E2F0016-19-43

Preliminary

OKI Semiconductor

This version: Apr. 1999

MSM9225

CAN (Controller Area Network) Controller

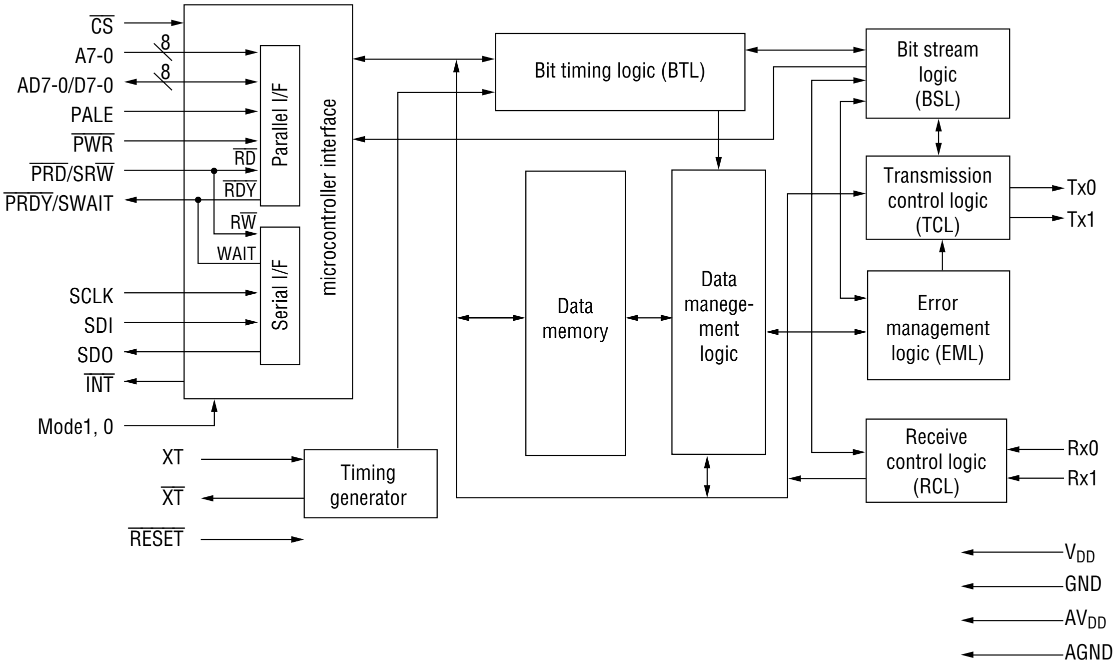
GENERAL DESCRIPTION

The MSM9225 is a microcontroller peripheral LSI which conforms to the CAN protocol for high-speed LANs in automobiles.

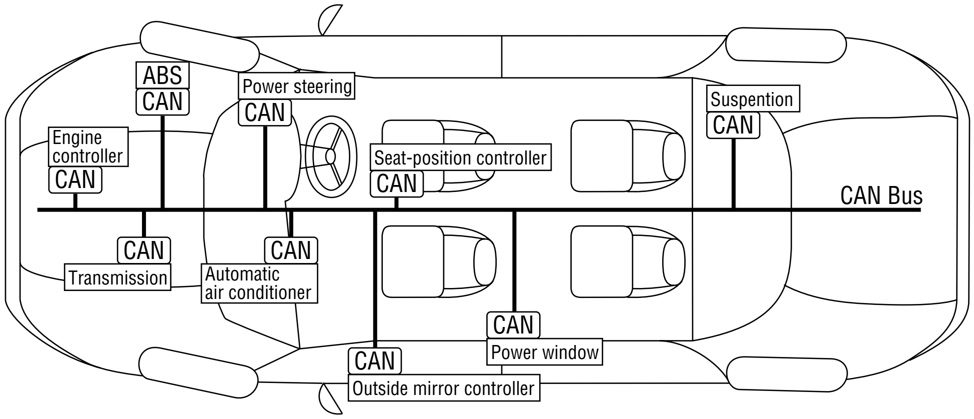
FEATURES

- Conforms to CAN protocol specification (Bosch Co., V.2.0 part b/Full CAN)
- Maximum 1 Mbps real-time communication control (programmable)
- Communication system:
  - Transmission line is bi-directional, two-wire serial communications
  - NRZ (Non-Return to Zero) system using bit stuff function
  - Multi-master system
  - Broadcast system
- Maximum 16 messages × 8 bytes of message buffer
  - Number of messages can be extended by group message function (max: 2 groups)
- Priority control by identifier
  - Normally 2032 types, 2032×2^{18} types at extension
- Microcontroller interface
  - Corresponding to both parallel and serial interface
    - Parallel interface: separate address/data bus type (with address latch signal/no address latch signal) and multiplexed address/data bus type.
    - Serial interface: Synchronous communication type
  - Interrupt is used for three outputs: transmission/receive/error
- Error control:
  - Bit error/stuff error/CRC error/form error/acknowledge error detection functions
  - Retransmission / error status monitoring function when error occurs
- Communication control by transmission request function
- Sleep/Stop mode function
- Supply voltage: 5 V ±10%
- Operating temperature: -40 to +115°C
- Package: 44-pin plastic QFP (QFP44-P-910-0.80-2K) (Product name: MSM9225GA-2K)

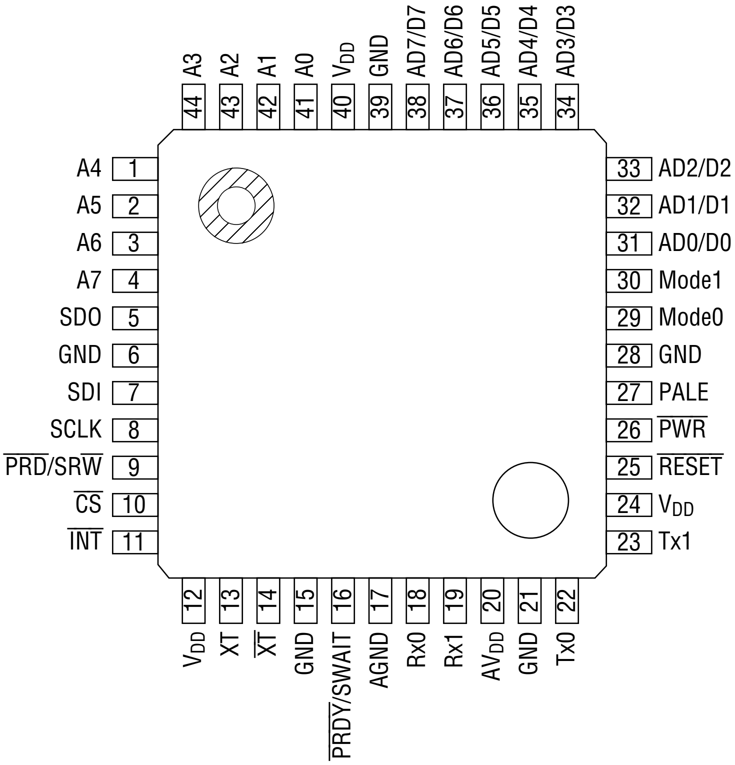
BLOCK DIAGRAM



CONFIGURATION EXAMPLE



PIN CONFIGURATION (TOP VIEW)



