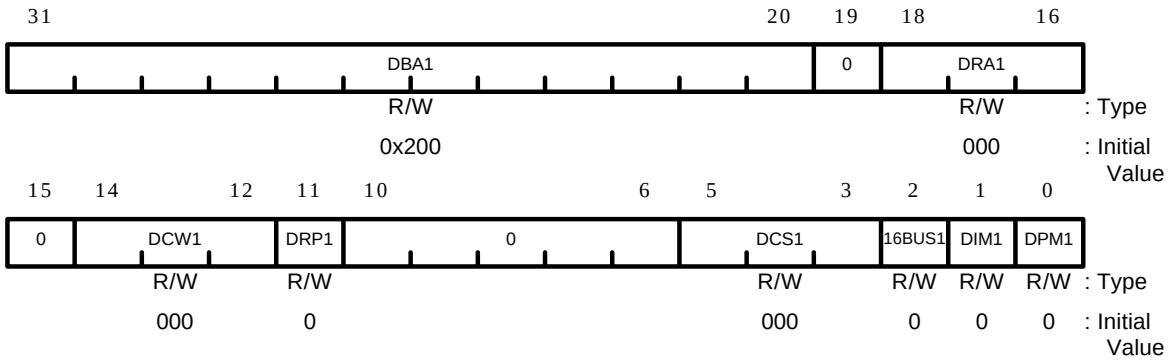
Bit	Mnemonic	Name of Field	Description
22:20	PWTE0	DRAM channel 0 external bus master page mode wait	Page Mode Wait Cycle for External Bus Master Designates the number of wait cycles with which the external bus master accesses the Channel 0 DRAM in the page mode (burst mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
18:16	WTE0	DRAM channel 0 external bus master normal mode wait	Normal Mode Wait Cycle for External Bus Master Designates the number of wait cycles with which the external bus master accesses the Channel 0 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
14:12	PWTD0	DRAM channel 0 internal DMAC page mode wait	Page Mode Wait Cycle for Internal DMAC Designates the number of wait cycles with which the internal DMAC accesses the Channel 0 DRAM in the page mode (burst mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits

Fig. 8-6 DRAM Channel 0 Wait Register (1/2)

Bit	Mnemonic	Name of Field	Description
10:8	WTD0	DRAM channel 0 internal DMAC normal mode wait	Normal Mode Wait Cycle for Internal DMAC Designates the number of wait cycles with which the internal DMAC accesses the Channel 0 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
5:4	PWTC0	DRAM channel 0 CPU page mode wait	Page Mode Wait Cycle for CPU Designates the number of wait cycles with which the CPU accesses the Channel 0 DRAM in the page mode (burst mode). 00: 0 wait 10: 2 waits 01: 1 wait 11: 3 waits
2:0	WTC0	DRAM channel 0 CPU normal mode wait	Normal Mode Wait Cycle for CPU Designates the number of wait cycles with which the CPU accesses the Channel 0 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits

Fig. 8-7 DRAM Channel 0 Wait Register (2/2)

8.4.4 Channel control register 1 (DCCR1)



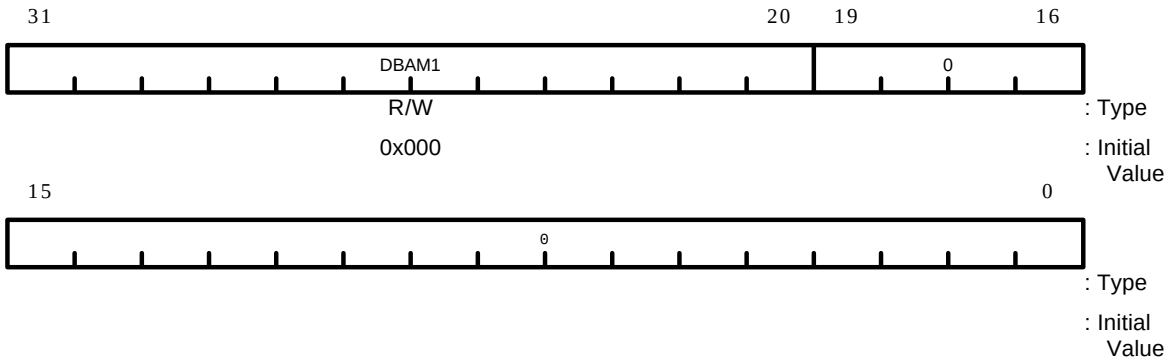
Bit	Mnemonic	Name of Field	Description
31:20	DBA1	DRAM channel 1 base address	DRAM Control Base Address on Channel 1 Designates the base address of Channel 1 using physical address.
18:16	DRA1	DRAM channel 1 row address size	DRAM Control Row Address on Channel 1 Designates the row direction size of the DRAM to be connected to each bank on Channel 1. 000: 512 rows 011: 4096 rows 001: 1024 rows 1**: Reserved 010: 2048 rows
14:12	DCW1	DRAM channel 1 column address size	DRAM Control Column Word on Channel 1 Designates the column direction size of the DRAM to be connected to each bank on Channel 1. 000: 128 words 10*: 2048 words 001: 256 words 11*: Reserved 010: 512 words 011: 1024 words
11	DRP1	DRAM channel 1 RAS precharge time	DRAM RAS Precharge Wait Time on Channel 1 RAS precharge for the following numbers of cycles is performed when a page hit miss occurs while in the page mode, and when recovering from a refresh cycle. 0 : 3 cycles 1 : 2 cycle

Fig. 8-8 DRAM Control Register Channel 1 (1/2)

Bit	Mnemonic	Name of Field	Description
5:3	DCS1	DRAM channel 1 size	DRAM Control Channel Size on Channel 1 Designates the total memory size to be assigned to all of four banks on Channel 1. 000: 1 Mbytes 100: 16 Mbytes 001: 2 Mbytes 101: 32 Mbytes 010: 3 Mbytes 11*: Reserved 011: 8 Mbytes
2	16BUS1	DRAM channel 1 bus size	DRAM Control 16-bit Width Bus Size on Channel 1 Designates the bus width of the memory to be connected to Channel 1. 1: 16-bit width bus size 0: 32-bit width bus size
1	DIM1	Reserved	This bit is reserved. Do not set to 1.
0	DPM1	DRAM channel 1 page mode select	DRAM Control Page Mode on Channel 1 Designates the page mode of the DRAM connected to Channel 1. 1: Extended data output (EDO) mode 0: Fast page mode

Fig. 8-9 DRAM Control Register Channel 1 (2/2)

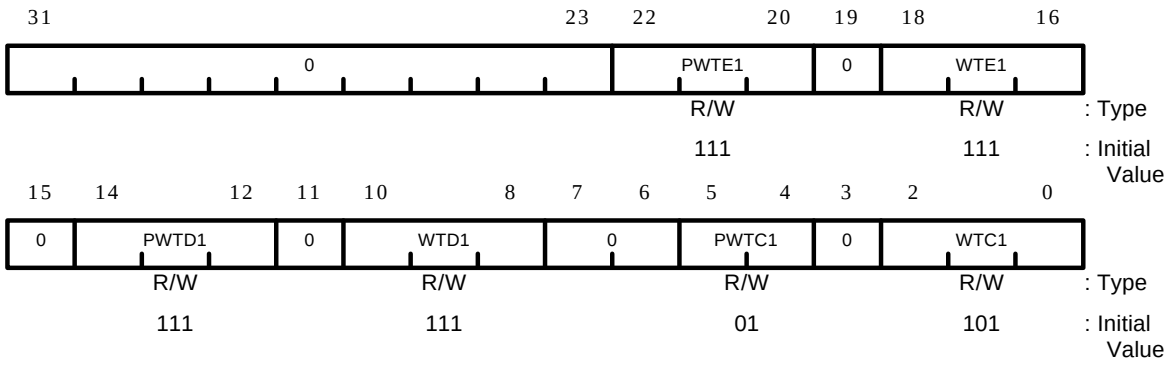
8.4.5 Base address mask register 1 (DBMR1)



Bit	Mnemonic	Name of Field	Description
31:20	DBAM1	DRAM channel 1 base address mask	DRAM Control Base Address Mask on Channel 1 Specifies the valid bit of the address comparison by the DBA1 field of the channel control register. 1: Bit of the corresponding DBA1 field is not compared. 0: Bit of the corresponding DBA1 field is compared.

Fig. 8-10 DRAM Channel 1 Base Address Mask Register

8.4.6 Wait register 1 (DWR1)



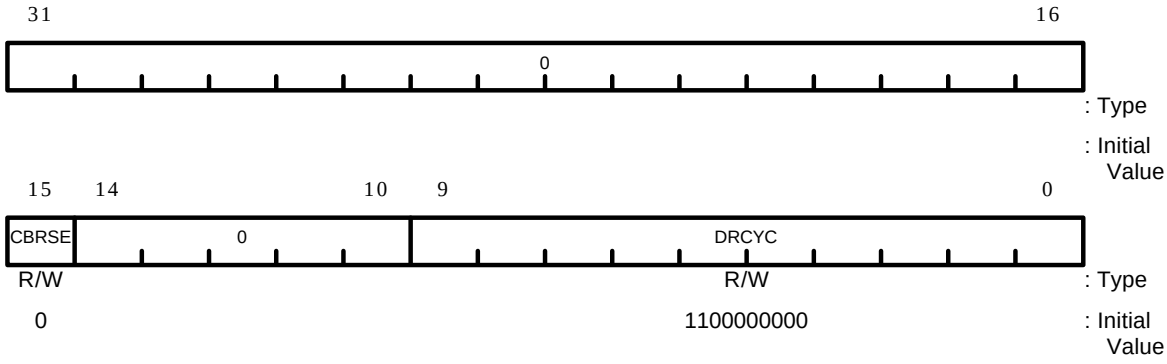
Bit	Mnemonic	Name of Field	Description
22:20	PWTE1	DRAM channel 1 external bus master page mode wait	Page Mode Wait Cycle for External Bus Master Designates the number of wait cycles with which the external bus master accesses the Channel 1 DRAM in the page mode (burst mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
18:16	WTE1	DRAM channel 1 external bus master normal mode wait	Normal Mode Wait Cycle for External Bus Master Designates the number of wait cycles with which the external bus master accesses the Channel 1 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
14:12	PWTD1	DRAM channel 1 internal DMAC page mode wait	Page Mode Wait Cycle for Internal DMAC Designates the number of wait cycles with which the internal DMAC accesses the Channel 1 DRAM in the page mode (burst mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits

Fig. 8-11 DRAM Channel 1 Wait Register (1/2)

Bit	Mnemonic	Name of Field	Description
10:8	WTD1	DRAM channel 1 internal DMAC normal mode wait	Normal Mode Wait Cycle for Internal DMAC Designates the number of wait cycles with which the internal DMAC accesses the Channel 1 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits
5:4	PWTC1	DRAM channel 1 CPU page mode wait	Page Mode Wait Cycle for CPU Designates the number of wait cycles with which the CPU accesses the Channel 1 DRAM in the page mode (burst mode). 00: 0 wait 10: 2 waits 01: 1 wait 11: 3 waits
2:0	WTC1	DRAM channel 1 CPU normal mode wait	Normal Mode Wait Cycle for CPU Designates the number of wait cycles with which the CPU accesses the Channel 1 DRAM in the normal mode (single mode). 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: 7 waits

Fig. 8-12 DRAM Channel 1 Wait Register (2/2)

8.4.7 Refresh control register (DREFC)



Bit	Mnemonic	Name of Field	Description
15	CBRSE	Self refresh enable	CBR Self Refresh Enable Designates whether or not to use the DRAM self-refresh function in the halt mode. CBRSE Function 0 Disable self refresh 1 Enable self refresh
9:0	DRCYC	Refresh cycle	DRAM Control Refresh Cycle Specifies the refresh cycle. Default is $(1100000000)_2$. It's 15.36 µsec when the internal system clock is 50 MHz (20 nsec per cycle). Specify the clock count of the internal system clock in binary.

Fig. 8-13 Refresh Cycle Register

8.5 Operations

8.5.1 Channel select

The channel select for DRAM access is conducted by the physical address. Of the addresses, the high-order 12 bits are compared if they are within the range from the base address DBAn (n=1, 0) that is set up in the channel control register DCCRn (n=1, 0) to the size that is set up in the size register DCSn (n=1, 0), and if they are in the range, the channel is selected.
The base address can be masked. The bit masked is not compared as an address. Masking is conducted by the base address mask register that designates at will a mask with 12-bit base address.
The DRAMC selects the channel and bank according to the following table:

Table 8-2 Address Decode

Ch. Size	Bank Size	Ch. Select Base Addr.bit	Bank3	Bank2	Bank1	Bank0
1M	256K	A31:20	A19:18=11	A19:18=10	A19:18=01	A19:18=00
2M	512K	A31:21	A20:19=11	A20:19=10	A20:19=01	A20:19=00
4M	1M	A31:22	A21:20=11	A21:20=10	A21:20=01	A21:20=00
8M	2M	A31:23	A22:21=11	A22:21=10	A22:21=01	A22:21=00
16M	4M	A31:24	A23:22=11	A23:22=10	A23:22=01	A23:22=00
32M	8M	A31:25	A24:23=11	A24:23=10	A24:23=01	A24:23=00
64M	16M	A31:26	A25:24=11	A25:24=10	A25:24=01	A25:24=00

SIZE	A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18
1M	*	*	*	*	*	*	*	*	*	*	*	*	#	#
2M	*	*	*	*	*	*	*	*	*	*	*	#	#	
4M	*	*	*	*	*	*	*	*	*	*	#	#		
8M	*	*	*	*	*	*	*	*	*	#	#			
16M	*	*	*	*	*	*	*	*	#	#				
32M	*	*	*	*	*	*	*	#	#					
64M	*	*	*	*	*	*	#	#						

*: The bit to be compared with the base address to select the channel
#: To be used to select the bank

Depending on the set-up of the base address and size of each channel, two channels are sometimes selected. In such a case, the channel to which a higher base address is set up is selected. When accessing the addresses shown in Figure 8-10, Channel 1 with a higher base address value is selected. When the same base address is set up, Channel 0 has the first priority to be selected.

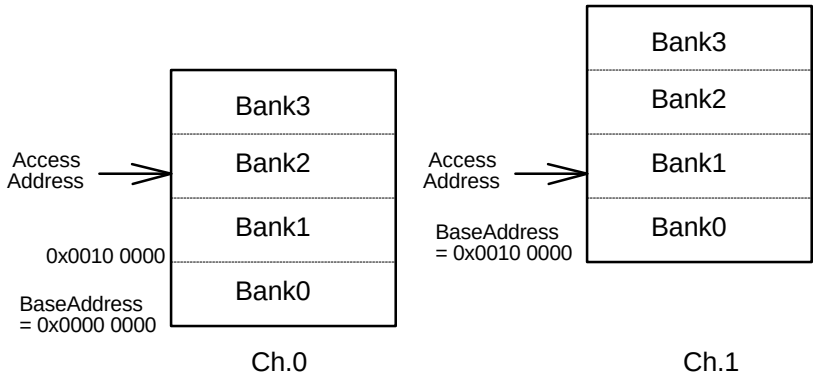


Fig. 8-14 Process at the Time of Dual Channel Select

8.5.2 Address multiplex

The multiplex of the address allocates the upper addresses to the row address and the lower addresses to the column address to make page mode access effective.
The number of the column address bits is determined by the DCWn (n=0,1) of the DCCRn. When the bus width of the DRAM is 16 bits, A[m:1] (m=11,10,9,8,7) becomes the column address, and the high-order bit that starts from the A[m+1] (m=11,10,9,8,7) becomes the row address. The number of bits of the row address is designated by the DRAn (n=1,0).

Table 8-2 Row Address and Column Address

Row Address			Column Address		
DRAn	Row Size	Number of Bit	DCWn	Number of Words in Column Direction	Number of Bit
000	512	9	000	128	7
001	1024	10	001	256	8
010	2048	11	010	512	9
011	4096	12	011	1024	10
1**	Reserved	N/A	10*	2048	11
-	-	-	11*	Reserved	N/A

8.5.3 Operation modes

Table 8-3 Bus Operation of Bus Master and Memory Operation Mode

Memory Device Operation Mode	Access Type	Data Length	Device	Bus Master (CPU, DMAC)			
				Single Operation		Burst Operation	
				32-bit Bus	16-bit Bus	32-bit Bus	16-bit Bus
Normal Mode	R/W	W HW B	FP HP	O		X	
Fast Page Mode	R/W	W	FP	O*		O	
Hyper Page Mode (EDO)	R/W	W	HP	O*		O	

O: Supported with DRAMC X: Not supported

Data length: W=word, HW=half word, B=byte

Device: FP=Fast page DRAM, HP=Hyper page DRAM (EDO)

*: The single operation is the same as the normal mode.

Single Operation

At the time of access to uncache area.

When the data cache refill size is 1.

Burst Operation

At the time of data cache burst refill

At the time of instruction cache refill

At the time of burst access by the DMAC

8.5.4 32/16-bit static bus sizing

Supports the DRAM of 16-bit bus width by setting up 16BUS0 and 16BUS1 of the DCCRn (n=1,0).

32-bit Bus Access
Indicates that the DRAM connected to the channel is 32-bit bus when the 16BUSn of the DRAM channel control register is 0. It can conduct the most efficient memory access.

Word/triple-byte access with 16-bit width DRAM
Indicates that the DRAM connected to the channel is 16-bit bus when the 16BUSn of the DRAM channel control register is 1. At this time, the DRAMC conducts a memory access twice at the timing set up in the register when a word or triple-byte access is requested. The second access shall be page mode.

Half-word/byte access with 16-bit width DRAM
If a half-word or byte access is requested when the 16BUSn of the DRAM channel control register is 1, the DRAMC conducts a memory access once at the timing set up in the register.

8.5.5 Support for external bus master

In the TX3904, the DRAM memory address is multiplexed with the address bus signal. When the external bus master conducts a memory access using the DRAMC, it is necessary to avoid conflicts between the DRAM memory address output and the address signal that is output by the external bus master.
The external bus master is expected to manage not to generate a page hit miss when it uses the DRAMC. The accessing address must be word (32-bit) boundary. The number of data transfer in burst mode must be a multiple of 32 bits.

8.5.6 Support for half speed bus

The signals generated for DRAM (address, data, RAS* and CAS*) is synchronized to a rising edge of a processor clock (internal clock) in half speed bus mode. So these signals are not synchronized to a rising edge of the SYSCLK.
When the external bus master conducts a memory access through the DRAMC in the half speed mode, set up a wait in the DRAMC channel control register such that the number of the DRAM access cycles (the number of GCLK cycles) is an even number.

8.5.7 Page mode support and page hit detection

The page mode of the DRAM is a method to access while keeping asserting RAS* and changing the column address under the fixed row address. In the DRAMC, when the bus master conducts the burst mode access, the page mode of the DRAM is used. At the single word access, the page mode access of the DRAM is not conducted. The DRAMC supports two modes--the fast page mode and the hyper page mode (EDO). Selection of the fast page mode and hyper page mode is conducted in the DPMn (n=1,0) of the channel control register.
When the row address changes (at a page hit miss), the DRAMC negates the RAS*. The page hit miss is caused when the row address changes and the page mode access still continues. After the page hit miss, one access cycle of the normal mode is executed and the page mode access is continued.

8.5.8 Column address counter

A 12-bit column address counter is used for the fast page mode and the hyper page mode. When the page mode access continues beyond the number of the column words that is set up in the DCWn (n=0,1) of the channel control register, a page hit miss occurs. Users cannot access this column address counter.
When the bus master conducts the burst mode access, the column address counter increments address like as 0-4-8-C-10-... if the lower four-bit of burst starting address is 0x0. On the other hand, it decrements address like as C-8-4-0-... if the starting address is 0xC.

8.5.9 Timing control

The timing control controls the switch timing of the row/column addresses and the timing of RAS*/CAS* and WE*.

8.5.10 Refresh timing

The refresh supports the CAS before RAS refresh (CBR) and the CAS before RAS self refresh (CBRS).
Normally, the CBR refresh is conducted. When a DRAM with the self refresh function is used and the self refresh enable CBRSE of the refresh control register DREFC is 1, a self refresh is conducted in the first refresh cycle after the TX3904 entered into the halt mode, and the refresh timer is halted. When the halt mode is resolved, it recovers from the self refresh to the CBR refresh. Please note that the contents of the DRAM are not retained when having shifted into the self refresh mode using a DRAM without the self refresh function.
The refresh is conducted for Channel 0 and Channel 1 simultaneously. The refresh timer is 10 bits and programmable. Designate the number of clocks of the internal system clock with a binary value for desired refresh cycle. For example, when the internal system clock is 50 MHz (20 ns cycles), 15.6 μsec is 780 clocks so that the set-up should be $(1100001100)_2$. The default is $(1100000000)_2$ for 768 clocks (when @50 MHz) at 15.36 μsec. When using at a frequency other than 50 MHz, the user is required to set up the appropriate number of cycles, taking the DRAM access and refresh arbitration into consideration.

8.5.11 Arbiter

For the arbitration of the refresh and DRAM access, the request that came first has the priority. If one of them is in operation, a new request is made to wait. The refresh during a DRAM access waits until the DRAMC replys ACK* to processor core. Therefore, refresh is conducted after the operation completion when it is a single operation and after the ACK* is asserted when it is a burst access, and the burst access is suspended. After the refresh, the suspended burst operation is resumed. The first access after the resumption is at the same timing as the normal mode.

8.5.12 Operations at the time of reset

After reset, a 8-cycle CBR refresh dummy cycle is executed such that the CBR refresh is conducted.

8.6 Timing Diagrams

8.6.1 32-bit bus single read operation

(1) DRAM access by the internal bus master

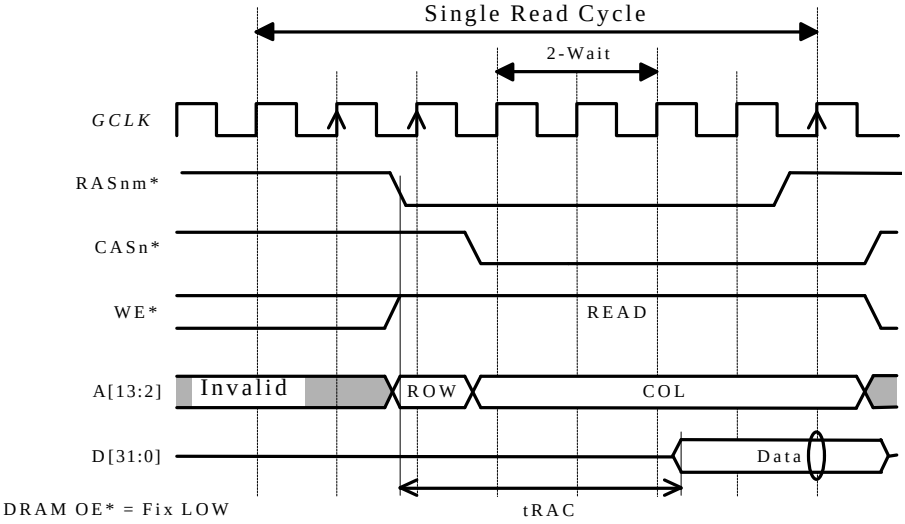


Fig. 8-15 32-bit Bus DRAM Single Read Operation

(Internal Bus Master)

(2) When the external bus master conducts a DRAM access
(Half speed bus mode: HALF*=low)

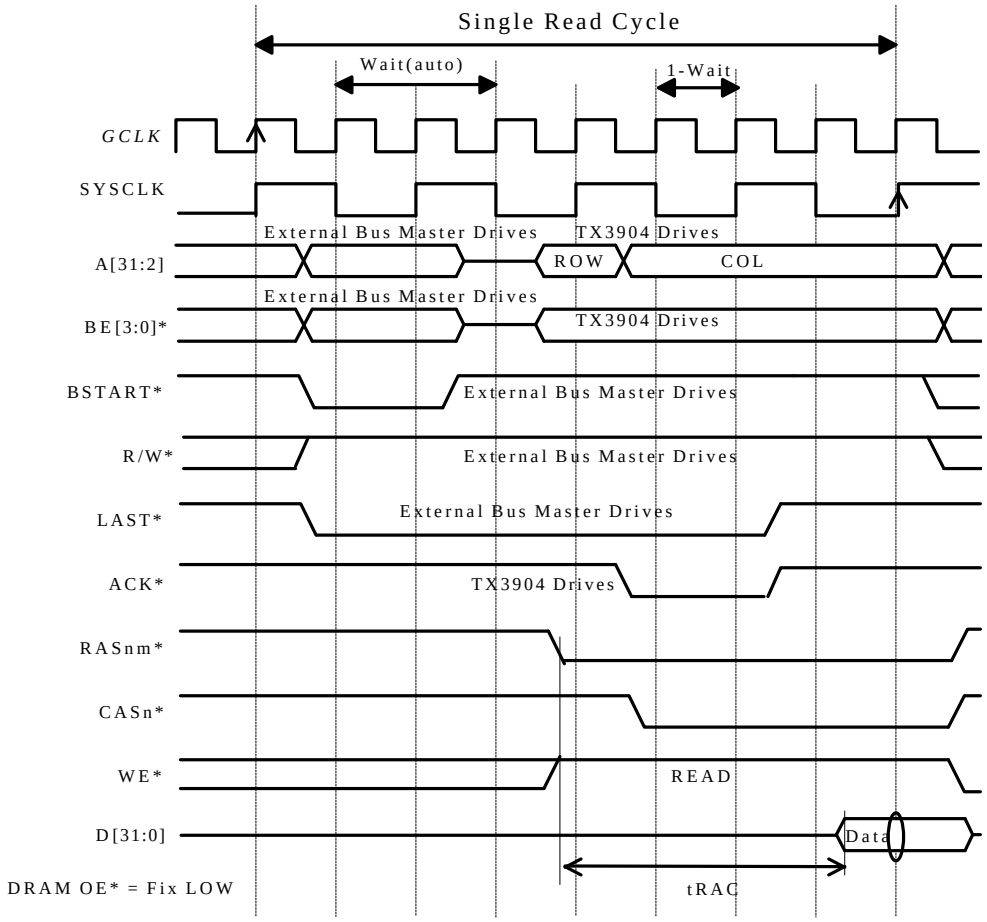


Fig. 8-16 32-bit Bus Single Read Operation

(External Bus Master; Half Speed Bus)

When the external bus master conducts a memory access using the DRAMC of the TX3904, it is necessary to avoid conflicts of address signals. The DRAMC automatically inserts a wait cycle of 1 SYSCLK to avoid a conflict when it has received an access request from the external bus master. Stop the external bus master's drive of address at a rising of BSTART*.
In the half speed bus mode, output signals for memory access, except the ACK*, are not synchronized to the SYSCLK but to the internal GCLK.

(3) When the external bus master conducts a DRAM access
(Full speed bus mode; HALF*=high)

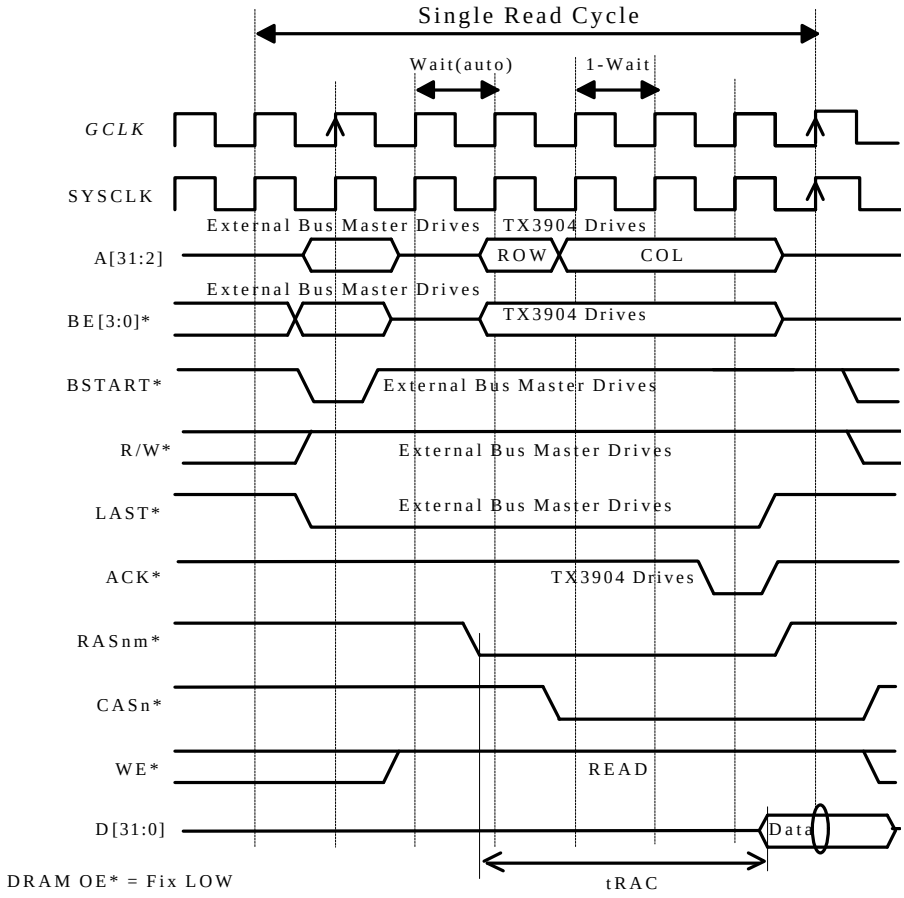


Fig. 8-17 32-bit Bus Single Read Operation

(External Bus Master; Full Speed Bus)

8.6.2 32-bit word single read operation with 16-bit bus

(1) Fast page DRAM

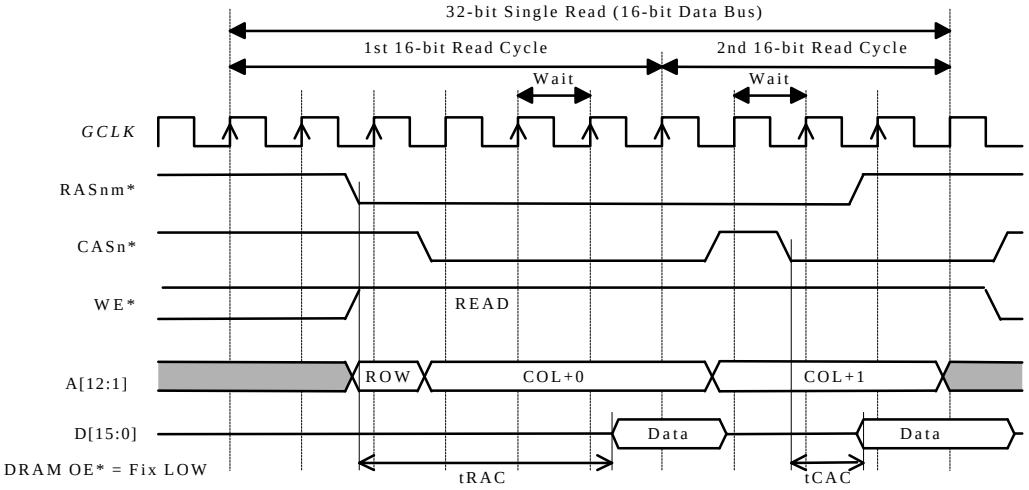


Fig. 8-18 16-bit Bus DRAM Single Read Operation

(Fast Page Mode DRAM; 1-1 Wait)

(2) Hyper page DRAM

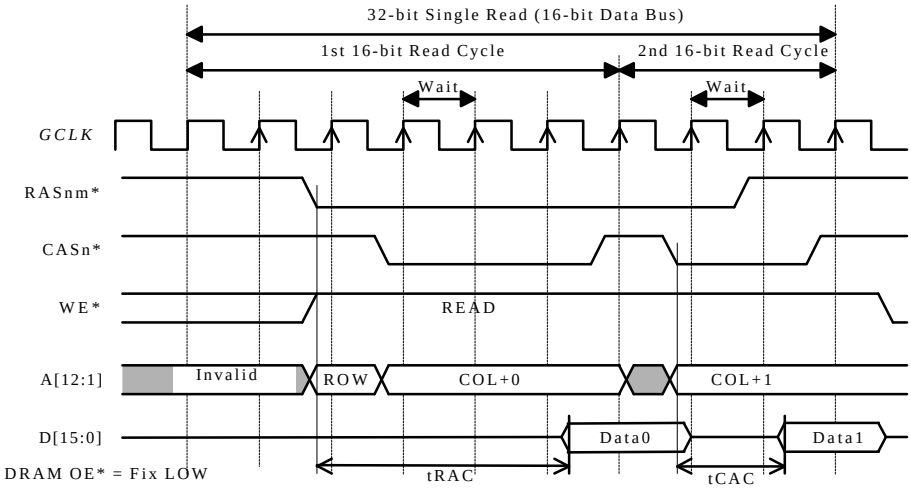


Fig. 8-19 16-bit Bus DRAM Single Read Operation

(Hyper Page Mode DRAM; 1-1 Wait)

If a 32-bit single read request is accepted when the 16BUSn of the control register is 1, the DRAMC reads 16-bit half word twice at the page read. The mode of the page read is set up in the DPM of the channel control register.

8.6.3 32-bit bus fast page mode read (Burst mode)

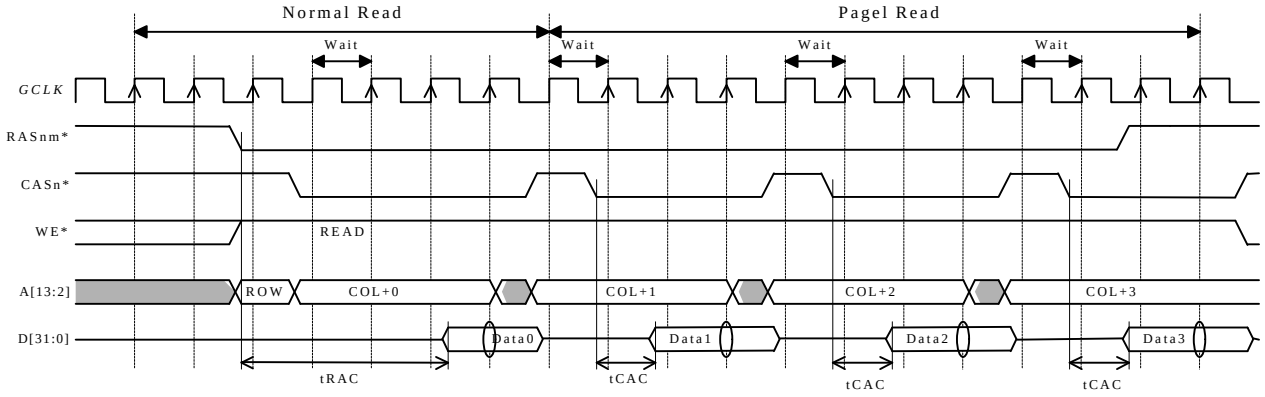


Fig. 8-20 32-bit Fast Page Read Operation

(1-1 Wait; 6-4-4-4; Burst size=4)

8.6.4 16-bit bus fast page mode word read (Burst read)

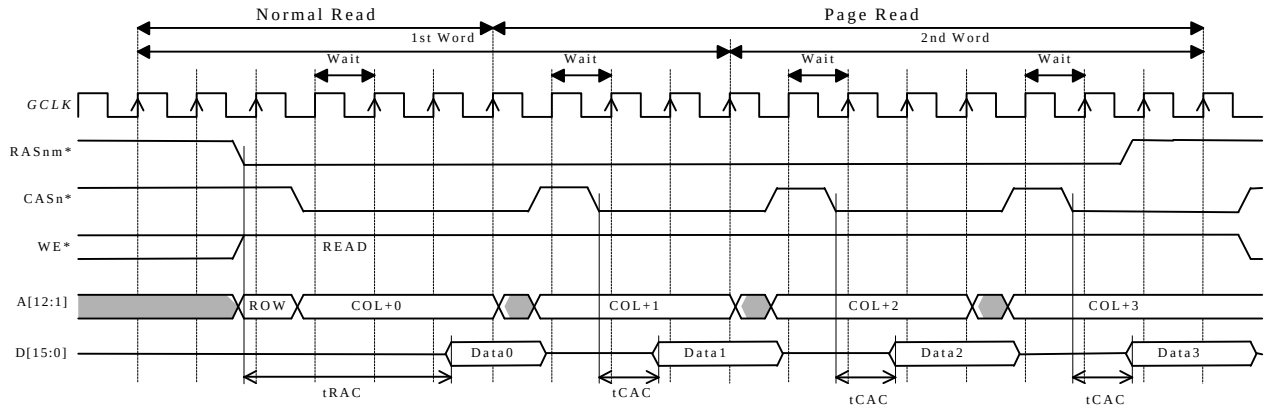


Fig. 8-21 16-bit Fast Page Read Operation

(1-1 Wait; 6-4-4-4; Burst size=4)

8.6.5 32-bit bus fast page mode read (Burst read) page hit miss

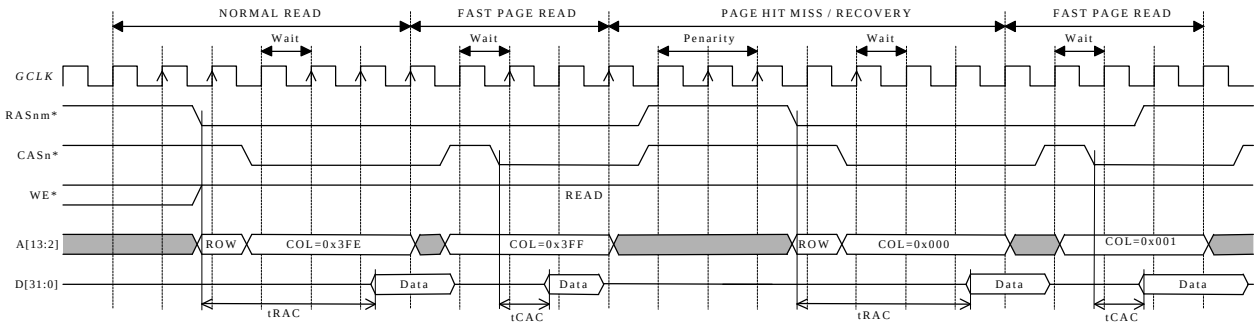


Fig. 8-22 Fast Page Read Page Hit Miss Operation

8.6.6 32-bit bus hyper page mode read (Burst read)

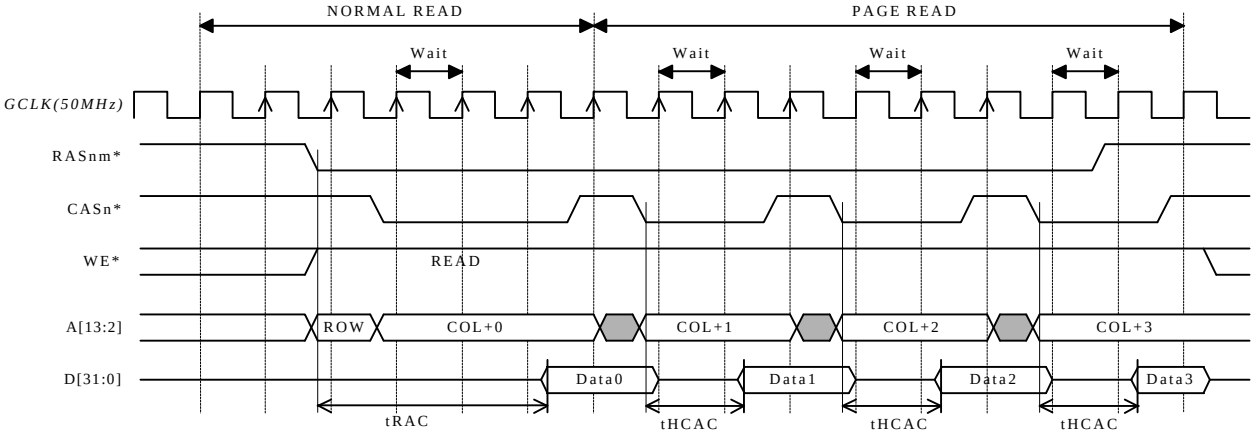


Fig. 8-23 32-bit Bus Hyper Page Mode Read Operation

(1-1 Wait; 6-3-3-3; Burst size=4)

8.6.7 32-bit bus hyper page mode read (Page hit miss)

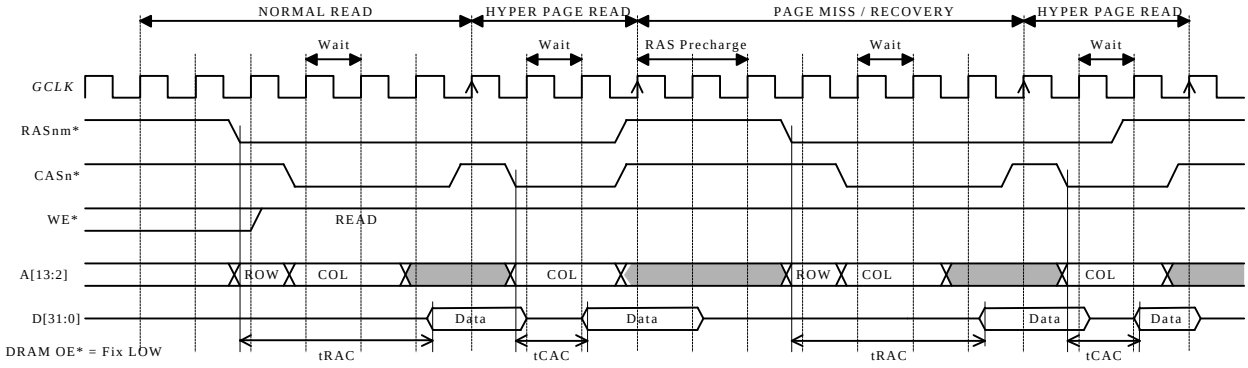


Fig. 8-24 Hyper Page Read Page Hit Miss Operation

8.6.8 32-bit bus single write (Early write)

(1) DRAM access by the internal bus master

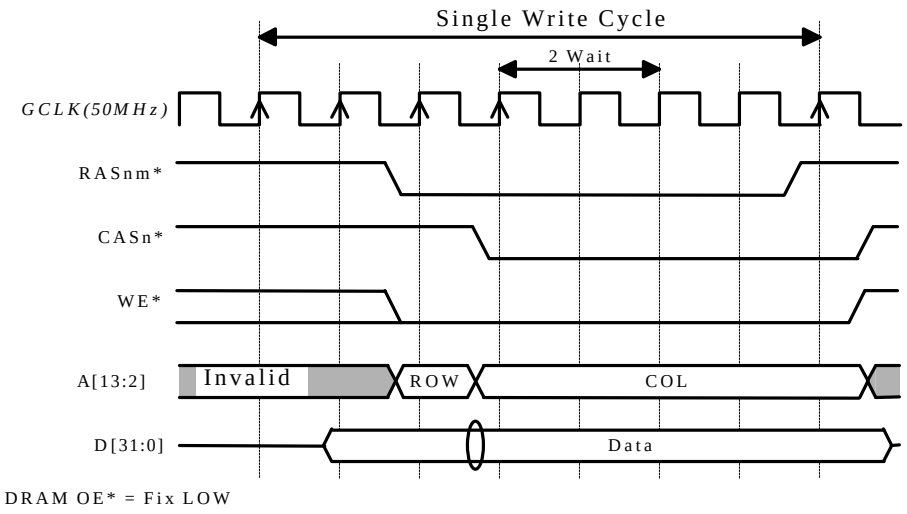


Fig. 8-25 32-bit Bus Single Write Operation

(Internal Bus Master)

(2) DRAM access by the external bus master

(Half speed bus mode; HALF*=low)