44-Pin Plastic QFP

PIN DESCRIPTIONS

Symbol	Pin	Type	Description
$\overline{\text{CS}}$	10	I	Chip select pin. When "L", PALE, $\overline{\text{PWR}}$, $\overline{\text{PRD}}$/SR$\overline{\text{W}}$, SCLK and SDO pins are valid.
A7-0	41-44, 1-4	I	Address bus pins (when using separate buses). If used with a multiplexed bus or if used in the serial mode, fix these pins at "H" or "L" levels.
AD7-0 /D7-0	31-38	I/O	Multiplexed bus: Address/data pins Separate buses: Data pins If used in the serial mode, fix these pins at "H" or "L" levels.
$\overline{\text{PWR}}$	26	I	Write input pin during parallel mode. Data is captured when this pin is at a "L" level. If used in the serial mode, fix this pin at a "H" or "L" level.
$\overline{\text{RPD}}$/SR$\overline{\text{W}}$	9	I	Parallel mode: Read signal pin. When at a "L" level, data is output from the data pin. Serial mode: Read/write signal pin. When at a "H" level, data is output from the SDO pin. When at a "L" level, the SDO pin is at high impedance, and data is captured beginning with the second byte of data input from the SDI pin.
PALE	27	I	Address latch signal pin. When at a "H" level, addresses are captured. If used in the parallel mode and the address latch signal is unnecessary or in the serial mode, fix this pin at a "H" or "L" level.
SDI	7	I	Serial data input pin. Addresses (1st byte) and data (beginning from the 2nd byte) are input to this pin, LSB first. If used in the parallel mode, fix this pin at a "H" or "L" level.
SDO	5	O	Serial data output pin. When the $\overline{\text{CS}}$ pin is at a "H" level, this pin is at high impedance. When $\overline{\text{CS}}$ is at a "L" level, data is output from this pin LSB first. If used in the parallel mode, fix this pin at a "H" or "L" level.
SCLK	8	I	Shift clock input pin for serial data. At the rising edge of the shift clock, SDI pin data is captured. At the falling edge, data is output from the SDO pin.
$\overline{\text{PRDY}}$ /SWAIT	16	O	Ready output pin. If the microcontroller's bus cycle is fast, a signal is output to extend the bus cycle until the internal access is completed.

	Internal access in progress	After completion of access
Parallel mode	"L" level output	High impedance output
Serial mode	"H" level output	"L" level output

<table>
<tr><th>Symbol</th><th>Pin</th><th>Type</th><th>Description</th></tr>
<tr><td>Mode1, 0</td><td>29, 30</td><td>I</td><td>Microcontroller interface select pins.<table>
<tr><th>Mode1</th><th>Mode0</th><th colspan="3">Interface</th></tr>
<tr><td>0</td><td>0</td><td rowspan="3">Parallel mode</td><td rowspan="2">Separate buses</td><td>No address latch signal</td></tr>
<tr><td>0</td><td>1</td><td>With address latch signal</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">Multiplexed buses</td></tr>
<tr><td>1</td><td>1</td><td colspan="3">Serial mode</td></tr>
</table></td></tr>
<tr><td>$\overline{\text{INT}}$</td><td>11</td><td>O</td><td>Interrupt request output pin.
When an interrupt request occurs, a "L" level is output.
Three types of interrupts share this pin: transmission complete, reception complete, and error.</td></tr>
<tr><td>$\overline{\text{RESET}}$</td><td>25</td><td>I</td><td>Reset pin.
System is reset when this pin is at a "L" level.</td></tr>
<tr><td>XT</td><td>13</td><td>I</td><td rowspan="2">Clock pins. If internal oscillator is used, connect a crystal oscillator. If external clock is input, input clock via XT pin. The $\overline{\text{XT}}$ pin should be left open.</td></tr>
<tr><td>$\overline{\text{XT}}$</td><td>14</td><td>O</td></tr>
<tr><td>RX0, RX1</td><td>18, 19</td><td>I</td><td>Receive input pin. Differential amplifier included.</td></tr>
<tr><td>TX0, TX1</td><td>22, 23</td><td>O</td><td>Transmission output pin.</td></tr>
<tr><td>V_{DD}</td><td>12, 24, 40</td><td>—</td><td>Internal logic power supply pin.</td></tr>
<tr><td>GND</td><td>6, 15, 21 28, 39</td><td>—</td><td>Internal logic GND pin.</td></tr>
<tr><td>AV_{DD}</td><td>20</td><td>—</td><td>Power supply pin for receive input differential amplifier.</td></tr>
<tr><td>AGND</td><td>17</td><td>—</td><td>GND pin for receive input differential amplifier.</td></tr>
</table>

FUNCTIONAL DESCRIPTION

Data Memory

Before starting communication, messages for communication and various control registers must be set at the data memory.

Addresses X0hex to XDhex are the message memory, which stores control registers, identifiers and the contents of each message.

In this address space, the higher 4 bits of an address corresponds to the number of messages, and a maximum of 16 (0Xhex to FXhex) can be stored. Each message has an area to store a maximum of 8 bytes of data memory, 1 byte of control register, and a maximum of 5 bytes of an identifier.

This means that the data memory capacity for messages is: 16 messages × (8 bytes for a message + 1 byte for a control register + 5 bytes for an identifier) = 224 bytes.

Addresses XEhex to XFhex are allocated to the control registers.

The configuration of data memory is as follows

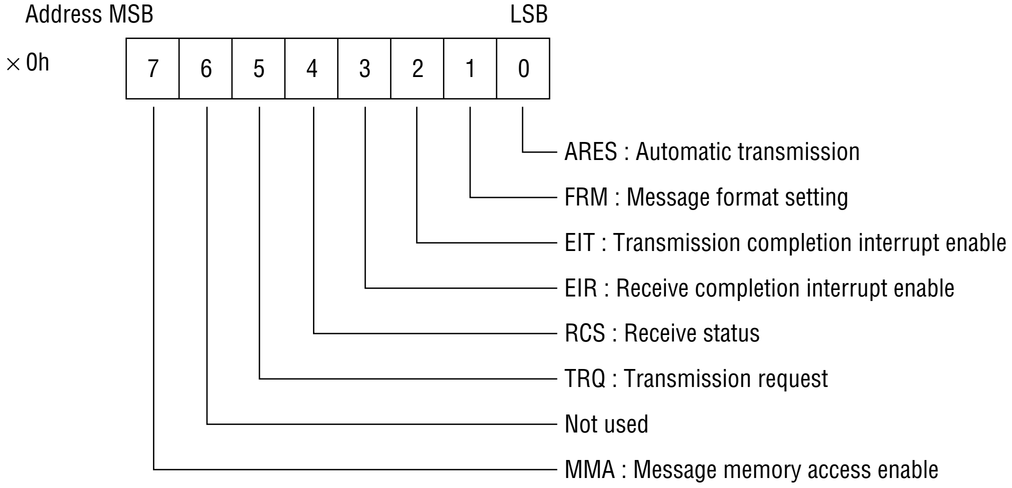
Data memory configuration

Address								Function	
A7	A6	A5	A4	A3	A2	A1	A0	IDFM = 0 (standard)	IDFM = 1 (extended)
				0	0	0	0	Message control register	
Corresponds to number of messages				0	0	0	1	Identifier 0	
				0	0	1	0	Identifier 1	
				0	0	1	1	Message 0	Identifier 2
0	0	0	0	0	1	0	0	Message 1	Identifier 3
↓				0	1	0	1	Message 2	Identifier 4
1	1	1	1	0	1	1	0	Message 3	Message 0
				0	1	1	1	Message 4	Message 1
				1	0	0	0	Message 5	Message 2
				1	0	0	1	Message 6	Message 3
				1	0	1	0	Message 7	Message 4
				1	0	1	1	—	Message 5
				1	1	0	0	—	Message 6
				1	1	0	1	—	Message 7
0	0	0	0	1	1	1	0	Various control registers	
↓									
1	1	1	1	1	1	1	1		

Message Memory

The message memory stores messages to be transmitted/received.
For transmission, only messages stored in the message memory can be transmitted. A message with the highest priority among messages requested for transmission is sent.
For receiving, only messages that have an identifier stored in the message memory can be received. When a message is received normally, and its identifier matches with the identifier stored in the message memory, data of the received message is written to the message area of the corresponding message in the message memory.
The message memory can store a maximum of 16 messages. Set messages at the NMES register.

1. Inside message control register (X0hex)
   This register performs various controls for a message.
   Set this register for each message.
   The bit configuration is as follows:



(1) Message memory access request/enable bit: MMA
This bit prevents contention between the microcontroller and CAN when accessing the message memory.
When the microcontroller accesses the message memory, "1" is written to the MMA bit regularly. The microcontroller confirms that the MMA bit is "1", and then accesses the message memory. Write "0" to the MMA bit when the microcontroller accessing ends.
Operations by the MMA bit are shown in the following table.
At reset, the MMA bit is set to "0".

MMA	Microcontroller	Reception	Transmission
0	Accesses from microcontroller to message memory are disabled	Operate Reading of received data	Operate
1	Accesses from microcontroller to message memory are enabled	Stop Rewriting of control area	Stop Rewriting of control area Rewriting of transmission data

(2) Transmission request: TRQ

When a message is transmitted, the microcontroller writes "1" to this bit. When transmission ends normally, CAN writes "0". This means that the TRQ bit is "1" during transmission. Therefore, to request transmission, confirm that the TRQ bit is "0" first, then write "1" to the TRQ bit. When the remote frame is received while the ARES bit is "1", the TRQ bit is set to "1".

At reset, the TRQ bit is set to "0".

(3) Receive status: RCS

When receiving completes, the RCS bit becomes "1". Write "0" to the RCS bit before the microcontroller calls up receive data. When receiving the remote frame, the RCS bit becomes "1" just after the reception.

At reset, the RCS bit is set to "0".

(4) Receive completion interrupt enable: EIR

The microcontroller sets interrupt request signal generation disable/enable when receiving completes.

The EIR bit is valid when the EINTR bit of the CANI register is “1”.

At reset, the EIR bit is set to "0".

(5) Transmission completion interrupt enable: EIT

The microcontroller sets interrupt request signal generation disable/enable when transmission completes.

The EIT bit is valid when the EINTT bit of the CANI register is “1”.

At reset, the EIT bit is set to "0".

(6) Message format setting: FRM

The microcontroller sets the format of the message to be sent/received. A message in a format other than the specified format cannot be sent/received.

For the relationship between setting and format, see the table below.

When a message specified to a group message is received, the content of the RTR bit is written.

At reset, the FRM bit is set to "0".

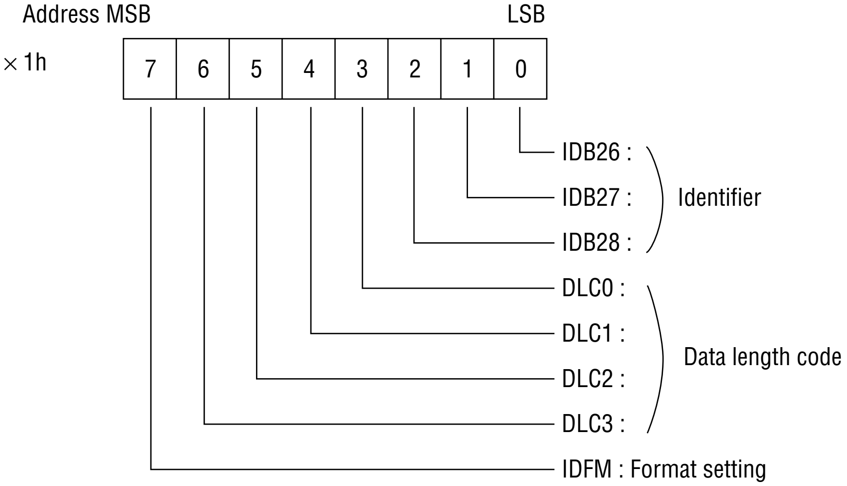
FRM	Message Type	Transmission Format	Receive Format
0	Standard message	Data frame	Remote frame
	Group message	Transmission disable	Data frame
1	Standard message	Remote frame	Data frame
	Group message	Transmission disable	Remote frame

(7) Automatic transmission : ARES

If the data frame is automatically transmitted after remote frame reception, the ARES bit should be set to "1".