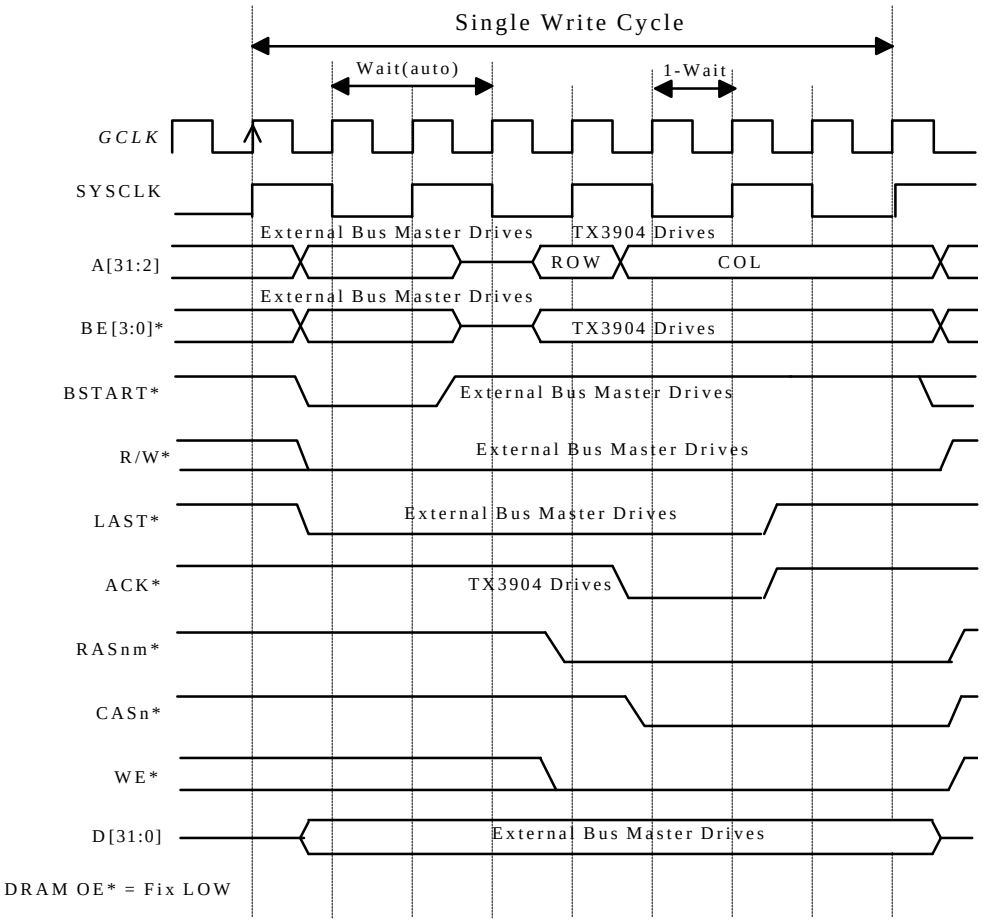


Fig. 8-26 32-bit Bus Single Write Operation

(Half Speed Bus; External Bus Master)

(3) DRAM access by the external bus master

(Full speed bus mode; HALF*=high)

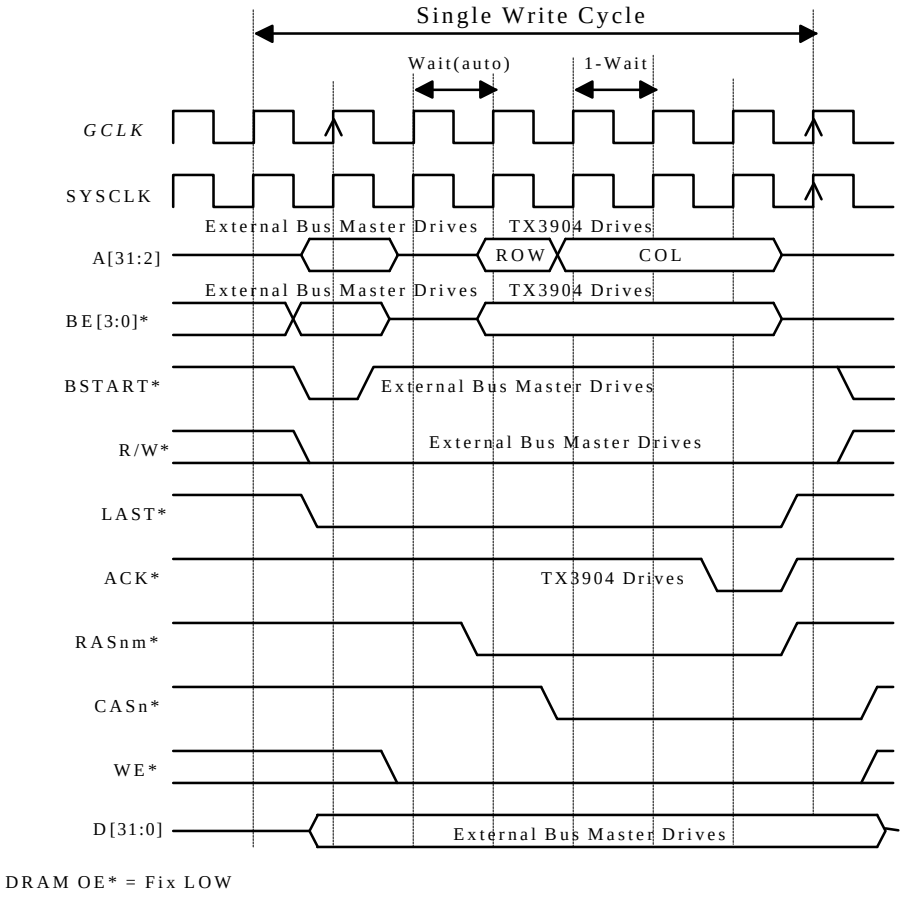


Fig. 8-27 32-bit Bus Single Write Operation

(Full Speed Bus; External Bus Master)

8.6.9 32-bit bus fast page mode write (Early write)

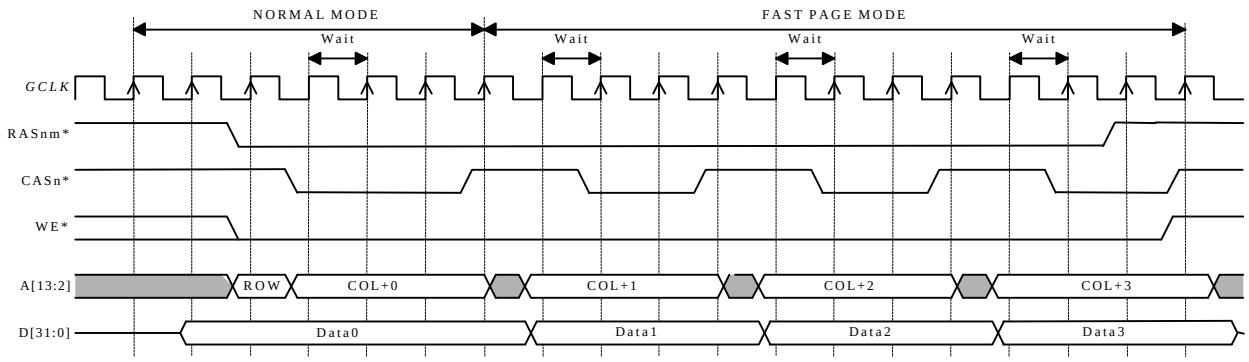


Fig. 8-28 32-bit Bus Fast Page Mode Write Operation (1-1 Wait)

8.6.10 32-bit bus hyper page mode write (Early write)

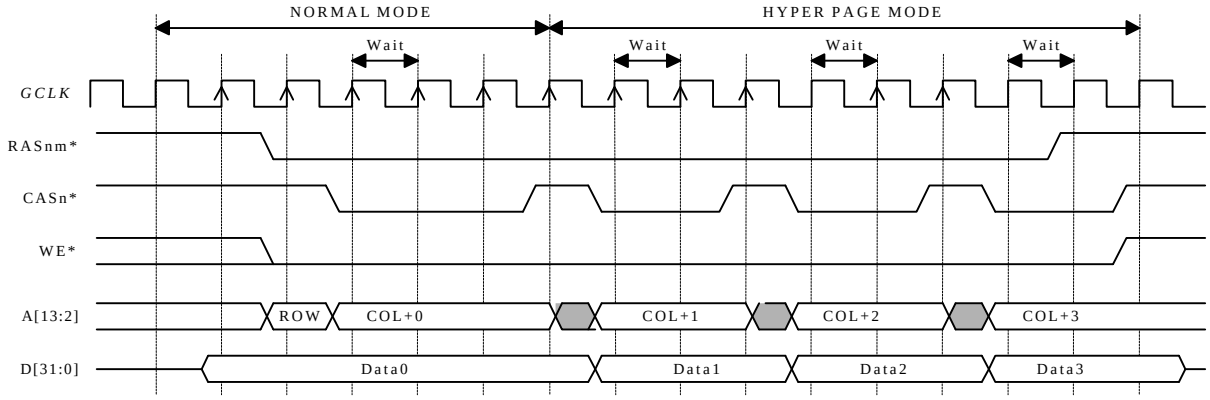


Fig. 8-29 32-bit Bus Hyper Page Mode Write Operation (1-1 Wait)

8.6.11 CBR refresh

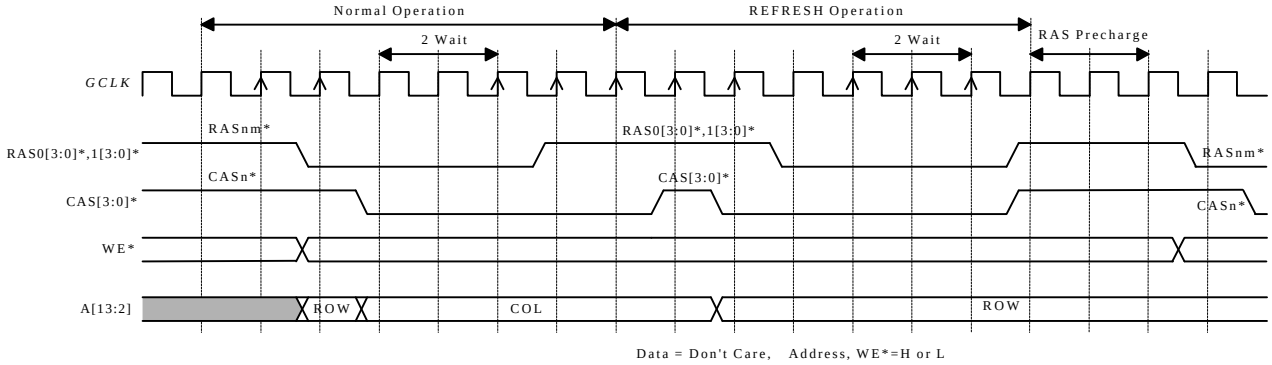


Fig. 8-30 CAS Before RAS Refresh Operation

A wait during a CBR refresh is the same as a wait in the normal mode. (WTCn of the channel to which the greater value is set up.)

The number of RAS precharge cycles after a refresh is set up in the DRPT. (DRPn of the channel to which the greater value is set up.)

8.7 External Circuit Connections

(1) 16-bit width DRAM connection

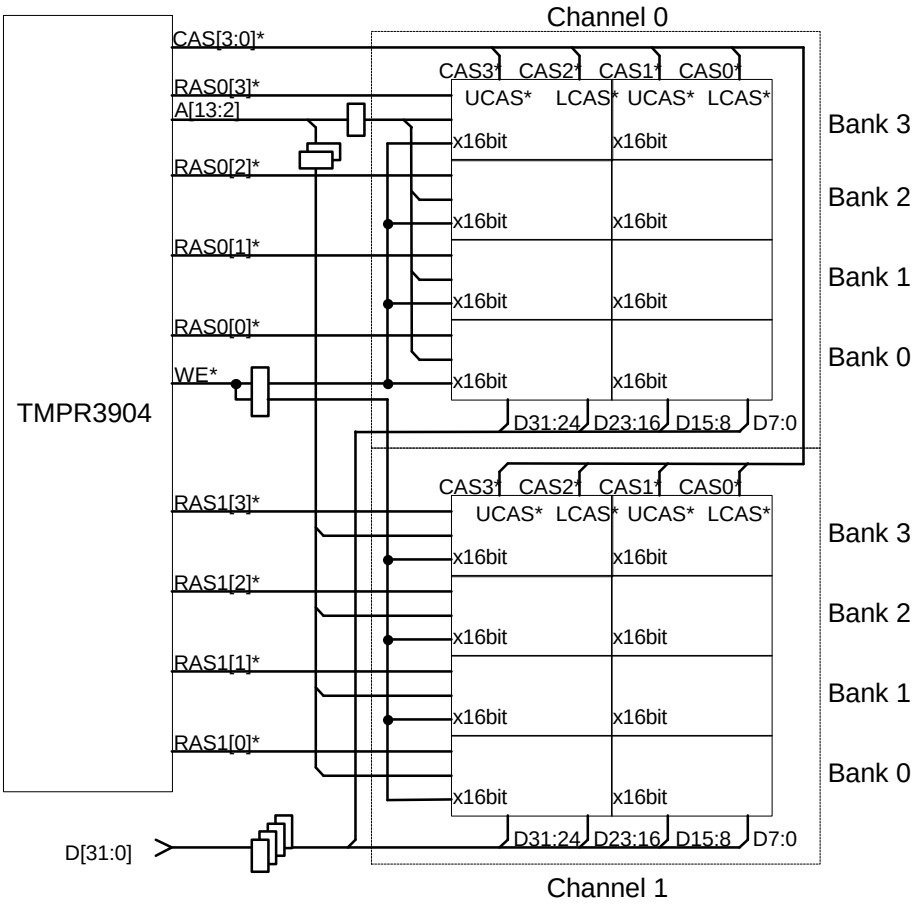


Fig. 8-31 16-bit width DRAM Connection

A necessity of buffer for address, data and WE* depends on number of DRAM devices.

(2) 32-bit width DRAM connection

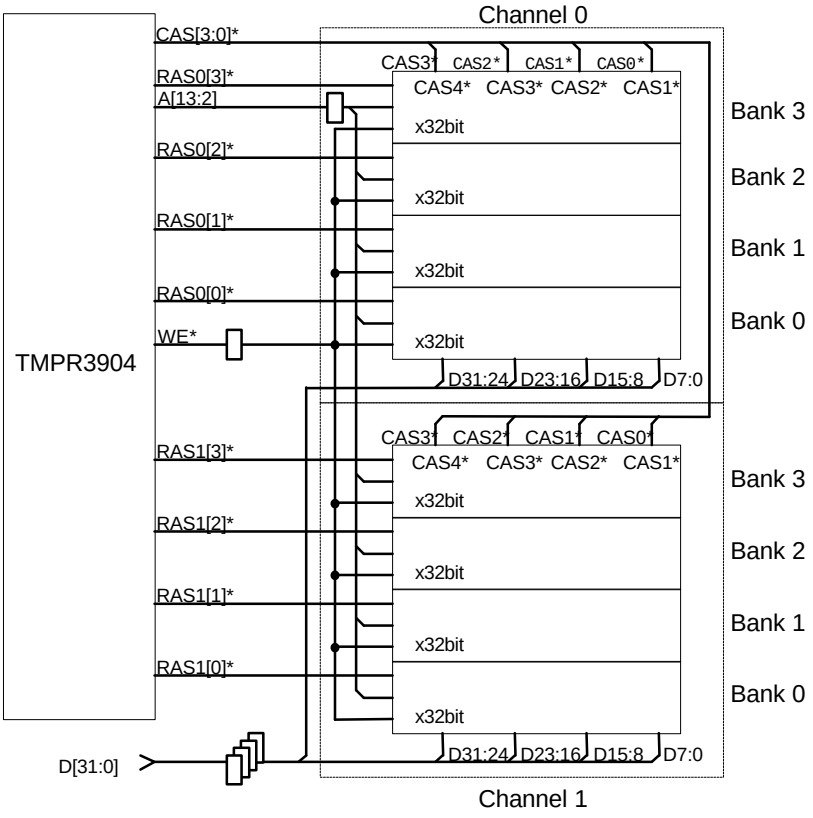


Fig. 8-32 32-bit width DRAM Connection

(3) Connection of 16-bit bus configuration

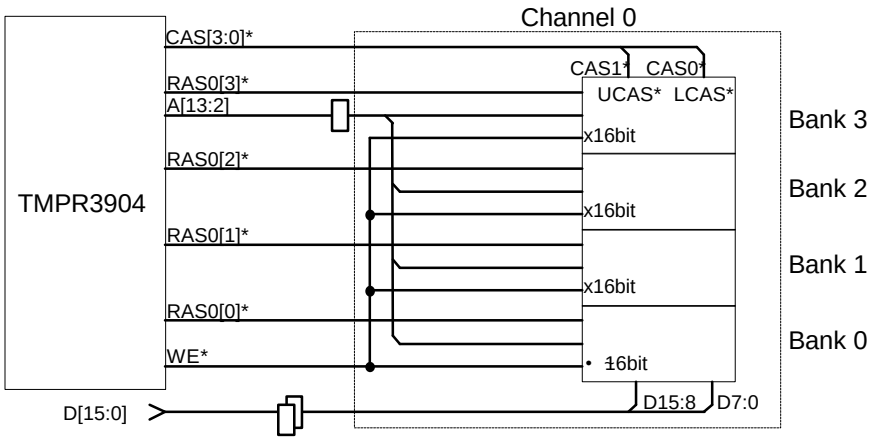


Fig. 8-33 16-bit Bus Configuration

A necessity of buffer for address, data and WE* depends on number of DRAM devices.

9 ROM CONTROLLER (ROMC)

9.1 Features

The ROM Controller generates signals and timings that are to be necessary to control the ROM and the SRAM.

(1) Supports the two-channel ROM control

Mask ROM, EPROM, FLASH, and SRAM support

Supports read operations only for the Mask ROM/EPROM.

Supports read/write operations for the FLASH/SRAM.

Each channel can form up to two banks.

(2) Timing can be set up for each channel

The number of waits can be set up in seven steps--from 0 to 6.

(3) Size can be set up for each channel

The memory size to be assigned to the channels can be set up in six steps--1/2/4/8/16/32 Mbytes

(4) Static bus sizing can be set up for each channel

Supports static bus sizing of 32/16 bits

(5) Support for page read mode

Supports the page mode Mask ROM.

(6) Interleave can be set up for each channel

Two-way interleave systems can be formed.

9.2 Block Diagrams

Figure 9-1 shows the ROMC connection inside the TX3904 and Figure 9-2 shows the internal blocks.

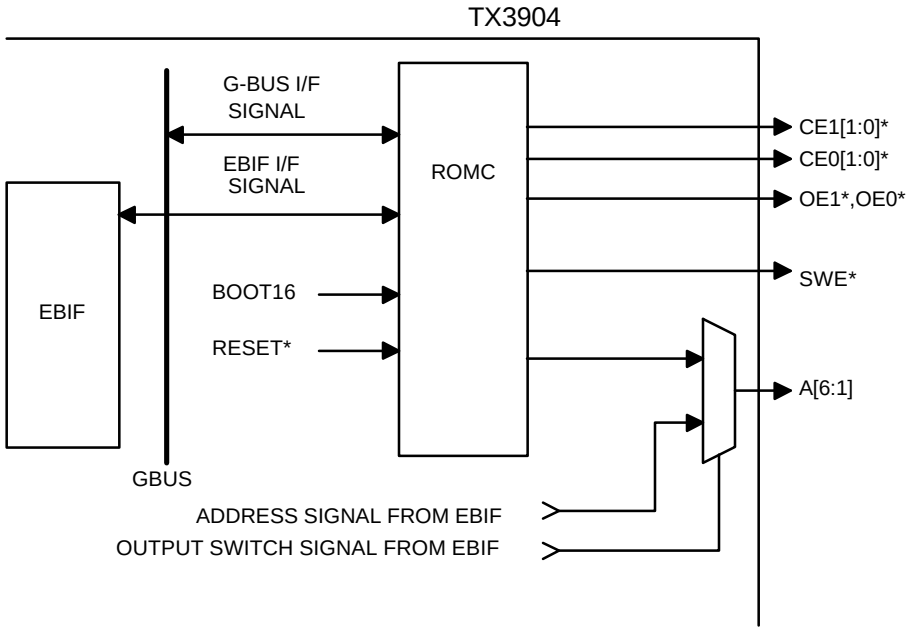


Fig. 9-1 Connection of ROMC inside TX3904

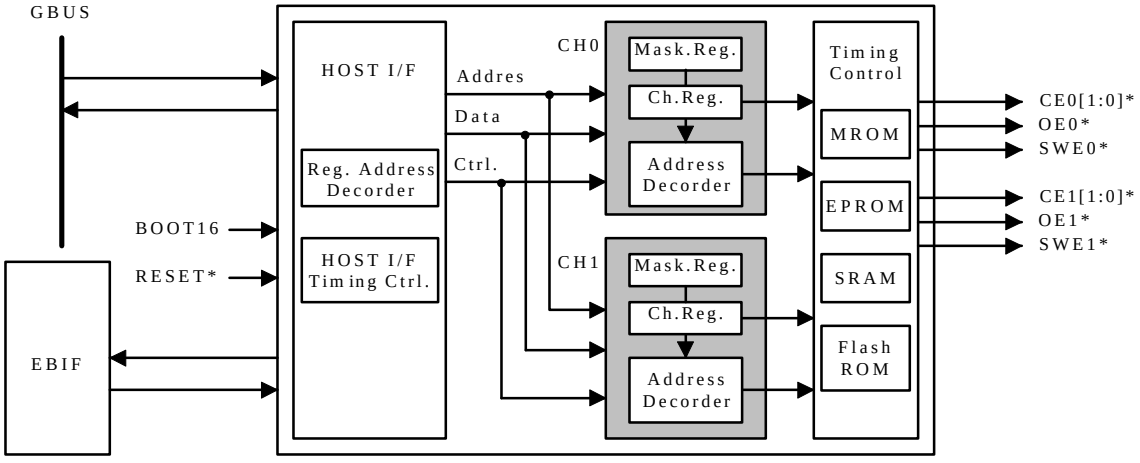


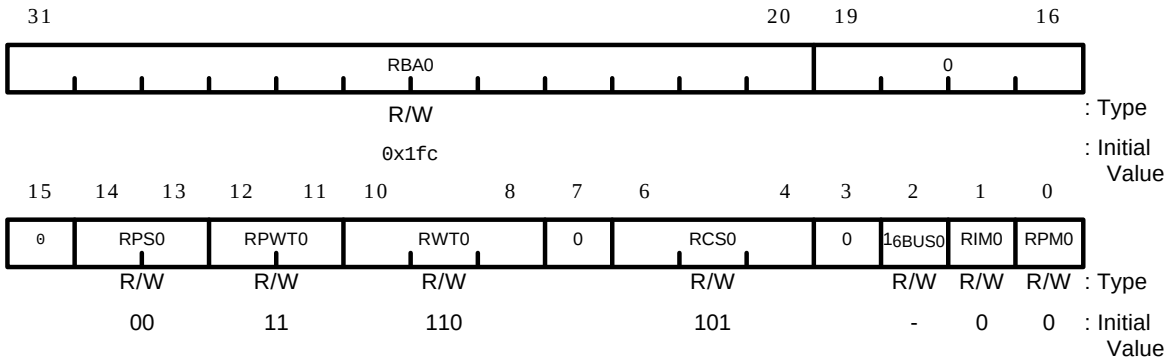
Fig. 9-2 ROMC Block Diagram

9.3 Registers

Table 9-1 ROM Controller Registers

Address	Register Symbol	Register Name
0xFFFF_9000	RCCR0	Channel Control Register 0
0xFFFF_9004	RBMR0	Base Address Mask Register 0
0xFFFF_9100	RCCR1	Channel Control Register 1
0xFFFF_9104	RBMR1	Base Address Mask Register 1

9.3.1 Channel control register 0



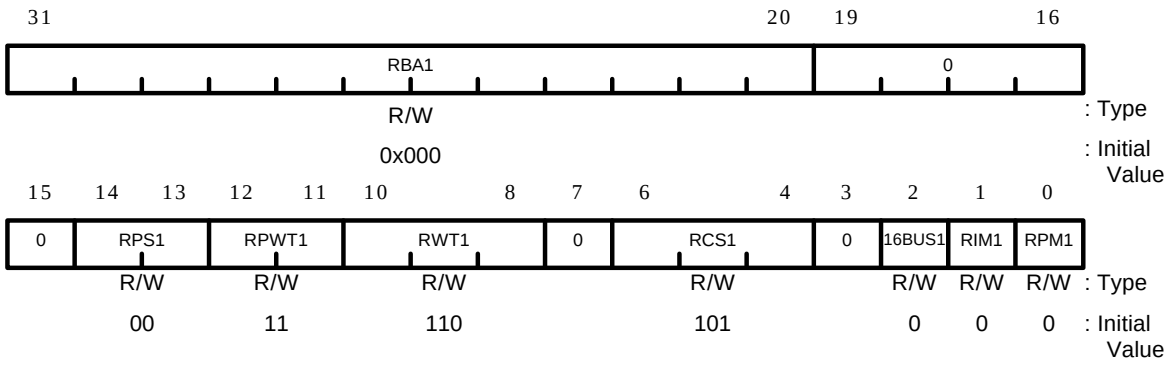
Bit	Mnemonic	Name of Field	Description
31:20	RBA0	ROM channel 0 base address	ROM Control Base Address on Channel 0 Designates the base address of Channel 0 using physical address.
14:13	RPS0	ROM channel 0 page mode ROM page size	ROM Control Page Mode ROM Page Size Designates the page size when the page mode Mask ROM is used for Channel 0. 00: 4-word 01: 8-word 1*: Reserved
12:11	RPWT0	ROM channel 0 page mode wait time	ROM Control Page Read Mode Wait Time on Channel 0 Designates the number of wait cycles during the page read when the page read Mask ROM is used for Channel 0. 00: 0 wait 01: 1 wait 10: 2 waits 11: 3 waits
10:8	RWT0	ROM channel 0 normal mode wait time	ROM Control Wait Time on Channel 0 Designates the number of wait cycles in the single operation of Channel 0 and in the burst operation with the Mask ROM that does not support the page mode. 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: Reserved

Fig. 9-3 ROM Channel 0 Control Register (1/2)

Bit	Mnemonic	Name of Field	Description
6:4	RCS0	ROM Channel 0 memory size	ROM Control Channel Size on Channel 0 Designates the memory size to be assigned to Channel 0. 000: 1 Mbytes 100: 16 Mbytes 001: 2 Mbytes 101: 32 Mbytes 010: 4 Mbytes 11*: Reserved 011: 8 Mbytes
2	16BUS0	ROM Channel 0 memory bus width	ROM Control 16-bit Width Bus Size on Channel 0 Sets up the memory bus width of Channel 0. The default setting of Channel 0 is given by the external input pin (BOOT16). 1: 16-bit width bus size 0: 32-bit width bus size
1	RIM0	Reserved	This bit is reserved. Do not set to 1.
0	RPM0	ROM Channel 0 page mode ROM enable	ROM Control Page Mode on Channel 0 Sets up its type when using the Mask ROM for Channel 0. 1: Page mode 2: Normal mode

Fig. 9-4 ROM Channel 0 Control Register (2/2)

9.3.2 Channel control register 1



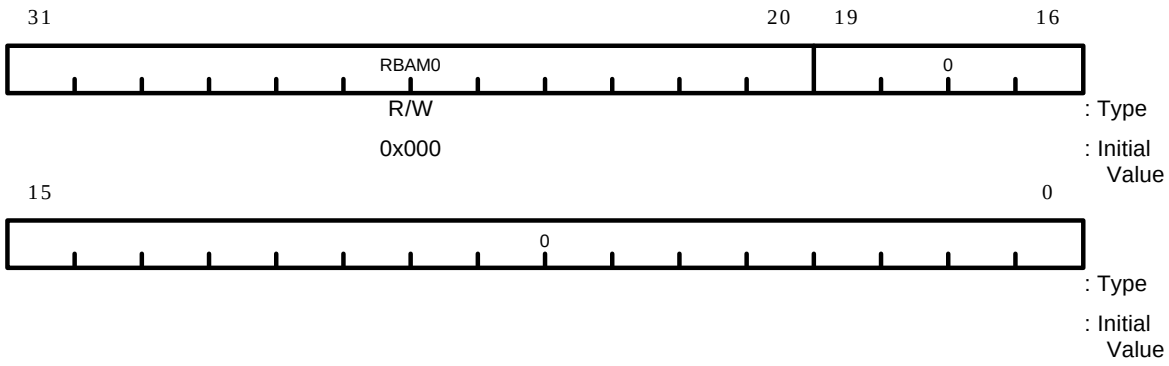
Bit	Mnemonic	Name of Field	Description
31:20	RBA1	ROM channel 1 base address	ROM Control Base Address on Channel 1 Designates the base address of Channel 1 using physical address.
14:13	RPS1	ROM channel 1 page mode ROM page size	ROM Control Page Mode ROM Page Size Designates the page size when the page mode Mask ROM is used for Channel 1. 00: 4-word 01: 8-word 1*: Reserved
12:11	RPWT1	ROM channel 1 page mode wait time	ROM Control Page Read Mode Wait Time on Channel 1 Designates the number of wait cycles during the page read and wait cycles during interleaving when the page read Mask ROM is used for Channel 1. 00: 0 wait 01: 1 wait 10: 2 waits 11: 3 waits
10:8	RWT1	ROM channel 1 normal mode wait time	ROM Control Wait Time on Channel 1 Designates the number of wait cycles in the single operation of Channel 1 and in the burst operation with the Mask ROM that does not support the page mode. 000: 0 wait 100: 4 waits 001: 1 wait 101: 5 waits 010: 2 waits 110: 6 waits 011: 3 waits 111: Reserved

Fig. 9-5 ROM Channel 1 Control Register (1/2)

Bit	Mnemonic	Name of Field	Description
6:4	RCS1	ROM Channel 1 memory size	ROM Control Channel Size on Channel 1 Designates the memory size to be assigned to Channel 1. 000: 1 Mbytes 100: 16 Mbytes 001: 2 Mbytes 101: 32 Mbytes 010: 4 Mbytes 11*: Reserved 011: 8 Mbytes
2	16BUS1	ROM Channel 1 memory bus width	ROM Control 16-bit Width Bus Size on Channel 1 Sets up the memory bus width of Channel 1. 1: 16-bit width bus size 0: 32-bit width bus size
1	RIM1	Reserved	This bit is reserved. Do not set to 1.
0	RPM1	ROM Channel 1 page mode ROM enable	ROM Control Page Mode on Channel 1 Sets up its type when using the Mask ROM for Channel 1. 1: Page mode 2: Normal mode

Fig. 9-6 ROM Channel 1 Control Register (2/2)

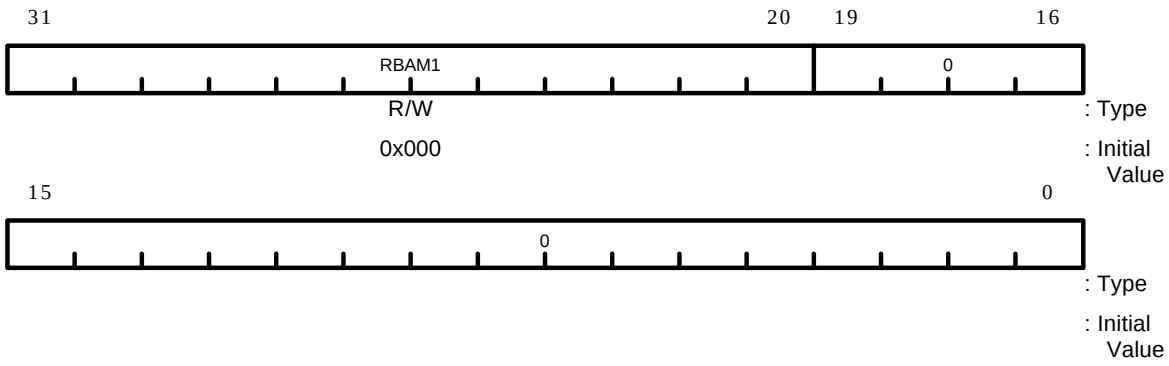
9.3.3 Base address mask register 0



Bit	Mnemonic	Name of Field	Description
31:20	RBAM0	ROM Channel 0 base address mask	ROM Control Base Address Mask on Channel 0 Specifies the valid bit of the address comparison by the RBA0 field of the channel control register. 1: Bit of the corresponding RBA0 field is not compared. 0: Bit of the corresponding RBA0 field is compared.

Fig. 9-7 ROM Channel 0 Base Address Mask Register

9.3.4 Base address mask register 1



Bit	Mnemonic	Name of Field	Description
31:20	RBAM1	ROM Channel 1 base address mask	ROM Control Base Address Mask on Channel 1 Specifies the valid bit of the address comparison by the RBA1 field of the channel control register. 1: Bit of the corresponding RBA1 field is not compared. 0: Bit of the corresponding RBA1 field is compared.

Fig. 9-8 ROM Channel 1 Base Address Mask Register

9.4 Operations

9.4.1 Channel select

The channel select for memory access is conducted by the address that is given onto the GBUS of the internal system bus.
Of the addresses, the high-order 12 bits are compared if they are within the range from the base address that is set up in the channel control register to the size that is set up in the size register, and if they are in the range, the channel is selected. The base address can be masked. Masking is conducted by the base address mask register that designates at will a mask with 12-bit base address.

Table 9-2 Address Decode

Ch. Size	Bank Size	Ch. Select Base Addr. bit	Bank Sel.(Ch0)		Bank Sel.(Ch1)	
			Bank 1	Bank 0	Bank 1	Bank 0
1M	512K	A31:20	A19=1	A19=0	A19=1	A19=0
2M	1M	A31:21	A20=1	A20=0	A20=1	A20=0
4M	2M	A31:22	A21=1	A21=0	A21=1	A21=0
8M	4M	A31:23	A22=0	A22=1	A22=1	A22=0
16M	8M	A31:24	A23=0	A23=1	A23=1	A23=0
32M	16M	A31:25	A24=0	A24=1	A24=1	A24=0

Size	A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19
1M	*	*	*	*	*	*	*	*	*	*	*	*	#
2M	*	*	*	*	*	*	*	*	*	*	*	#	
4M	*	*	*	*	*	*	*	*	*	*	#		
8M	*	*	*	*	*	*	*	*	*	#			
16M	*	*	*	*	*	*	*	*	#				
32M	*	*	*	*	*	*	*	#					

*: The bit to be compared with the base address to select the channel
#: To be used to select the bank

Depending on the set-up of the base address and the size of each channel, two channels are sometimes selected. In such a case, the channel to which a higher base address is set up is selected. When accessing the addresses shown in Figure 9-9, Channel 1 with a higher base address value is selected. When the same base address is set up, Channel 0 has the priority to be selected.

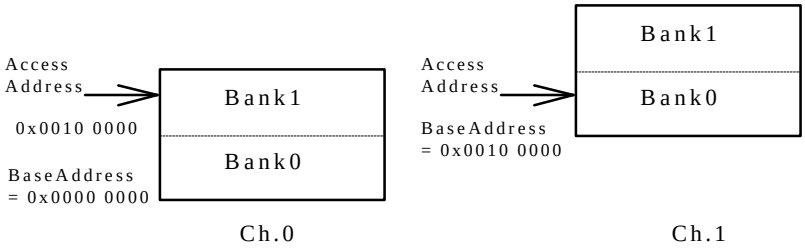
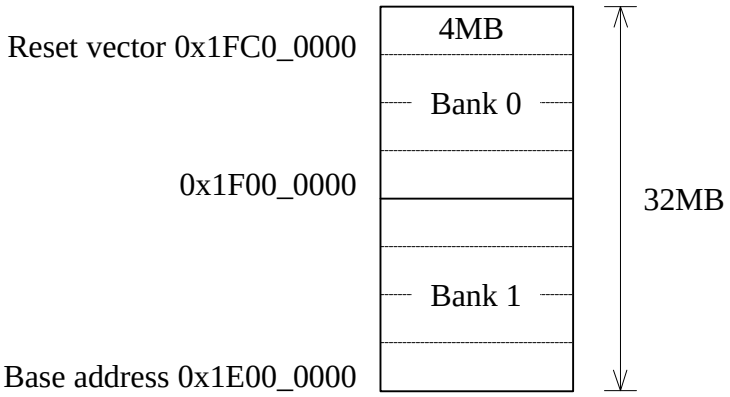


Fig. 9-9 Process at the Time of Dual Channel Select

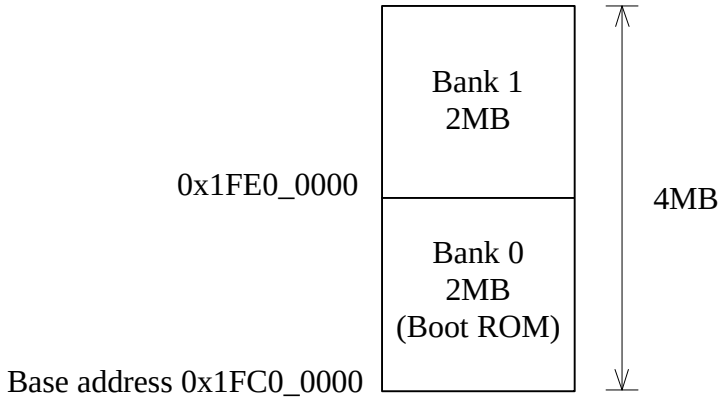
Please note that the range of the bit which compares with the base address gets narrower as the channel memory size gets larger. The base address default for channel 0 is 0x1FC, but the actual base address becomes 0x1E00_0000 when the memory size is 32 Mbytes.
Also note that, in the case of channel 0 only, the positions of bank 0 and bank 1 on the memory map are reversed (bank 1 is allocated to the lower address) when the memory size is set to 8 Mbytes or larger.

The following figure is an example of a memory map in which the default of the channel 0 base address is left at 0x1FC and the memory size was changed from 32 Mbytes to 4 Mbytes.

Memory size 32MB (default)



Memory size 4MB



When two 2-Mbyte ROMs are used like above, a boot ROM (containing initialization program) has to be connected to bank 0 and another one to bank 1. The boot ROM can be accessed properly in default state immediately after reset. Then a memory size is set 4 Mbyte, two ROMs are accessible as a continuous address space.

9.4.2 Operation Modes

Table 9-3 Relationship between the Bus Operation by Bus Master and Memory Access Mode

Memory Operation Mode	Access Type	Device	Bus Master (CPU, DMAC)			
			Single Operation		Burst Operation	
			32-bit Bus	16-bit Bus	32-bit Bus	16-bit Bus
Normal Mode	R	PM,M, E,S,F	O		O	
	W	S	O		O	
Page Mode	R	PM	O*		O	
	W	-	X		X	

O: Supported with ROMC X: Not supported
*: The single operation is the same as the normal mode.
Single Operation
At the time of access to non-cache area.
When the data cache refill size is 1.
Burst Operation
At the time of data cache burst refill
At the time of instruction cache refill
At the time of burst access by the DMAC
PM: Page mode Mask ROM, M: Mask ROM
E: EPROM, F: Flash ROM, S: SRAM

9.4.3 32/16-bit Static Bus Sizing

The bus width 32/16 bits of the ROM that is connected to the ROM Channel 0 is designated by an external input pin (BOOT16). The value of this input pin will be taken in to the 16BUS0 of the channel control register 0 at the time of reset. The bus width of the ROM channel 1 is set up by the program.

9.4.4 16-bit Bus Access

Word/triple-byte access with 16-bit width ROM

If the 16BUSn of the ROM channel control register is 1 when a word/triple-byte access is requested, the ROMC executes two bus cycles with the timing set up in the register.

Half-word/byte access with 16-bit width ROM

If the 16BUSn of the ROM channel control register is 1 when a 16-bit data access is requested, the ROMC executes one bus cycle with the timing set up in the register.

9.4.5 Access by External Bus Master

The ROMC drives address(es) with the built-in address counter at the times of page mode access (page mode mask ROM) and interleave access. When the external bus master conducts a memory access using the ROMC of the TX3904, an address bus conflict may occur. The external bus master must stop the drive of the address bus and the BE* at a rising of the BSTART*. When having received a memory access request from the external bus master, the ROMC automatically inserts a wait cycle to avoid driving the address, CE*, OE*, and SWE* at the S1 state. The number of the automatically inserted wait cycles is 1 SYSCLK. AT a single access, the address output by the external bus master is given to the memory. Therefore, at this time, please do not stop the address drive.
In the half speed bus mode, please adjust the wait such that the number of the cycles (GCLK) of the internal system clock should be an even number.

9.4.6 Page mode support

In the case of a page mode mask ROM, the ROM controller conducts page mode access when the bus master executes the burst mode access. The bus master issues a burst starting address for the first word, then the ROM controller generates the successive address for the second word and after. The ROM controller increments address like as 0-4-8-C-10-... if the lower four-bit of burst starting address is 0x0. On the other hand, it decrements address like as C-8-4-0-... if the starting address is 0xC.

9.5 Timing Diagrams

9.5.1 32-bit bus single read operation (ROM/SRAM)

(1) Memory access by internal bus master

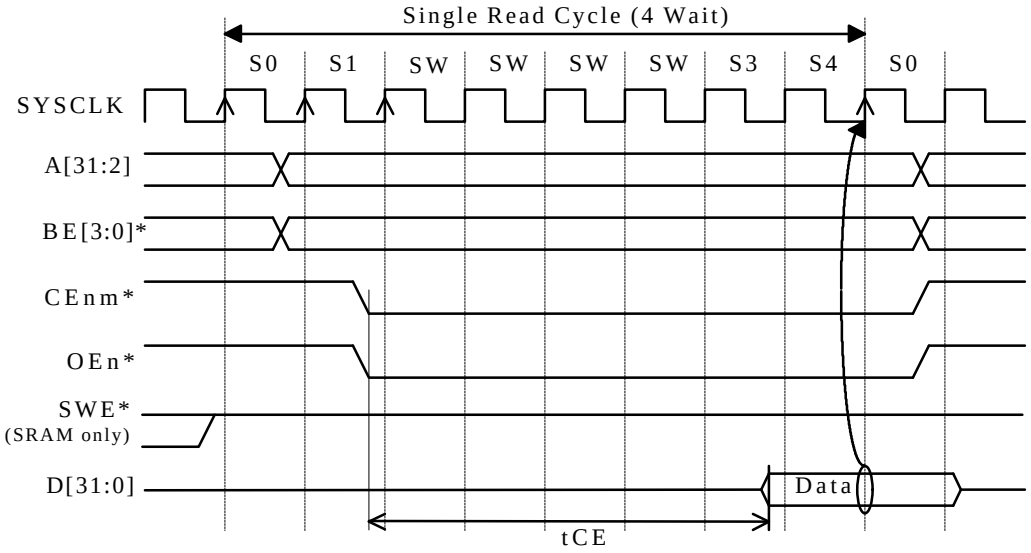


Fig. 9-10 32-bit Bus Memory (ROM/SRAM) Single Read Operation

(Internal Bus Master)

(2) Memory access by external bus master (Half speed bus mode; HALF*=low)

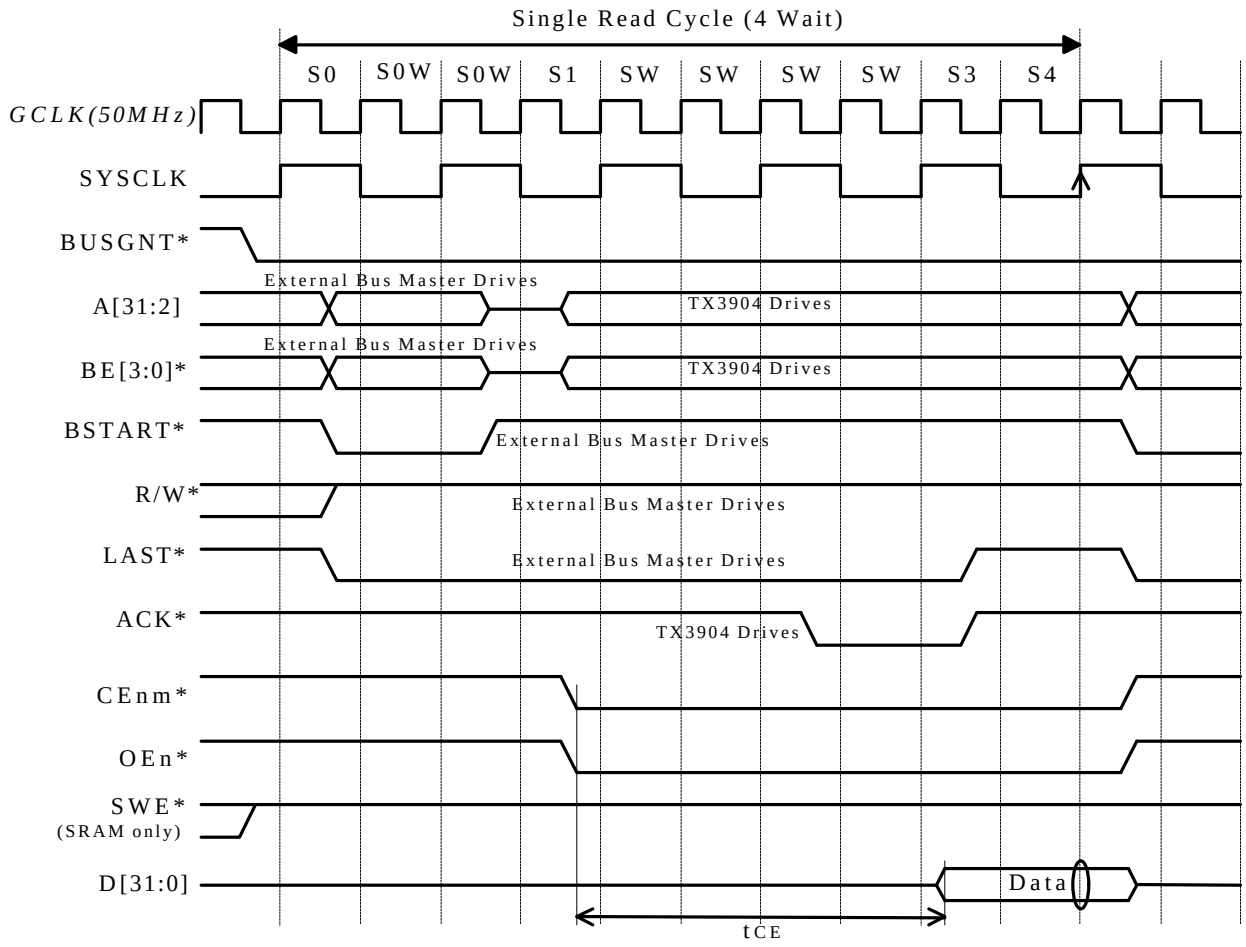


Fig. 9-11 32-bit Bus Memory (ROM/SRAM) Single Read Operation

(Half Speed Bus; External Bus Master)

In the half speed bus mode, output signals for memory access, except the ACK*, are not synchronized to the SYSCLK but to the internal GCLK.

(3) Memory access by external bus master (Full speed bus mode; HALF*=high)

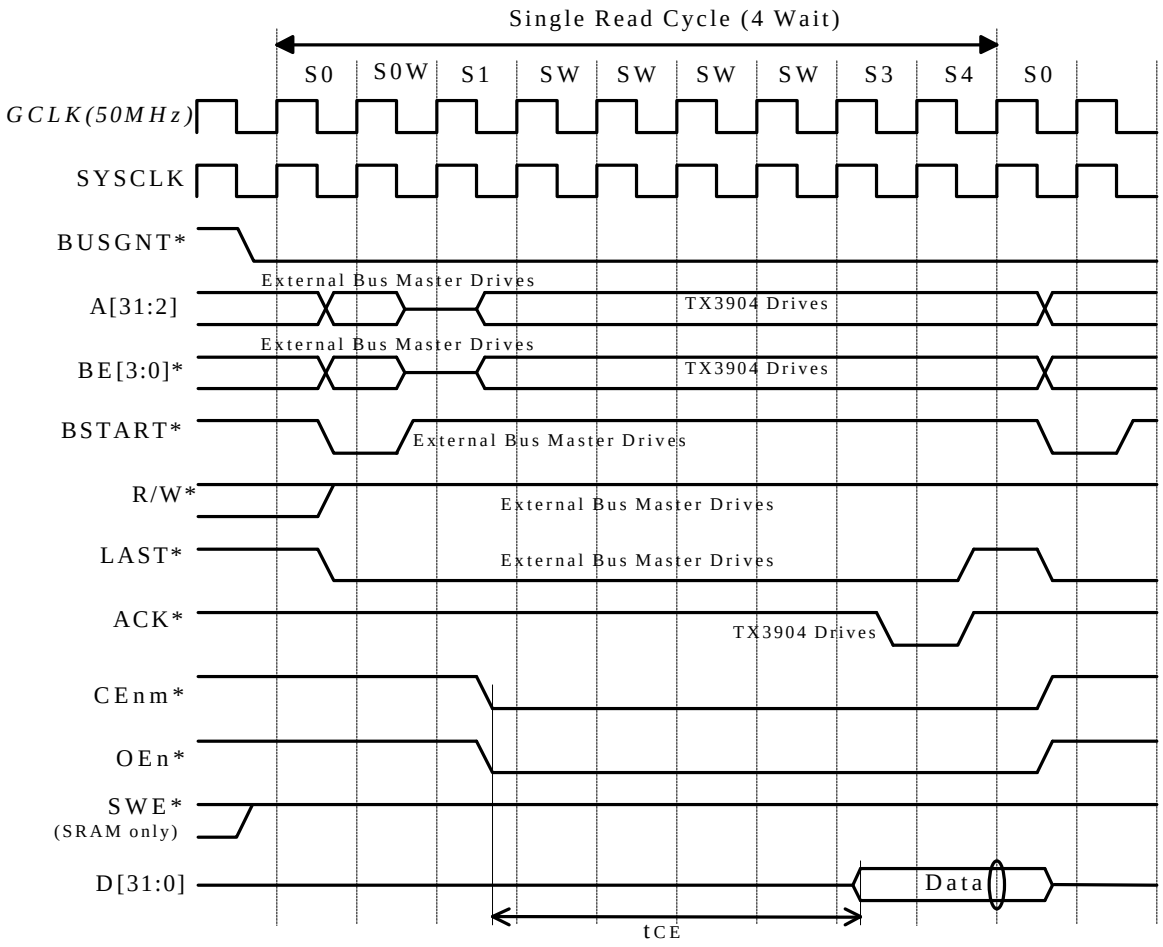


Fig. 9-12 32-bit Bus Memory (ROM/SRAM) Single Read Operation

(Full Speed Bus; External Bus Master)

9.5.2 16-bit bus single read (32-bit word) operation (ROM/SRAM)

(1) ROM and SRAM other than page mode MROM

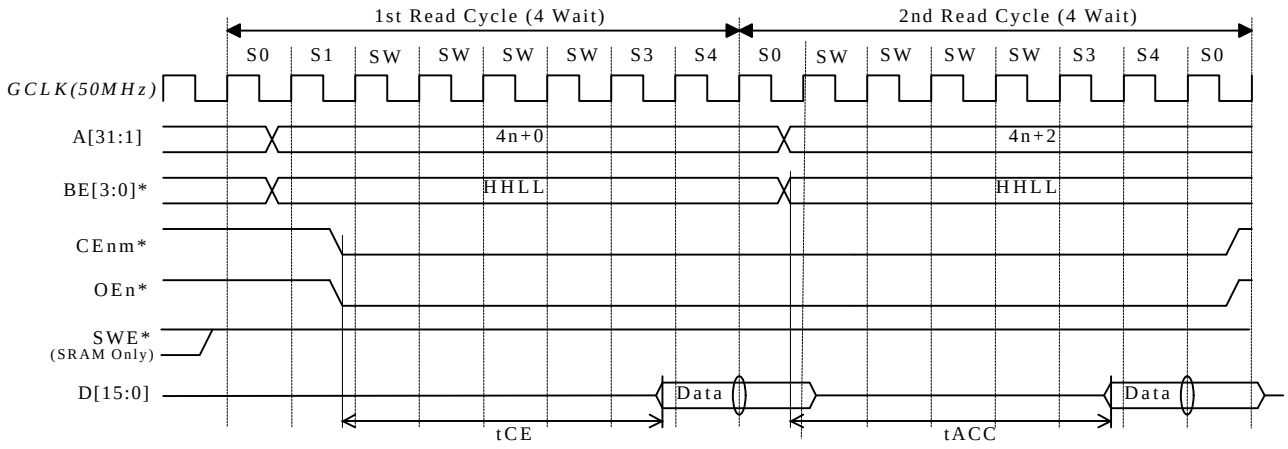


Fig. 9-13 16-bit Bus Memory Single Read (Word) Operation

(Other Than Page Mode MROM)

(2) Page mode MROM

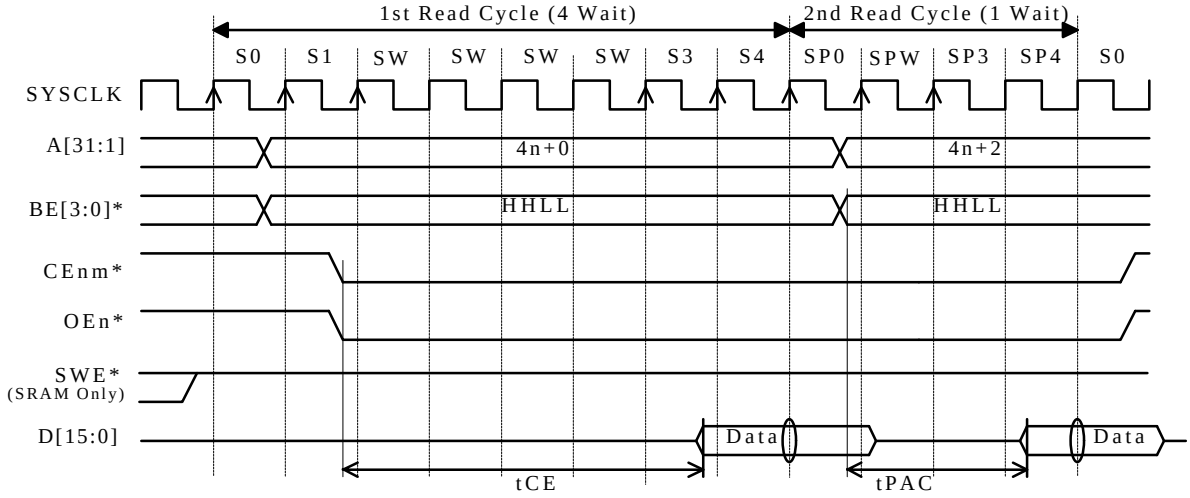


Fig. 9-14 16-bit Bus Memory Single Read (Word) Operation

(Page Mode MROM)

When a 16-bit page mode MROM is used to read a 32-bit word, the access is conducted in the page mode. The built-in page counter is not used in the single read operation.

9.5.3 16-bit bus single read (half word) operation (ROM/SRAM)

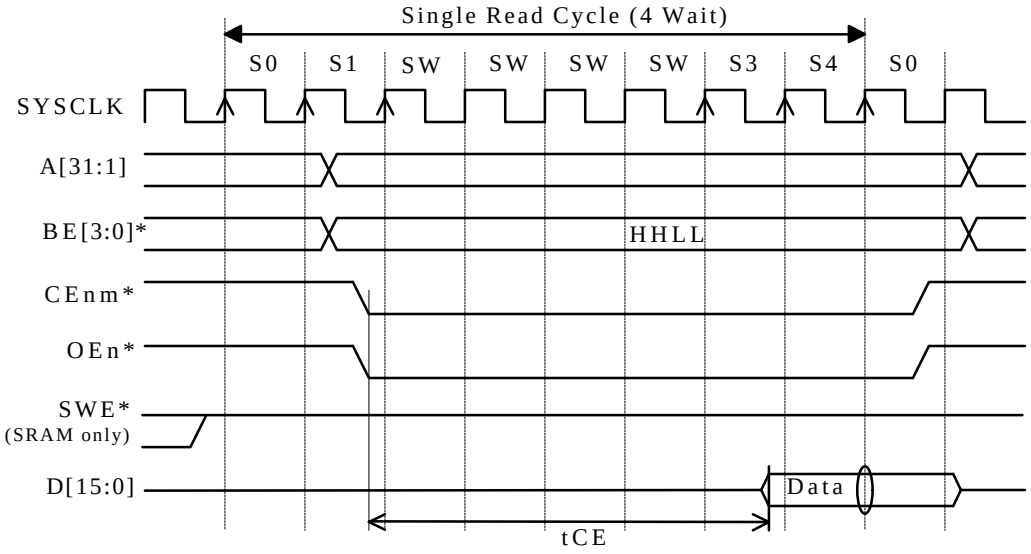


Fig. 9-15 16-bit Bus Single Read (Half Word) Operation

(ROM/SRAM)

9.5.4 32-bit bus single write operation (SRAM/Flush)

(1) Memory access by internal bus master

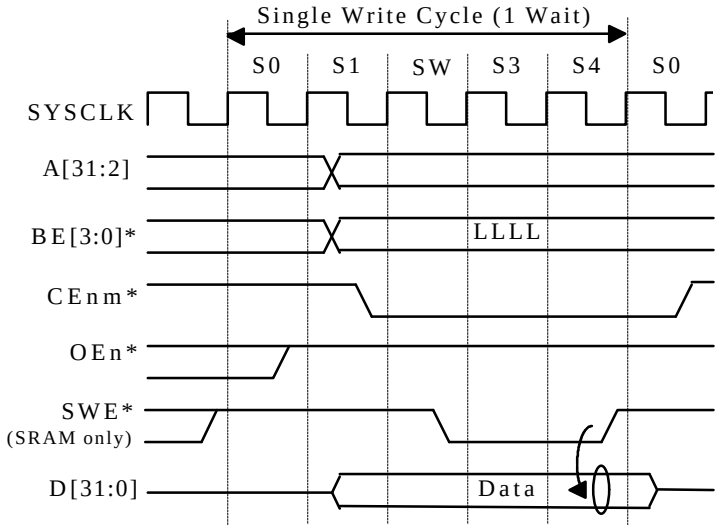


Fig. 9-16 32-bit Bus Memory Single Write Operation (SRAM/Flush)

(2) Memory access by external bus master (Half speed bus mode; HALF*=low)

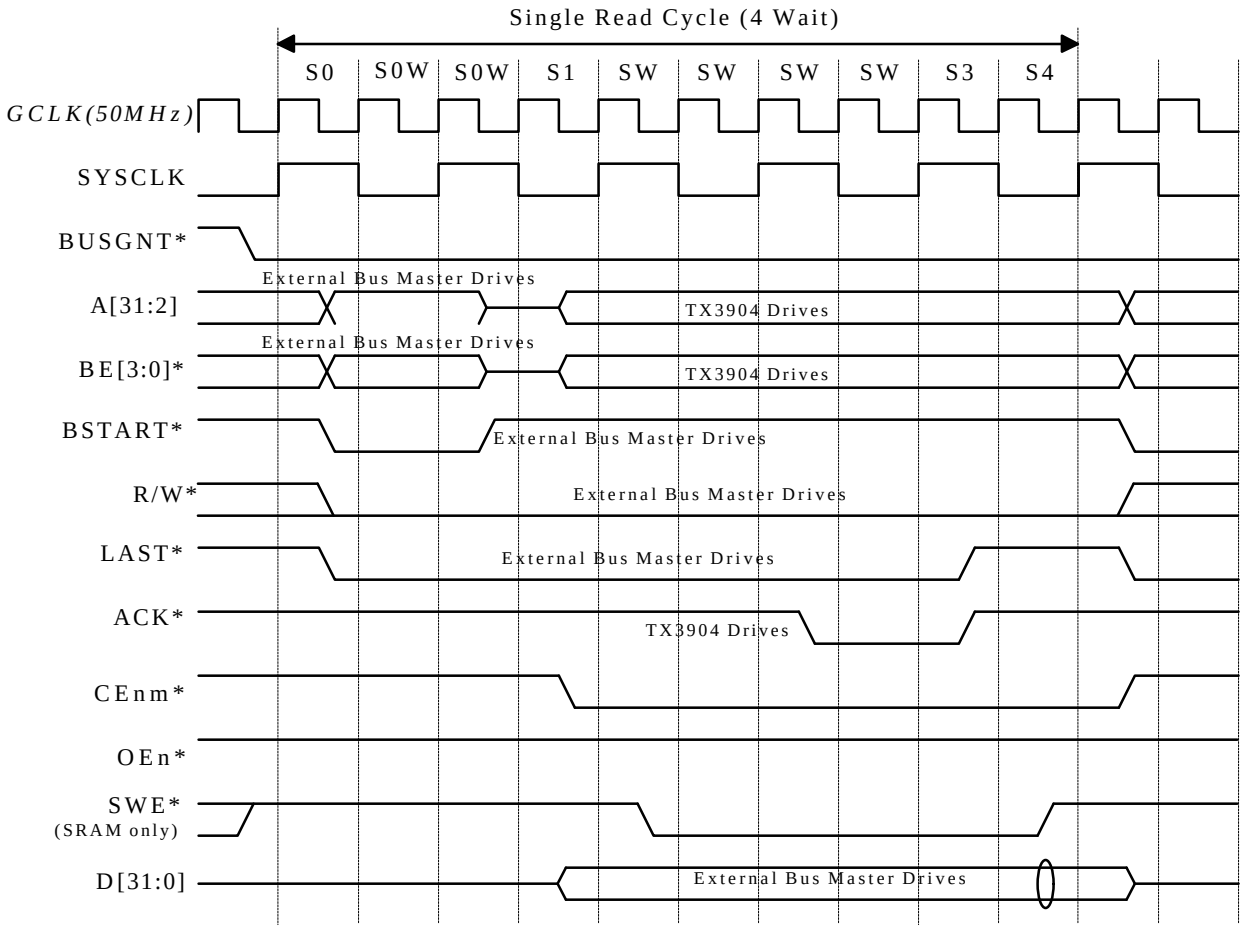


Fig. 9-17 32-bit Bus Memory (SRAM/Flash) Single Read Operation

(Half Speed Bus; External Bus Master)

9.5.5 16-bit bus single write (word) operation (SRAM/Flush)

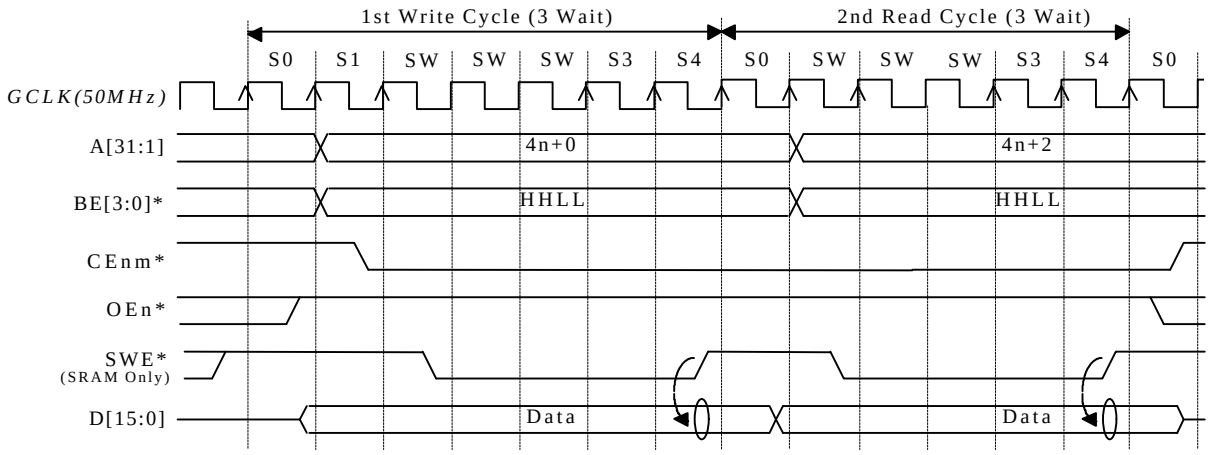


Fig. 9-18 16-bit Bus Memory Single Write (Word) Operation

(SRAM/Flush)

9.5.6 16-bit bus single write (half word) operation (SRAM/Flush)

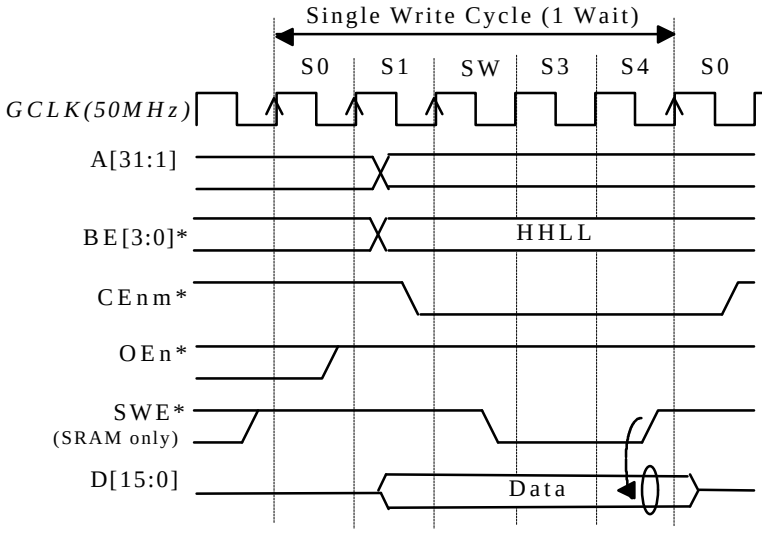


Fig. 9-19 16-bit Bus Memory Single Write (Half Word) Operation

(SRAM/Flush)

9.5.7 32-bit bus normal mode burst read operation (ROM/SRAM)

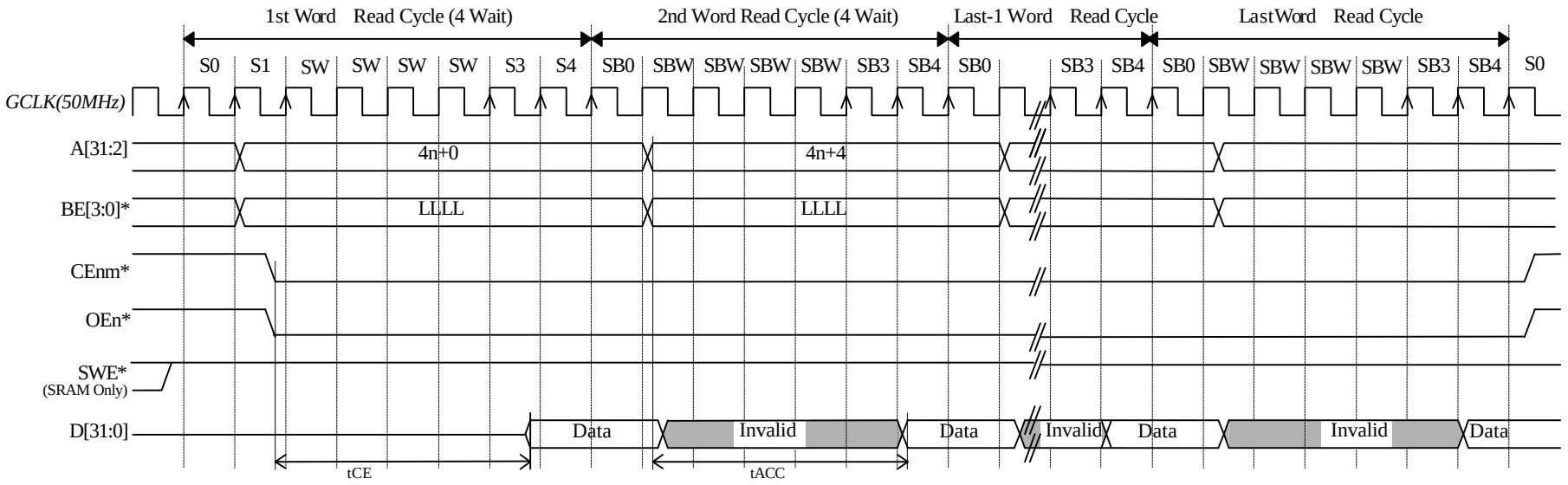


Fig.9-20 32-bit Bus Normal Mode Burst Read Operation

9.5.8 32-bit bus page mode burst read operation (Page mode MROM)

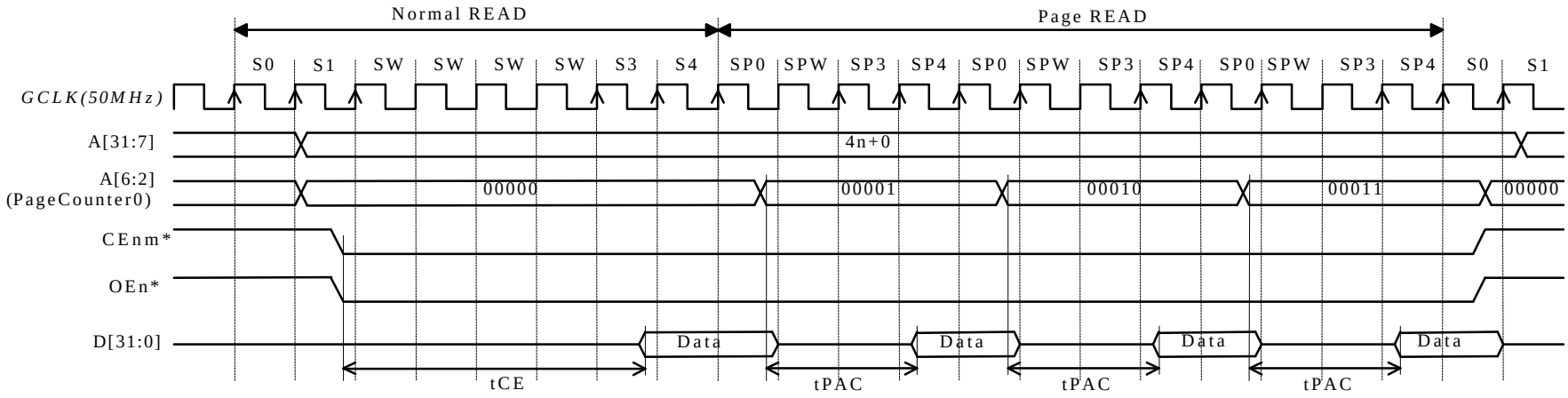


Fig. 9-21 32-bit Bus Page Mode Burst Read (Page Mode Mask ROM) Operation

9.5.9 16-bit bus word normal mode burst read operation (ROM/SRAM)

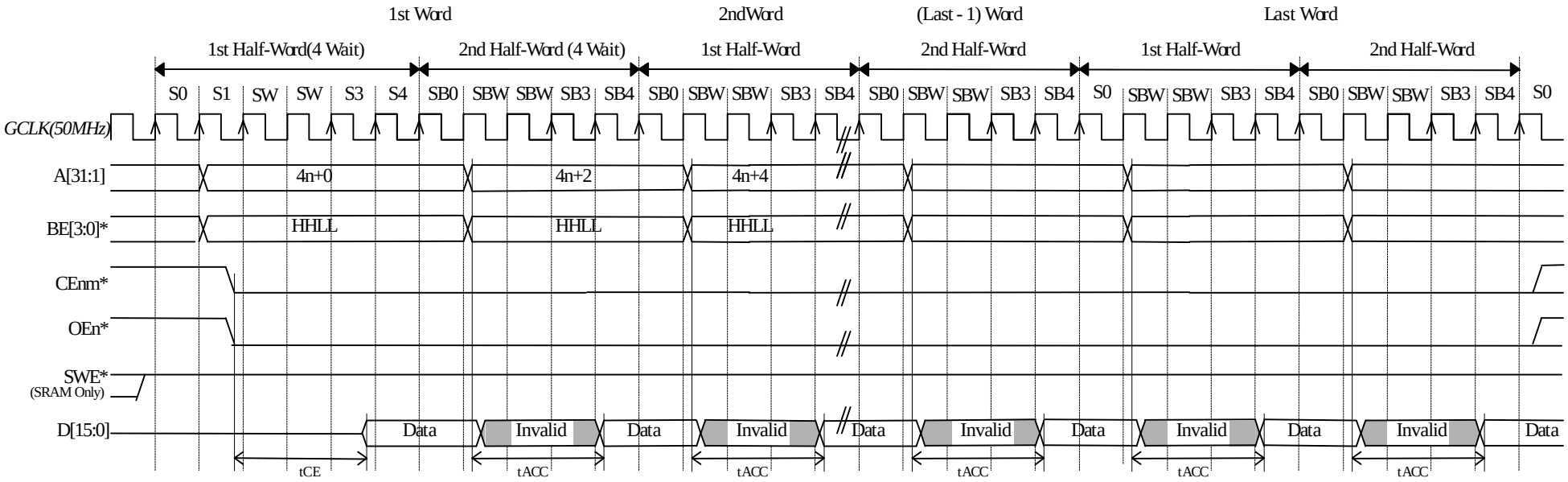


Fig. 9-22 16-bit Bus Normal Mode Burst Read Operation

9.5.10 16-bit bus page mode burst read (word) operation (Page mode MROM)

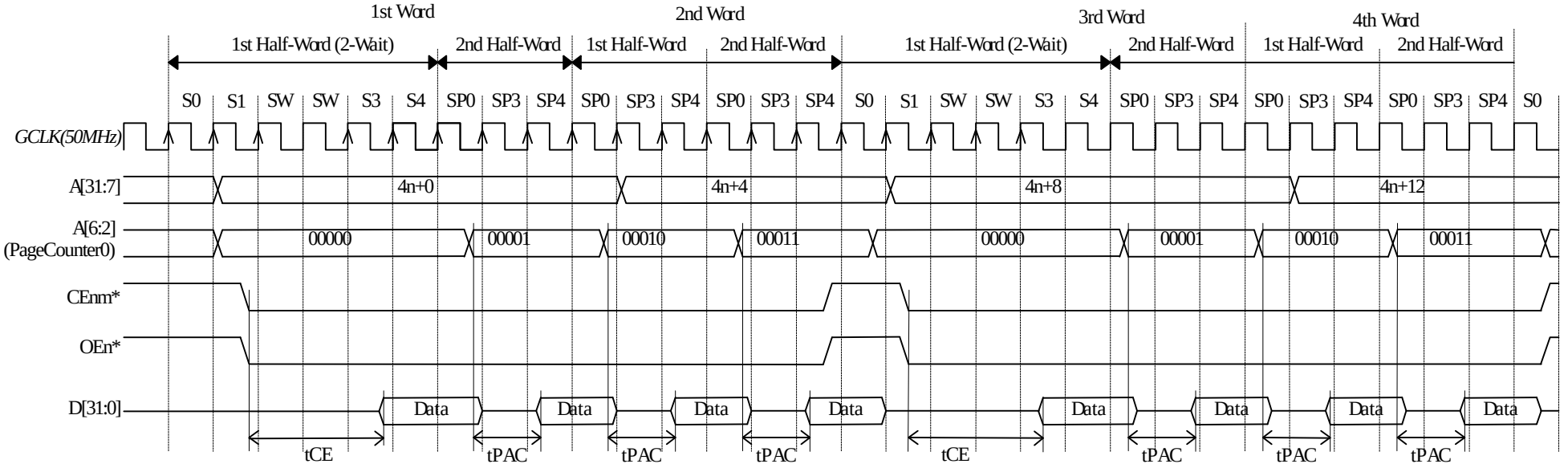


Fig. 9-23 16-bit Bus Page Mode Burst Read (Word) Operation (Page Mode Mask ROM)

Fig. 9-23 shows the case where the burst size of the TX3904 is 4 words. With a Mask ROM of the page size 16 bits and 4 words, a 4-word page read is repeated twice. With a Mask ROM of the page size 16 bits and 8 words, a 8-word page read is conducted once to complete.

9.5.11 32-bit bus normal mode burst write (SRAM)

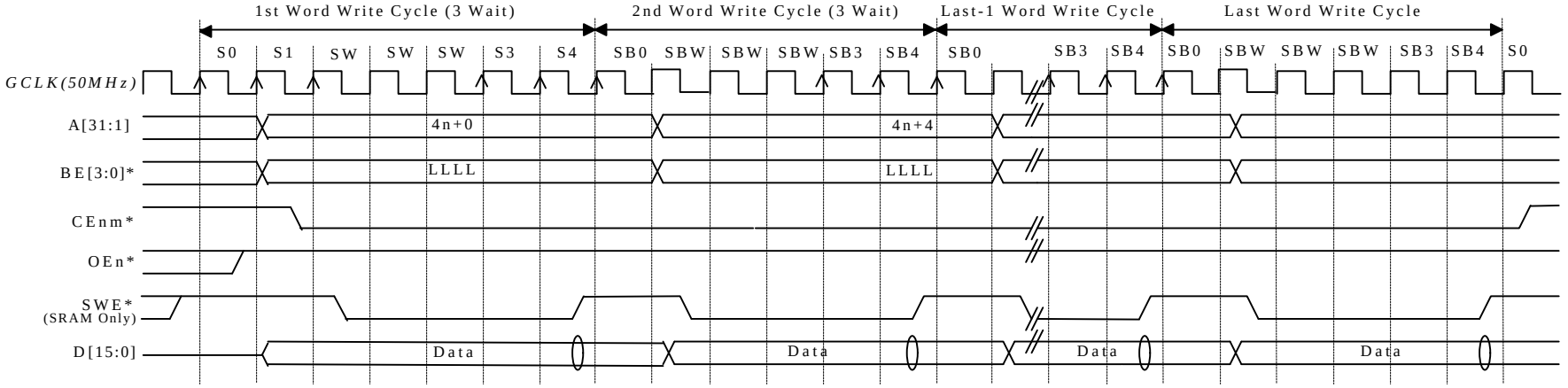


Fig. 9-24 32-bit Bus Normal Mode Burst Write Operation

9.5.12 16-bit bus normal mode burst write (word) (SRAM; WE control write)

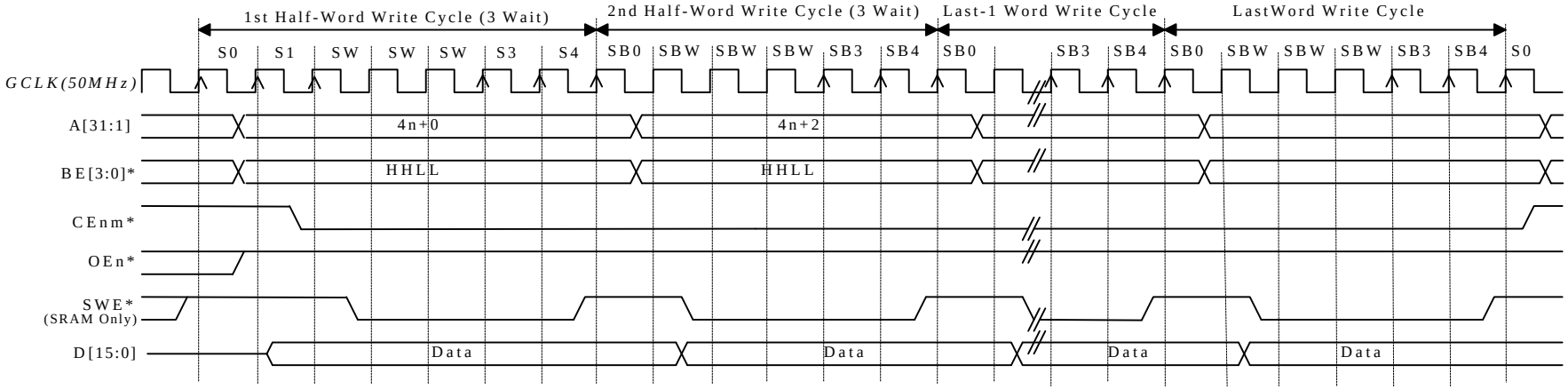


Fig. 9-25 16-bit Bus Normal Mode Burst Write Operation

9.5.13 16-bit bus normal mode burst write (half word) (SRAM; WE control)

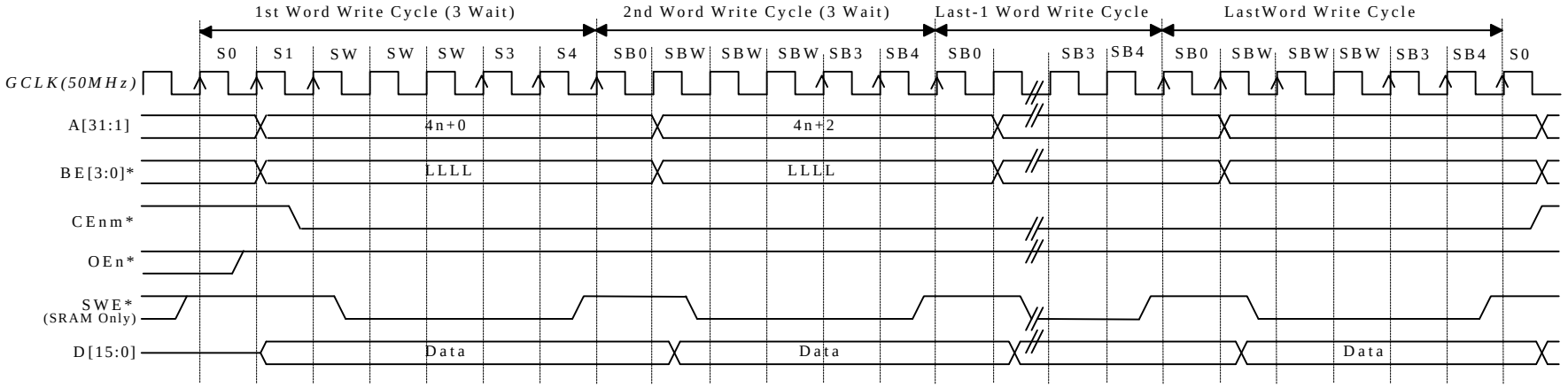


Fig. 9-26 16 bit Bus Normal Mode Burst Write (Half Word) Operation

9.6 Examples of MROM/EPROM Usage

The following figures are the usage examples of MROM/EPROM.

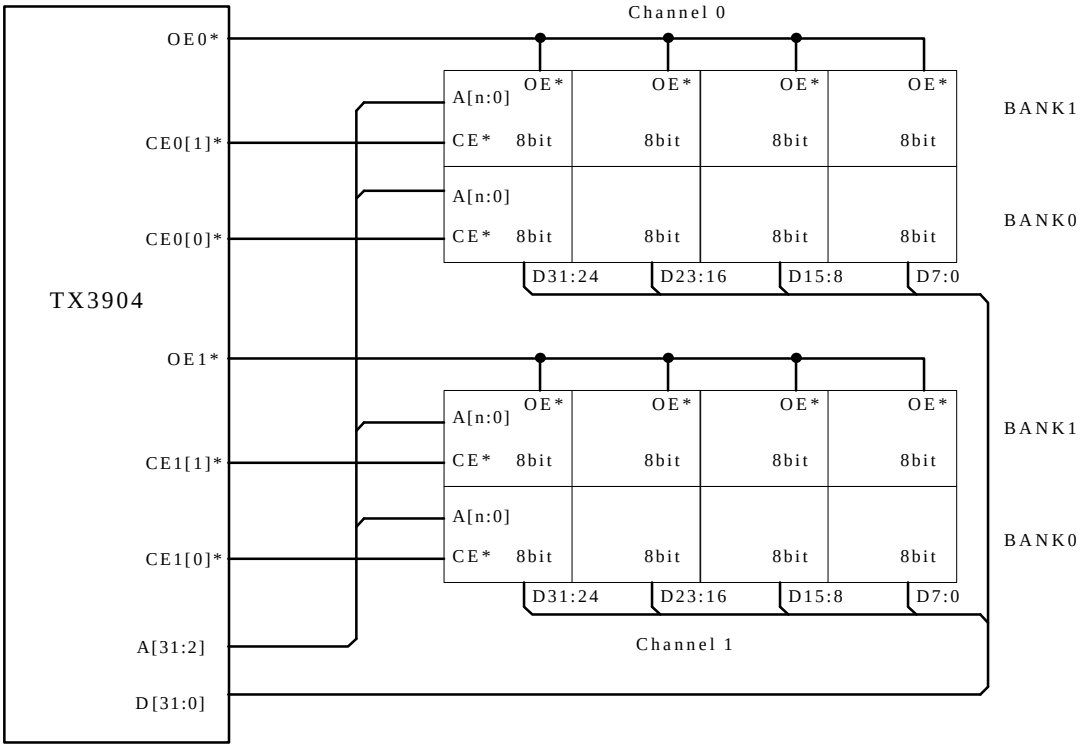


Fig. 9-27 Connection Example of 8-bit width Memory Chips

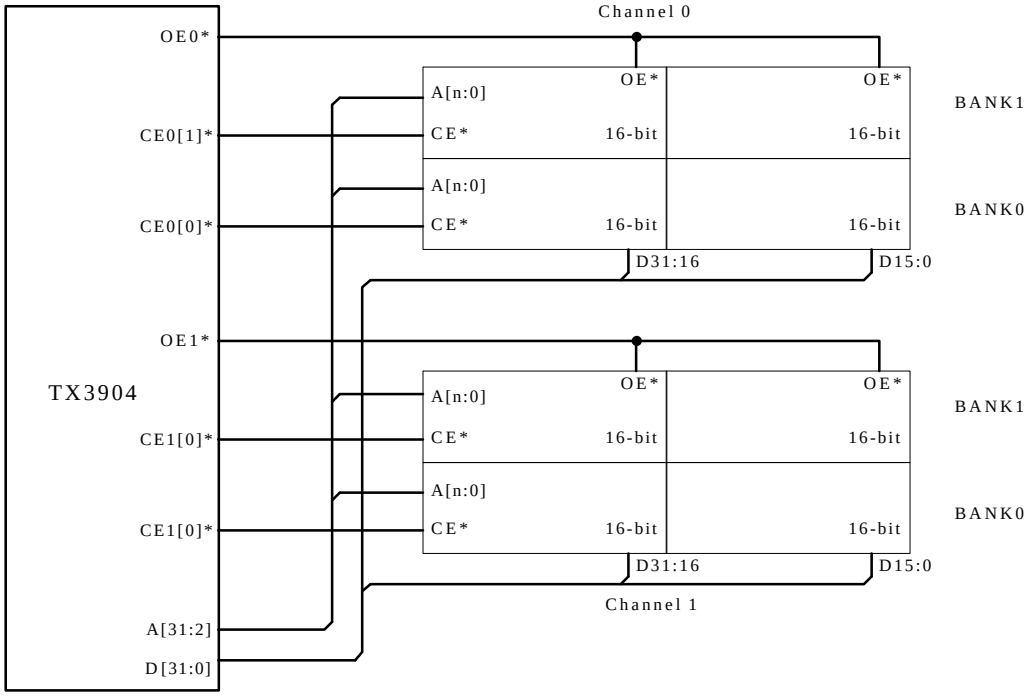


Fig. 9-28 Connection of 16-bit width Memory Chips

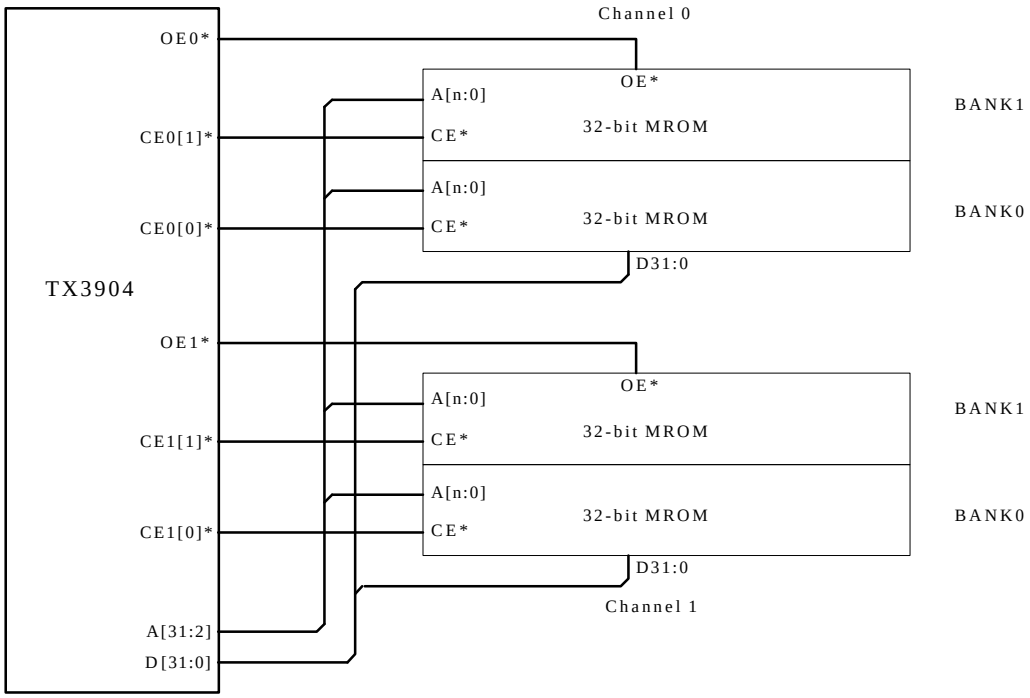


Fig. 9-29 Connection of 32-bit width Memory Chips

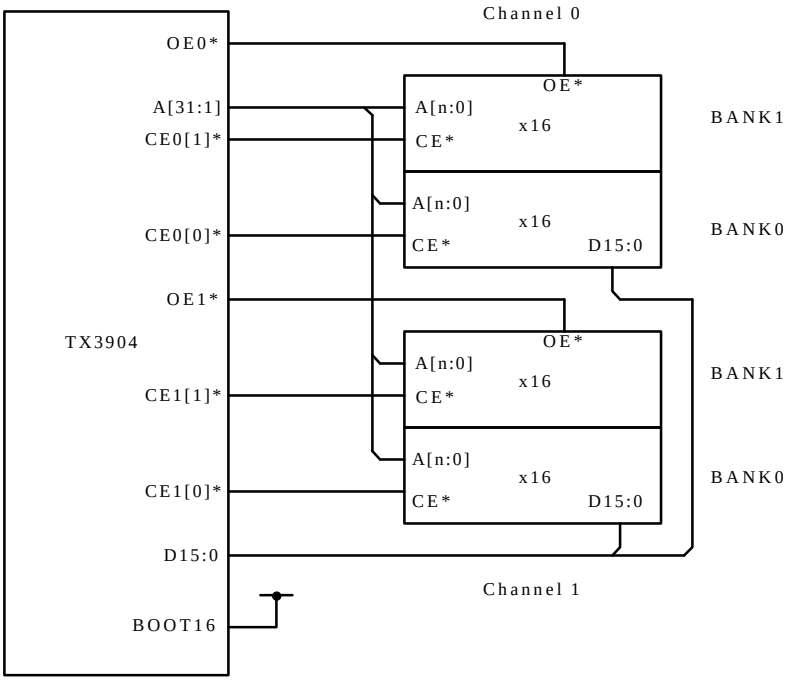


Fig. 9-30 16-bit Bus Sizing Connection Using 16-bit width Memory Chips

9.7 Examples of SRAM Usage

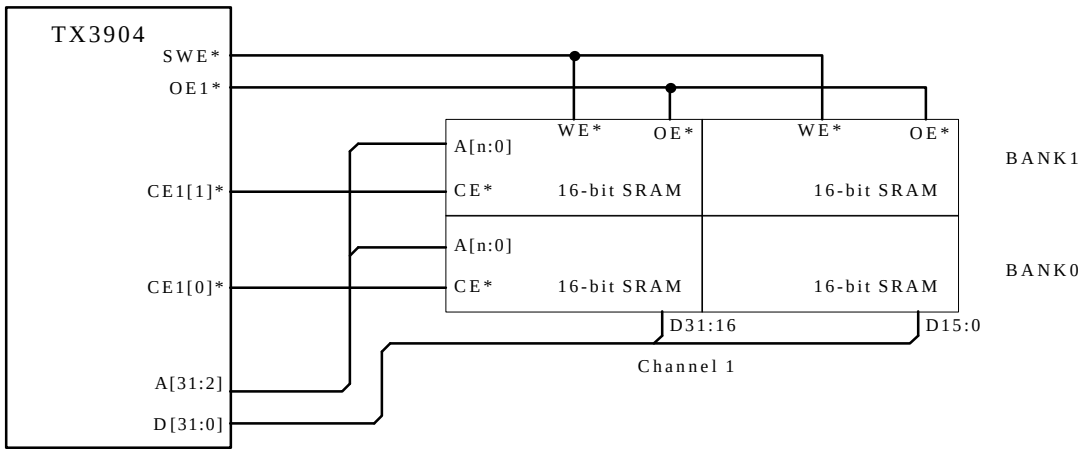


Fig. 9-31 Connection of x16-bit SRAM Chips

10 DMA CONTROLLER (DMAC)

The TX3904 has two modules of built-in DMA controllers (DMAC) with two channels.