At reset, the ARES bit is set to "0".

2. Identifier 0 (X1hex)
   This register sets the data length code and a part of the identifier.
   Set this register for each message.
   The bit configuration is as follows:



(1) Format setting: IDFM
The microcontroller sets the message format.
At reset, the IDFM bit is undefined.

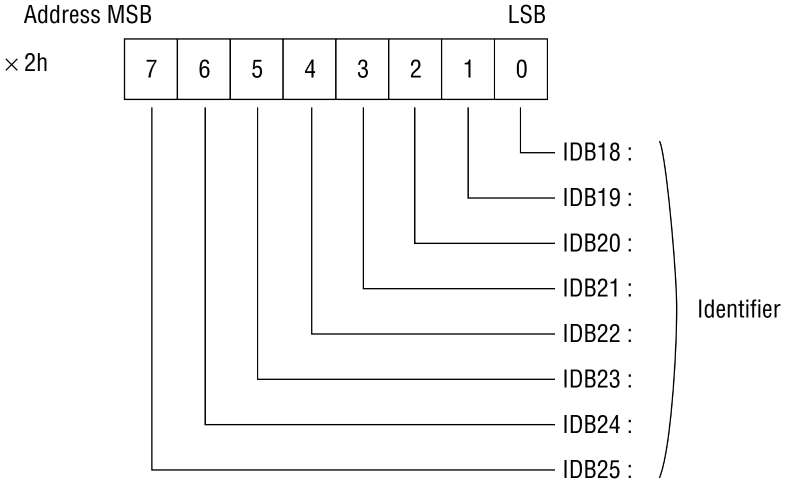
IDFM	Operation
0	Standard format (ID = 11 bits)
I	Extended format (ID = 29 bits)

(2) Data length code: DLC3 to DLC0
This is control field data to set the number of bytes of a data field. 0 to 8 can be set.
At reset, these bits are undefined.

(3) Identifier: IDB28 to IDB26
These bits set the identifier field.
For standard format (IDFM = 0), the higher 3 bits of the 11 bits are set.
For extended format (IDFM = 1), the higher 3 bits (ID28 to ID26) of the 29 bits (ID28 to ID0) are set.
At reset, these bits are undefined.

3. Identifier 1 (X2hex)

   This register sets the identifier.
   Set this register for each message.
   The bit configuration is as follows:



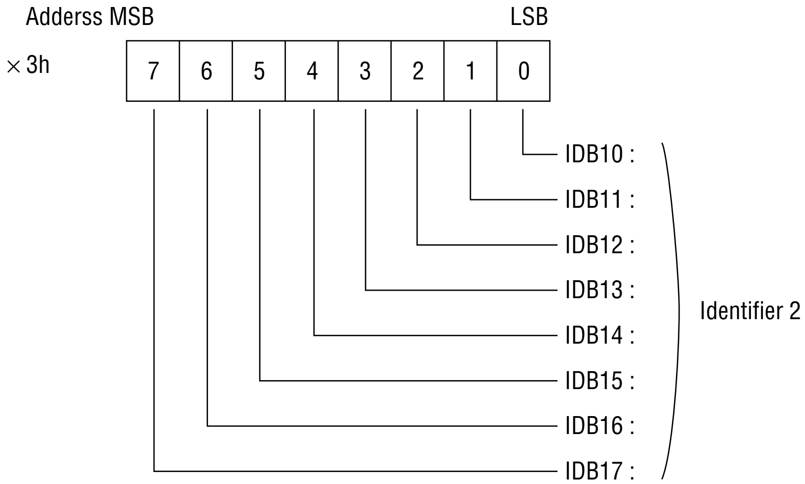
(1) Identifier: IDB25 to IDB18

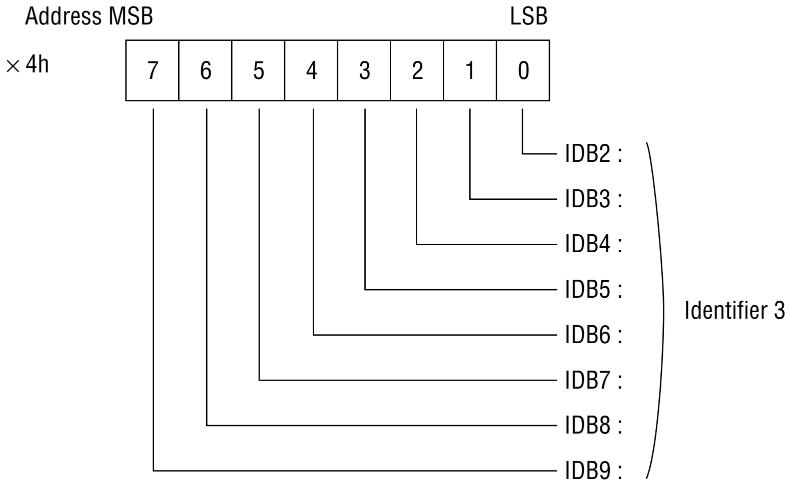
These bits set the lower 8 bits of the 11 bits of the identifier field.
For standard format (IDFM = 0), the lower 8 bits of the 11 bits are set.
For extended format (IDFM = 1), ID25 to ID18 of the 29 bits (ID28 to ID0) are set.
At reset, these bits are undefined.

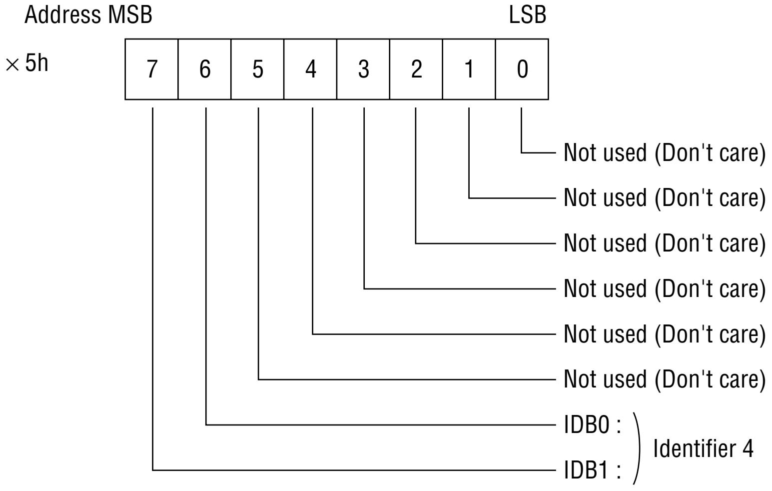
4. Address: X3 to XDhex

   For standard format (IDFM = 0), addresses X3 to XAhex become the transmission/receive data storage area.
   For extended format (IDFM = 1), addresses X3 to X5hex are used to set the identifier field, and addresses X6 to XDhex become the transmission/receive data storage area.
   For both, a maximum of 8 bytes of transmission/receive data can be stored, but the number of transmittable/receivable bytes must have been set by data length code.
   At reset, message content is undefined.