10.1 Features

The following are the features of the DMAC built-in in the TX3904.

(1) Independent two-channel DMA (Two modules are connected with a daisy-chain)
(2) Two kinds of bus ownership requests: With and without snoop request
(3) Transfer requests: Internal request/External request
(4) Transfer modes: Dual address mode/Single address mode
(5) Transfer devices: Memory to memory, memory to I/O, I/O to memory
(6) Device sizes: Memory - 32 bits (16 bits available by ROMC/DRAMC);
I/O - 8, 16, 32 bits
(7) Address changes: Increase/Decrease/Fix
(8) Priority between channels: Fixed
(9) Big endian/Little endian can be set up
(10) Next transfer address can be set up (Single address mode only)

10.2 Configuration

10.2.1 TX3904 internal connection

The following Figure 10-1 shows the DMAC connection inside the TX3904.

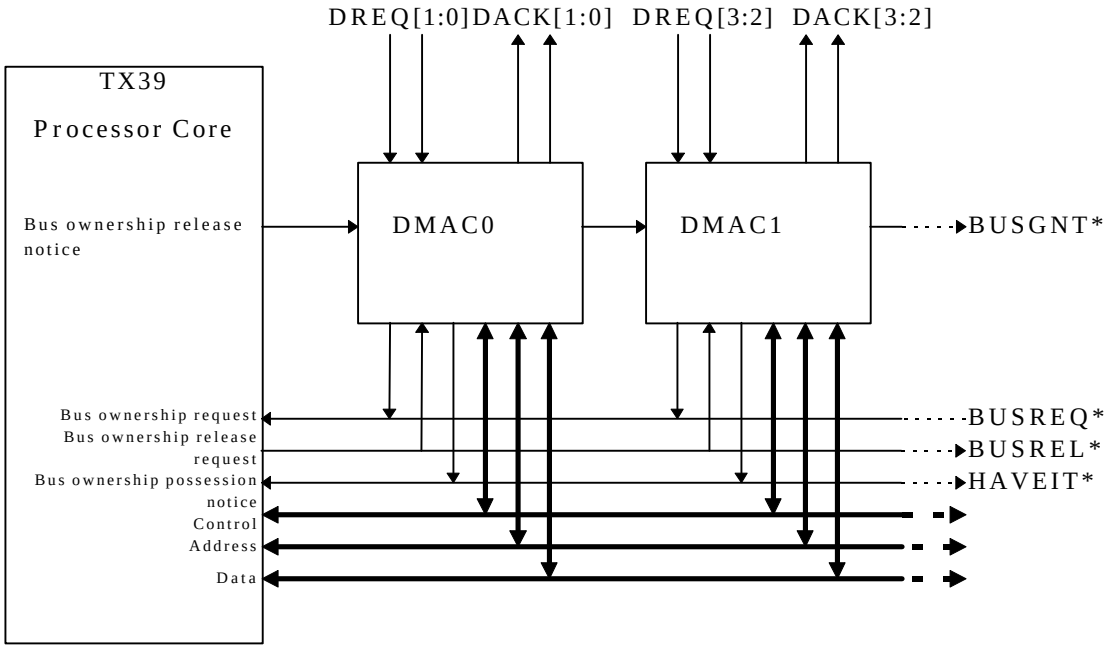


Fig. 10-1 DMAC Connection Inside TX3904

Inside the TX3904, two DMAC modules are daisy-chain connected. The module closer to the TX39 Processor Core (of higher priority) is DMAC0 and the module further to it (of lower priority) is DMAC1.

Each DMAC has two DMA channels; so that the two modules have four channels together. Each of the channels has a data transfer request signal DREQn from the external devices and an acknowledge signal DACKn* to the DREQn. The “n” is the channel number and is replaced by 0-3. Channel 0 has a higher priority than Channel 1 and Channel 2 has a higher priority than Channel 3.

When transferring data, the DMAC can snoop the data cache inside the TX39 Processor Core. Snooping is a function to make invalid the data inside the data cache when the data of the transfer destination address are inside the data cache. By doing so, the data inside the data cache and the data in the external memory are made compatible to each other. The DMAC can select whether or not to use this snoop function. As for the details of the snoop function, please refer to “7.5.4. Snoop Function.”

The DMAC has two kinds of bus ownership (GREQ and SREQ) which does/does not use the snoop function. The GREQ is the bus ownership request that does not use the snoop function and the SREQ is the bus ownership request that uses the snoop function. Of these two kinds of

bus ownership requests, the SREQ has a higher priority than the GREQ. Therefore, the priority between the two DMAC modules depends on the daisy-chain connection when the bus ownership request mode is the same, and it depends on the bus ownership request mode when the kinds of the bus ownership differ.

10.2.2 DMAC internal blocks

The following Figure 10-2 shows the internal blocks of DMAC0.

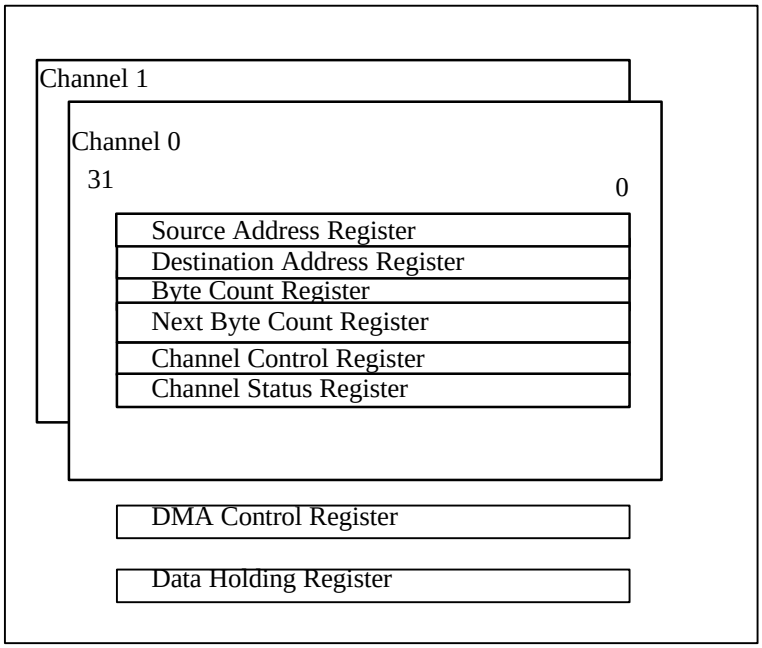


Fig. 10-2 DMAC Internal Blocks

10.2.3 Priority between modules

In the priority between the modules, there are the priority by the daisy-chain and the priority by the difference in the kinds of bus ownership.

(1) Priority by daisy-chain

DMAC0 and DMAC1 are daisy-chain connected. When DMAC0 and DMAC1 are both using the snoop function or are both not using it, the priority is determined by the daisy-chain connection; and DMAC0 has a higher priority than DMAC1.
When DMAC0 and DMAC1 simultaneously request of the TX39 Processor Core for the bus ownership, DMAC0 is granted the bus ownership to start a transfer operation. DMAC1 cannot start a transfer until DMAC0 releases the bus ownership.
Even if DMAC0 requests the bus ownership when DMAC1 has the bus ownership, the bus ownership shall not be transferred to DMAC0 immediately. DMAC0 is made to wait until DMAC1 releases the bus ownership.

(2) Priority by bus ownership request modes

When the bus ownership requests are different; namely, when one DMAC is using the SREQ and the other DMAC the GREQ, the DMAC using SREQ has a higher priority.
When the DMAC using SREQ and the DMAC using GREQ simultaneously request the bus ownership, the DMAC using SREQ obtains the bus ownership.
If the DMAC using SREQ requests the bus ownership while the DMAC using GREQ is transferring data, the bus arbiter in the TX39 processor core requests the bus ownership at a break of the data transfer. If the DMAC using GREQ answers to the request from the bus arbiter, the bus ownership is transferred to the DMAC using SREQ. However, if the RelEn is cleared to 0 in the CCRn of the DMAC using GREQ (if the bus ownership release request is not answered), the DMAC using GREQ does not release the bus ownership and continues the data transfer.

10.3 Registers

The following diagram shows fourteen built-in 32-bit registers of the DMAC. Tables 10-1 and 10-2 show the register maps of DMAC0 and DMAC1.

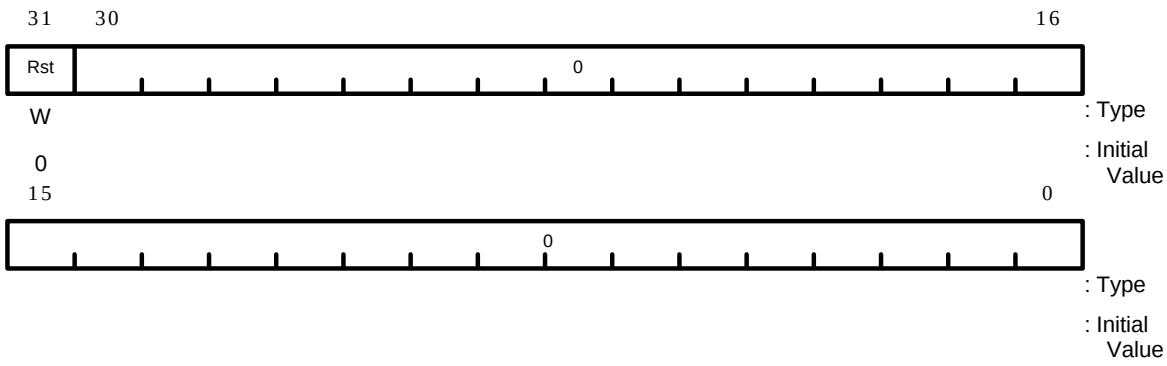
Table 10-1 DMAC0 Registers

Address	Register Symbol	Name of Register
0xFFFF-A08C	DHR0	Data holding register (DMAC0)
0xFFFF-A080	DCR0	DMA control register (DMAC0)
0xFFFF-A034	NCR1	Next byte count register (Ch.1)
0xFFFF-A030	BCR1	Byte count register (Ch.1)
0xFFFF-A02C	DAR1	Destination address register (Ch.1)
0xFFFF-A028	SAR1	Source address register (Ch.1)
0xFFFF-A024	CSR1	Channel status register (Ch.1)
0xFFFF-A020	CCR1	Channel control register (Ch.1)
0xFFFF-A014	NCR0	Next byte count register (Ch.0)
0xFFFF-A010	BCR0	Byte count register (Ch.0)
0xFFFF-A00C	DAR0	Destination address register (Ch.0)
0xFFFF-A008	SAR0	Source address register (Ch.0)
0xFFFF-A004	CSR0	Channel status register (Ch.0)
0xFFFF-A000	CCR0	Channel control register (Ch.0)

Table 10-2 DMAC1 Registers

Address	Register Symbol	Name of Register
0xFFFF-B08C	DHR1	Data holding register (DMAC1)
0xFFFF-B080	DCR1	DMA control register (DMAC1)
0xFFFF-B034	NCR3	Next byte count register (Ch.3)
0xFFFF-B030	BCR3	Byte count register (Ch.3)
0xFFFF-B02C	DAR3	Destination address register (Ch.3)
0xFFFF-B028	SAR3	Source address register (Ch.3)
0xFFFF-B024	CSR3	Channel status register (Ch.3)
0xFFFF-B020	CCR3	Channel control register (Ch.3)
0xFFFF-B014	NCR2	Next byte count register (Ch.2)
0xFFFF-B010	BCR2	Byte count register (Ch.2)
0xFFFF-B00C	DAR2	Destination address register (Ch.2)
0xFFFF-B008	SAR2	Source address register (Ch.2)
0xFFFF-B004	CSR2	Channel status register (Ch.2)
0xFFFF-B000	CCR2	Channel control register (Ch.2)

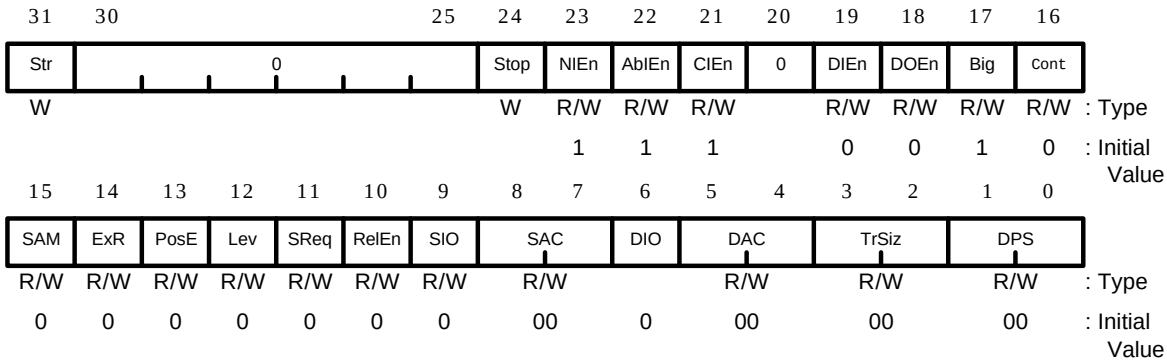
10.3.1 DMA control register (DCR)



Bit	Mnemonic	Name of Field	Description
31	Rst	Reset	Reset Conducts software reset of the DMAC. The DMAC shall be initialized when the Rst bit is set to 1. All the values of the DMAC internal registers become the initial values. Also, all transfer requests are canceled; and the two channels become the stop status. 0: Ignores. 1: Initializes DMAC.

Fig. 10-3 DMA Control Register (DCR)

10.3.2 Channel control register (CCRn)



Bit	Mnemonic	Name of Field	Description
31	Str	Channel start	Start Starts the channel operation. By setting up this bit to 1, the channel becomes the wait status and starts a data transfer in response to a transfer request. "1" is only valid to write in to the Str bit and the writing-in of "0" is ignored. When read out, it is always "0." 1: Starts the channel operation.
24	Stop	Channel stop	Stop Completes the channel operation to change the channel to the halt status. The SARn and the DARn maintain the address following the address with which the last transfer was made. The BCRn maintains the rest of the number of transfer bytes. When the channel is not in the wait status, writing-ins are ignored. "1" is only valid to write in to the Stop bit and the writing-in of "0" is ignored. When read out, it is always "0." Do not set this bit "0" when the channel is not in the wait satate. When the Str bit and the Stop bit are simultaneously set up to 1, the channel becomes the wait status. 1: Competes the channel operation.
23	NIEn	Normal completion interrupt enable	Normal Completion Interrupt Enable 1: Grants a normal completion interrupt 0: Inhibits a normal completion interrupt
22	AbEn	Abnormal completion interrupt enable	Abnormal Completion Interrupt Enable 1: Grants an abnormal completion interrupt 0: Inhibits an abnormal completion interrupt

Fig. 10-4 Channel Control Registers (CCRn) (1/4)

21	CIEn	Continue interrupt enable	Continuous Mode Interrupt Enable 1: Grants the continue mode interrupt 0: Inhibits the continue mode interrupt
19	DIEn	DONE input enable	DONE Interrupt Enable Validates the input of DONE* signal. 1: Finishes transfer operation when DONE* has become low. 0: Does not finish transfer operation even when DONE* has become low.
18	DOEn	DONE output enable	DONE Output Enable Makes the DONE* signal output valid. 1: Assert the DONE* signal when transfer operation ends. 0: Do not assert the DONE* signal when transfer operation ends.
17	Big	Big endian	Big Endian 1: The channel operates with the big endian. 0: The channel operates with the little endian
16	Cont	Continue mode	Continuous Mode 1: Operates in the continue mode. 0: Does not operate in the continue mode. The TX3904 doesn't support the little endian mode.
15	SAM	Single address mode	Single Address Mode Designates the transfer address mode. 1: Single address mode 0: Dual address mode
14	ExR	External request mode	External Request Mode Designates the transfer request mode. 1: External transfer request 0: Internal transfer request

Fig. 10-5 Channel Control Registers (CCRn) (2/4)

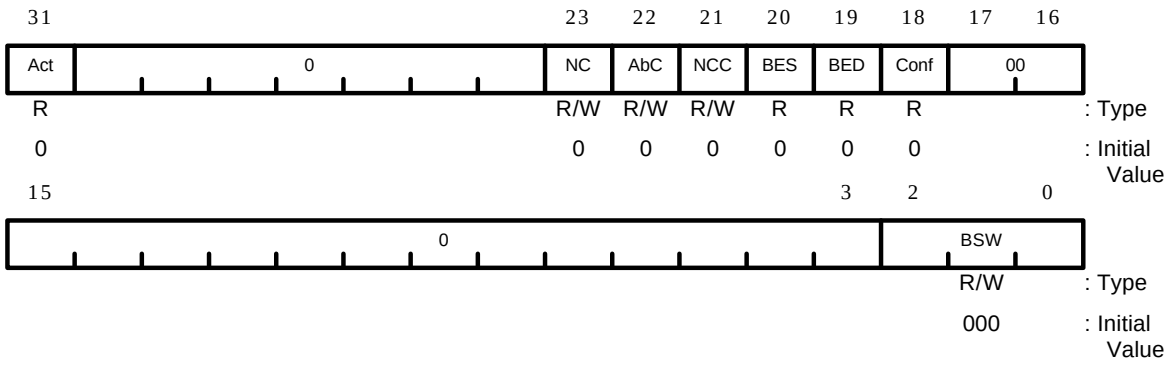
13	PosE	Rising edge	Positive Edge Designates the valid level of the transfer request signal DREQn. This is valid only when the transfer request is an external transfer request (the ExR bit is 1). When it is an internal transfer request (the ExR bit is 0), the value of the PosE shall be ignored. There are the edge detection and the level detection as the methods to acknowledge the DREQn signal; and they are set up in the Lev bit. The active level of the transfer acknowledge signal DACKn is the same as the active level of the DREQn signal. 1: The rising or the high level of the DREQn signal is valid. The active level of the DACKn signal is high. 0: The falling or the low level of the DREQn signal is valid. The active level of the DACKn signal is low.
12	Lev	Level mode	Level Mode Designates the request method of the external transfer request. This is valid only when an external transfer request is set up (the ExR bit is 1) as the transfer request. When an internal transfer request (the ExR bit is 0) is set up, the value of the Lev bit shall be ignored. The valid level of the DREQn signal is set up in the PosE bit. 1: Level mode. Acknowledges the levels (the low level when the PosE bit is 0 and the high level when the PosE bit is 1) of the DREQn signal as data transfer requests. 0: Edge mode. Acknowledges changes (the falling edge when the PosE bit is 0 and the rising edge when the PosE bit is 1) in the DREQn signal as data transfer requests.
11	SReq	Snoop request	SReq (Snoop Request) Designates whether or not to use the snoop function as the bus ownership request mode. When using it, the snoop function of the TX39 Processor Core becomes valid; and the TX39 Processor Core watches the address of the DMA transfer. When not using it, the snoop function of the TX39 Processor Core does not function. 1: Snoop function is used (SREQ). 0: Snoop function is not used (GREQ). The GREQ is not able to be used in the TX3904F.

Fig. 10-6 Channel Control Registers (CCRn) (3/4)

10	RelEn	Bus ownership release request enable	Release Request Enable Designates the response to the bus ownership release request from the TX39 Processor Core. 1: Bus ownership release request is answered. When the TX39 Processor Core issues a bus ownership release request, the DMAC temporarily returns the bus ownership to the TX39 Processor Core at a break of the bus operation. 0: Bus ownership release request is not answered.
9	SIO	Source 1/O	Source Type I/O Designates the source type. 1: I/O device 0: Memory
8:7	SAC	Source address count	Source Address Count Designates the address change of the source. 1x: Address fix 01: Address decrease 00: Address increase
6	DIO	Destination I/O	Destination Type: I/O Designates the destination device. 1: I/O device 0: Memory
5:4	DAC	Destination address count	Destination Address Count Designates the address change method of the destination. 1x: Address fix 01: Address decrease 00: Address increase
3:2	TrSiz	Transfer size	Transfer Size Indicates the transfer data quantity to a transfer request in the external transfer request mode. 11: 8 bits (Byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)
1:0	DPS	Device port size	Device Port Size Designates the bus width of the I/O device that has been designated as the source device or the destination device. 11: 8 bits (Byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)

Fig. 10-7 Channel Control Registers (CCRn) (4/4)

10.3.3 Channel status register (CSRn)



Bit	Mnemonic	Name of Field	Description
31	Act	Channel active	Channel Active Indicates that the channel is in the wait status. 1: Channel is in the wait status. 0: Channel is not in the wait status.
23	NC	Normal completion	Normal Completion Indicates that the channel operation has completed normally. If interrupts at the time of a normal completion are permitted in the CCR register, the DMAC requests an interrupt when the NC bit becomes 1. In the continue mode, the NC bit never becomes 1. By writing "0" in the NC bit, it is cleared to 0. When an interrupt has been requested by a normal completion, the interrupt request shall be withdrawn when the NC bit becomes 0. If the Str bit is attempted to be set to 1 when the NC bit is 1, an error occurs. Please clear the NC bit to 0 when starting the next transfer. The writing-in of "1" shall be ignored. 1: Channel operation has completed normally. 0: Channel operation has not completed normally.

Fig. 10-8 Channel Status Registers (CSRn) (1/3)

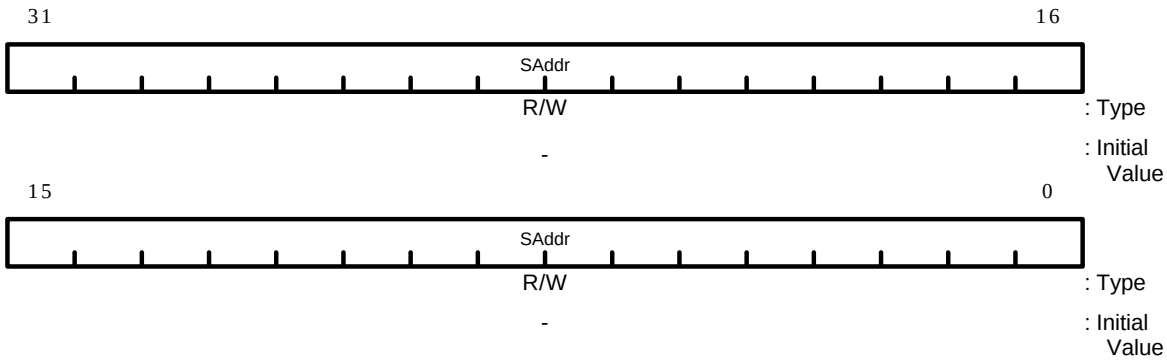
22	AbC	Abnormal completion	Abnormal Completion Indicates that the channel operation has completed abnormally. If interrupts at the time of an abnormal completion are permitted by the CCR register, the DMAC requests an interrupt when the AbC bit becomes 1. By writing "0" in, the AbC bit can be cleared to 0. When an interrupt has been requested by an abnormal completion, the interrupt request shall be withdrawn when the AbC bit becomes 0. When the AbC bit is cleared to 0, the BES, BED, and Conf bits are cleared to 0. If the Str bit is attempted to be set to 1 when the AbC bit is 1, an error occurs. Please clear the AbC bit to 0 when starting the next transfer. The writing-in of "1" shall be ignored. 1: Channel operation has completed abnormally. 0: Channel operation has not completed abnormally.
21	NCC	Continue mode completion	Normal Completion of Continuous Mode Indicates that the data transfer has completed normally in the continue mode. When permitted in the CIEn of the CCR, an interrupt is requested as the continue mode interrupt. The NCC can be cleared by writing "0" in. Being different from the NC or AbC, the next transfer starts without clearing the NCC to 0. The writing-in of "1" shall be ignored. 1: Data transfer has normally completed at the time of the continue mode. 0: Data transfer has not completed normally.
20	BES	Source bus error	Source Bus Error 1: Bus error has occurred at the time of the source access. 0: Bus error has not occurred at the time of the source access.
19	BED	Destination bus error	Destination Bus Error 1: Bus error has occurred at the time of the destination access. 0: Bus error has not occurred at the time of the destination access.
18	Conf	Configuration error	Configuration Error 1: There was a configuration error. 0: There was no configuration error.

Fig. 10-9 Channel Status Registers (CSRn) (2/3)

2:0	BSW	BSTART wait	BSW(BSTART Wait Cycle) Specifies the BSTART* wait clock count in single address mode I/O to memory transfer. 000: 0 wait cycle 001: 1 wait cycle : : 111: 7 wait cycles

Fig. 10-10 Channel Status Registers (CSRn) (3/3)

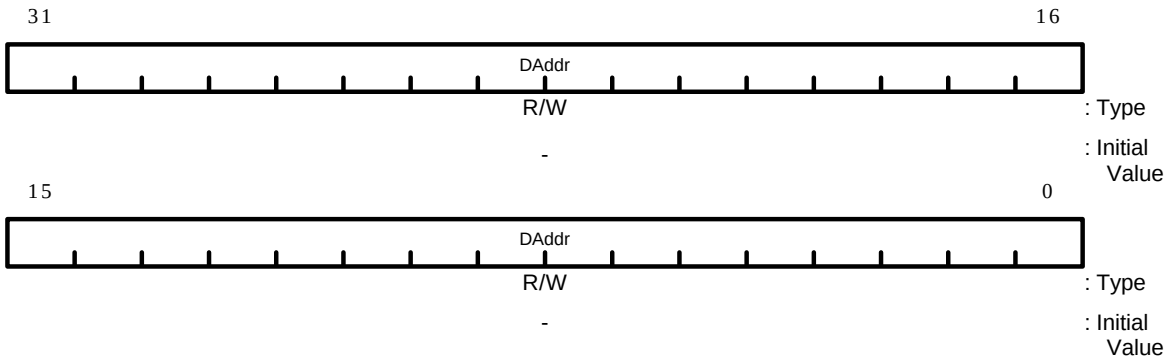
10.3.4 Source address register (SARn)



Bit	Mnemonic	Name of Field	Description
31:0	SAddr	Source address	Source Address Sets up the source address that is to be the data transfer origin with a physical address. The value changes by the value designated in the DPS of the CCRn. If the mode is set to the continue mode when the source is an I/O device in the single address mode, the SAddr indicates the next transfer start address. When the data transfer has completed, the SAddr value is loaded to the DAR to start the next data transfer.

Fig. 10-11 Source Address Registers (SARn)

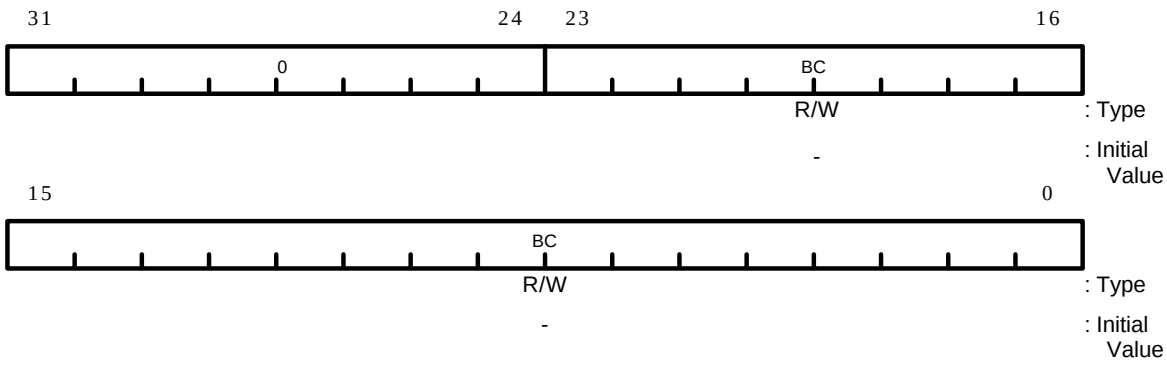
10.3.5 Destination address register (DARn)



Bit	Mnemonic	Name of Field	Description
31:0	DAddr	Destination address	Destination Address Sets up the destination address that is to be the data transfer destination with a physical address. The value changes by the value designated in the DPS of the CCRn. If the mode is set to the continue mode when the destination is an I/O device in the single address mode, the DAddr indicates the next transfer start address. When the data transfer has completed, the DAddr value is loaded to the SAR to start the next data transfer.

Fig. 10-12 Destination Address Registers (DARn)

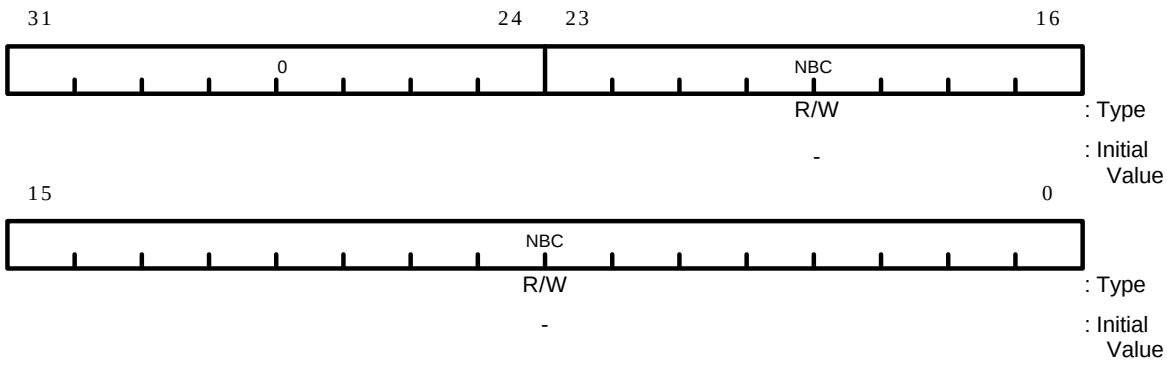
10.3.6 Byte count register (BCR0n)



Bit	Mnemonic	Name of Field	Description
23:0	BC	Byte count	Byte Count Sets up the number of bytes to be data transferred. The value decreases by the number of data transferred (by the value designated in the TrSiz of the CCRn).

Fig. 10-13 Byte Count Registers (BCRn)

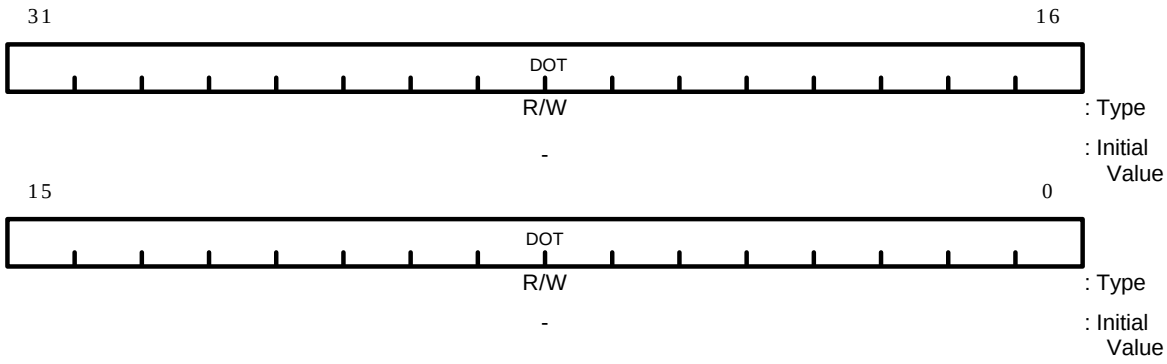
10.3.7 Next byte count register (NCR0/1)



Bit	Mnemonic	Name of Field	Description
23:0	NBC	Next byte count	Next Byte Count Sets up the number of transfer bytes of the next data transfer in the continue mode. When the data transfer has completed, the NBC value is loaded to the BCRn to start the next data transfer.

Fig. 10-14 Next Byte Count Registers (NCR)

10.3.8 Data holding register (DHR)



Bit	Mnemonic	Name of Field	Description
31:0	DOT	Transfer data	Data on Transfer The data that were transferred in the dual address mode and were read from the source.

Fig. 10-15 Data Holding Registers

10.4 Functions

This section explains the functions of DMAC0. DMAC1 has the same functions as DMAC0.

10.4.1 Overview

The DMAC is a 32-bit DMA controller that can transfer data inside the system using the TX39 Processor Core at a high speed without the TX39 Processor Core.

(1) Source and destination

The DMAC conducts the data transfer between memories or between a memory and an I/O device. The device at the data transfer origin is called a source device and the device at the data transfer destination is called a destination device. Memories and I/O devices can be designated as a source device or a destination device. The DMAC transfers from a memory to an I/O device, from an I/O device to a memory, and from a memory to another memory; and it cannot transfer from an I/O device to another I/O device.
The difference between a memory and an I/O device is the access method to the device. When the DMAC accesses an I/O device, it asserts the DACKn signal. There is only one DACKn signal in one channel so that there is only one I/O device that can be used when transferring. Therefore, the transfer between I/O devices cannot be done.
Channel 1 (DMAC0) and Channel 2 (DMAC1) can select the TX3904 built-in serial I/O as the I/O device. This selection of the serial I/O is conducted in the CConR of the EBIF.

(2) Transfer of bus ownership (Bus arbitration)

By a transfer request from inside or outside the DMAC, the DMAC requests of the TX39 Processor Core for the bus ownership. When a responding signal comes from the TX39 Processor Core, the DMAC is granted the bus ownership to execute the bus cycles of data transfer.
In the bus ownership request of the DMAC, there are modes that requests/does not request for snooping of the TX39 Processor Core built-in cache. The selection of the mode is set up in the register of each channel.
The TX39 Processor Core sometimes requests the release of the bus ownership. Whether or not to respond to this request is set up in the register of each channel.
When the transfer request is no longer there, the DMAC releases the bus ownership.

(3) Transfer request mode

In the transfer requests of the DMAC, there are an internal request mode and an external request mode.
The internal request mode is a mode that generates transfer requests inside the DMAC. When the start bit of the DMAC built-in register (Str bit of the channel control register) is set to 1, a transfer request is generated and the DMAC starts a transfer operation.
The external request mode is a mode that generates transfer requests by the input of the transfer request signal (DREQn) that the I/O device outputs. There are a level mode that generates

transfer requests by the level detection of the DREQn signal and a edge mode that generates transfer requests by the edge detection of the DREQn signal.

(4) Address modes

There are two address modes--a single address mode and a dual address mode.
The single address mode conducts the data transfer between a memory and an I/O device. The memory is accessed by the address that the DMAC outputs and the I/O device is accessed through the DACKn signal. This mode conducts reading from the source device and writing into the destination device in one bus operation.
The dual address mode conducts the data transfer between memories or between a memory and an I/O device. The addresses of the source device and the destination device are output by the DMAC. When accessing the I/O device, the DMAC asserts the DACKn signal. This mode implements the two bus operations--the read operation and the write operation. The transfer data that were read out from the source device is temporarily taken in to the data holding register (DHR) inside the DMAC and then is written into the destination device.

(5) Next transfer address can be set up (Continue mode)

In the single address mode, the next transfer starting address can be set up in the register in advance (the continue mode). When the previous transfer has completed normally, the channel being operated in the continue mode starts another data transfer, regarding the next transfer address set up in the register as the transfer address of the new memory.

(6) Channel operations

DMAC0 has two channels (Channel 0 and Channel 1). The channel is turned on by setting the start (Str) bit of the channel control register (CCRn) to 1; and it shall be in the wait status.
If a transfer request occurs when the channel is in the wait status, the DMAC is granted the bus ownership to conduct the data transfer. When the transfer request is no longer there, the DMAC releases the bus ownership to become the wait status. When the transfer completes, the channel becomes the halt status. In the transfer completion, there are a normal completion and an abnormal completion by causes such as an error occurrence. At the completion of a transfer, an interrupt signal can be generated.
The following Figure 10-10 shows the outline of the status changes of the channel operations.

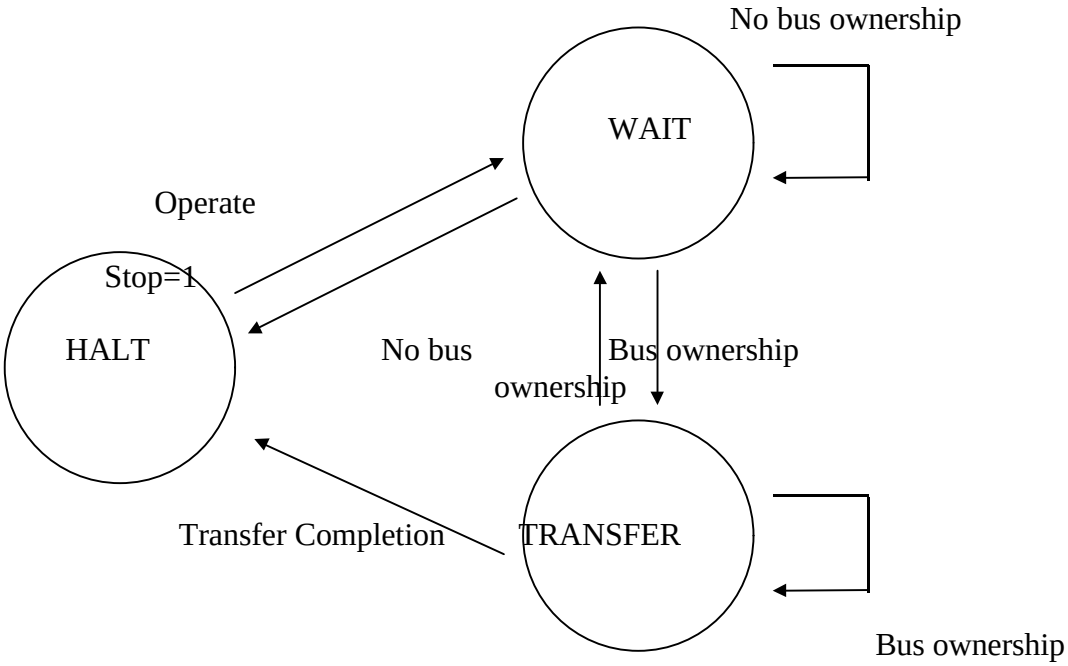


Fig.10-16 Status Shifts between Channel Operations

(7) Summary of transfer method combinations

The following table shows the transfers that can be done by the DMAC by combining the modes.

Transfer Request	Edge/Level	Address Mode	Continue Mode	Transfer Device
Internal	--	Dual	Inhibited	Memory → Memory
External	Edge	Dual	Inhibited	Memory → I/O
				I/O → Memory
		Single	Enabled	Memory → I/O
				I/O → Memory
	Level	Dual	Inhibited	Memory → I/O
				I/O → Memory
		Single	Enabled	Memory → I/O
				I/O → Memory

10.4.2 Transfer requests

In order to conduct a data transfer by the DMAC, it is necessary to generate a transfer request to the DMAC. There are two kinds in the DMAC's transfer requests--an internal transfer request and an external transfer request. The transfer requests can be set up for each channel.
For either transfer request, if a transfer request occurs after the channel operation is started, the DMAC is granted the bus ownership to conduct the data transfer.

Internal transfer request

If "1" is set to the Str bit of the CCR when the ExR bit of the CCRn is 0, a transfer request occurs immediately. This transfer request is called the internal transfer request.
In the internal transfer request, a transfer request exists until the channel operation has completed so that data are transferred consecutively unless a shift to a channel whose priority is higher or a shift of the bus ownership to a bus master whose priority is higher occurs.

External transfer request

When the ExR bit of the CCRn is 1, if a transfer request is informed from the external circuit through the DREQn signal that supports to the channel after the Str bit of the CCR has been set to 1 and the channel has become the wait status, a transfer request is generated. This transfer request is called the external transfer request. The external transfer request is used for the transfer between a memory and an I/O device.
In the acknowledgment method of the DREQn signal, there are an edge mode that acknowledges edges and a level mode that acknowledges levels. The polarity of the edge or level that is to be acknowledged by the PosE bit of the CCRn can be designated.
The data transfer unit for one transfer request is designated in the TrSiz field of the CCRn. Either 32-bit, 16-bit, or 8-bit can be designated.
In the edge mode, it is necessary at each transfer request to deassert the DREQn signal and then to assert it to made a valid edge; however, in the level mode, consecutive transfer requests can be acknowledged by maintaining the valid level.
Do not write "1" into the stop bit in the channel control register during continuous transfer in the level mode. If it is necessary to suspend a data transfer by the stop bit in the level mode, the DREQn signal must be asserted and deasserted for each word transferred.
There are two DMAC operation, synchronized to the SYSCLK and to the internal clock (GCLK). The SYSCLK and GCLK is the same in the full speed bus mode. On the other hand, it is necessary to take care of the external signals in the half speed bus mode.

- DREQn (input)

Since three of DREQ[3:1] is synchronized by the SYSCLK, they are noticed to the DMAC with a delay of one SYSCLK. The DREQ[0] is not synchronized so it is directly informed to the DMAC.

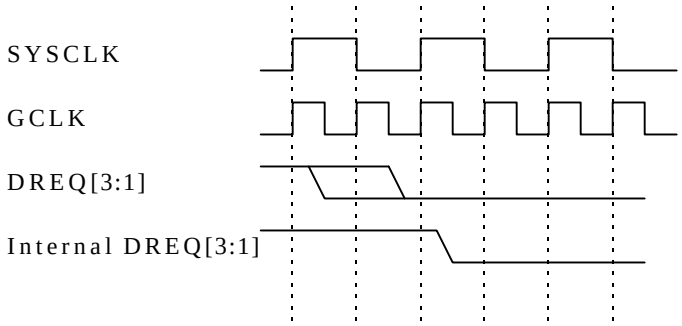


Fig. 10-17 DREQ[3:1] in the half speed bus mode

- DACKn (output)

This signal is synchronized to the GCLK, so it is changed at the rising edge of the SYSCLK or falling edge.

- BSTART* (output)

This signal is synchronized to the SYSCLK.

- R/W* (output)

This signal is synchronized to the GCLK.

- ACK* (input)

This signal is synchronized by the SYSCLK. It is used at a time of I/O access in the dual address mode. Do not assert the ACK* from outside of the TX3904 since the on-chip memory controllers generate an internal acknowledge signal during memory access in the dual address mode, and in the single address mode. Note that the internal acknowledge is invisible from outside.

Level Mode

In the level mode, the internal DREQn signal (dreq) is level-detected at a rising of the internal clock (GCLK). If the active level is detected in the dreq signal when the channel is in the wait status, the DMAC switches to the transfer status to start data transfer. The active level of the DREQn signal is set up in the PosE bit of the CCRn. The active level of the DACKn signal is the same as the active level of the DREQn signal.

If the external circuit has asserted the DREQn signal, please maintain the DREQn signal at the active level until the DACKn signal is asserted in the I/O device access cycle. If the DREQn signal is deasserted before the DACKn signal is asserted, the transfer request may not be acknowledged.

If the dreq signal is at the active level at a rising of the GCLK that acknowledges the assertion of the acknowledge signal (which is generated by the on-chip memory controllers at a memory access in the dual address mode or in the single address mode, or is the same as the ACK* signal at an I/O access in the dual address mode), the next data transfer is conducted immediately afterwards. However, if a transfer request is generated on another channel with a higher priority, a channel transit takes place.

If the dreq signal is not at the active level at a rising of the GCLK that acknowledges the assertion of the acknowledge signal, it is understood that there is no transfer request so that a transfer operation on another channel may be started or the bus ownership may be released to become the wait status.

The acknowledge signal is recognized at the rising edge of the GCLK which precedes the GCLK cycle when the final LAST* signal for a data transfer is negated.

The unit of the transfer request is designated in the TrSiz field of the CCRn.

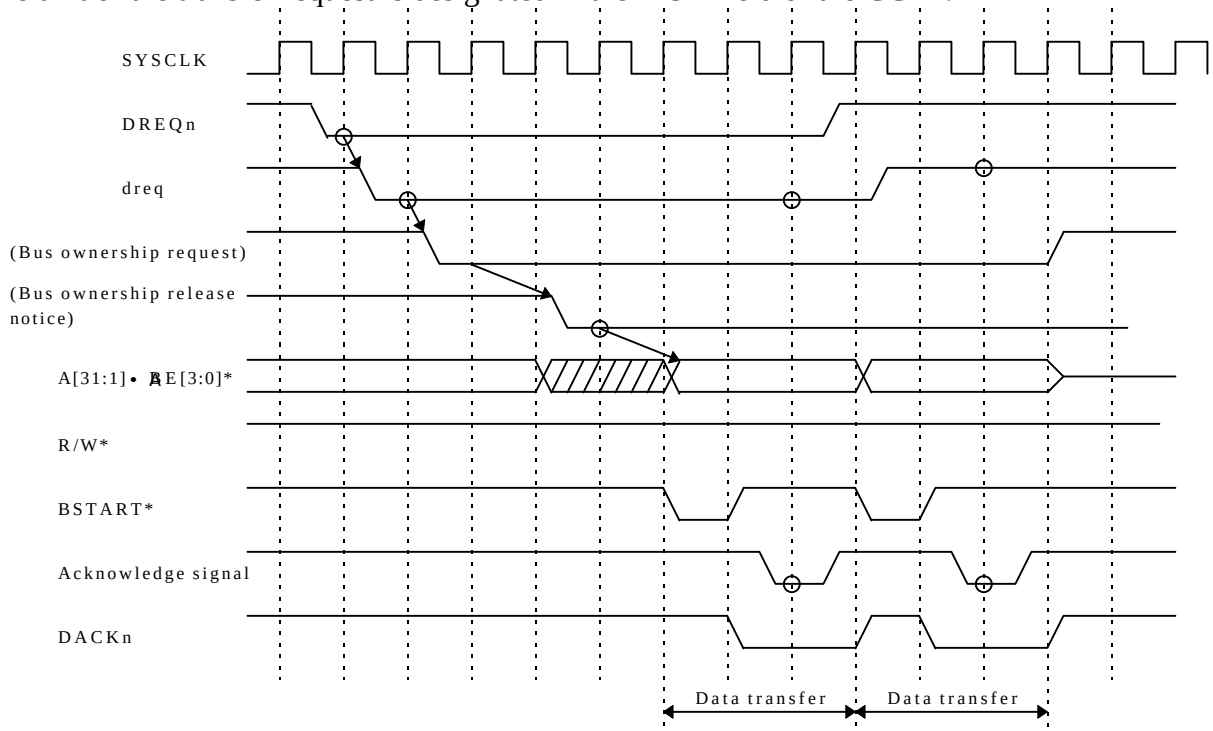


Fig. 10-18 Transfer Request Timing (Level Mode)

Edge Mode

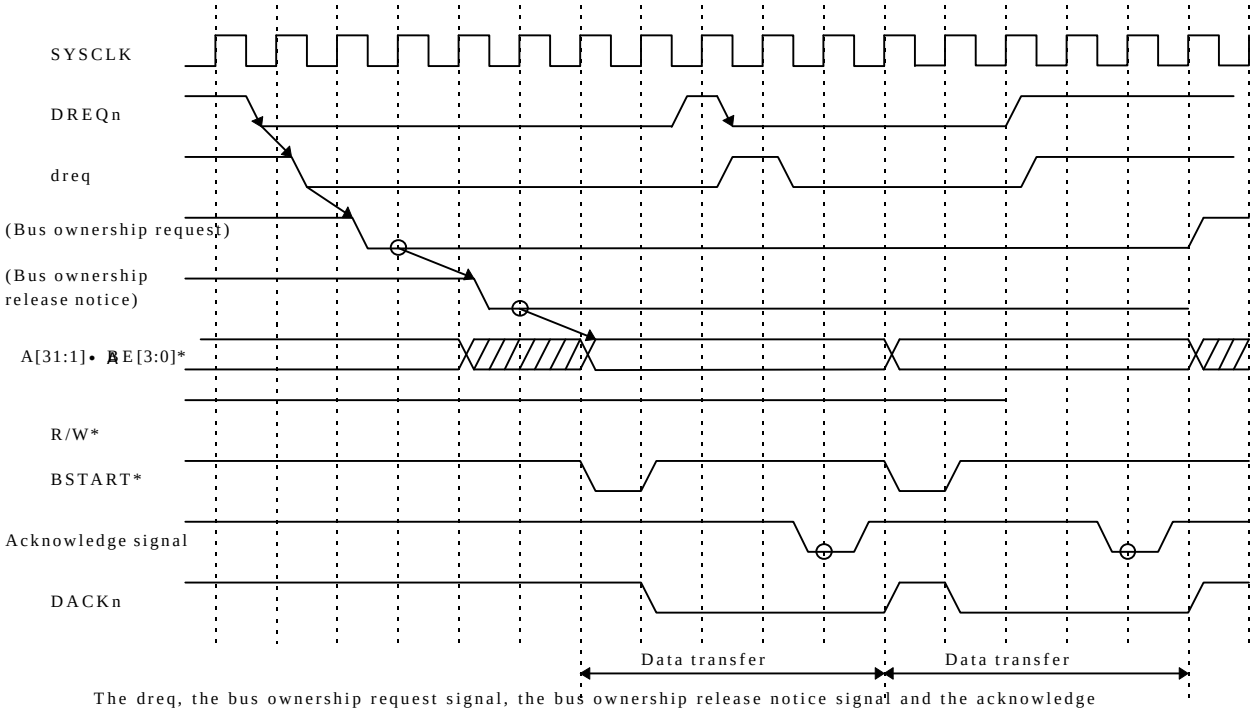
In the edge mode, the internal DREQn signal (dreq) is edge-detected. If the valid edge of the dreq signal is acknowledged at a rising of the internal clock (GCLK) (if it is presently at the active level although it was not at the active level at the rising of one GCLK before) when the channel is in the wait status, the DMAC acknowledges that there is a transfer request and switches to the transfer status to start the transfer operation. The active edge (falling or rising) of the DREQn signal is set up in the PosE bit of the CCRn. The active level of the DACKn signal is at high when a rising of the DREQn signal is on the valid edge, and at low when a falling of the DREQn signal is on the active edge.

If the valid edge of the dreq signal is acknowledged by a rising of the GCLK that acknowledges the assertion of the acknowledge signal (which is generated by the on-chip memory controllers at a memory access in the dual address mode or in the single address mode, or is the same as the ACK* singal at an I/O access in the dual address mode), the next data transfer is conducted immediately afterwards. However, if a transfer request is generated on another channel with a higher priority, a channel transit takes place.

If there is no valid edge of the dreq signal by the rising of the GCLK that acknowledges the assertion of the acknowledge signal, it is understood that there is no transfer request so that a transfer operation on another channel may be started or the bus ownership may be released to be in the wait status.

The acknowledge signal is recognized at the rising edge of the GCLK which precedes the GCLK cycle when the final LAST* signal for a data transfer is negated.

The unit of the transfer request is designated in the TrSiz field of the CCRn.



The dreq, the bus ownership request signal, the bus ownership release notice signal and the acknowledge signal are internal signals of the TX3904.

Fig. 10-19 Transfer Request Timing (Edge Mode)

10.4.3 Address modes

At the address mode, the DMAC outputs addresses to both the source device and the destination device to conduct transfer operations or it outputs its address to one of the devices to designate whether or not the transfer operation shall be implemented. The former is called a dual address mode, and the latter is called a single address mode.

In the dual address mode, the DMAC first implements a read operation to the source device. At this time, the data being output by the source device are temporarily stored in the register (DHR) inside the DMAC. Then, the DMAC writes in these data by implementing a write operation to the destination device to realize the data transfer from the source device to the destination device.

In the single address mode, the DMAC outputs its address to the memory and the DACKn signal to the I/O device to conduct the data transfer between the memory and the I/O device in one bus cycle.

The address mode is designated in the SAM bit of the CCRn.

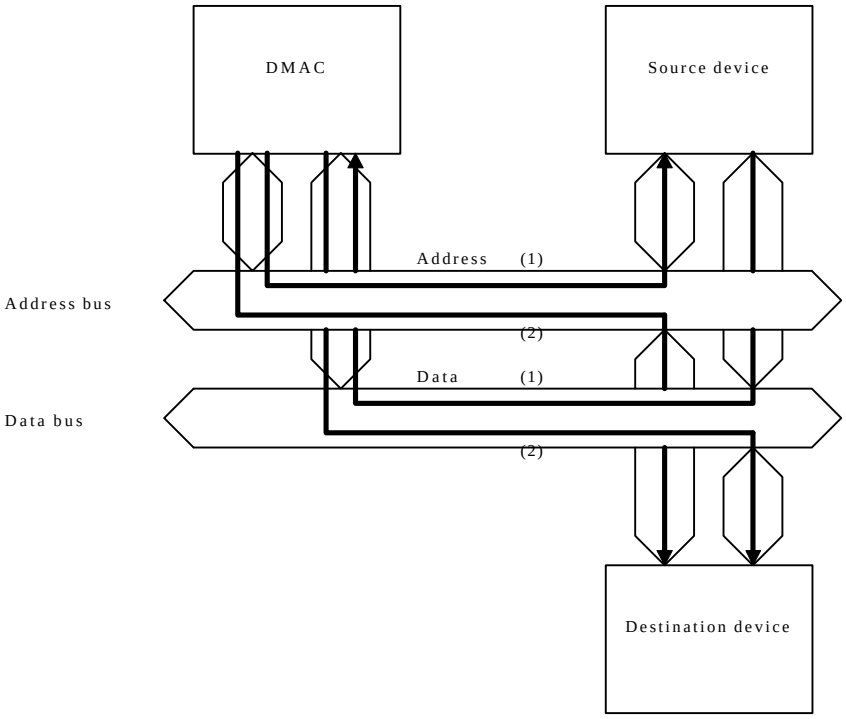


Fig. 10-20 Dual address mode transfer

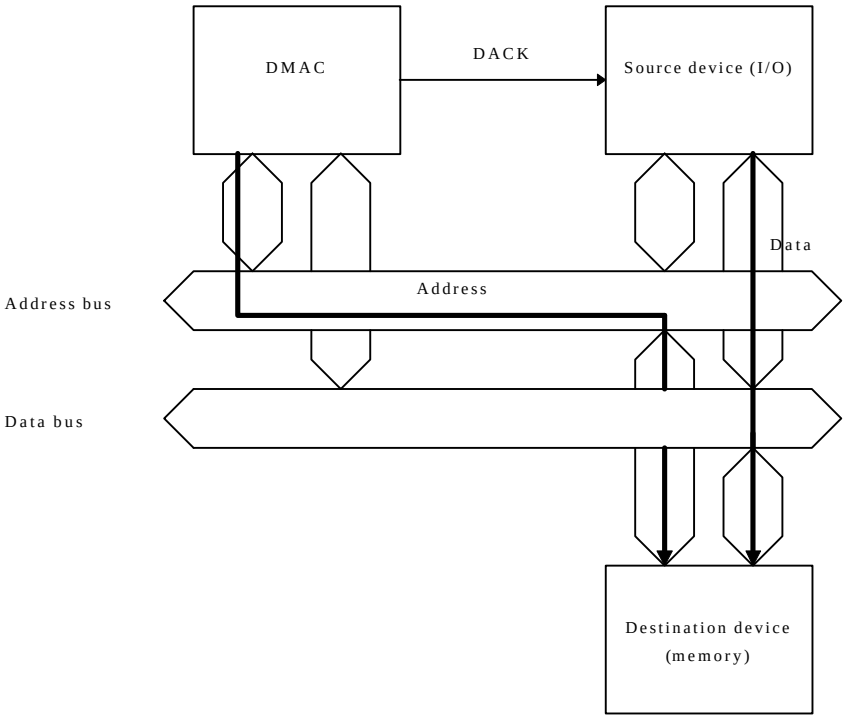


Fig. 10-21 Single address mode transfer

Dual address mode

The dual address mode is a mode that executes the transfer by the following two bus operations:
— A read operation to output the address of the source device and to read the data from the source device to take them in to the register (DHR) inside the DMAC.
— A write operation to output the address of the destination device and to write the data of the DHR to the destination device.
In the dual address mode, the following three kinds of transfers are possible:
— Memory → Memory
— Memory → I/O device
— I/O device → Memory

The data transfer unit of the DMAC is the data amount that was designated in the TrSiz field of the CCRn (32 or 16 bits). In the external transfer request, the data of the amount of the transfer unit are transferred every time a transfer request is acknowledged. It is always a word (4 bytes, 32 bits) for the transfer of the internal transfer request.
In the dual address mode, the data of the amount of the data transfer unit are read in to the DHR from the source device, and then the data are written in to the destination device.
A memory access occurs according to the data transfer unit. However, two 16-bit memory accesses occur when the data transfer unit is 32 bits and 16-bit width memory is designated by memory controller.
Separated from the data transfer unit, the bus width (the device port size) of the I/O device is designated in the DPS field of the CCRn for the data transfer from a memory to an I/O or from an I/O to a memory. The device port size is 32, 16, or 8 bits.
When the data transfer unit and the device port size are the same, the DMAC conducts one read operation or write operation to the I/O device. The bus operation to a memory is always one time; so, in this case, one transfer completes with two bus operations.
When the device port size is smaller than the data transfer unit, the DMAC conducts multiple read operations or write operations to the I/O device. For example, to transfer data from an I/O device whose data transfer unit is 32 bits and whose device post size is 8 bits to a memory, 8-bit data are read out from the I/O device four times to be stored in the DHR; then the 32-bit data are written in to the memory at one time (twice for 16-bit width memory).
The address and the value of BCRn are changed with the data transfer unit.
The set-up where the device port size is larger than the data transfer unit is inhibited.

The following table shows the summary:

Table 10-3 Data Transfer Unit and Device Port Size (Dual Address Mode)

TrSiz	DPS	Bus Operation for I/O Device
32	32	One time
32	16	Two times
32	8	Four times
16	32	Set up inhibited
16	16	One time
16	8	Two times
8	32	Set up inhibited
8	16	Set up inhibited
8	8	Set up inhibited

Single address mode

The single address mode enables the transfer between a memory and an I/O device. The DMAC outputs the memory address to the memory and outputs the DACKn signal to the I/O device. The data transfer between the memory and the I/O device is conducted in one bus operation.
In the single address mode, the R/W* signal indicates the bus operation for the memory. The R/W* signal is high at a memory to I/O device transfer and is low at a I/O device to memory transfer.
The data transfer unit is designated in the TrSiz field of the CCRn.
If the data transfer size and the device port size are set up in different sizes in the single address mode, it is operated in the burst mode of (the data transfer size / the device port size)-times. For example, if the data transfer unit is 32 bits and the device port size is 8 bits, it is a burst operation of 4 words. The address changes by the data transfer unit. The BCRn value also changes by the data transfer unit. The set-up where the device port size is larger than the data transfer unit is inhibited. When the data transfer unit is 32 bits and the device port size is 8 bits, bus operation becomes a burst transfer.
The following table shows the relationship between the data transfer unit and the device port size.

Table 10-4 Data Transfer Unit and Device Port Size (Dual Address Mode)

TrSiz	DPS	Bus Operation for I/O Device
32	32	One time
32	16	Set up inhibited
32	8	Four times: Burst
16	32	Set up inhibited
16	16	One time
16	8	Set up inhibited
8	32	Set up inhibited
8	16	Set up inhibited
8	8	One time

In the single address mode, the DMAC does not input and output data so that it does not drive the data bus.

10.4.4 Burst transfer

In the single address mode, the burst transfer is supported for the fast data transfer. It is available for the data transfer between 32-bit width memory and 32-bit width I/O device. Set 0x (32 bits) to the TrSiz and 11 (8 bits) to the DPS in the single address mode in order to use burst transfer.

In the burst transfer, four words data transfer are occurred for one data transfer request. The lower four bits of a start address of a burst transfer is restricted as follows.

Start address	address change
0	0→4→8→C
C	C→8→4→0
others	inhibited

10.4.5 Continue mode

In the single address mode, the continue mode can be used for operation. The set-up of the continue mode is designated in the Cont bit of the CCRn.

In the continue mode, immediately after a data transfer has completed normally, the next data transfer starts. The DARn value is loaded to the SARn when a memory is the source device; and the SARn value is loaded to the DARn when a memory is the destination device; and a new data transfer start address is implemented. Also, the NCRn value is loaded to the BCRn to change to the number of a new transfer bytes.

In the continue mode, the NCC bit of the CSRn is set to 1 when a data transfer completes normally. A continue interrupt occurs when the CIEn bit of the CCRn is 1. Then, the next transfer address and the number of bytes are set up, and automatically the Cont bit is cleared to 0. After this series of operations, when the bus ownership has returned to the TX39 Processor Core, set up the next transfer address and the number of the next transfer bytes and set the Cont bit to 1 as a process to cope with the continue interrupt. If the Cont bit is not set to 1, the channel operation finishes when the data transfer has completed.

The channel operation in the continue mode can be finished also by setting the Stop bit to 1.

10.4.6 Channel operation

The channels are turned on when the Str bit of the CCRn of each channel is set to 1. When a channel is turned on, a starting check-up is conducted; and if there is no error, the channel shall be in the wait status.
If a transfer request occurs when the channel is in the wait status, the DMAC is granted the bus ownership to start the transfer operation.
In the completion of the channel operation, there are a normal completion and a abnormal completion due to causes such as a forceful termination and a error occurrence. The status of the completion is indicated in the CSRn.

Start of channel operation

A channel is turned on when the Str bit of the CCRn is set to 1.
When the channel is turned on, configuration errors are checked for; and if there is no error, the channel becomes the wait status. If an error is detected, the channel completes abnormally.
When a channel becomes the wait status, the Act bit of the CSRn of this channel becomes 1.
When the channel is set up for the internal transfer request, transfer request(s) immediately occur; and the DMAC is granted the bus ownership to start the data transfer.
If the channel is set up for an external transfer request, data transfer starts when the DREQn is asserted.

Completion of channel operation

In the channel operation completion, there are a normal completion and an abnormal completion. If a completion is a normal completion or an abnormal completion is indicated in the CSRn.
If it is attempted to set 1 to the Str bit when the NC bit or the AbC bit is 1, the channel operation does not start and completes abnormally.

Normal completion

The channels complete normally in the following three cases. However, in the continue mode, the completion by the Str bit is the only completion of the channel operation; and in the other two cases, the current transfer operation is completed to start the next transfer operation. In the normal completion, the completion invariably occurs after the completion of the transfer of the data transfer unit (the value that was set up in the TrSiz field of the CCRn).
When the contents of the BCRn has become 0 to complete the data transfer
When 1 is set to the Stop bit of the CCRn while in the wait status
When a low is input to the DONE* signal during the data transfer (Read the BCRn in order to check whether the data transfer is terminated by DONE* signal or not.)

Abnormal completion

The following cases are the abnormal completion of the DMAC.

Completion by configuration error

The configuration error is a mistake in the set-up of the DMA transfer. A configuration error occurs before a data transfer operation is started so that the values of SARn, DARn, and BCRn are the same as the set-up time. When a channel completes abnormally by a configuration error, the AbC bit of the CSRn is set to 1 and, at the same time, the Conf bit is set to 1. The following are the causes of the configuration error:

- To have set up I/O devices for both the source device and the destination device.
- To have set up memories for both the source device and the destination device in the single mode.
- To have set up the continue mode when it is the dual address mode.
- To have set 1 to the Str bit of the CCRn when the value of NC bit or the AbC bit of the CSRn is 1.
- To have set up the BCRn with a value that is indivisible by the data transfer unit.
- To have set up the SARn and the DARn with a value that is indivisible by the data transfer unit.
- To have set up the data transfer unit larger than the device port size.
- To have set 1 to the Str bit of the CCRn when the BCRn value is 0.

Completion by bus error

By an abnormal completion by a bus error, the AbC bit of the CSRn is set to 1 and, at the same time, 1 is set to the BES bit or the BED bit

- A bus error is informed during the data transfer.

If the BUSERR* is low at a rising of the clock that acknowledges the acknowledge signal, the DMAC terminates the data transfer.

Priority of channels

The priority of Channel 0 is always higher of the two channels of the DMAC. Therefore, if Channel 0 and Channel 1 simultaneously generate transfer requests, the transfer operation to the transfer request of Channel 0 is conducted first. If Channel 1 still has the transfer request when the transfer request of Channel 0 is no longer there, the transfer operation of Channel 1 is executed (Internal transfer requests are maintained unless the transfer requests are cleared; and external transfer requests are maintained in the edge mode and they are not maintained in the level mode: Please keep asserting the DREQn signal in the level mode).

If a transfer request occurs on Channel 0 during the data transfer of Channel 1, a channel transit takes place. The data transfer of Channel 1 is suspended and the transfer of Channel 0 starts. When the transfer request of Channel 0 is no longer there, the transfer operation of Channel 1 resumes.

A channel transit takes place at the transfer completion of the data transfer unit; namely in:

- Single address mode: At the completion of each bus operation
- Dual address mode: When the data inside the DHR have all been written in.

Interrupts

The DMAC can request interrupts of the TX39 Processor Core at the channel operation completion.
In interrupt, the are three kinds--a normal interrupt, an abnormal interrupt, and a continue interrupt.

Normal interrupt

When a channel operation has completed normally, the NC bit of the CSRn is set to 1. At this time, if the normal completion interrupt is permitted in the NIEn bit of the CCRn, an interrupt is requested of the TX39 Processor Core.
In the continue mode (when the Cont bit is 1), a normal completion interrupt shall not occur.

Abnormal interrupt

When a channel operation has completed abnormally, the AbC bit of the CSRn is set to 1. At this time, if the abnormal completion interrupt is permitted in the AbIEn bit of the CCRn, an interrupt is requested of the TX39 Processor Core.

Continue interrupt

When a transfer operation has completed while operating in the continue mode, the NCC bit of the CSRn is set to 1. At this time, if the continue interrupt is permitted in the CIEn bit of the CCRn, an interrupt is requested of the TX39 Processor Core.
The continue interrupt is an interrupt that the TX39 Processor Core generates to set up the address and the number of bytes of the next transfer (the second transfer after the completed transfer), to set the Cont bit, and to clear the NCC bit . Therefore, the next transfer starts even if the NCC bit remains 1.

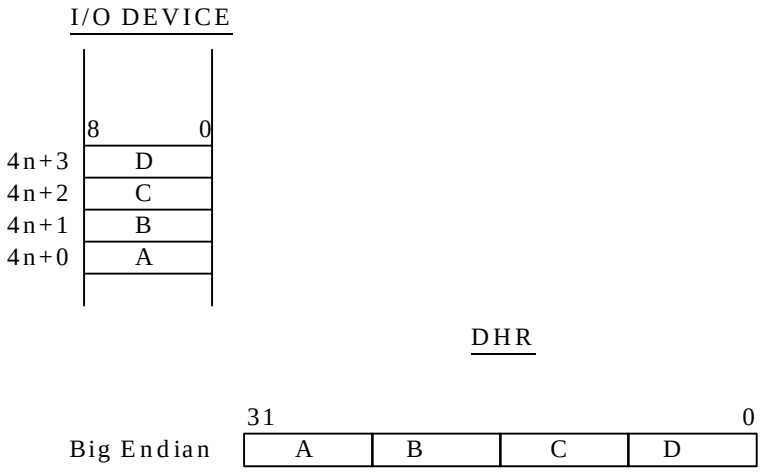
10.4.7 Endian switch function

When the data transfer unit and the device port size are different in the dual address mode, the DMAC combines or disassembles the data in the DHR.

For example, if the port size of the I/O device that is the source device is 8 bits and the data transfer unit of the memory that is the destination device is 32 bits, the DMAC reads out 8-bit data four times from the I/O device and assembles them as 32-bit data in the DHR to write into the memory.

The following diagram shows an example of a relationship of the data order with an 8-bit I/O device and a 32-bit DHR.

The TX3904 supports only the big endian.



10.5 Operations

The DMAC's operation is conducted synchronously with the rising edges of the SYSCLK.

10.5.1 Dual address mode

Memory to memory transfer

The following Fig. 9-22 shows the timing of one word in the case where the data are transferred from a DRAM to another DRAM.

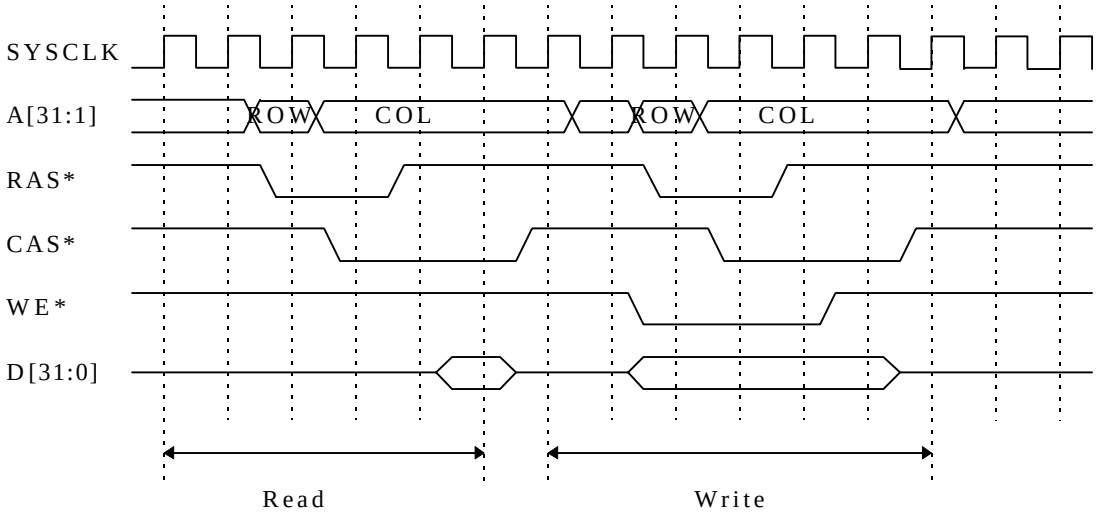


Fig. 10-22 Dual Address Mode (Memory → Memory)

Memory to I/O device transfer

Fig. 9-23 shows the timing of a memory to I/O device transfer when the data transfer unit is set at 32 bits and the device port size at 16 bits:

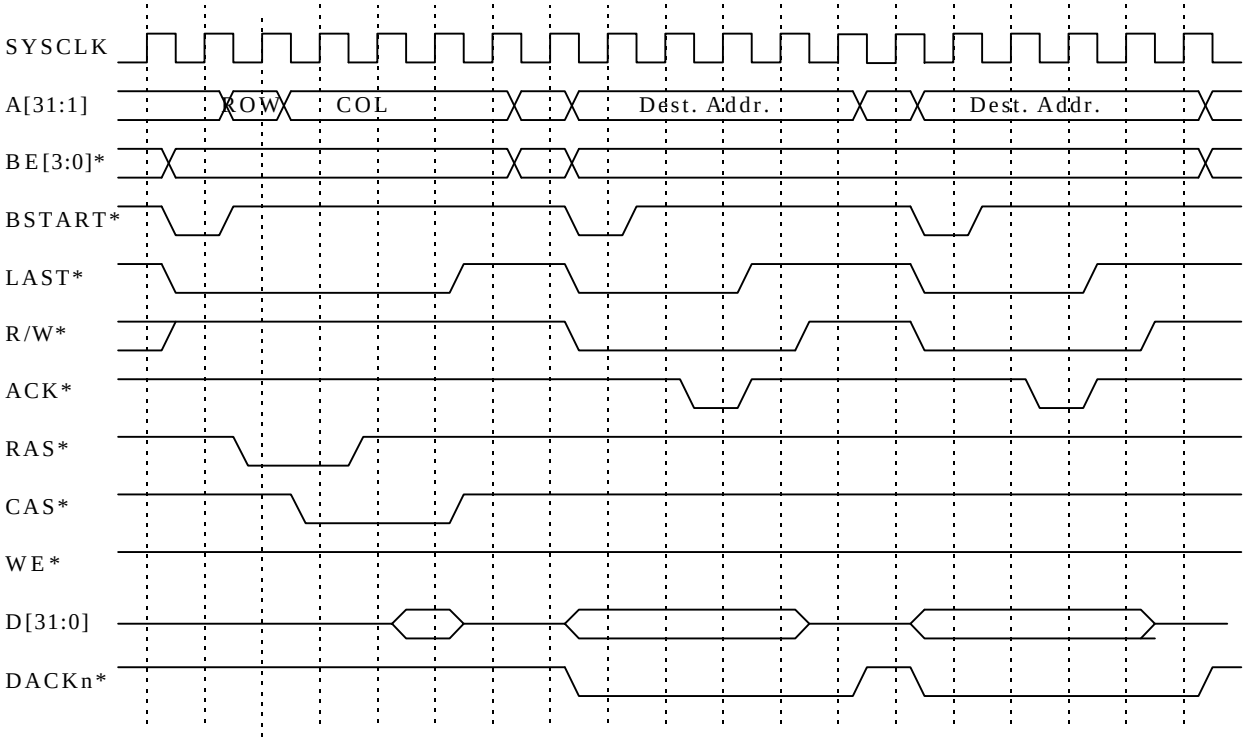


Fig. 10-23 Dual Address Mode (Memory → I/O Device)

I/O device to memory transfer

Fig. 9-24 shows the timing of an I/O device to memory transfer when the data transfer unit is set at 32 bits and the device port size at 32 bits:

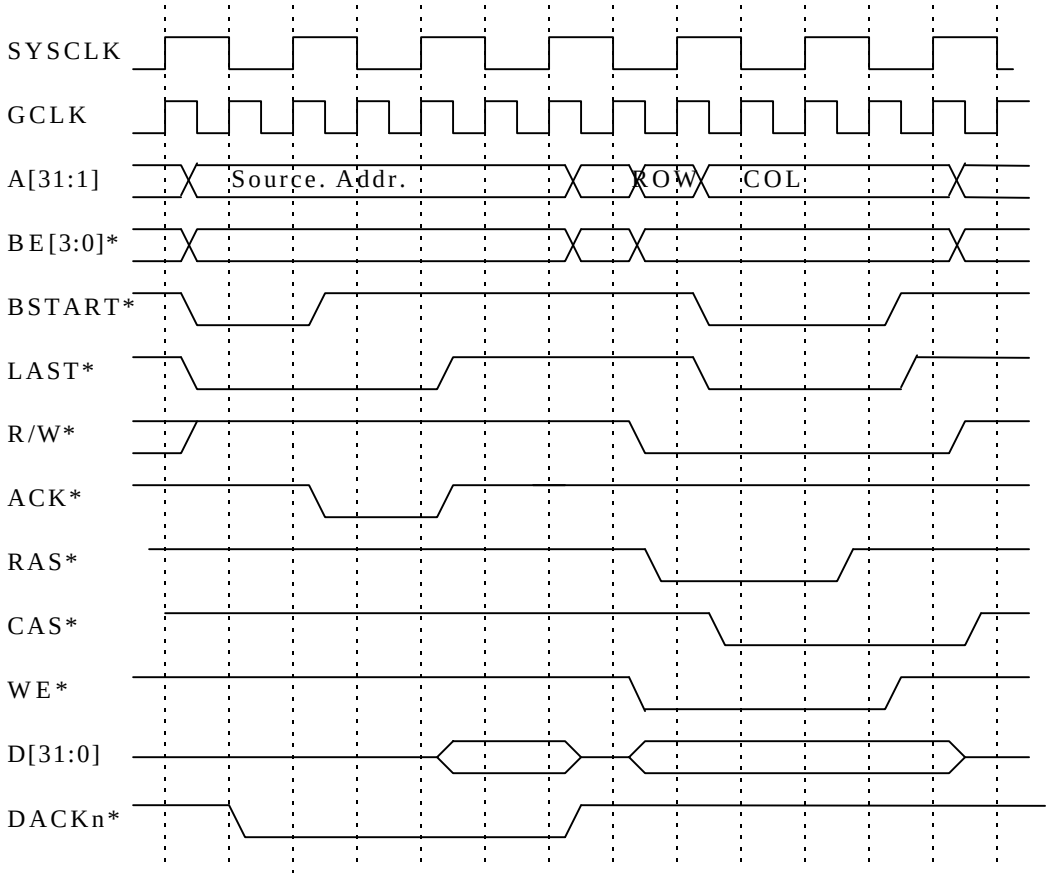


Fig. 10-24 Dual Address Mode (I/O Device → Memory)

10.5.2 Single address mode

In the single address mode, the acknowledge signal is driven by the TX3904 built-in memory controller so that it is not necessary to input from outside. The R/W* signal indicates the bus operation for the memory.

Memory to I/O device

Fig. 9-25 shows the timing of a memory to I/O device transfer when the data transfer unit is set at 32 bits and the device port size at 32 bits:

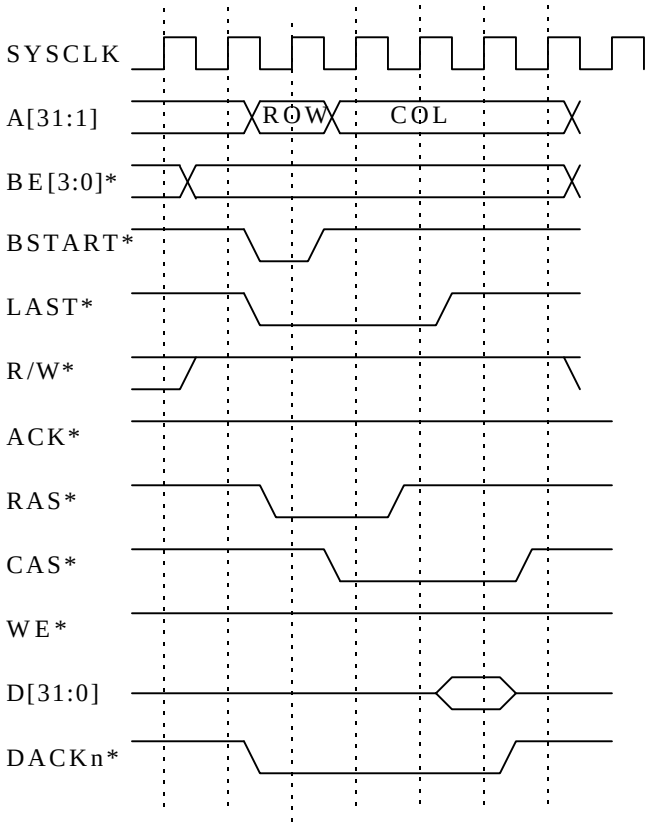


Fig. 10-25 Single Address Mode (Memory → I/O Device)

I/O device to Memory

Fig. 9-26 shows the timing of an I/O device to memory transfer when the data transfer unit is set at 32 bits and the device port size at 32 bits (half speed bus mode). This figure illustrates when 2 wait cycles are set to the channel status register BSW field.

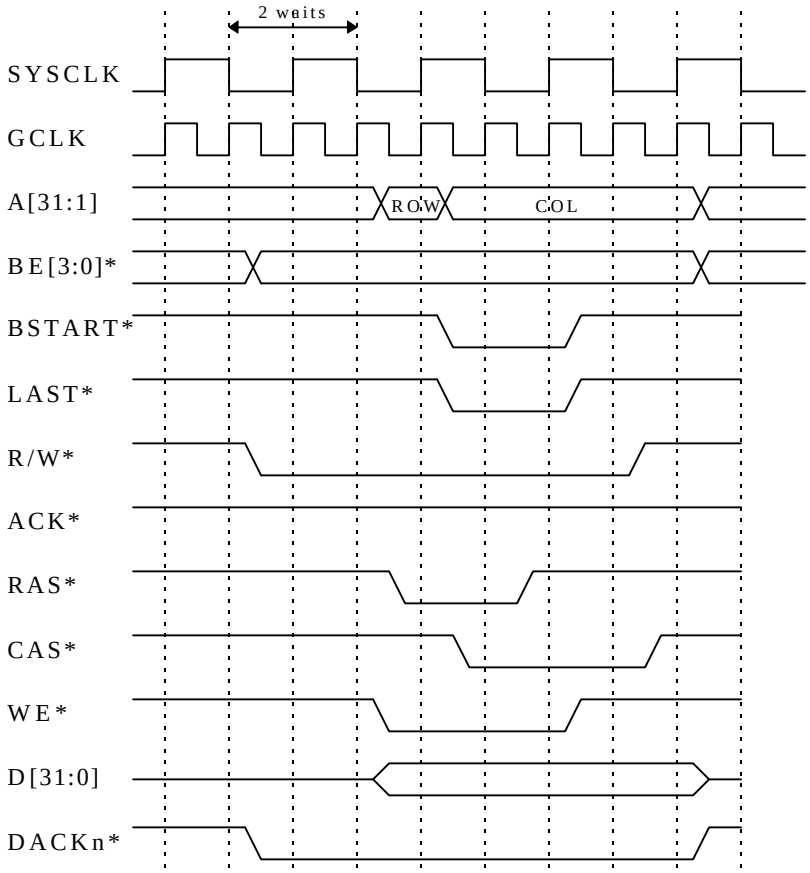


Fig. 10-26 Single Address Mode (I/O Device → Memory)

Burst mode

Fig. 9-27 shows the timing of a memory to I/O device burst transfer operation when the data transfer unit is set at 32 bits and the device port size at 8 bits:

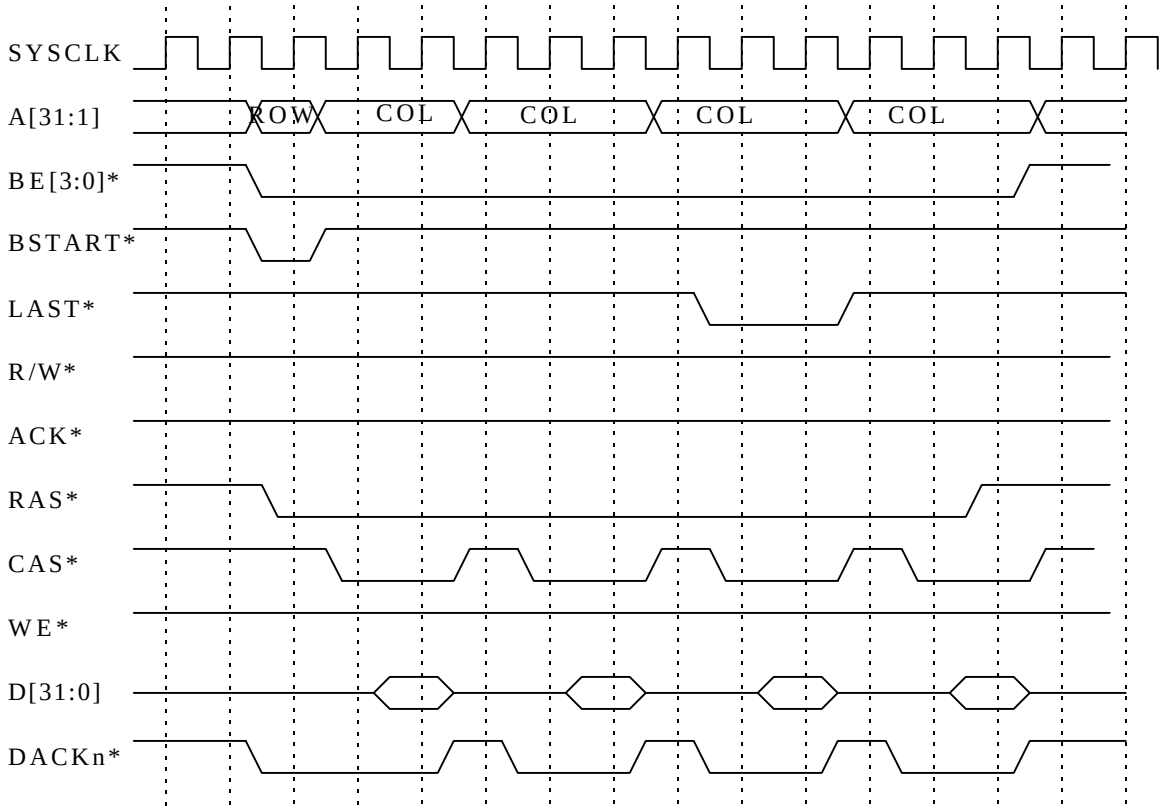


Fig. 10-27 Single Address Mode (Burst Mode)

10.5.3 Input of DONE* signal

When the DIEn bit in the CCRn is set to 1, the DMAC ends the data transfer if the DONE* signal is asserted during a data transfer.

The timing to acknowledge the DONE* signal is at the rising of the GCLK recognizing the last internal acknowledge signal in the transfer unit. In addition to the ACK* input from outside, acknowledge signals that are generated in the internal memory controllers (RAMC and ROMC) and that are generated in the EBIF at the SCS accesses are also subject to this.

The timing where the DONE* signal becomes valid is shown in Fig. 9-28 that presents an example of a memory to I/O device transfer (dual address mode) when the data transfer unit is set at 32 bits and the device port size at 16 bits.

In a normal completion by the DONE* signal, the transfer address to be output in the next bus operation is stored in the SARn and DARn; and the number of residual transfer bytes is stored in the BCRn.