   The relationship between address and identifier bits for extended format (IDFM = 1) is as follows:


Adderss MSB
LSB
× 3h
7
6
5
4
3
2
1
0
IDB10 :
IDB11 :
IDB12 :
IDB13 :
IDB14 :
IDB15 :
IDB16 :
IDB17 :
Identifier 2


Address MSB
LSB
× 4h
7
6
5
4
3
2
1
0
IDB2 :
IDB3 :
IDB4 :
IDB5 :
IDB6 :
IDB7 :
IDB8 :
IDB9 :
Identifier 3

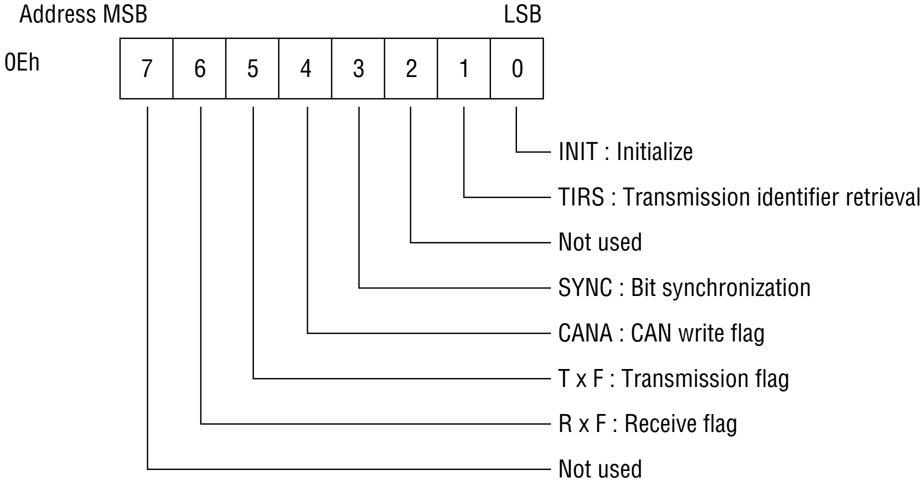

Address MSB
LSB
× 5h
7
6
5
4
3
2
1
0
Not used (Don't care)
Not used (Don't care)
Not used (Don't care)
Not used (Don't care)
Not used (Don't care)
Not used (Don't care)
IDB0 :
IDB1 :
Identifier 4

Control Register

These registers listed below control various operations of CAN.

Address	Symbol	Name
0EH	CANC	CAN control register
0FH	CANI	CAN interrupt control register
1EH	NMES	Number of message specification registers
1FH	BTR0	CAN bus timing register 0
2EH	BTR1	CAN bus timing register 1
2FH	TIOC	Communication input/output control register
3EH	GMR0	Group message register 0
3FH	GMR1	Group message register 1
4EH	GMSK00	Message mask register 00
4FH	GMSK01	Message mask register 01
5EH	GMSK02	Message mask register 02
5FH	GMSK03	Message mask register 03
6EH	GMSK10	Message mask register 10
6FH	GMSK11	Message mask register 11
7EH	GMSK12	Message mask register 12
7FH	GMSK13	Message mask register 13
8EH	STBY	Standby control register
8FH		Not used (reserve area)
9EH	TMN	Communication message number register
9FH	CANS	CAN status register
AEH	TEC	Transmission error counter
AFH	REC	Receive error counter
BEH		Not used (reserve area)
BFH		
CEH		
CFH		
DEH		
DFH		
EEH		
EFH		
FEH		
FFH		

1. CAN control register (CANC: 0Ehex)
   This register controls the operation of CAN.
   The bit configuration is as follows:



(1) Initialize: INIT
This bit is used to initialize the communication control area. To start initialization, write "1" to INIT, read INIT and confirm that INIT is "1", then initialize. To end initialization, write "0" to INIT, read INIT, and confirm that INIT is "0". For both, initialization mode is not set/cleared until the above procedure is executed.

If the INIT bit is set to "1" during the transmission or receive operation, the initialization will start after the communication completes.
When the INIT bit is set to "1", the communication operation stops but the error counter and data memory are held.

To initialize message memory, write the number of messages to be used to the number of messages setting register, NMES, then write the inside message control register, identifier 1, and identifier 2 sequentially from message 0 for all messages.
At reset, INIT is set to "1".

(2) Transmission identifier retrieval: TIRS
This bit is used to scan identifiers sequentially from message 0 to the last message of the message memory, detecting priority of the message for which the transmission request TRQ is "1" and starting to transmit the messages. TIRS will be set to "0" when there are no transmission request messages after scanning or transmitting.
At reset, TIRS is set to "0".

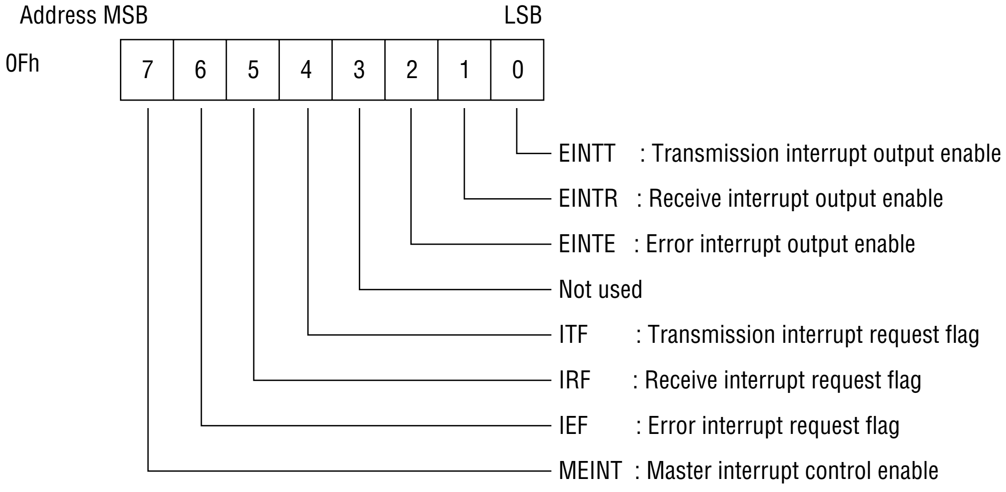
(3) Bit synchronization: SYNC
This bit is used to set the bit synchronization edge to synchronize at the CAN bus.
When SYNC is "0", the synchronization edge is set at the falling edge of data.
When SYNC is "1", the synchronization edge is set at both the rising and falling edges of data.
At reset, SYNC is set to "0".

(4) CAN write flag: CANA
This bit is used to indicate receive data write status to the message memory. CANA is "1" while CAN is writing receive data to the message memory.
This is a read-only flag.

(5) Transmission flag: TxF
This bit is used to indicate transmission operation status.
When TxF is "0", CAN is in transmission operation stop status.
When TxF is "1", CAN is in transmission operation status. TxF becomes "0" when transmission completes.
This is a read-only flag.

(6) Receive flag: RxF
This bit is used to indicate receive operation status.
When RxF is "0", CAN is in receive operation stop status.
When RxF is "1", CAN is in receive operation status. RxF becomes "0" when receiving completes.
This is a read-only flag.

2. CAN interrupt register (CANI: 0Fhex)
This register controls CAN interrupts.
The bit configuration is as follows:



(1) Transmission interrupt output enable: EINTT
This bit is used to output transmission interrupt signal INTT from interrupt pin $\overline{\text{INT}}$ when transmission completes.
When EINTT is "0", a transmission interrupt signal is not output from the interrupt pin.
When EINTT is "1", a transmission interrupt signal is output from the interrupt pin.
At reset, EINTT is set to "0".

(2) Receive interrupt output enable: EINTR
This bit is used to output receive interrupt signal INTR from interrupt pin $\overline{\text{INT}}$ when reception completes.
When EINTR is "0", a receive interrupt signal is not output from the interrupt pin.
When EINTR is "1", a receive interrupt signal is output from the interrupt pin.
At reset, EINTR is set to "0".

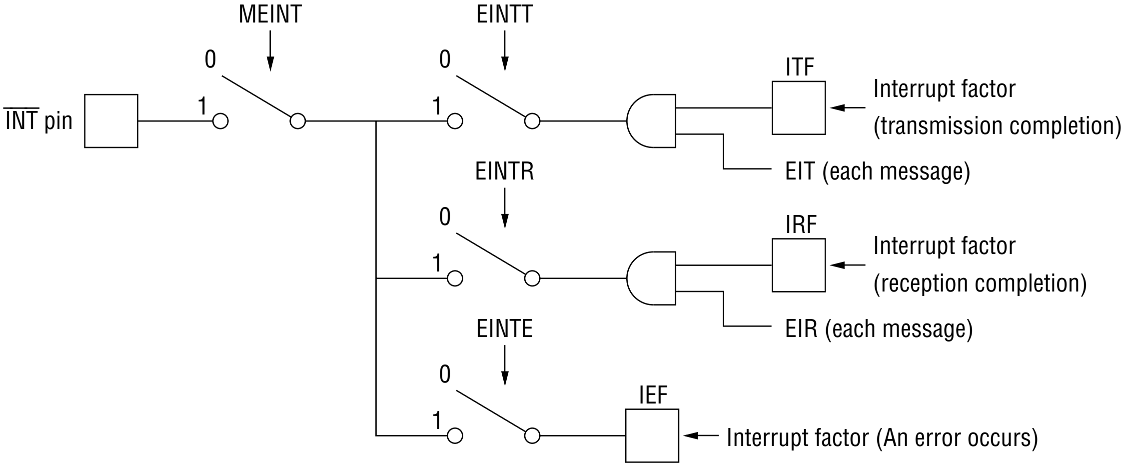
(3) Error interrupt output enable: EINTE
When an error occurs, this bit is used to output error interrupt signal INTE from interrupt pin $\overline{\text{INT}}$.
When EINTE is "0", an error interrupt signal is not output from the interrupt pin.
When EINTE is "1", an error interrupt signal is output from the interrupt pin.
At reset, EINTE is set to "0".

(4) Transmission interrupt request flag: ITF
ITF becomes "1" when a transmission interrupt is generated. Only "0" can be written to this bit.
At reset, ITF is set to "0".

(5) Receive interrupt request flag: IRF
IRF becomes "1" when a receive interrupt is generated. Only "0" can be written to this bit.
At reset, IRF is set to "0".

(6) Error interrupt request flag: IEF
IEF becomes "1" when an error occurs. Only "0" can be written to this bit.
At reset, IEF is set to "0".

(7) Master interrupt control enable: MEINT
This bit is used to set enable/disable of communication interrupts.
The flowchart of interrupt control is shown below.
When MEINT is "0", interrupt request control is disabled.
When MEINT is "1", interrupt request control is enabled.
At reset, MEINT is set to "0".



3. Number of messages specification register (NMES: 1Ehex)

   This is a register to set the number of messages to be stored in the message memory.
   A maximum of 16 messages can be set, with message numbers 0 to 15.
   Writing to NMES is enabled when initialize bit INIT of the CAN control register (CANC: OEhex) is "1".
   At reset, NMES is set to "0000".
   The bit configuration and relationship between message number and number of messages are as follows:

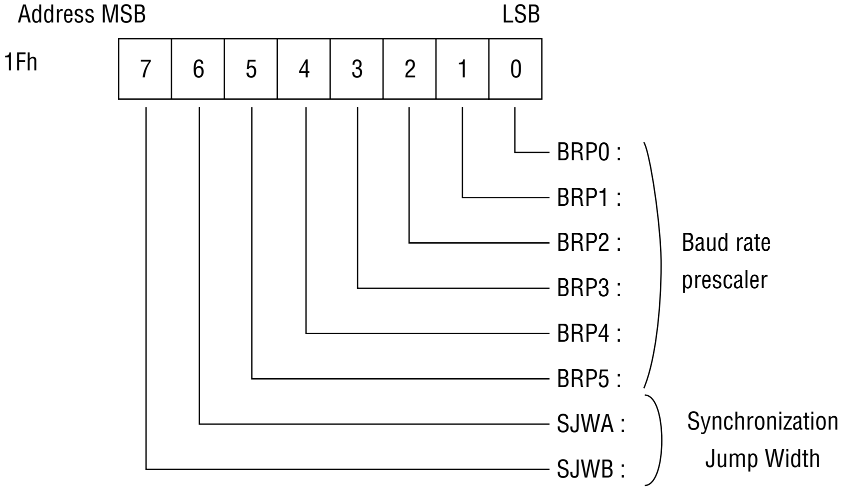
Address	MSB							LSB
1Eh	*	*	*	*	NMES3	NMES2	NMES1	NMES0

								Number of message
*	*	*	*	0	0	0	0	1
*	*	*	*	0	0	0	0	2
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
*	*	*	*	1	1	1	0	15
*	*	*	*	1	1	1	1	16

* : Don't Care

4. CAN bus timing register 0 (BTR0: 1Fhex)

   This register sets the baud rate prescaler and synchronization jump width (SJW) used for bus timing. Writing to the BTR0 bit is enabled, when the INIT bit of the CAN control register (CANC: 0Ehex) is "1".
   The bit configuration is as follows:



(1) Baud rate prescaler: BRP5 to BRP0

This is a 6-bit prescaler to set the BTL cycle time and SJW of the basic clock for communication operation.