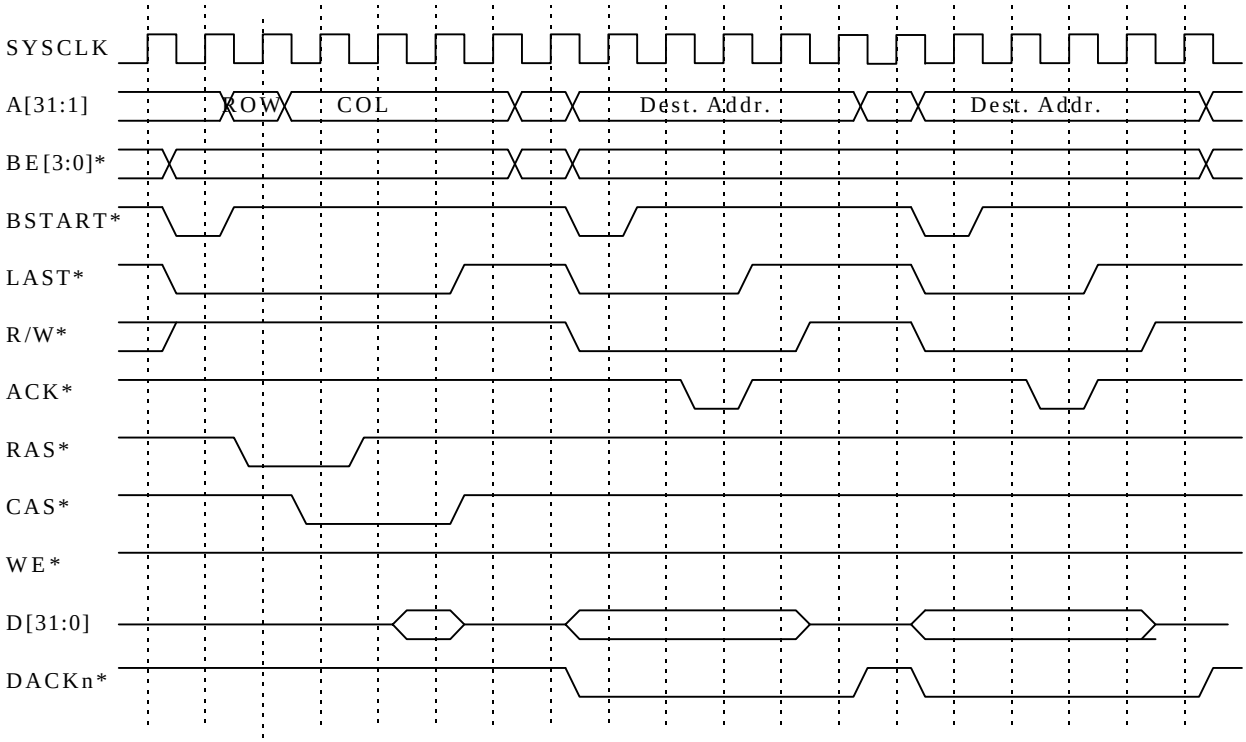


Fig.10-28 Transfer Completion by DONE* Signal

10.5.4 Output of DONE* signal

When the DOEn bit in the CCRn is set to 1, the DONE* signal is driven at the last data transfer in the single address mode or at the last I/O access of the last data transfer in the dual address mode. A timing DONE* is driven is the same as the DACK signal.

10.5.5 Note for DRAM refresh during DMA

A refresh operation for DRAM is sometime inserted at the start of data transfer or during a burst transfer. A timing that a refresh operation is inserted at the start of data transfer is shown in Fig. 9-29. It is an example of a data transfer of memory to I/O device in the single address mode.

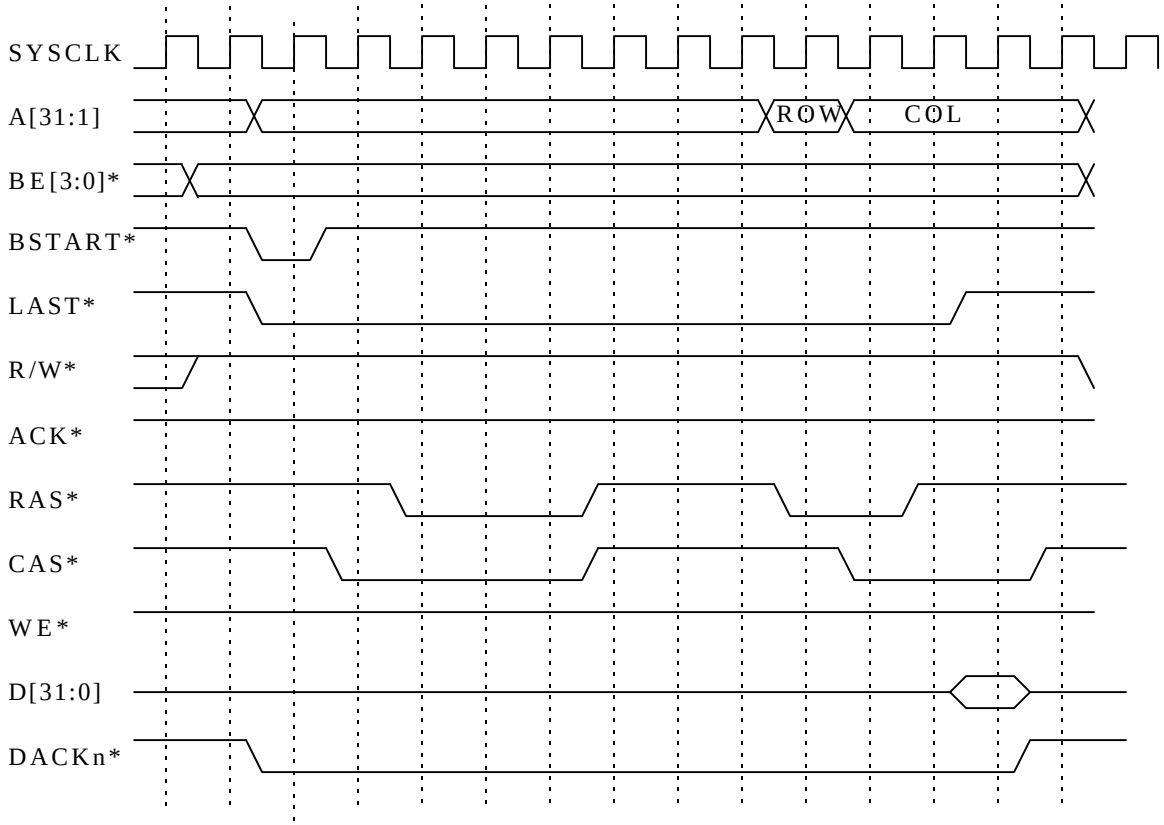


Fig. 10-29 Insertion of refresh operation

11 INTERRUPTS

The TX3904 has one non-maskable interrupt (NMI*) and eight external interrupt (INT[7:0]) pins. In addition to these, there are other interrupts that occur inside the TX3904. This chapter handles these interrupts.

11.1 Features

The TX3904 interrupts have the following characteristics:

(1) One non-maskable interrupt and 17 (nine internal and eight external) interrupts
(2) Informs the TX39 Processor Core of interrupt sources
(3) Seven interrupt levels (16 interrupts)
(4) Interrupt mask for each source and the interrupt mask by the level

(3) and (4) are the functions for 16 of the 17 interrupts.

11.2 Configuration

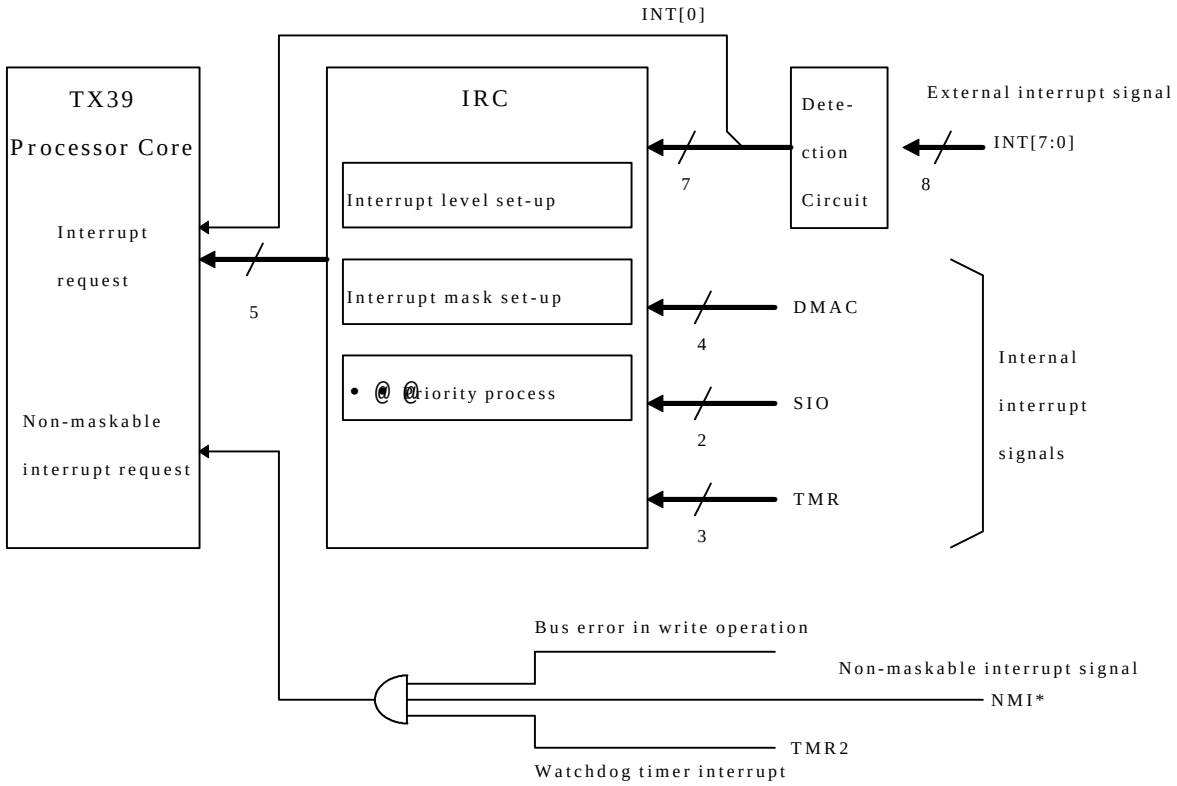


Fig. 11-1 TX3904 Interrupt Block Diagram

The “detection circuit” detects the active status of the external interrupt request signal (INT[7:0]). The active status can be selected from four kinds--the rising edge, the falling edge, the high level, and the low level, and it is set up in the CConR of the EBIF.

The “IRC” is an interrupt controller. It arbitrates the priority of the seven external interrupts and nine internal interrupts and informs the TX39 Processor Core of the interrupt with the highest priority.

11.3 Functions

11.3.1 Interrupt sources

The TX3904 has the following interrupt sources.

(Maskable) interrupts
Table 11-1 shows the interrupt sources of the TX3904.
The priority of Numbers 0-15 interrupts is arbitrated by the IRC and the interrupt with the highest priority is informed to the TX39 Processor Core. When one of the Numbers 0-15 interrupts is informed to the TX39 Processor Core, IP[4] in the TX39 Cause register is set and, as the same time, the interrupt number is indicated in the IP[3:0].
Interrupts by the INT[0] are not arbitrated for the priority unlike the Numbers 0-15 interrupts. Interrupts by the INT[0] are not masked in the TX3904 internal circuits. When an interrupt by the INT[0] is informed of, IP[5] is set.

Table 11-1 Interrupt Sources

Interrupt Number	IP[5:0]	Interrupt Request Source	Name of Interrupt
15	x11111	TMR2	Timer Interrupt
14	x11110	TMR1	Timer Interrupt
13	x11101	TMR0	Timer Interrupt
12	x11100	SIO1	SIO Interrupt
11	x11011	SIO0	SIO Interrupt
10	x11010	DMAC0	DMA Interrupt (Ch.0)
9	x11001	DMAC0	DMA Interrupt (Ch.1)
8	x11000	DMAC1	DMA Interrupt (Ch.2)
7	x10111	DMAC1	DMA Interrupt (Ch.3)
6	x10110	INT[7]	External Interrupt
5	x10101	INT[6]	External Interrupt
4	x10100	INT[5]	External Interrupt
3	x10011	INT[4]	External Interrupt
2	x10010	INT[3]	External Interrupt
1	x10001	INT[2]	External Interrupt
0	x10000	INT[1]	External Interrupt
	1xxxxx	INT[0]	External Interrupt

Non-maskable interrupts
The sources for non-maskable interrupts are divided into the interrupts by the NMI* signal, watchdog timer interrupts by the TMR2 and bus error in a write operation by the TX39 processor core.
The TX3904 acknowledges falling edges of the NMI* as a non-maskable interrupt request.
The TMR2 requests of the TX39 Coprocessor Core for a non-maskable interrupt if the TMR requests for a watchdog interrupt in the watchdog timer mode. However, if the WR bit of the CConR is set to 1, reset is implemented to the TX3904 at a watchdog interrupt time.

11.3.2 Interrupt detection

The active status of an interrupt by the INT[7:0] can be set up in the EIM field of the CConR. There are the high level, low level, rising edge, and falling edge in the active status. The TX3904 interrupt detection circuit notifies of an interrupt request to the TX39 Processor Core or the IRC when it has acknowledged the set up status.
Negating of an interrupt signal should be done in the interrupt handler after confirming the interrupt source. When the active status is set up at the high level or the low level, operate for the external circuit that is asserting the INTn signal to deassert it. When the active status is set up at the rising edge or the falling edge, negate the interrupt by writing in the value that supports the ElClr of the CConR.

11.3.3 Interrupt priority arbitration process

This section describes the Numbers 0-15 interrupts that are arbitrated by the IRC.

16 interrupt sources

Arbitrates interrupt requests of the nine internal sources and seven external sources.

Seven interrupt levels

Has seven levels of priority and can set up the priority for each interrupt source.
The interrupt level is set up in the interrupt level register. There is a 3-bit level set-up field for each interrupt source. The greater the value (the interrupt level) in this field is, the higher the priority is. When the value is 000 (the interrupt level 0), the interrupt by the source does not occur.

Interrupt mask

Other than setting-up 000 to the interrupt level, masking by the interrupt mask register is also possible. If the interrupt level is equal to or less than the value that is set up in the interrupt mask register, the interrupt shall be masked and shall not be informed to the TX39.

Interrupt source notice

When an interrupt occurs, the IRC informs the TX39 Processor Core of the of the interrupt number. The TX39 Processor Core can know the interrupt source by reading-out the value in the IP field of the cause register. When an interrupt from among the Numbers 0-15, the IP[4] is set to 1, and at the same time, the interrupt number is indicated in IP[3:0].
When multiple interrupts (of the same level) are occurring simultaneously, the source with the greater source number is informed. Interrupt requests by the INT[0] are informed to the TX39 without going through the IRC so that an interrupt by the INT[0] and another interrupt issue requests simultaneously. In such a case, implement a priority process by the software.
When once the IRC informs the TX39 of the interrupt source, the interrupt number to be informed does not change until the process for the interrupt completes (until the corresponding interrupt status bit becomes 1). When the process has completed, the interrupt number with the highest priority at that point shall be informed.

11.4 Registers

This section describes the registers of the IRC that arbitrates interrupts. For the set-up of the interrupt detection circuit, please refer to the explanations of the chip configuration register (CConR) in Chapter 5.

11.4.1 Register map

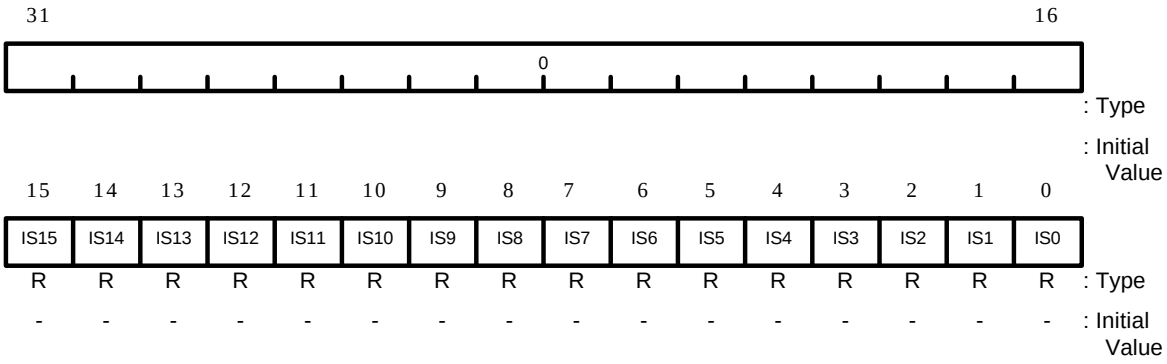
The IRC has the following registers:

Table 11-2 IRC Register Map

Address	Register Symbol	Register	Corresponding Interrupt Number
0xFFFF_C01C	ILR3	Interrupt Level Register 3	15-12
0xFFFF_C018	ILR2	Interrupt Level Register 2	11-8
0xFFFF_C014	ILR1	Interrupt Level Register 1	7-4
0xFFFF_C010	ILR0	Interrupt Level Register 0	3-0
0xFFFF_C004	IMR	Interrupt Mask Register	All (15-0)
0xFFFF_C000	ISR	Interrupt Status Register	All (15-0)

11.4.2 Interrupt status register (ISR)

Indicates the status of each interrupt source. There is an interrupt request when it is 0, and no interrupt request when it is 1. Please note that a suffix number of interrupt status indicates an interrupt number. For example, the interrupt status[0] corresponds with the INT[1] of external interrupt.

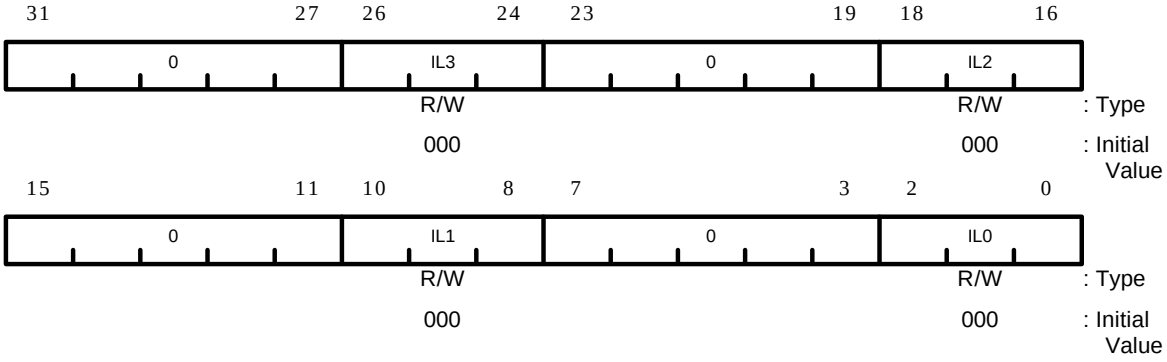


Bit	Mnemonic	Name of Field	Description
15	IS15	Interrupt status [15]	Interrupt Status [15] Indicates the status of the interrupt number 15 (TMR2) 1: There is no interrupt request. 0: There is an interrupt request.
14	IS14	Interrupt status [14]	Interrupt Status [14]
13	IS13	Interrupt status [13]	Interrupt Status [13]
12	IS12	Interrupt status [12]	Interrupt Status [12]
11	IS11	Interrupt status [11]	Interrupt Status [11]
10	IS10	Interrupt status [10]	Interrupt Status [10]
9	IS9	Interrupt status [9]	Interrupt Status [9]
8	IS8	Interrupt status [8]	Interrupt Status [8]
7	IS7	Interrupt status [7]	Interrupt Status [7]
6	IS6	Interrupt status [6]	Interrupt Status [6]
5	IS5	Interrupt status [5]	Interrupt Status [5]
4	IS4	Interrupt status [4]	Interrupt Status [4]
3	IS3	Interrupt status [3]	Interrupt Status [3]
2	IS2	Interrupt status [2]	Interrupt Status [2]
1	IS1	Interrupt status [1]	Interrupt Status [1]
0	IS0	Interrupt status [0]	Interrupt Status [0]

Fig. 11-2 Interrupt Status Register (ISR)

11.4.3 Interrupt level registers (ILR3-ILR0)

The interrupt level registers set up the level of the interrupt for each source. The total number of interrupt level registers is four.
The following diagram shows the example of the level register 0 (ILR0).

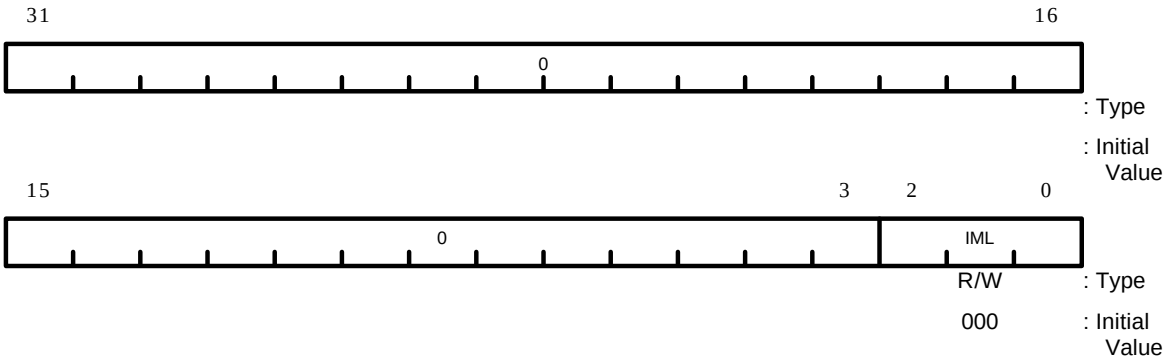


Bit	Mnemonic	Name of Field	Description
26:24	IL3	Interrupt level [3]	Interrupt Level [3] Sets up the interrupt level for the interrupt number 3. 111: Interrupt level 7 110: Interrupt level 6 101: Interrupt level 5 100: Interrupt level 4 011: Interrupt level 3 010: Interrupt level 2 001: Interrupt level 1 000: Interrupt inhibited
18:16	IL2	Interrupt level [2]	Interrupt Level [2]
10:8	IL1	Interrupt level [1]	Interrupt Level [1]
2:0	IL0	Interrupt level [0]	Interrupt Level [0]

Fig. 11-3 Interrupt Level Register 0 (ILR0)

11.4.4 Interrupt mask register (IMR)

Sets up the interrupt level to be masked.



Bit	Mnemonic	Name of Field	Description
2:0	IML	Interrupt mask level	Interrupt Mask Level Sets up the interrupt mask level. The interrupt sources of the levels that are equal to or less than the set-up values shall be masked.

Fig. 11-4 Interrupt Mask Register (IMR)

12 SERIAL PORTS (SIO)

12.1 Features

Non-synchronous serial interface (all-dual RS-232C protocol) controllers.

(1) Baud rate generator
(2) Modem flow control (CTS/RTS)
(3) FIFO (8 bits x eight steps)
(4) Multi-controller system support

12.2 Block Diagrams

Fig. 9-1 shows the SIO connection inside the TX3904 and Fig. 9-2 shows internal blocks of the SIO.

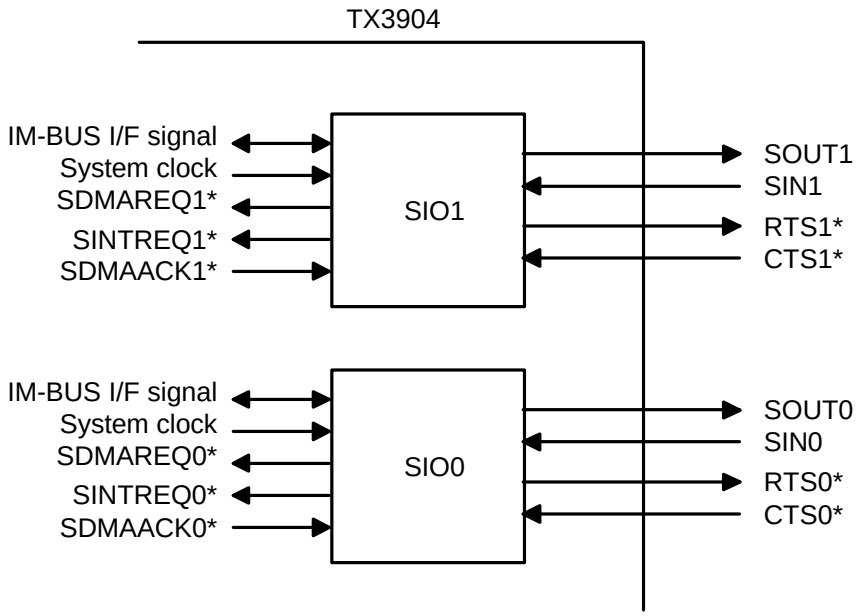


Fig. 12-1 SIO Connection inside TX3904

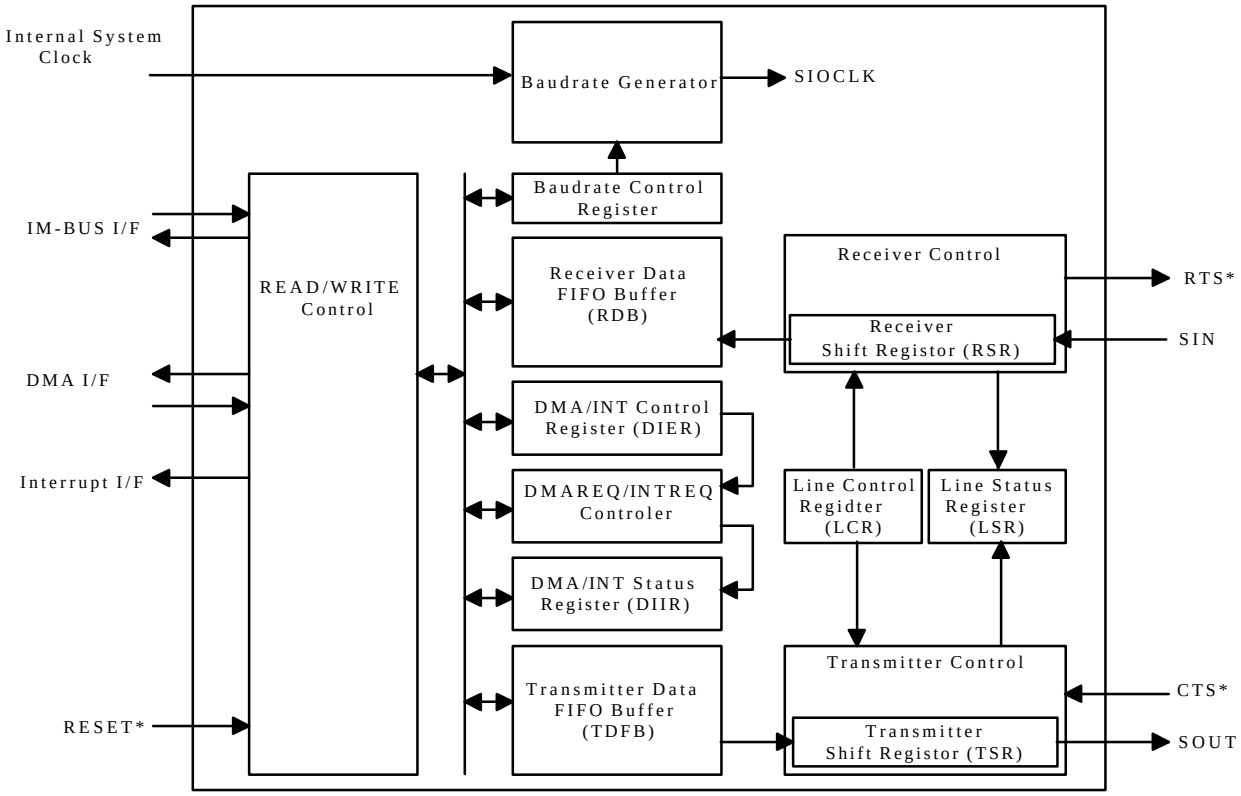


Fig. 12-2 SIO Internal Block Diagram

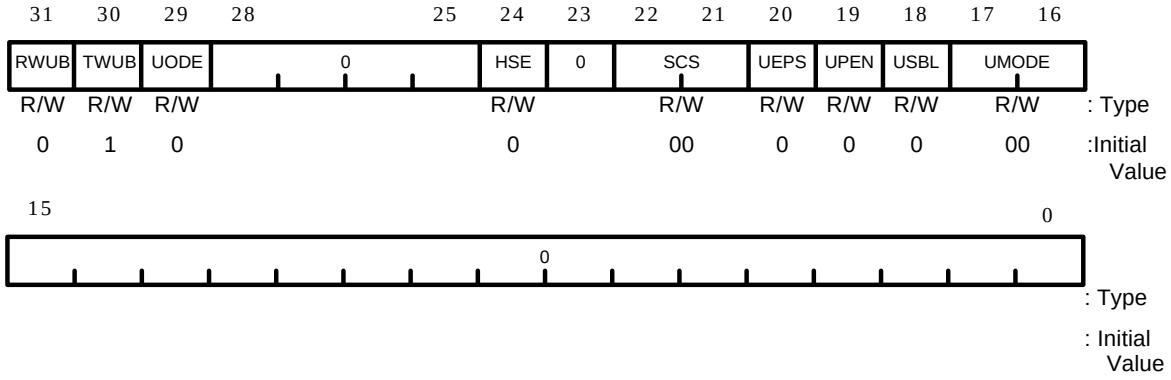
12.3 Registers

Table 12-1 Registers

Address	Register Symbol	Register Name
0xFFFF-F300	SLCR0	Line Control Register 0
0xFFFF-F304	SLSR0	Line Status Register 0
0xFFFF-F308	SDICR0	DMA/Interrupt Control Register 0
0xFFFF-F30C	SDISR0	DMA/Interrupt Status Register 0
0xFFFF-F310	SFCR0	FIFO Control Register 0
0xFFFF-F314	SBGR0	Baud Rate Control Register 0
0xFFFF-F320	TFIFO	Transfer FIFO Buffer 0
0xFFFF-F330	SFIFO	Receive FIFO Buffer 0
0xFFFF-F400	SLCR1	Line Control Register 1
0xFFFF-F404	SLSR1	Line Status Register 1
0xFFFF-F408	SDICR1	DMA/Interrupt Control Register 1
0xFFFF-F40C	SDISR1	DMA/Interrupt Status Register 1
0xFFFF-F410	SFCR1	FIFO Control Register 1
0xFFFF-F414	SBGR1	Baud Rate Control Register 1
0xFFFF-F420	TFIFO	Transfer FIFO Buffer 1
0xFFFF-F430	SFIFO	Receive FIFO Buffer 1

12.3.1 Line control register (SLCRn)

The line control register designates the format of the non-synchronous sending/receiving data.



Bit	Mnemonic	Name of Field	Description
31	RWUB	Receive wake up bit	Wake Up Bit for Receive 0: Set this bit to 0 when itself is a slave controller in the multi-controller system mode and when itself is selected. When the RWUB is 0, the slave controller receives data from the master controller. 1: Set this bit to 1 when itself is a slave controller in the multi-controller system mode; namely, when conducting an address (ID) frame receiving from the master controller. When the WUB of the receiving data frame is 1 (the address frame), it is received to make an interrupt to the host. When the WUB is 0 (the data frame), the received data are read and thrown away.
30	TWUB	Transmit wake up bit	Wake Up Bit for Transmit Designates the wake up bit to attach to the transfer address (ID)/data when itself is the master controller in the multi-controller system mode. 0: Data frame transfer 1: Address (ID) frame transfer (default)
29	UODE	Open drain enable	SOUT Open Drain Enable In the multi-controller system mode, the slave controller must make the SOUT an open drain. 0: SOUT open drain disable 1: SOUT open drain enable

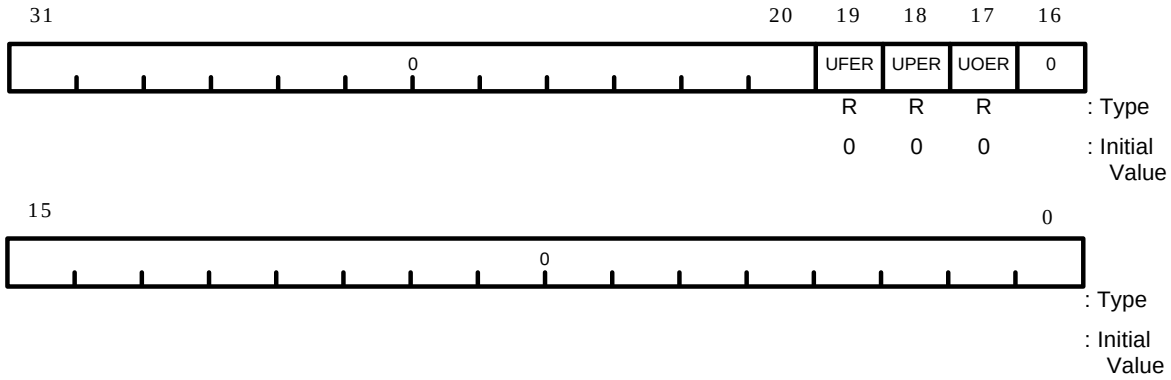
Fig. 12-3 Line Control Register (1/2)

Bit	Mnemonic	Name of Field	Description
24	HSE	Hand shake enable	Hand Shake Enable Hand shake function control using the CTS. 0: Disable (Sending can always be done.) 1: Enable
22:21	SCS	Clock select	SIO Clock Select Selects the serial transfer clock. The serial transfer clock is a clock with the frequency 16 times of the baud rate (bps). 00: T0 (Internal system clock) 01: Baud rate generator 1*: Set-up impossible
20	UEPS	Even parity select	UART Even Parity Select Makes the parity mode to the even number (Even) parity. 0: Odd parity 1: Even parity UPEN UEPS Description 1 0 Odd Parity 1 1 Even Parity 0 * Parity Disable
19	UPEN	Parity check enable	UART Parity Enable Must be 0 in the multi-control system mode (UMODE=10, 11). 0: Disable parity check 1: Enable parity check
18	USBL	Stop bit length	UART Stop Bit Length Designates the Stop bit length. 0: 1-bit 1: 2-bit
17:16	UMODE	Mode	UART Mode Sets up the SIO mode. A data length in the multi-controller system mode includes a wake up bit. 00: 8-bit data length 01: 7-bit data length 10: Multi-controller 9-bit data length 11: Multi-controller 8-bit data length

Fig. 12-4 Line Control Register (2/2)

12.3.2 Line status register (SLSRn)

The status information about the serial data sending/receiving is given to the line status register.

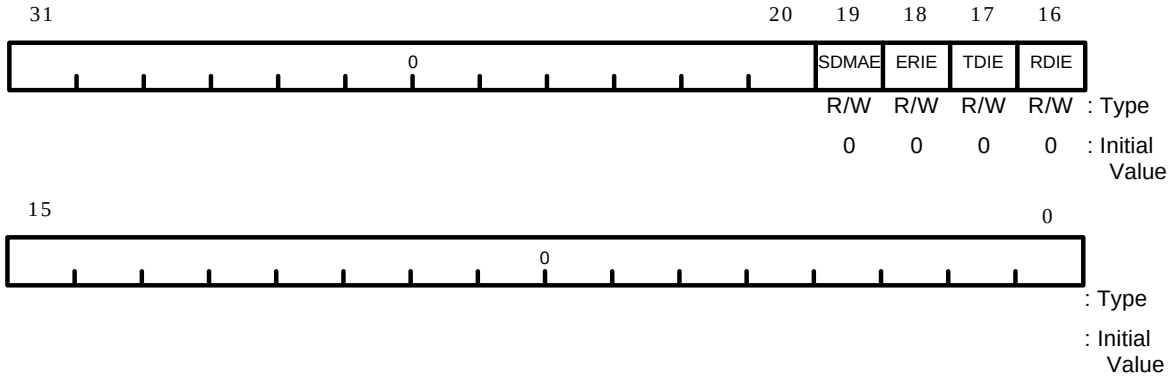


Bit	Mnemonic	Name of Field	Description
19	UFER	Frame error	UART Frame Error Shall be set when the Stop bit is not detected. It shall be cleared by writing "0." The writing of "1" is invalid. 1: Frame error occurred.
18	UPER	Parity error	UART Parity Error Shall be set when the UPEN of the line control register is 1 and when an error is detected in the receiving data in the parity bit that was added to the receiving data. It shall be cleared by writing "0." The writing of "1" is invalid. 1: Parity error occurred.
17	UOER	Overrun error	UART Overrun Error Shall be set when the overrun error occurs. It shall be cleared by writing "0." The writing of "1" is invalid. 1: Over run error occurred.

Fig. 12-5 Line Status Register

12.3.3 DMA/Interrupt control register (SDICRn)

Designates to conduct the host interface either by the DMA or by the interrupt.



Bit	Mnemonic	Name of Field	Description
19	SDMAE	DMA request enable	SIO DMA Request Enable The I/F with the host is conducted by the DMA. 0: Interrupt mode 1: DMA mode
18	ERIE	Error interrupt enable	Error Interrupt Enable Error interrupt enable. In the error, there are a frame error, a parity error, and an over-run error. The kind of the error is verified in the line status register. 0: Disable 1: Enable
17	TDIE	Transmission data DMA /Interrupt enable	Transmit DMA/Interrupt Enable **When in interrupt mode (SDMAE = 0)** Data write request interrupt enable for transmit FIFO. Asserts the SINTREQ* signal when the transmit FIFO is empty. 0: Disable 1: Enable **When in DMA mode (SDMAE = 1)** Write DMA request enable. Generates a write request when the DMA request trigger level count in the transmit FIFO has a space. 0: Asserts the SINTREQ* signal. 1: Asserts the SDMAREQ* signal.

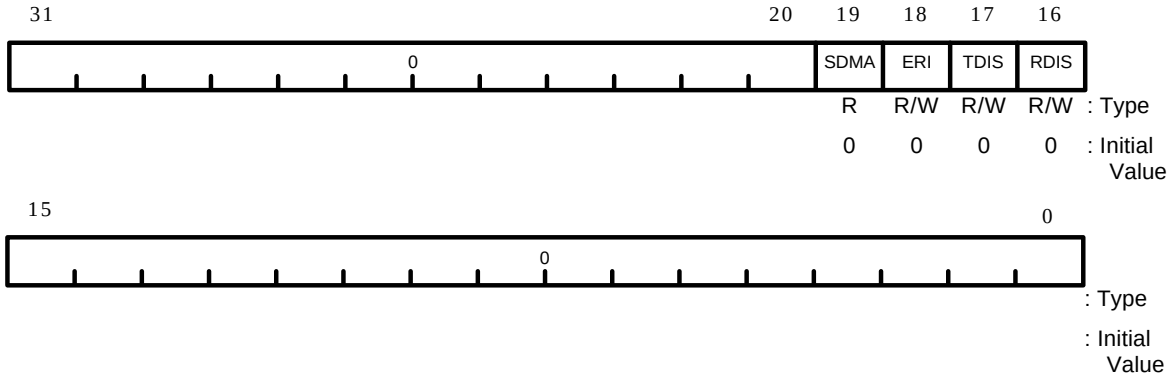
Fig. 12-6 DMA/Interrupt Control Register (1/2)

16	RDIE	Receive data DMA /Interrupt enable	Receive DMA/Interrupt Enable **When in interrupt mode (SDMAE = 0)** Valid data read request interrupt enable for receive FIFO. Asserts the SINTREQ* signal when there is valid data in the receive FIFO. 0: Disable 1: Enable **When in DMA mode (SDMAE = 1)** Read DMA request enable. Generates a read request when the number of valid data in the receive FIFO reaches the DMA request trigger level. 0: Asserts the SINTREQ* signal. 1: Asserts the SDMAREQ* signal.

Fig. 12-7 DMA/Interrupt Control Register (2/2)

12.3.4 DMA/Interrupt status register (SDISRn)

Indicates the status information of the DMA/interrupt.



Bit	Mnemonic	Name of Field	Description
19	SDMA	DMA request	DMA Request Indicates that the DMA request is being asserted.
18	ERI	Error interrupt	Error Interrupt Indicates that there is an error interrupt. This bit shall be cleared by writing "0." The writing of "1" is invalid. 1: ERI occurs
17	TDIS	Transmission data empty	Transmit DMA/Interrupt Status **When in the Interrupt mode (SDMAE=0)** Set to "1" when there is a space in the transmit FIFO. This bit is cleared when a 0 is written to it. Also, when the TDIE bit of the DMA/Interrupt control register is set to "1," the SINTREQ* signal is negated simultaneous with the clearing of this bit. **When in the DMA mode (SDMAE=1)** Set to "1" when the space in the transmit FIFO reaches the DMA request trigger level. When the TDIE bit of the DMA/Interrupt control register is set to "1," TDIS is cleared and the SDMAREQ* signal is negated when the SDMAACK* signal is asserted. When TDIE is set to "0," the SINTREQ* signal is negated when a "0" is written to TDIS.

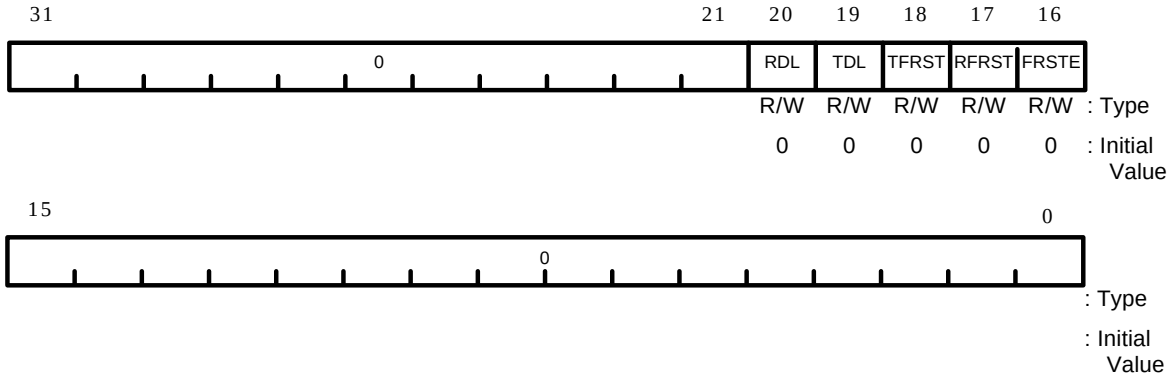
Fig. 12-8 DMA/Interrupt Status Register (1/2)

16	RDIS	Reception data full	Receive DMA/Interrupt Status **When in the Interrupt mode (SDMAE=0)** Set to “1” when there is valid data in the receive FIFO. This bit is cleared when a “0” is written to it. Also, when the RDIE bit of the DMA/Interrupt control register is set to “1,” the SINTREQ* signal is negated simultaneous with the clearing of this bit. **When in the DMA mode (SDMAE=1)** Set to “1” when the number of valid data in the receive FIFO reaches the DMA request trigger level. When the RDIE bit of the DMA/Interrupt control register is set to “1,” RDIS is cleared and the SDMAREQ* signal is negated when the SDMAACK* signal is asserted. When RDIE is set to “0,” the SINTREQ* signal is negated when a “0” is written to RDIS.

Fig. 12-9 DMA/Interrupt Status Register (2/2)

12.3.5 FIFO control register (SFCRn)

Sets up the control of the transfer/receive FIFO buffers.

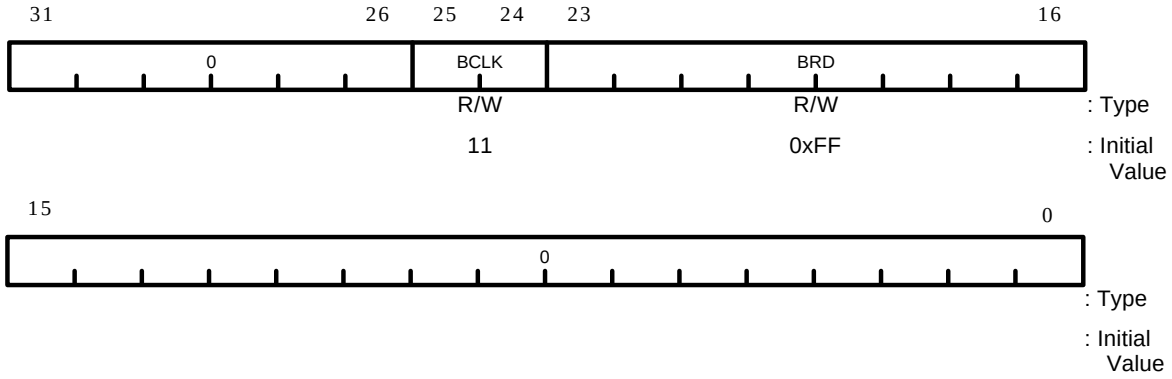


Bit	Mnemonic	Name of Field	Description
20	RDL	Receive FIFO DMA request trigger level	Receive FIFO DMA Request Trigger Level Sets up the DMA transfer level from the receive FIFO buffer. 0: 4 bytes 1: 8 bytes
19	TDL	Transfer FIFO DMA request trigger level	Transfer FIFO DMA Request Trigger Level Sets up the DMA transfer level to the transfer FIFO buffer. 0: 4 bytes 1: 8 bytes
18	TFRST	Transmit FIFO reset	Transmit FIFO Reset Resets the transfer FIFO buffer. It is valid when FRSTE is 1. 0: In operation 1: Reset transfer FIFO
17	RFRST	Receive FIFO reset	Receive FIFO Reset Resets the receive FIFO buffer. It is valid when FRSTE is 1. 0: In operation 1: Reset receive FIFO
16	FRSTE	FIFO reset enable	FIFO RESET Enable Reset enable of the FIFO buffer. Resets the transfer/receive FIFO by combining with TFRST and RFRST. 0: In operation 1: Reset enable

Fig. 12-10 FIFO Control Register

12.3.6 Baudrate control register (SBGRn)

Conducts the selection of clocks and the set-up of divide values to be given to the baudrate generator.

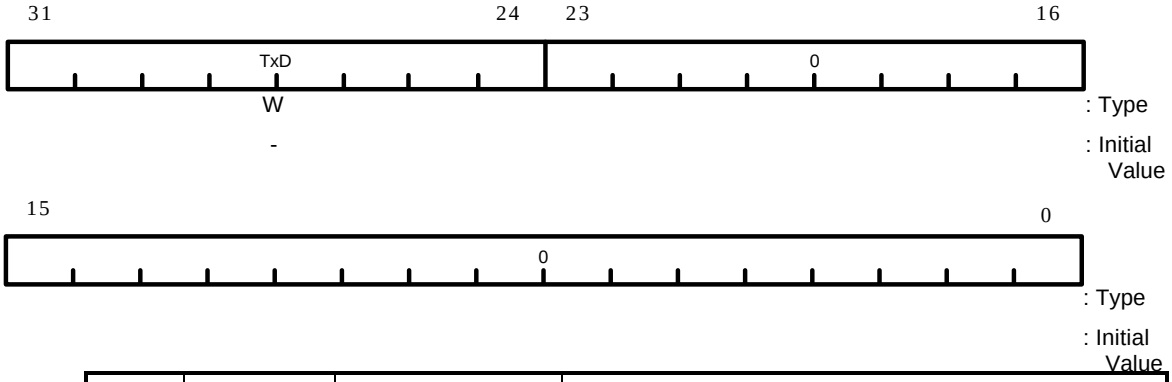


Bit	Mnemonic	Name of Field	Description
25:24	BCLK	Baudrate generator clock	Baud rate Generator Clock Specifies the Baud rate generator input clock. 00: Selects prescalar output T0 (fc/4). 01: Selects prescalar output T2 (fc/16). 10: Selects prescalar output T4 (fc/64). 11: Selects prescalar output T6 (fc/256). The fc corresponds with a processor clock when the RF bit in the Config register is set 00.
23:16	BRD	Baudrate divide value	Baudrate Divide Value Sets up the divider BRG of the baudrate generator. Shall be designated with binary values.

Fig. 12-11 Baudrate Control Register

12.3.7 Transmit FIFO buffer (TFIFOn)

Writes transmission data.

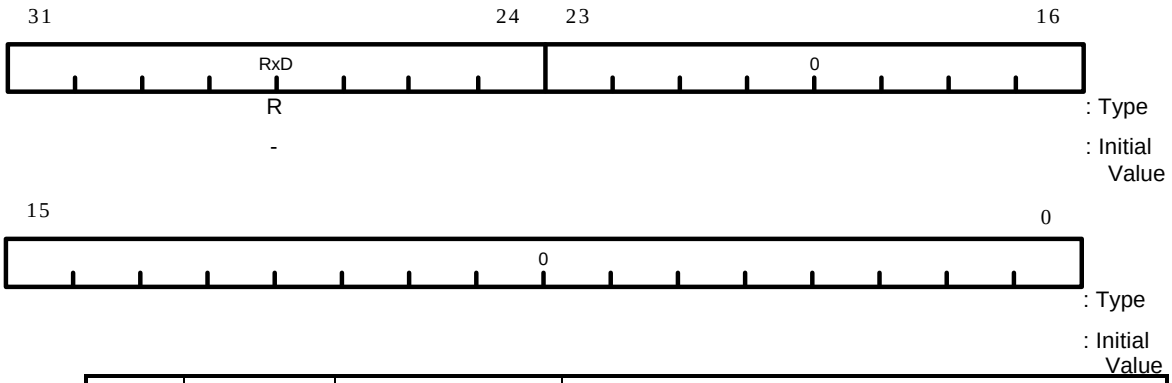


Bit	Mnemonic	Field name	Explanation
31:24	TxD	Transmission data	Transmit Data Writes the transmission data.

Fig. 12-12 Transmit FIFO buffer

12.3.8 Receive FIFO buffer (SFIFOn)

Reads reception data.



Bit	Mnemonic	Field name	Explanation
31:24	RxD	Reception data	Receive Data Reads the reception data.

Fig. 0-13 Receive FIFO buffer

12.4 Operations

12.4.1 Overview

The TX3904 has two channels of SIO's. The SIO converts serial data from outside that are to be input to the SIN to parallel data by the shift register. The converted parallel data shall be stored in the receive FIFO buffer. The parallel data being stored in the FIFO buffer are taken out by a DMA transfer or an interrupt.

At the time of sending, the parallel data shall be written into the transfer FIFO buffer from the memory or the CPU by a DMA transfer or an interrupt and are loaded to the shift register in due order. The parallel data are converted to serial data by the shift register to be output from the SOUT.

The clock that is to be the basis of the sending and receiving is made by the baudrate generator. The baudrate generator generates clocks with a frequency 16 times of the sending baudrate. The frequency is set by the register.

For the sending and receiving flow control, one can be selected from the hardware flow control (RTS/CTS) and the software flow control (XON/XOFF).

12.4.2 Data format

The TX3904 built-in SIO can use the following data format:

Data length: 9, 8, and 7 bits (9-bit data support the multi-controller system)
Stop bit: 1 and 2 bits
Parity bit: There is parity bit/There is no parity bit (There is no parity bit when the data length is 9 bits)
Parity method: Even number/Odd number
The start bit is fixed to 1 bit.

The following diagrams show the data frame configuration.

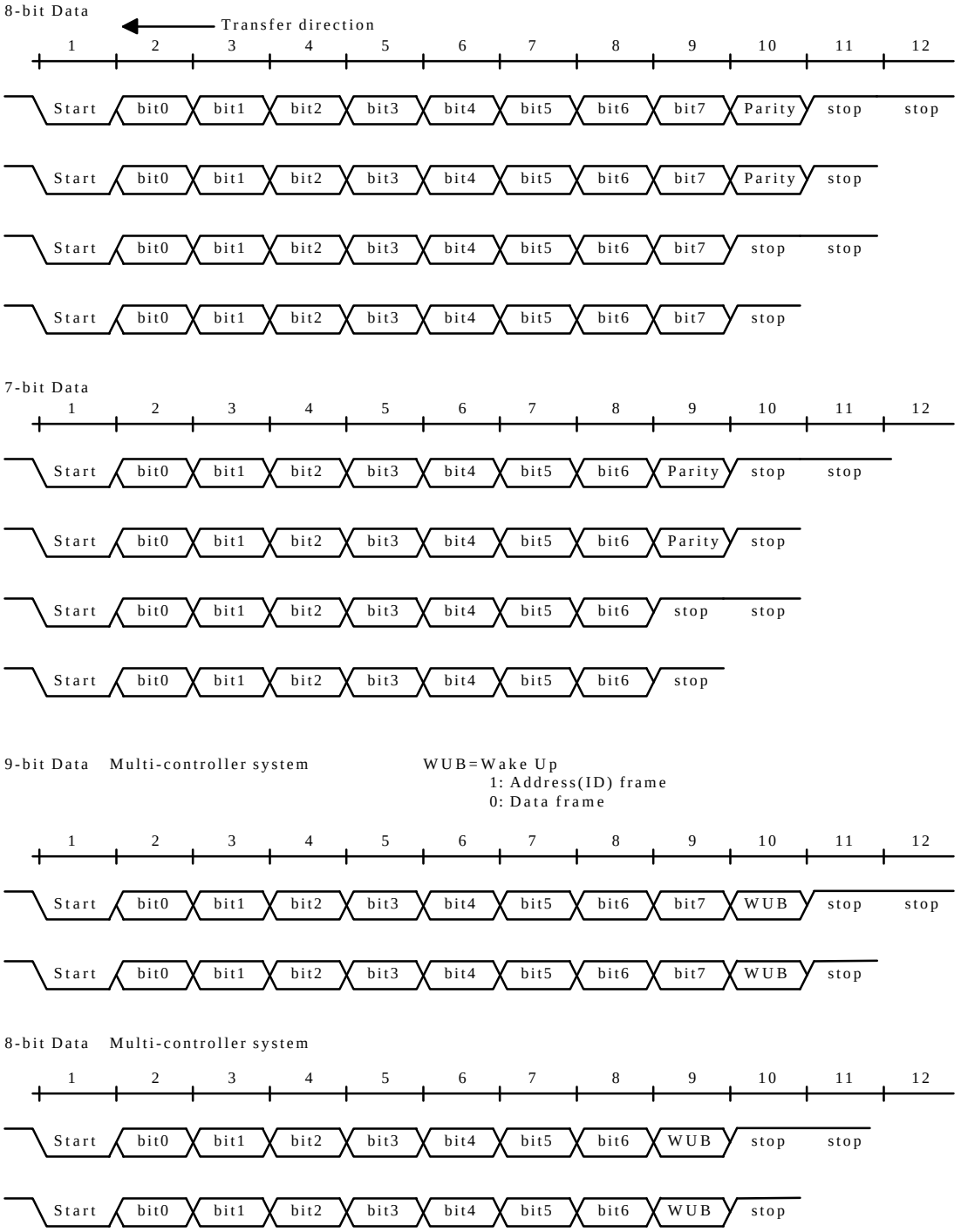


Fig. 12-14 Data Frame Configuration

12.4.3 Serial clock generator

Select the baud rate generator circuit output or internal system clock IMCLK. Then, make it the SIOCLK (transmit/receive clock which determines the serial I/O transfer speed).

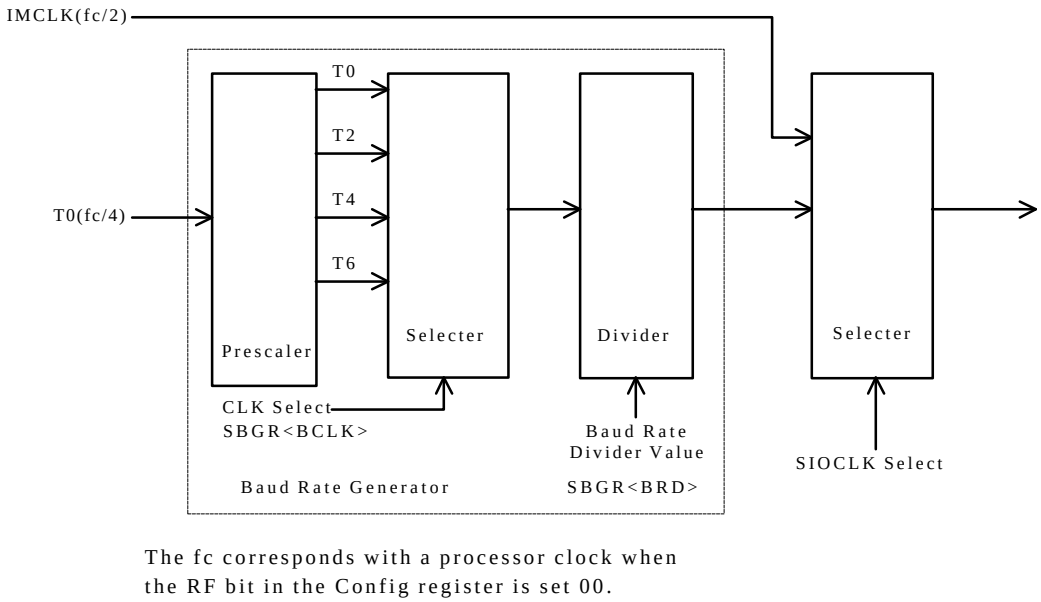


Fig. 12-15 Baudrate Generator and SIOCLK Generator

12.4.4 Baudrate generator

The baudrate generator is a circuit to generate the sending/receiving clock that rules the transfer speed of the serial I/O.

$$\text{Baud rate} = \frac{\text{The input clock of the baud rate generator}}{\text{Divider of the baud rate generator}} \div 16$$

The input clock to the baudrate generator shall be input by selecting from T0, T2, T4, and T6 of the pre-scaler.
The selected baud rate generator input clock is divided by the value that is set up in the baud rate control register. (1, 2, 3,.....255 dividing)
Table 12-2 shows the relationship between the dividing latch's value and the baudrate; and Table 12-3 shows an example of a typical divider set-up for baudrate.

Table 12-2 Value of Dividing Latch and Baudrate (in: kbps)

	Baudrate Generator Input Cloc				
fc[MHz]	BRG	T0	T2	T4	T6
	Divider	fc/4	fc/16	fc/64	fc/256
50.00	5	156.25	39.06	9.77	2.44
50.00	10	78.13	19.53	4.88	1.22
50.00	41	19.05	4.76	1.19	0.30
50.00	55	14.20	3.55	0.89	0.22
49.15	5	153.60	38.40	9.60	2.40
49.15	10	76.80	19.20	4.80	1.20
49.15	20	38.40	9.60	2.40	0.60
49.15	40	19.20	4.80	1.20	0.30
49.15	55	13.96	3.49	0.87	0.22
49.15	225	3.41	0.85	0.21	0.05
25	5	78.13	19.53	4.88	1.22
25	10	39.06	9.77	2.44	0.61
25	20	19.53	4.88	1.22	0.31
25	27	14.47	3.62	0.90	0.23
25	40	9.77	2.44	0.61	0.15
25	80	4.88	1.22	0.31	0.08
25	110	3.55	0.89	0.22	0.06

Table 12-3 Divider Set-Up Example

			BRG Divider		
fc[MHz]	BPS	T0	T2	T4	T6
		fc/4	fc/16	fc/64	fc/256
50.00	0.11				111
50.00	0.15				81
50.00	0.3				41
50.00	0.6			81	20
50.00	1.2		163	41	10
50.00	2.4		81	20	5
50.00	4.8	163	41	10	3
50.00	9.6	81	20	5	1
50.00	14.4	54	14	3	
50.00	19.6	40	10	2	
50.00	28.8	27	7	2	
50.00	38.4	20	5	1	
50.00	76.8	10	3		

12.4.5 Receiver Controller

For the receive control, the start bit is detected by the majority logic to start a data receiving operation. Data receiving is conducted also by the majority logic. The start bit and the receiving of the data 1 bit are sampled at SIOCLK 16 clocks. The majority logic is implemented based on the three results of the 7th, 8th, and 9th clocks. (The data that are the same for two or more of the three results are chosen.)
When a parity bit is attached to a 7-bit length, the parity is shifted in to the 8th bit of the receive shift register to be stored in the highest-order bit of the receive read buffer.
When a parity bit is attached to a 8-bit length, the parity is stored in the RWUB of the line control register.

12.4.6 Receiver shift register

This is a 8-bit shift register. It shifts in such that the first data bit that was received is bit0 (the lowest-order bit).

12.4.7 Receiver read buffer

This is a buffer placed between the receive shift register and the receive FIFO buffer. After receiving a data frame, the data are stored in this buffer and a parity check is conducted.

12.4.8 Transmitter controller

When the transfer shift register has taken out all of the previous data, one data frame is taken from the transfer FIFO buffer to the transfer shift register. 1 bit is sent out every SIOCLK 16 clocks.

12.4.9 Transmitter shift register

This is a 8-bit shift register. It shifts out from the lowest order-bit in due order to transfer data.

12.4.10 Host I/F

The sending data are written into the transfer FIFO buffer by the interrupt by the sending data empty or the DMA transfer. The writing-in by the interrupt is conducted by the byte by the CPU. Every time the read pointer of the transfer FIFO buffer changes, an interrupt is generated to request the CPU for sending data. In the writing-in to the transfer FIFO buffer by the DMA transfer, when sending data of 4 bytes or 8 bytes are sent to the transfer shift register (when the read pointer (0-7) becomes 4 and 0), a DMA request is generated and the data for sending are taken in from the memory.
The reading-out from the receive FIFO buffer is conducted by the interrupt process or the DMA transfer. Every time 1-byte receiving data are written in to the receive FIFO buffer (i.e., every time the write pointer of the receive FIFO buffer moves), an interrupt is generated and the CPU reads out the data.
In the interrupt mode, the FIFO functions as a simple one stage buffer and makes up a double-buffer with a read buffer.

In the reading-out by the DMA transfer, when receiving data of 4 bytes or 8 bytes are sent to the receive FIFO buffer (when the write pointer (0-7) becomes 4 and 0), a DMA request is generated and the receiving data are written in to the memory.

Specify the interrupt mode, DMA mode and DMA direction using the three SDMAE, TDIE and RDIE bits. Transmit and Receive cannot be simultaneously set to DMA when in the DMA mode.
The DMA controller settings must be dual address mode, external transfer request, and low-level mode.

Table 12-4 Register bit setting and transmit/receive operation

SDMAE	TDIE	RDIE	Transmit	Receive
0	0	0	TDIS polling	RDIS polling
0	0	1	TDIS polling	Interrupt (RDIS = 1)
0	1	0	Interrupt (TDIS=1)	RDIS polling
0	1	1	Interrupt (TDIS=1)	Interrupt (RDIS=1)
1	0	0	Setting prohibited	
1	0	1	Interrupt (TDIS=1)	Read DMA (SDMA,RDIS=1)
1	1	0	Write DMA (SDMA,TDIS=1)	Interrupt (RDIS=1)
1	1	1	Setting prohibited	

12.4.11 Hand shake function

By handshaking the CTS-RTS, the data transfer by the frame is made certain. This function is valid when the HSE of the line mode register is 1.
When the CTS* pin becomes high level in transfer operation, the transfer is halted until the CTS* pin becomes low after the completion of the currently conducted data transfer. At this time also, however, next data are requested of the CPU by an interrupt or a DMA request.
When the receiving of one-frame data is completed in a receive operation, the RTS* pin is set to high to request of the transferring side for a temporary halt of transfer. By changing the RTS* pin to low when the receiving preparation is done, the transfer resumes.

12.4.12 Parity control

In a transfer operation, the parity control circuit automatically generates a parity by the data that were written into the transfer shift register; and the parity is stored to the bit7 (the highest-order) of the transfer shift register when the data length is 7 bits and to the TWUB of the line control register when the data length is 8 bits to transfer it.
In a receive operation, a parity check is conducted when data have been written from the receive shift register to the receive read buffer. A comparison is made with the parity bit stored in the bit7 (the highest-order) of the read buffer when the data length is 7 bits and with the parity bit stored in the RWUB of the line control register when the data length is 8 bits; and when they do not match, a parity error occurs.

12.4.13 Error flag

Overrun error
When there are valid data in the receive read buffer, an overrun error occurs if all the bits of the next data are received by the received shift register. Also when the receive FIFO buffer has become full, it occurs. A flag is hoisted in the UOER of the line status register. When the ERIE of the interrupt control register is 1, an error interrupt (SINTREQ*) shall be asserted.
Parity error
When a parity error has occurred, a flag is hoisted in the UPER of the line status register. When the ERIE of the interrupt control register is 1, an error interrupt (SINTREQ*) shall be asserted.
Framing error
When a 0 is detected in the stop bit of the receive data (majority logic in the sampling at the 7th, 8th, and 9th clocks of the SIOCLK), a framing error occurs. A flag is hoisted in the UFER of the line status register. When the ERIE of the interrupt control register is 1, an error interrupt (SINTREQ*) shall be asserted.

12.4.14 Multi-controller system

When the register UMODE is 10 or 11, it is the multi-controller system mode. The multi-controller system sends data to the selected slave controller after the master controller sends an address (ID) to a slave controller to select the slave controller. Slave controllers that were not selected ignore the data. The sending of the address (ID) is conducted by setting the WUB of the data frame to 1. The data sending is conducted by setting the WUB to 0. The comparison of the ID's are conducted by the software. (Comparison between the ID that the slave controller received and the ID of the slave controller that the software acknowledges.)

Protocol
(1) The master and slave controllers change to the multi-controller system mode by setting the UMODE of the line control register to 10 or 11.

(2) The slave controller prepares the address (ID) frame for receiving from the master controller by setting the RWUB of the line control register to 1.
(3) The master controller sends the address (ID) (8-bit or 7-bit) of the slave controller by setting the WUB of the sending frame to 1 (the line control register TWUB = 1).
(4) The slave controller generates an interrupt to the host when the RWUB is 1 and when the WUB of the receiving data frame is 1 (the receiving data are the address frame). The host compares the data received (the address--ID) and its own address (ID); and when they match, the host clears the RWUB to 0.
(5) The master controller sends the data frame to the designated slave controller. At this time, the WUB of the data frame is set to 0 (the line control register TWUB = 0).
(6) The slave (the selected slave controller) whose RWUB is 0 receives the data. The slave (not-selected slave controller) whose RWUB remains 1 does not generate an interrupt because the WUB of the receiving data frame is 0. Therefore, it ignores the receiving data.
(7) The data sending from a slave controller is possible only to the master controller.

The following diagram shows a configuration example of the multi-controller system.

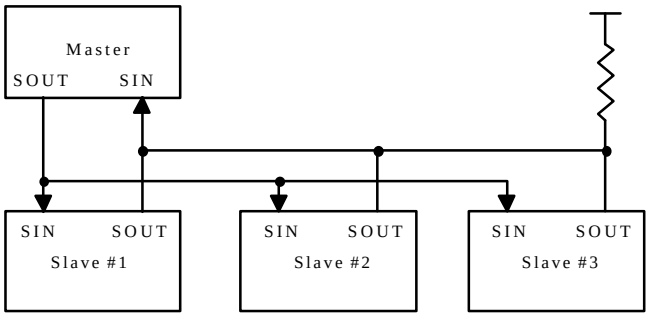


Fig. 12-16 Configuration Example of the Multi-Controller System

The output (SOUT) of the slave must be an open drain. When the UODE of the line control register is set to 1, the SOUT becomes the open drain. When the UODE is set to 0, the SOUT becomes the CMOS output.

12.5 Timing Explanations

12.5.1 Operation at the time of receiving (7 and 8 bit data length)

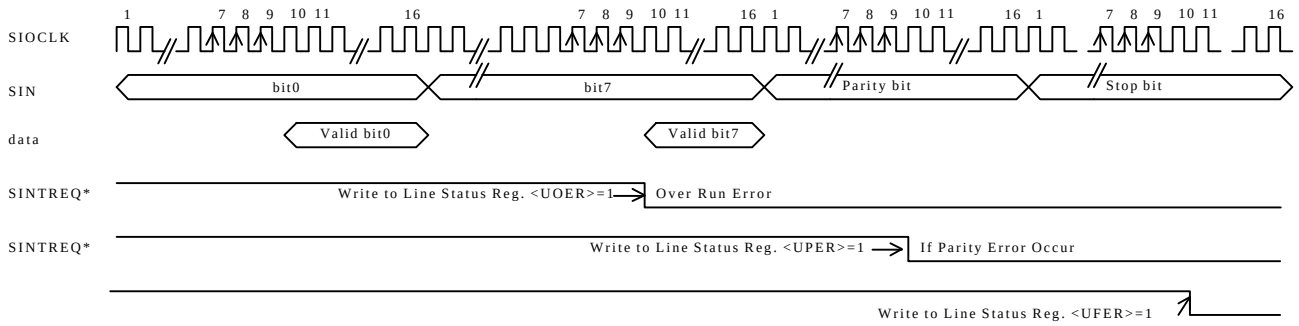


Fig. 12-17 Timing of Receiving (1)

12.5.2 Timing of SDMAREQ*/SMAACK* at the time of DMA I/F (at DMA level 4)

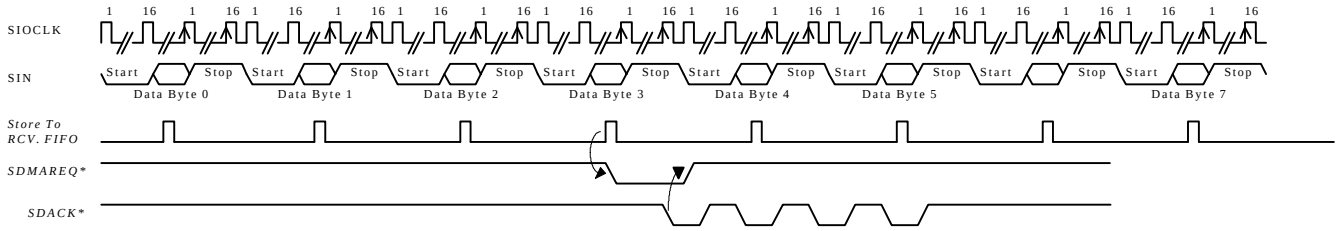


Fig. 12-18 SDMAREQ* and SDMAACK*

After having acknowledged the first SDMAACK* assertion, SDMAREQ* is deasserted. The SDMAACK* is sampled by the IMCLK.

12.5.3 Operation at the time of receiving (8 and 9 bit length multi-controller system; RWUB = 1 when standing by for ID receiving)

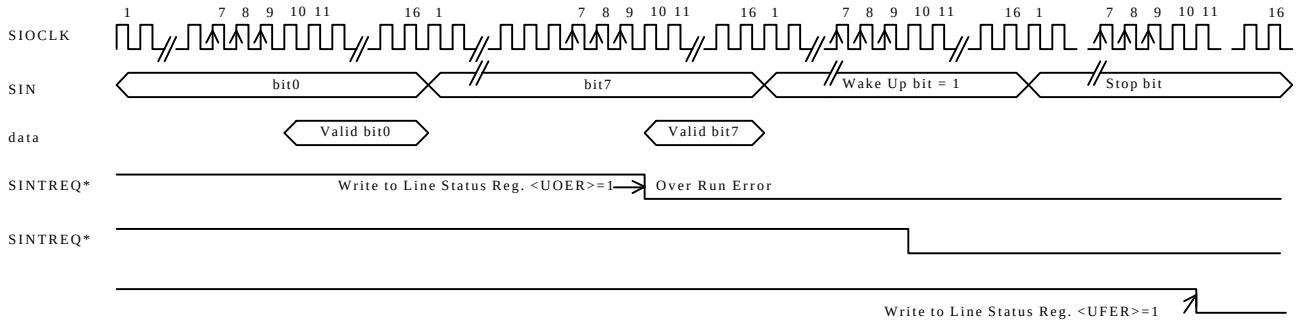


Fig. 12-19 Timing of Receiving (2)

12.5.4 Operation at the time of receiving (8 and 9 bit length multi-controller system; RWUB = 0 when standing by for data receiving)

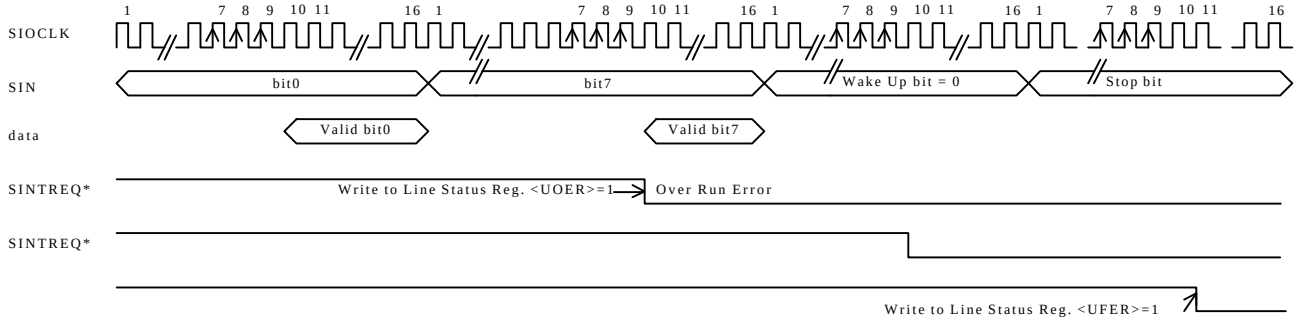


Fig. 12-20 Timing of Receiving (3)

12.5.5 Operation at the time of receiving (8 and 9 bit length multi-controller system;

RWUB = 1 when receiving data and discarding)

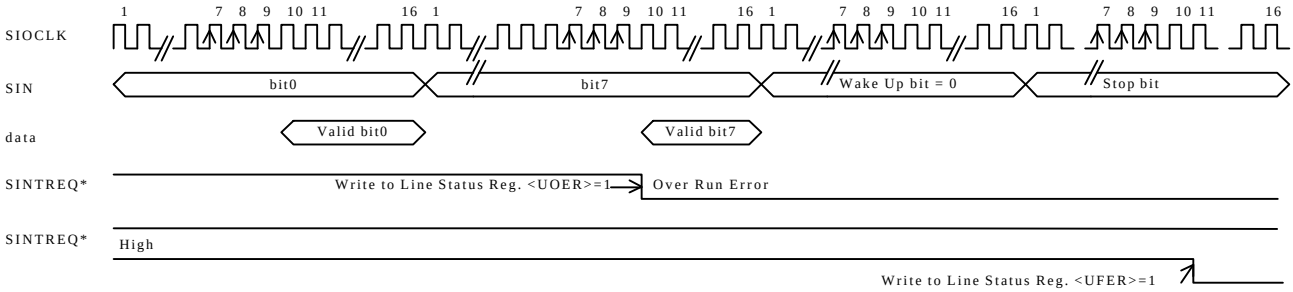


Fig. 12-21 Timing of Receiving (4)

12.5.6 Operation at the time of transmitting

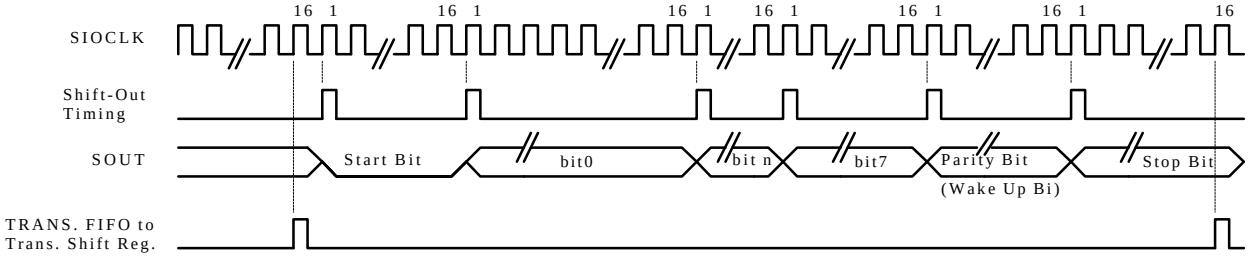


Fig. 12-22 Timing of Transmitting

12.5.7 Transmit halt timing by CTS*

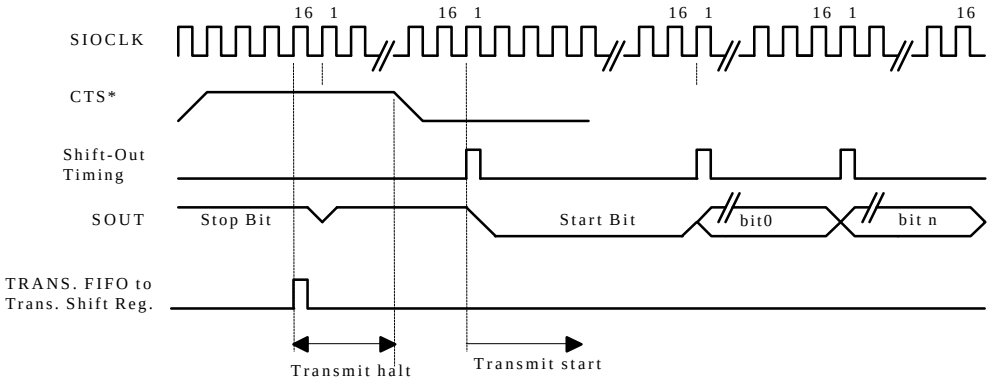


Fig. 12-23 Timing of CTS*

When the CTS* has become high while sending data, that transfer is done till the data byte, and after the transfer completion, the sending is halted. However, the next data are transferred from the FIFO to the transfer shift register.
The CTS* becomes low to resume the sending at the first shift-out start timing.
In a transfer operation, if the host I/F is the DMA mode and if the HSE of the control register is 1, the RTS* becomes high at the same timing as the SDMAREQ* assertion and it becomes low at the same timing as the SDMAREQ* negation.
In the interrupt mode, the RTS* becomes high at the SINTREQ* assertion and it becomes low at the SINTREQ* negation.

13 TIMERS/COUNTERS

13.1 Features

The timers/counters that are built-in in the TX3904 have the following three modes that use the 24-bit up counter:

(1) Interval Timer Mode
Generates regular interrupts
Can select the external input clock. Can count external events by using the external input clock.
Can select the rising and falling count of the clock in the external input clock.
(2) Pulse Generator Mode
Flip-flop output mode.
(3) Watchdog Timer mode
A timer to watch for runaways of the system

13.2 Block Diagrams

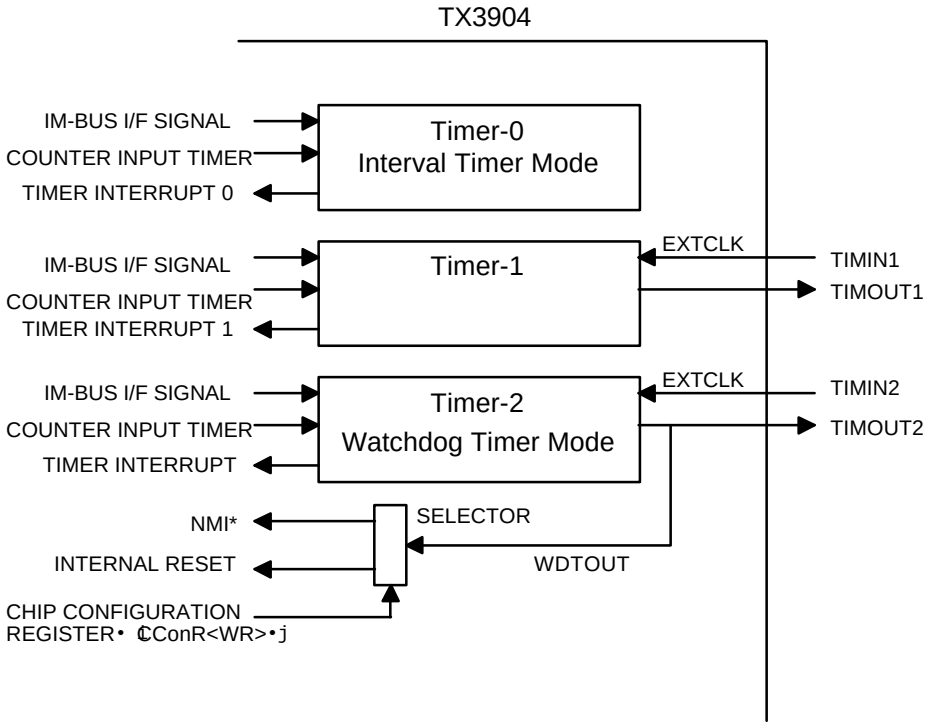


Fig. 13-1 Timer Module Connection inside the TX3904

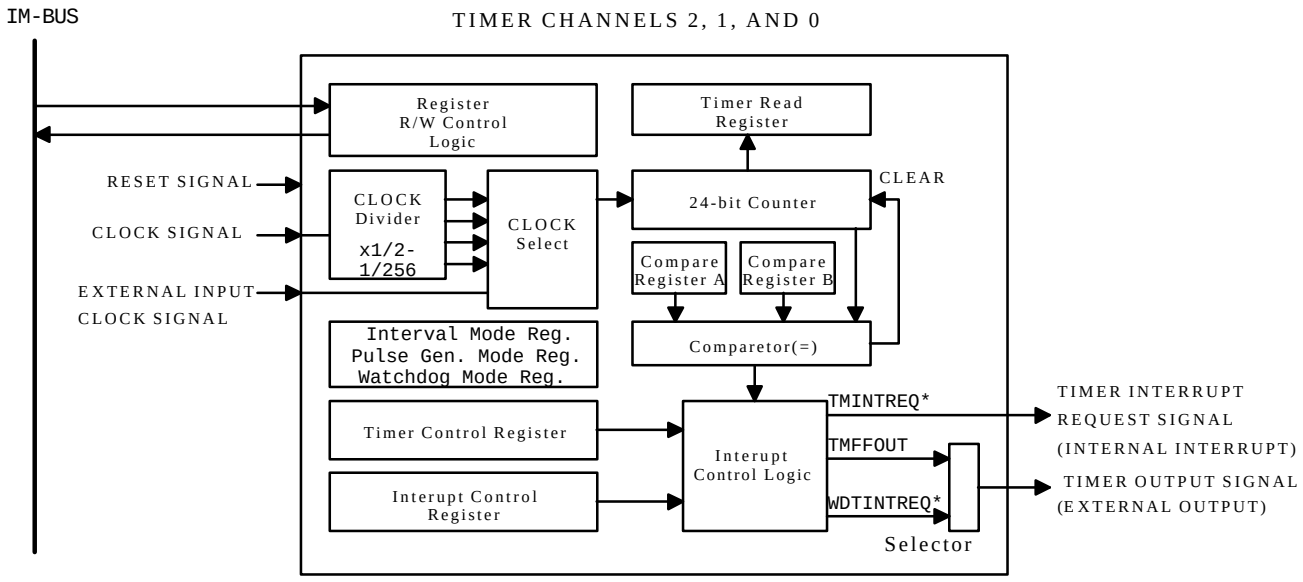


Fig. 13-2 PR39TMR Block Diagram

13.3 Registers

Table 13-1 Timer Registers (Timer 2)

Address	Register Symbol	Register Name
0xFFFF_F200	TCR2	Timer Control Register 2
0xFFFF_F204	TISR2	Timer Interrupt Status Register 2
0xFFFF_F208	CPRA2	Compare Register A2
0xFFFF_F20C	CPRB2	Compare Register B2
0xFFFF_F210	ITMR2	Interval Timer Mode Register 2
0xFFFF_F220	CCDR2	Divider Register 2
0xFFFF_F230	PGMR2	Pulse Generator Mode Register 2
0xFFFF_F240	WTMR2	Watchdog Timer Mode Register 2
0xFFFF_F2F0	TRR2	Timer Read Register 2

Table 13-2 Timer Registers (Timer 1)

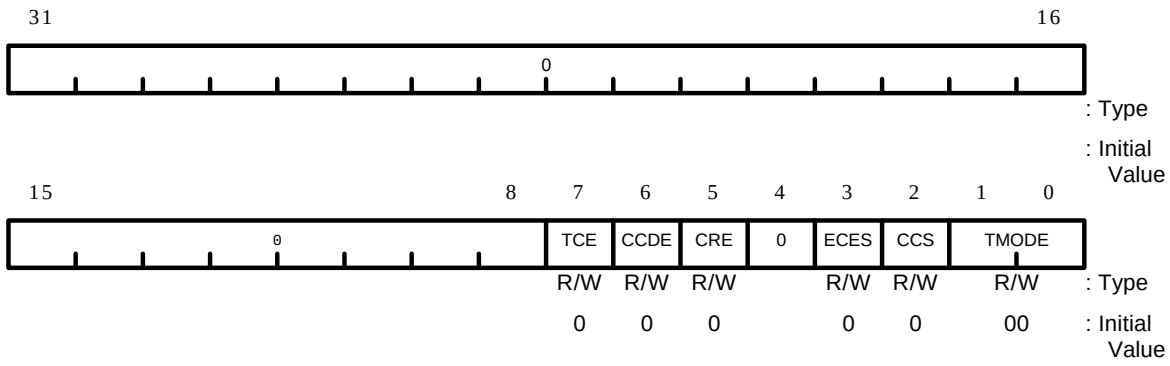
Address	Register Symbol	Register Name
0xFFFF_F100	TCR1	Timer Control Register 1
0xFFFF_F104	TISR1	Timer Interrupt Status Register 1
0xFFFF_F108	CPRA1	Compare Register A1
0xFFFF_F10C	CPRB1	Compare Register B1
0xFFFF_F110	ITMR1	Interval Timer Mode Register 1
0xFFFF_F120	CCDR1	Divider Register 1
0xFFFF_F130	PGMR1	Pulse Generator Mode Register 1
0xFFFF_F140	WTMR1	Watchdog Timer Mode Register 1
0xFFFF_F1F0	TRR1	Timer Read Register 1

Table 13-3 Timer Registers (Timer 0)

Address	Register Symbol	Register Name
0xFFFF_F000	TCR0	Timer Control Register 0
0xFFFF_F004	TISR0	Timer Interrupt Status Register 0
0xFFFF_F008	CPRA0	Compare Register A0
0xFFFF_F00C	CPRB0	Compare Register B0
0xFFFF_F010	ITMR0	Interval Timer Mode Register 0
0xFFFF_F020	CCDR0	Divider Register 0
0xFFFF_F030	PGMR0	Pulse Generator Mode Register 0
0xFFFF_F040	WTMR0	Watchdog Timer Mode Register 0
0xFFFF_F0F0	TRR0	Timer Read Register 0

Setting-up of the watchdog timer mode in Timers 1 and 0 has no meaning. Setting-up of the pulse generator mode in Timer 0 has no functional meaning.

13.3.1 Timer control registers 2, 1, and 0 (TCR2, 1, 0)



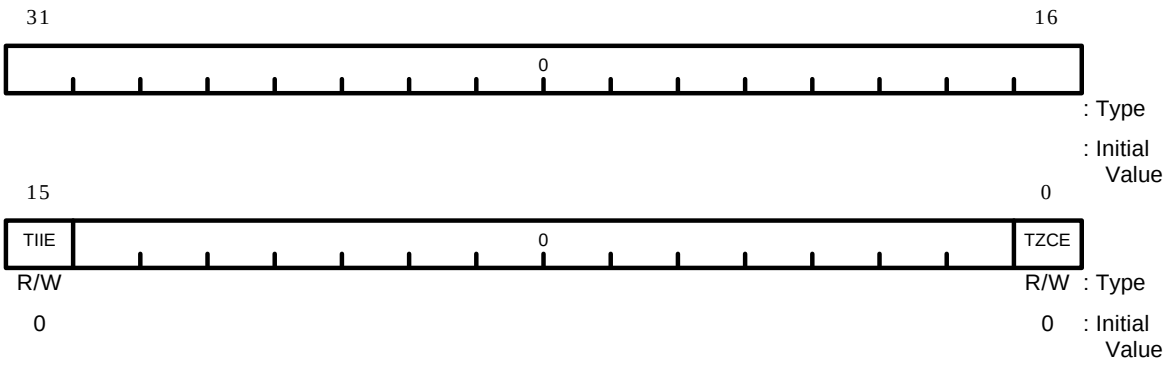
Bit	Mnemonic	Name of Field	Description
7	TCE	Timer counter enable	Timer Count Enable Controls the counter operation. Operations while the counter is being halted differ by the CRE bit. 0: Counter halted (When CRE=1, reset is also done.) 1: Counter in operation
6	CCDE	Timer clock divider enable	Counter Clock Divide Enable The enable set-up of the dividing operation of the internal system clock. When set at disable, the divide circuit halts. 0: Disable 1: Enable
5	CRE	Counter reset enable	Counter Reset Enable Controls reset of the 24-bit counter. By setting the TCE to 0 when the CRE is 1, the counter halts and resets. By setting the TCE to 0 when the CRE is 0, the counter halts.
3	ECES	External clock edge select	External Clock Edge Select Designates the operation edge of the counter when using the external input clock. 0: Falling edge 1: Rising edge

Fig. 13-3 Timer Control Registers (1/2)

Bit	Mnemonic	Name of Field	Description
2	CCS	Timer clock select	Counter Clock Select Selects the counter clock. 0: Internal system clock 1: External input clock
1:0	TMODE	Timer mode	Timer Mode Designates the operation mode of the timer. 11: Set-up disabled 10: Watchdog timer mode 01: Pulse generator mode 00: Interval timer mode
15:8 4	0		The execution of a writing is ignored. When read out, a zero is returned.