The relationship between the bit content and BTL is as follows:

At reset, BRP5 to BRP0 are set to "000000".

BRP5	**BRP4**	**BRP3**	**BRP2**	**BRP1**	**BRP0**	**BTL cycle time**
0	0	0	0	0	0	1X system clock cycle
0	0	0	0	0	1	2X system clock cycle
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1	1	1	1	1	0	63X system clock cycle
1	1	1	1	1	1	64X system clock cycle

The BTL cycle time is given by the following operation.

$$\text{BTL cycle time} = 2 \times (2^5 \times \text{BRP5} + 2^4 \times \text{BRP4} + 2^3 \times \text{BRP3} + 2^2 \times \text{BRP2} + 2^1 \times \text{BRP1} + \text{BRP0} + 1)/f_{OSC}$$

*) System clock is 1/2 division of oscillation frequency.

f_{OSC} is the oscillation frequency.

(2) SJW: SJWA, SJWB

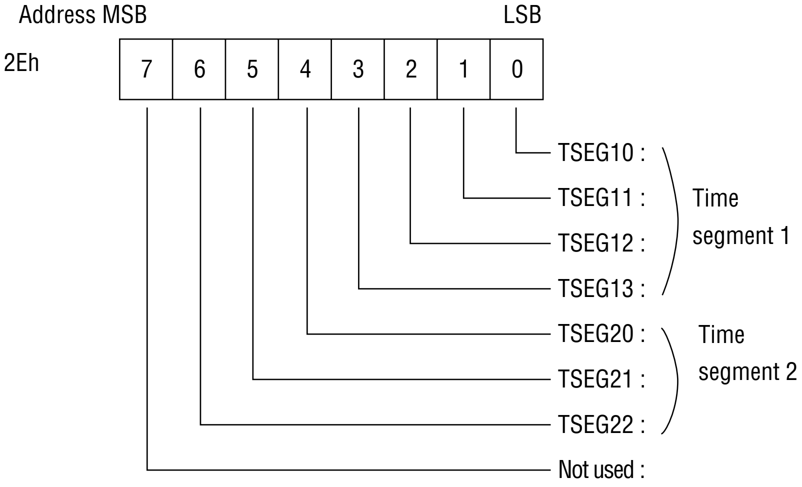
This is a 2-bit register to set SJW.

The relationship between bit content and SJW is as follows:

At reset, SJWA and SJWB are set to “00”.

SJWB	SJWA	SJW1, SJW2
0	0	1 × BTL cycle
0	1	2 × BTL cycle
1	0	3 × BTL cycle
1	1	4 × BTL cycle

5. CAN bus timing register 1 (BTR1: 2Ehex)
   This register sets the sampling count, sampling point and transmit point used for bus timing. Writing to the BTR1 bit is enabled, when the INIT bit of the CAN control register (CANC: 0Ehex) is "1".
   The bit configuration is as follows:



(1) Time segment 1: TSEG13 to TSEG10
This is a 4-bit register to set the sampling point.
The relationship between bit content and TSEG1 is as follows:
At reset, TSEG13 to TSEG10 are set to "0000".

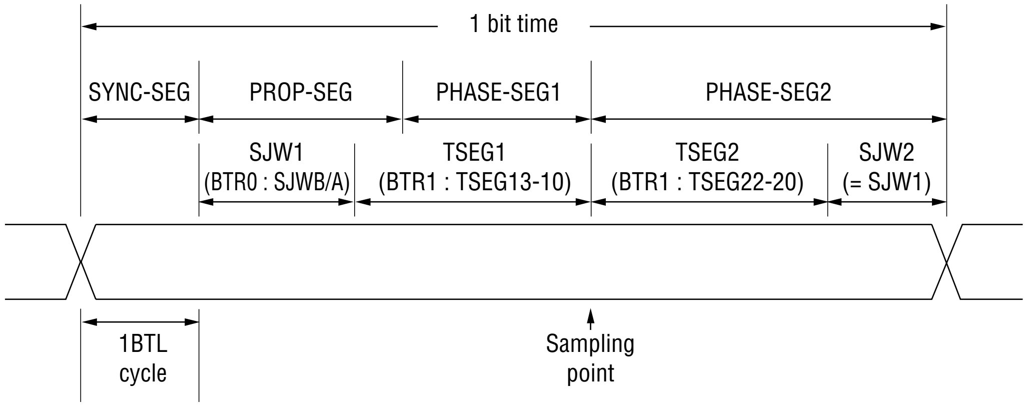
TSEG13	TSEG12	TSEG11	TSEG10	TSEG1
0	0	0	0	1 × BTL cycle
0	0	0	1	2 × BTL cycle
. .	. .	. .	. .	. .
1	1	1	0	15 × BTL cycle
1	1	1	1	16 × BTL cycle

(2) Time segment 2: TSEG22 to TSEG20
This is a 3-bit register to set the transmit point.
The relationship between the bit content and TSEG2 is as follows:
At reset, TSEG22 to TSEG20 are set to "000".