Fig. 13-4 Timer Control Registers (2/2)

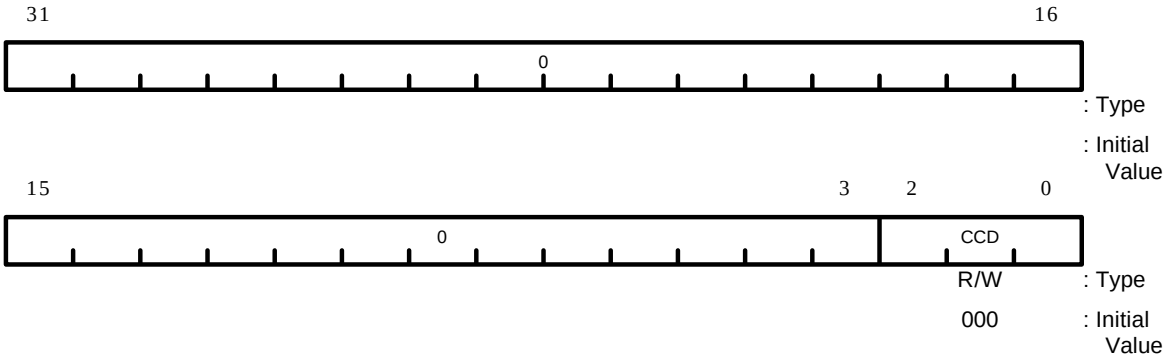
13.3.2 Interval timer mode registers 0, 1, and 2 (ITMR2, 1, 0)



Bit	Mnemonic	Name of Field	Description
15	TIIE	Interval timer interrupt enable	Timer Interval Interrupt Enable Sets up interrupt enable/disable in interval timer mode. 0: Disable (mask) 1: Enable
0	TZCE	Interval timer 0 clear enable	Interval Timer Zero Clear Enable Designates whether or not to clear the counter to 0 after the count value matches the compare register A0. If not cleared, the count is halted at that value. When the interrupt is recovered when the TZCE is 0 during the counter halt, the interrupt shall not reoccur. 0: Disable 1: Enable

Fig. 13-5 Interval Timer Mode Registers

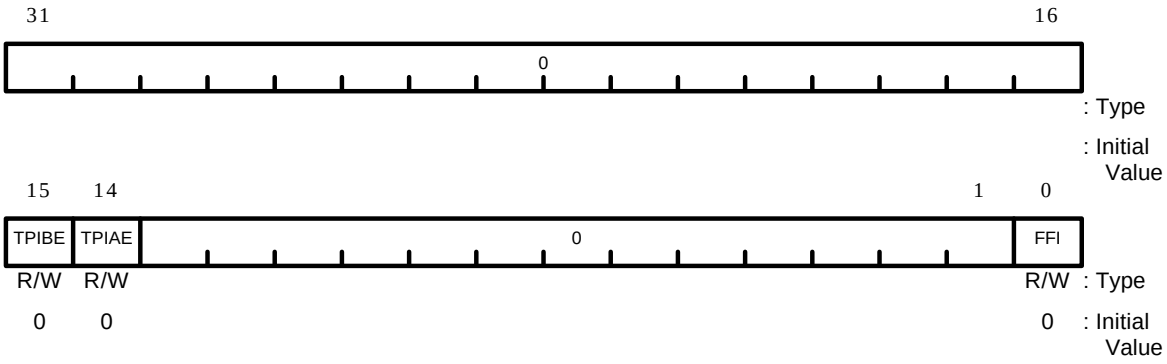
13.3.3 Divider registers 2, 1, and 0 (CCDR2, 1, 0)



Bit	Mnemonic	Name of Field	Description
2:0	CCD	Counter clock divider	Counter Clock Divide Designates the divider when the internal system clock is used for the counter input clock source. Shall be 2^{n+1} divided by the binary value n. 000: 2^1 dividing 001: 2^2 dividing 010: 2^3 dividing 011: 2^4 dividing 100: 2^5 dividing 101: 2^6 dividing 110: 2^7 dividing 111: 2^8 dividing

Fig. 13-6 Divider Registers

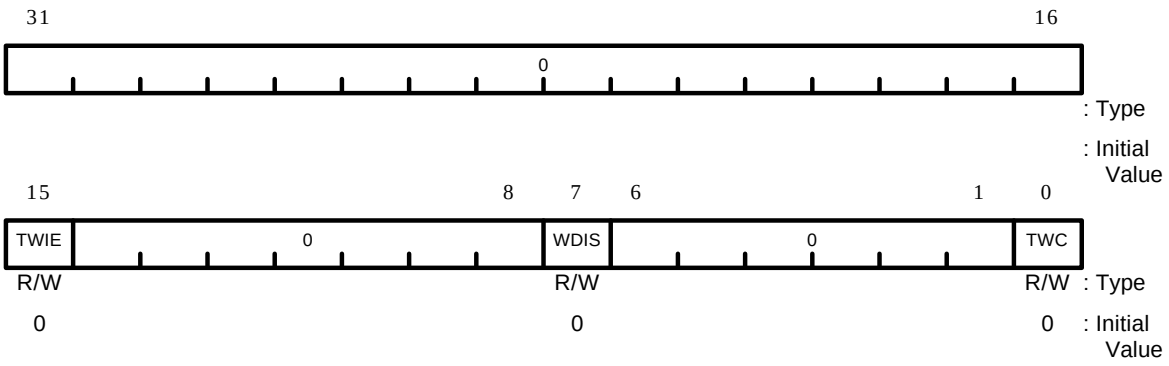
13.3.4 Pulse generator mode registers 2 and 1 (PGMR2, 1)



Bit	Mnemonic	Name of Field	Description
15	TPIBE	CPRB interrupt enable	Timer Pulse Generator Interrupt by CPRB Enable Sets up interrupt enable/disable when the CPRB and the counter value match in the pulse generator mode. 0: Disable (mask) 1: Enable
14	TPIAE	CPRA interrupt enable	Timer Pulse Generator Interrupt by CPRA Enable Sets up interrupt enable/disable when the CPRA and the counter value match in the pulse generator mode. 0: Disable (mask) 1: Enable
0	FFI	Flip-flop initial value	Timer Flip-Flop Initial Designates the initial value of timer flip-flop. 0: Low 1: High

Fig. 13-7 Pulse Generator Mode Registers

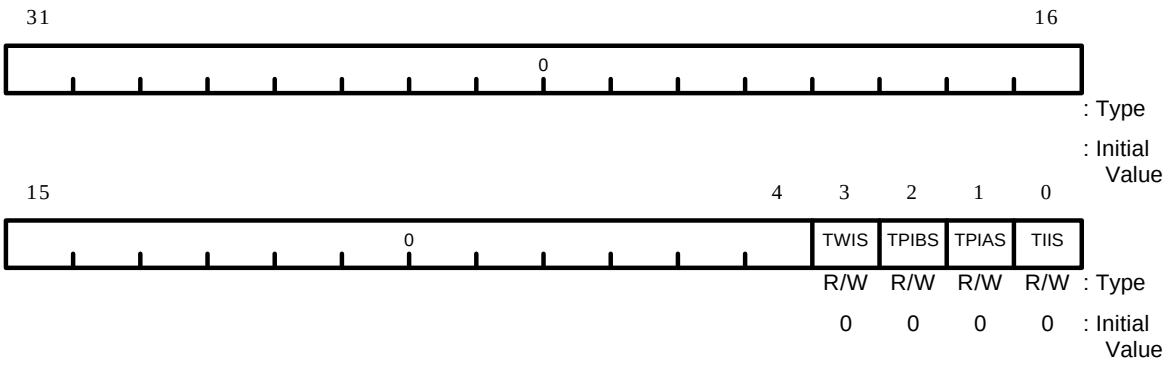
13.3.5 Watchdog timer mode register 2 (WTMR2)



Bit	Mnemonic	Name of Field	Description
15	TWIE	Watchdog timer interrupt enable	Timer Watchdog Interrupt Enable Sets up interrupt enable/disable in the watchdog timer mode. 0: Disable (mask) 1: Enable
7	WDIS	Watchdog timer disable	Watchdog Timer Disable The watchdog timer mode can be disabled by setting the WDIS to 1 and by setting the TCE of the timer control register to 0. The WDIS shall automatically be cleared to 0 after being disabled. The writing-in of "0" to the WDIS is invalid.
0	TWC	Watchdog timer clear	Timer Watchdog Clear 1: The counter shall be cleared when 1 is set to the TWC. The TWC shall automatically be reset to 0 after being cleared.

Fig. 13-8 Watchdog Timer Mode Register

13.3.6 Timer interrupt status registers 2, 1 and 0 (TISR2, 1, 0)



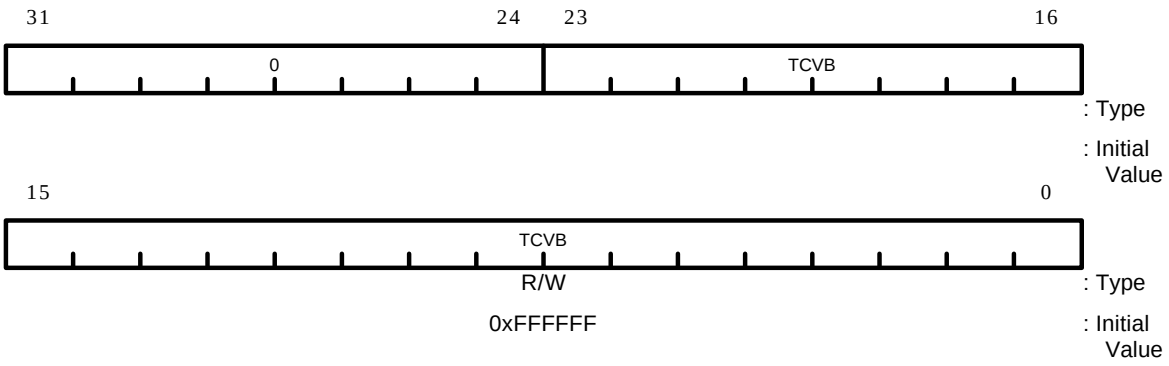
Bit	Mnemonic	Name of Field	Description
3	TWIS	Watchdog timer interrupt status	Timer Watchdog Interrupt Status When the TWIE is enable and when the counter value matches the compare register CPRA, the TWIS is set to assert the WDTINTREQ*. By writing "0" into the TWIS, the WDTINTREQ* is negated. The writing-in of "1" is invalid. 0: No interrupt has occurred when reading. Interrupt is negated when writing. 1: Matches with the compare register when reading so that an interrupt occurs. Invalid when writing.
2	TPIBS	Pulse generator CPRB interrupt status	Timer Pulse Generator Interrupt by CPRB Status When the TPIBS is enable and when the counter value matches the compare register CPRB, the TPIBS is set to assert the TMINTREQ*. By writing "0" into the TPIBS, the TMINTREQ* is negated. The writing-in of "1" is invalid. 0: When reading, no interrupt has occurred. When writing, interrupt is negated. 1: When reading, it matches with the compare register so that an interrupt occurs. When writing, invalid.

Fig. 13-9 Timer Interrupt Status Registers (1/2)

Bit	Mnemonic	Name of Field	Description
1	TPIAS	Pulse generator CPRA interrupt status	Timer Pulse Generator Interrupt by CPRA Status When the TPIAS is enable and when the counter value matches the compare register CPRA, the TPIAS is set to assert the TMINTREQ*. By writing "0" into the TIS, the TMINTREQ* is negated. The writing-in of "1" is invalid. 0: When reading, no interrupt has occurred. When writing, interrupt is negated. 1: When reading, it matches with the compare register so that an interrupt occurs. When writing, invalid.
0	TIIS	Interval timer interrupt status	Timer Interval Interrupt Status When the TIIE is enable and when the counter value matches the compare register CPRA, the TIIS is set to assert the TMINTREQ*. By writing "0" into the TIIS, the TMINTREQ* is negated. The writing-in of "1" is invalid. 0: When reading, no interrupt has occurred. When writing, interrupt is negated. 1: When reading, it matches with the compare register so that an interrupt occurs. When writing, invalid.

Fig. 13-10 Timer Interrupt Status Registers (2/2)

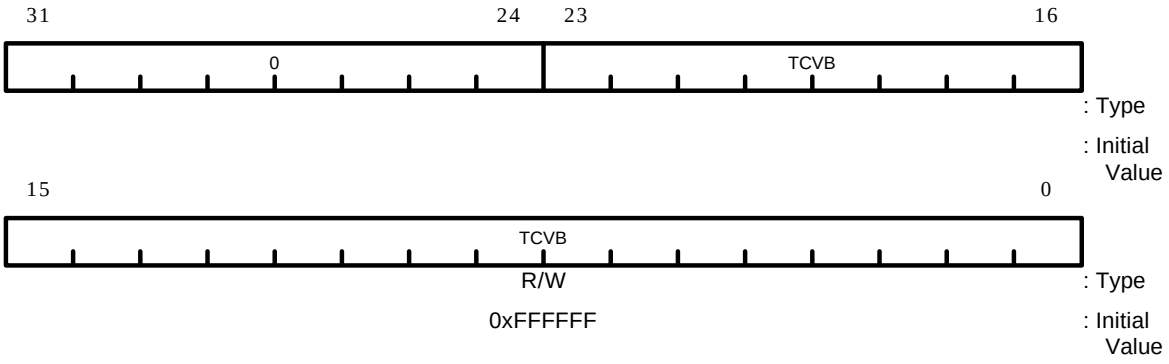
13.3.7 Compare registers A 2, 1 and 0 (CPRA2, 1, 0)



Bit	Mnemonic	Name of Field	Description
23:0	TCVA	Timer compare value A	Timer Compare Value Sets up the compare value of the timer. Sets up with binary values of 24 bits. Used in all modes.

Fig. 13-11 Compare Registers A

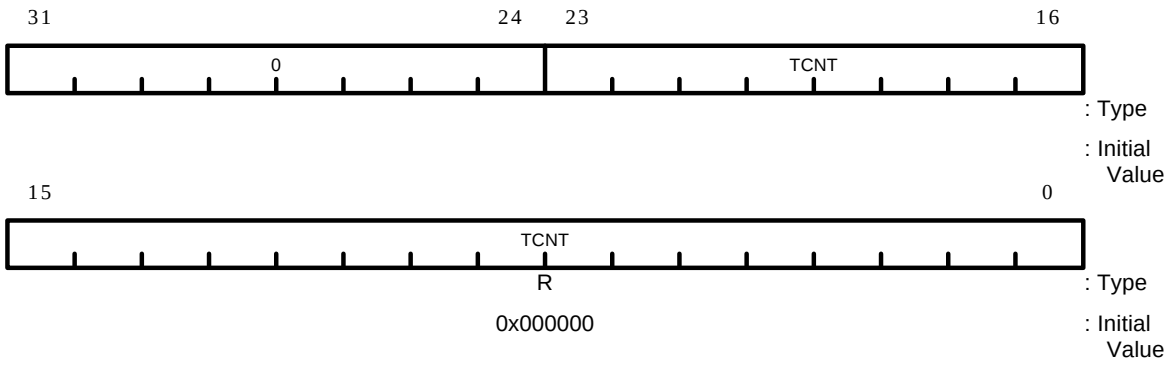
13.3.8 Compare registers B 2, 1 and 0 (CPRB2, 1, 0)



Bit	Mnemonic	Name of Field	Description
23:0	TCVB	Timer compare value B	Timer Compare Value The compare register B is used in the pulse generator mode. Please set up a value greater than the compare register A. Should be set up with binary values of 24 bits.

Fig. 13-12 Compare Registers B

13.3.9 Timer read registers 2, 1 and 0 (TRR2, 1, 0)



Bit	Mnemonic	Name of Field	Description
23:0	TCNT	Timer count value	Timer Count The 24-bit counter value is copied to this register. By reading this register, the count value is verified.

Fig. 13-13 Timer Read Registers

13.4 Operations

This section describes the case of Timer 2 because the operations are the same for Timers 2, 1 and 0. If there are differences on different channels, they shall be explained individually.

13.4.1 Interval timer mode

It is set up to the interval timer mode by the timer mode (TMODE)=00 of the timer control register (TCRn; n=0,1,2). The counter clock select (CCS) of the TCRn designates to use either the internal system clock or the external input clock. (Timer 0 can use only the internal system clock.)

When having selected the internal system clock, the clock that results from dividing the internal system clock can be regarded as the input of the 24-bit counter. The set up of the divider is conducted by the counter clock divide (CCD) of the divider register when the counter clock divide enable (CCDE) of the TCRn is 1. From 2^1 to 2^8 dividing of the internal system clock can be set up. When having selected the external input clock, the clock edge that operates to count can be selected in the external clock edge select (ECES) of the TCRn.

When the timer count enable (TCE) of the TCRn is set to 1, the 24-bit counter starts counting. When the count value matches the value of the compare register A (CPRA), a flag (“1”) is hoisted to the timer interval interrupt status (TIIS) of the status register (TISRn; n=2,1,0). The interrupt control logic asserts the timer interrupt request signal TMINTREQ* when 1 is set to the timer interval interrupt enable (TIIE) of the interval timer mode register (ITMRn; n=2,1,0). When 0 is set to the TIIE, the TMINTREQ* shall not be asserted. When “0” is written into the TIIS of the timer interrupt status register, the TIIS shall be cleared and the TMINTREQ* shall be deasserted. The writing-in of “1” to the TIIS is ignored.
When the timer zero clear enable (TZCE) of the ITMRn is set to 1, the 24-bit counter is cleared to 0 when the count value matches the value of the CPRA. When the TZCE is 0, the count operation is halted when the count value matches the value of the CPRA. The interrupt request occurrences are summarized in the Table 0-4 below.

Table 13-4 Interrupt Control by TIIE and TZCE

TIIE	TZCE	Interrupt operation when the counter has reached the set-up value.
0	*	Interrupt does not occur.
1	0	An interrupt occurs. If TZCE=0 when having recovered (write “0” to the TIIS) from the interrupt, another interrupt shall not occur. When the interrupt is recovered after TZCE=1 is set, it is the same as the cases of TIIE=1 and TZCE=1.
1	1	An interrupt occurs.

The reading of the count value can be done by reading the timer read register (TRRn; n=2,1,0). The value that was read from the TRRn is a copy of the count value of the 24-bit counter.
Fig. 9-14 shows the outline of the count operation and interrupt occurrences in the interval timer mode.
Fig. 9-15 shows the operation using the external input clock.

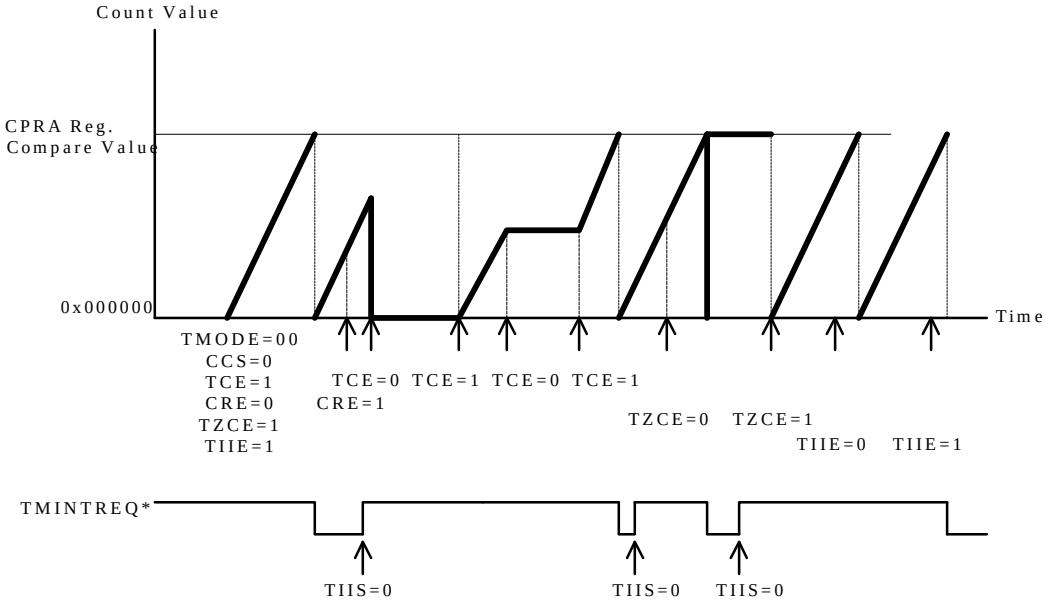


Fig. 13-14 Interval Timer Operation (Internal Clock Used)

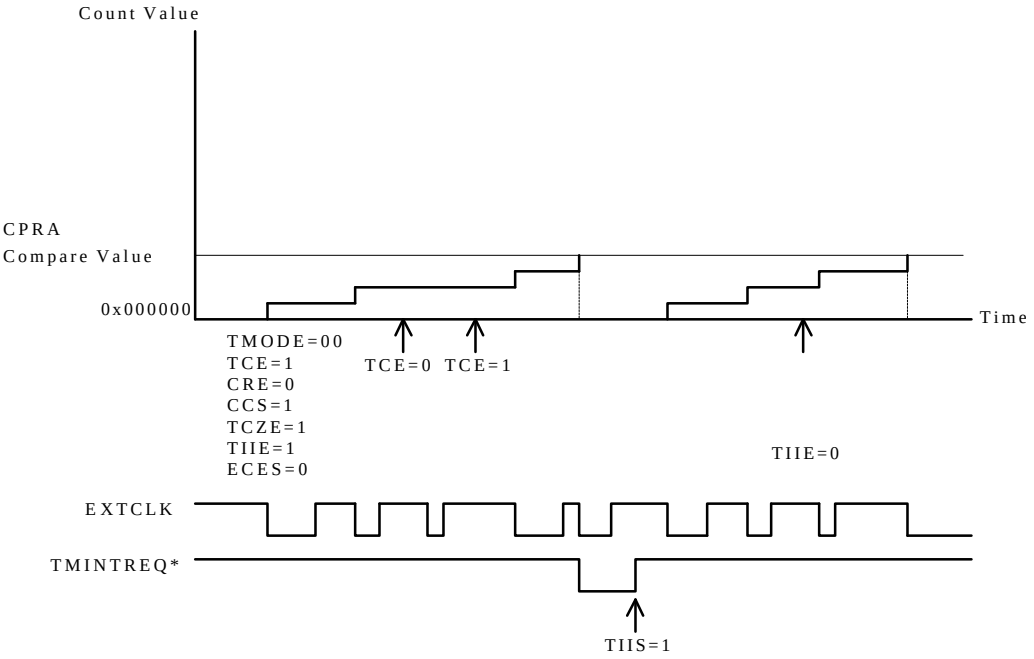


Fig. 13-15 Interval Timer Operation (External Input Clock; Falling Edge Operation)

Table 0-5 shows the count time when using the internal system clock.

Table 13-5 Divider and Count Number

Divider		IMCLK=25MHz			
Decimal	CCDR <CCD>	freq.(Hz)	Resolution (s)	24bit Max time(s)	Count of 1s
2	000	12.50E+6	80.00E-9	1.34	12500000
4	001	6.25E+6	160.00E-9	2.68	6250000
8	010	3.13E+6	320.00E-9	5.37	3125000
16	011	1.56E+6	640.00E-9	10.7	1562500
32	100	781.25E+3	1.28E-6	21.5	781250
64	101	390.63E+3	2.56E-6	42.9	390625
128	110	195.31E+3	5.12E-6	85.9	195313
256	111	97.66E+3	10.24E-6	171.8	97656

13.4.2 Pulse generator mode

When the TMODE of the TCRn (n=2,1) is 1, it is the pulse generator mode. In the pulse generator mode, quadrangular waves of at-will frequency and duty can be output using two compare registers--CPRA and CPRB. (The pulse generator mode cannot be used in Timer 0.)

When the TCE of the TCRn is set to 1, the 24-bit counter starts counting. When the value set to the CPRA and the count value match, the timer flip-flop is reversed. The output of the timer flip-flop is output to the TMFFOUT. After having matched with the CPRAn, the counter continues counting. When the value set to the CPRBn and the count value match, the timer flip-flop shall be reversed to clear the counter. The CPRA must be smaller than the CPRB. The initial status of the timer flip-flop can be set up in the flip-flop initialize (FFI) of the pulse generator mode register (PGMRn; n=2,1).

When the count value matches the value of CPRA, a flag (“1”) is hoisted to the timer pulse generator interrupt CPRA status (TPIAS) of the TISRn. The interrupt control logic asserts the timer interrupt request TMINTREQ* when 1 is set to the timer pulse generator interrupt CPRA enable (TPIAE) of the PGMRn. When 0 is set to the TPIAE, the TMINTREQ* shall not be asserted. By writing in “0” into the TPIAS of the TISRn, the TPIAS shall be cleared and the TMINTREQ* shall be deasserted. When the timer pulse generator interrupt CPRB enable (TPIBE) is set to 1, a flag (“1”) shall be hoisted in the TPIBS of the TISRn when matching with the CPRBn to assert TMINTREQ*. By writing in “0”, the TPIBS shall be cleared and the TMINTREQ* shall be deasserted.

The counter clock can select the internal system clock and the external input clock. The selection of the clock is conducted in the CCS of the TCRn. (Timer 0 can only be used with the internal system clock.)
When using the internal system clock, it can be divided. The set up of the divider is conducted by the CCD of the divider register when the CCDE of the TCRn is 1. From 2^1 to 2^8 dividing of the internal system clock can be set up. Count operations are conducted at the rising of the clock.
When using the external input clock, the edge of the clock can be selected in the ECES of the TCRn.

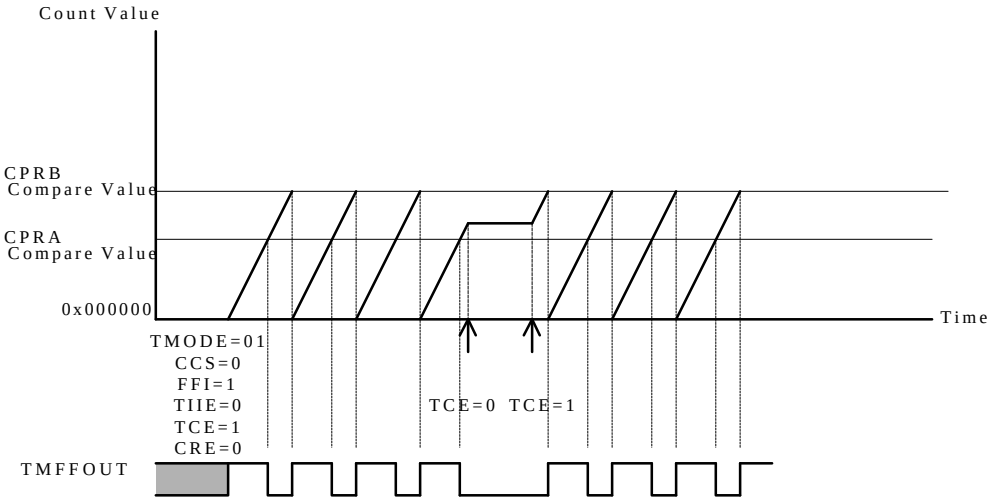


Fig. 13-16 Pulse Generator Mode Operation

13.4.3 Watchdog timer mode

In the TX3904, only the watchdog timer interrupt request signal (WDTINTREQ*) of the timer/counter Channel 2 can be connected to the internal NMI* or the reset circuit. The connection target of the WDTINTREQ* signal is selected in the WR bit of the chip configuration register CConR.

When the TMODE of the TCR2 is 10, it is the watchdog mode. When the TCE of the TCR2 is set to 1, the 24-bit counter starts counting. When the count value matches the CPRA, the comparator hoists a flag ("1") to the timer watchdog interrupt status (TWIS) of the TISR2. The interrupt control logic asserts the watchdog timer interrupt request WDTINTREQ* when 1 is set to the timer watchdog interrupt enable (TWIE) of the TCRn. When 0 is set to the TWIE, the WDTINTREQ* shall not be asserted. The WDTINTREQ* is deasserted by writing "0" into the TWIS. The writing-in of "1" to the TWIS shall be ignored. When 1 is set to the timer watchdog clear (TWC) of the watchdog timer mode register 2 (WTMR2), the 24-bit count shall be cleared. The TWC shall be cleared to 0 automatically after the 24-bit counter clears.
The watchdog timer can be disabled (counter halted) by setting the TCE to 0 when the watchdog timer disable (WDIS) of the WTMR2 is set to 1. When the WDIS is 0, the count operation cannot be halted even if the TCE is set to 0. If the TWIE of the TISR2 is set to 0 while WDIS is 1, the watchdog timer can also be disabled (interrupt masking). The TWIE cannot be set to 0 when the WDIS is 0. The WDIS shall automatically be cleared to 0 when the watchdog timer has become disabled by one of the above methods.
The count value can be read by reading the TRR2.

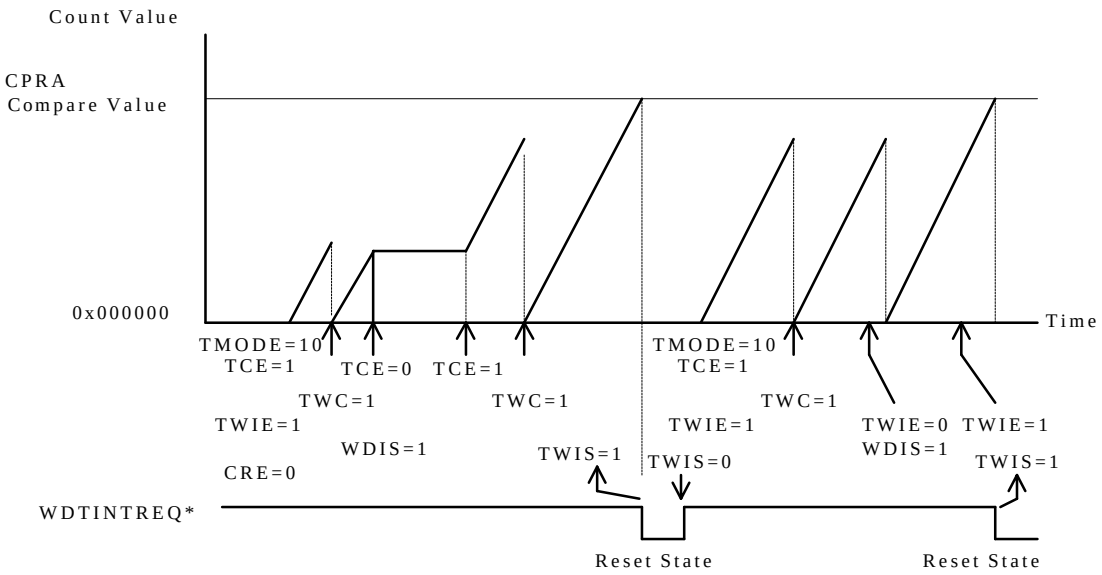


Fig. 13-17 Watchdog Timer Mode Operation

13.5 Timing Explanations

13.5.1 Interval timer mode interrupt timing

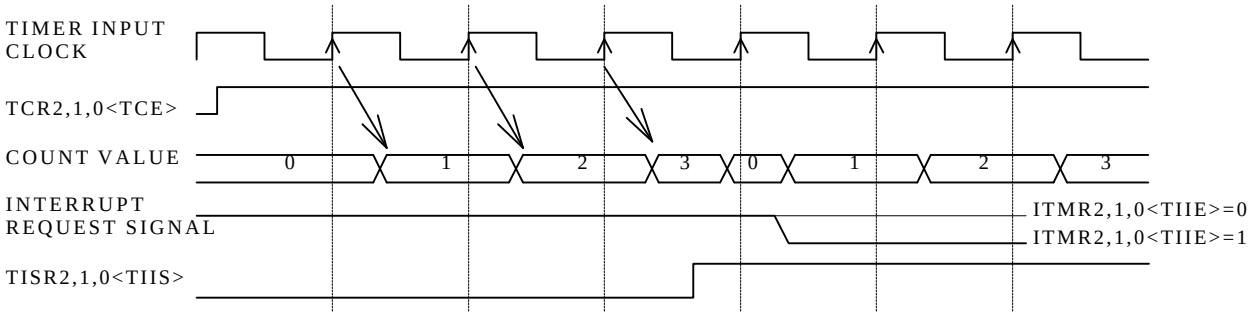


Fig. 13-18 Interval Timer Timing (Internal Clock)

The above diagram is the case of CPRA=3 and the counter clock IMCLK/2. The register TISR <TIIS> is set at the rising of the IMCLK where a matching is detected, and the TMINTREQ* shall be asserted synchronously with the internal system clock. At the same time, the counter is cleared to 0. (Register ITMR <TZCE> = 1)

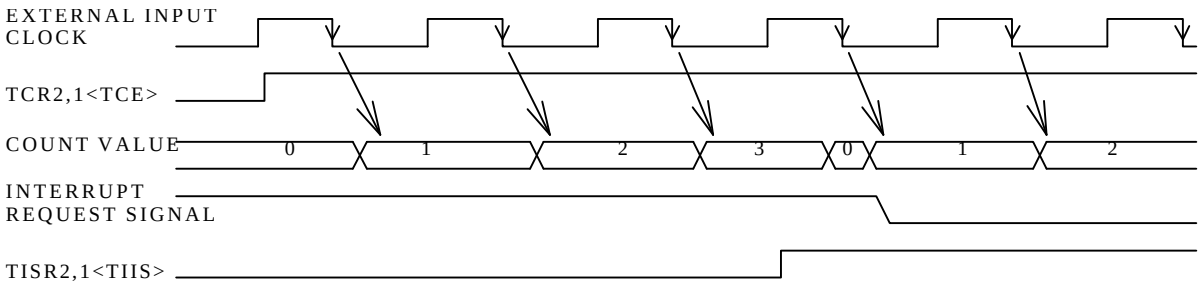


Fig. 13-19 Interval Timer Timing (External Input Clock)

The above diagram is the case of CPRA=3 and where the external input clock is selected for the counter clock. The register TICR <TIS> is set at the rising of the IMCLK where a matching is detected, and the TMINTREQ* shall be asserted. A the same time, the counter is cleared to 0. (Register ITMR <TZCE> = 1)

13.5.2 Pulse generator mode F/F output timing

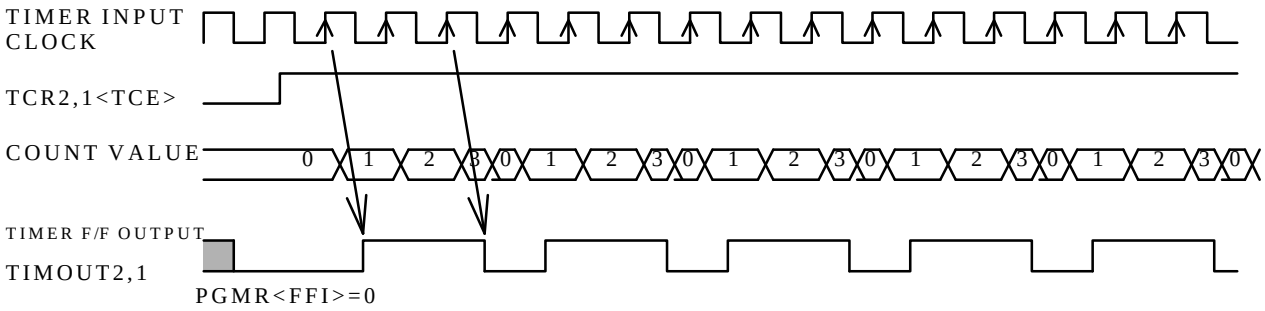


Fig. 13-20 Pulse Generator Mode Timing

The above diagram is the case where CPRA=1 and CPRB=3 in the pulse generator timing mode. The initial value of the timer F/F is set to 0. The timer F/F is initialized at the same time as the PGMR writing. In this diagram, a pulse waveform with 2:1 duty is being output at approximately 4.2 MHz.

13.5.3 Watchdog timer mode interrupt timing

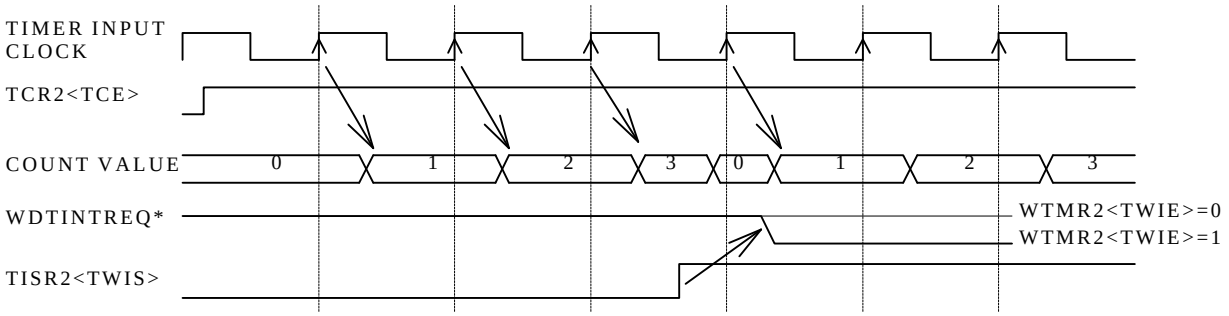


Fig. 13-21 Watchdog Timer Timing

14 IO PORTS (PIO)

The TX3904 has one channel exclusive PIO and two channels of shared PIO's. One channel comprises a 8-bit port.
The exclusive PIO (PIO0) uses the PIO0[7:0] signal. In the shared PIO's (PIO2 and PIO1), the PIO2[7:0] and PIO1[7:0] signals share pins with other signals.
The PIO can designate the input/output directions of data in each bit.

14.1 Set-Up of the Shared PIO's

The TX3904 has three channels of 8-bit PIO's. Of these, the pins of the two channels (PIO2 and PIO1) are shared with other signals. If these pins are used as I/O ports or as other functions is designated in the P2En bit and P1En bit of the CConR.
Table 0-1 and Table 0-2 show the correspondence of the shared pins of the PIO2 and PIO1:

Table 14-1 PIO2 Pins

PIO2[7]	PIO2[6]	PIO2[5]	PIO2[4]	PIO2[3]	PIO2[2]	PIO2[1]	PIO2[0]
A[31]	A[30]	A[29]	A[28]	A[27]	A[26]	A[25]	A[24]

Table 14-2 PIO1 Pins

PIO1[7]	PIO1[6]	PIO1[5]	PIO1[4]	PIO1[3]	PIO1[2]	PIO1[1]	PIO1[0]
DREQ[3]	DREQ[2]	DACK[3]	DACK[2]	BUSREL*	HAVEIT*	BUSREQ*	BUSGNT*

14.2 Functions

The signal of the PIO can be set up input/output in each bit. The set-up of input/output is designated by the PIO direction register. The status of the signal that is set up as an input signal is indicated in the PIO data register. The signal that is set up as an output signal outputs the value of the PIO data register.

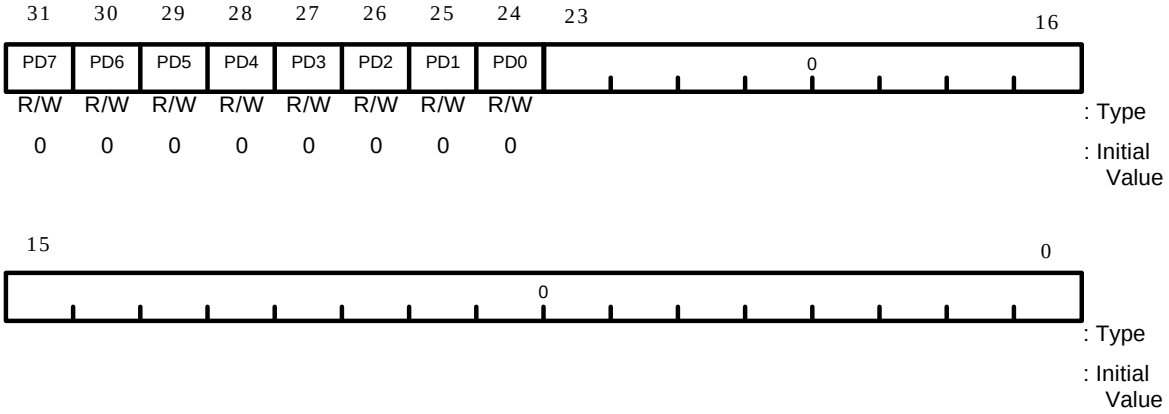
14.3 Registers

Figure 14-3 shows the register map of the PIO's.

Table 14-3 PIO Register Map

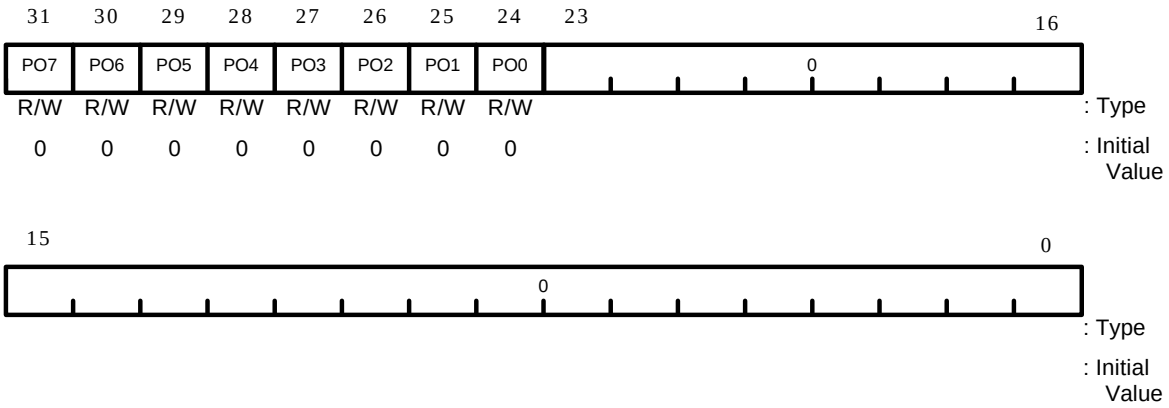
Address	Module	Register Name
0xFFFF_F704	PIO2	Data Register 2 (PDR2)
0xFFFF_F700	PIO2	Direction Register 2 (POR2)
0xFFFF_F604	PIO1	Data Register 1 (PDR1)
0xFFFF_F600	PIO1	Direction Register 1 (POR1)
0xFFFF_F504	PIO0	Data Register 0 (PDR0)
0xFFFF_F500	PIO0	Direction Register 0 (POR0)

14.3.1 PIO data registers (PDR2, 1, 0)



Bit	Mnemonic	Name of Field	Description
31	PD7	Port data 7	Port Data [7] Maintains the input/output data of the PIOn[7] signal. Indicates the output data when the PO7 bit of the direction register is 1, and indicates the input data when it is 0. Even if the TX39 Processor Core writes in a value to the PD7 when the PO7 bit is 0, it shall be ignored.
30	PD6	Port data 6	PIO Data [6]
29	PD5	Port data 5	PIO Data [5]
28	PD4	Port data 4	PIO Data [4]
27	PD3	Port data 3	PIO Data [3]
26	PD2	Port data 2	PIO Data [2]
25	PD1	Port data 1	PIO Data [1]
24	PD0	Port data 0	PIO Data [0]

14.3.2 PIO direction registers (POR2, 1, 0)



Bit	Mnemonic	Name of Field	Description
31	PO7	Port output mode [7]	Port Output Mode [7] Sets up the direction of the PIOn[7] signal. 1: PIOn[7] signal becomes the output signal. 0: PIOn[7] signal becomes the input signal.
30	PO6	Port output mode [6]	PIO Output Mode [6]
29	PO5	Port output mode [5]	PIO Output Mode [5]
28	PO4	Port output mode [4]	PIO Output Mode [4]
27	PO3	Port output mode [3]	PIO Output Mode [3]
26	PO2	Port output mode [2]	PIO Output Mode [2]
25	PO1	Port output mode [1]	PIO Output Mode [1]
24	PO0	Port output mode [0]	PIO Output Mode [0]