TSEG22	TSEG21	TSEG20	TSEG2
0	0	0	1 × BTL cycle
0	0	1	2 × BTL cycle
.	.	.	.
.	.	.	.
1	1	0	7 × BTL cycle
1	1	1	8 × BTL cycle

(3) Bit timing

Bit timing is set by CAN bus timing registers 0 and 1. The relationship between 1 bit time of a message and a CAN bus timing (the MSM9225 register) is as follows:



If setting is :

BTR0 = "01000001" ...SJWB = "0" SJWA = "1" BRP5-0 = "000001"
BTR1 = "00000001"...TSEG2 = "000" TSEG1 = "0001"

then the bit timing is as follows

Sync segment	1 BTL cycle (fixed)
SJW 1	2 BTL cycle
TSEG 1	2 BTL cycle
TSEG 2	1 BTL cycle
SJW 2	2 BTL cycle
1 bit time	8 BTL cycle

Sampling point = 5 BTL cycle

If f_{osc} = 16 MHz, then 1 BTL cycle is :

$$\text{BTL cycle} = 2 \times (2^5 \times 0 + 2^4 \times 0 + 2^3 \times 0 + 2^2 \times 0 + 2^1 \times 0 + 1 + 1) / 16\ \text{MHz} = 0.25\ \mu\text{s}$$

Therefore 1 bit time is :

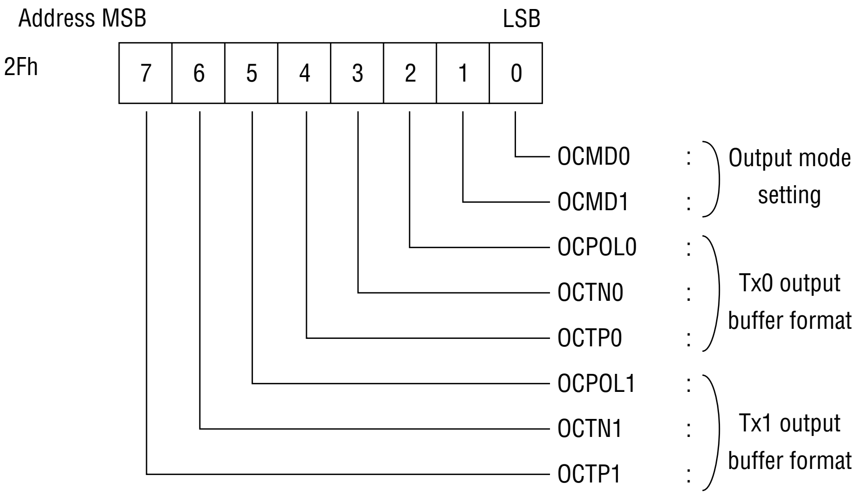
8 BTL cycle $= 8 \times 0.25\ \mu\text{s} = 2.0\ \mu\text{s}$
(= 500 Kb/s)

6. Communication input/output control register (TIOC: 2Fhex)

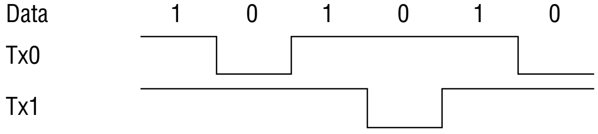
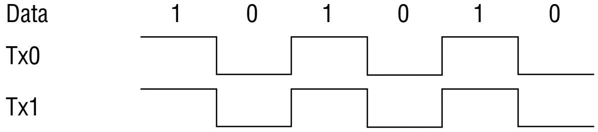
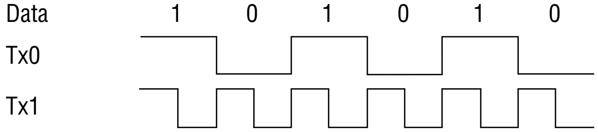
This register sets the communication mode and output buffer format.

Writing to the TIOC bit is enabled, when the INIT bit of the CAN control register (CANC: 0Ehex) is "1".

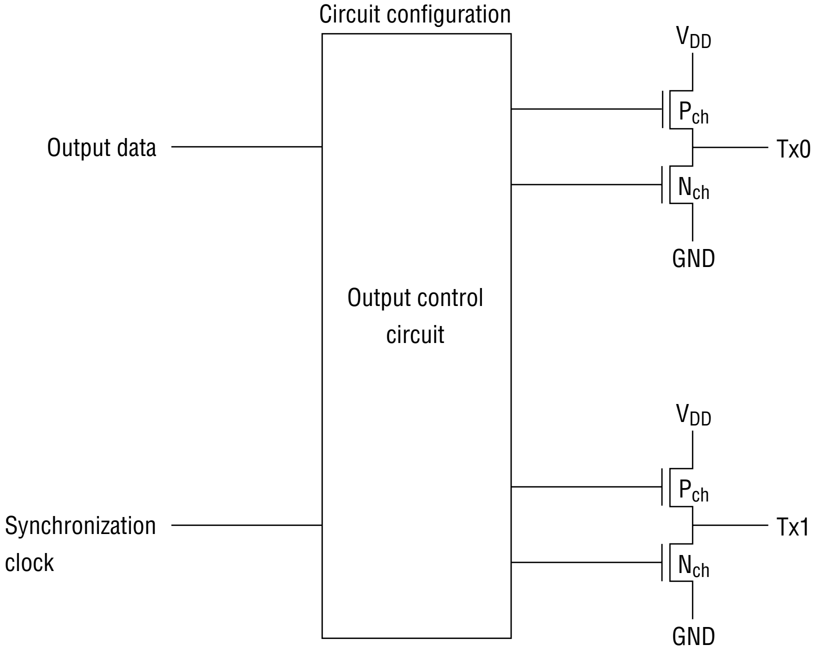
The bit configuration is as follows:



(1) Time segment 1: OCMD1 to OCMD0
These bits are used to set the output mode of output pins Tx0 and Tx1.
The relationship between the bit content and output mode is as follows:
At reset, OCMD1 to OCMD0 are set to "00".

OCMD1	OCMD0	Output mode of Tx0 and Tx1
0	0	[Double layer mode] Transmission data "0" is output from Tx0 and Tx1 altermately. Output example 
0	1	[Disabled]
1	0	[Single layer mode] Same bit string data is output from both Tx0 qnd Tx1. Output example 
1	1	[Clock output mode] Bit string data is output from Tx0. Synchrinization clock is output from Tx1. Output example 

(2) Output driver format setting: OCPOL, OCTN, OCTP
OCPOL is used to set the polarity of output.
OCTN is used to set the open/drain mode of the Nch transistor of the output driver.
OCTP is used to set the open/drain mode of the Pch transistor of the output driver.
The circuit configuration of the output driver and the relationship between bit content and output driver format are as follows:
At reset, all bits are set to "0".



Output driver format

Mode	OCTP	OCTN	OCPOL	Output data	Pch Tr	Nch Tr	Tx pin output level
Floating	0	0	0	0	off	off	Floating
	0	0	0	1	off	off	Floating
	0	0	1	0	off	off	Floating
	0	0	1	1	off	off	Floating
Pulldown	0	1	0	0	off	on	"0"
	0	1	0	1	off	off	Floating
	0	1	1	0	off	off	Floating
	0	1	1	1	off	on	"0"
Pullup	1	0	0	0	off	off	Floating
	1	0	0	1	on	off	"1"
	1	0	1	0	on	off	"1"
	1	0	1	1	off	off	Floating
Push-pull	1	1	0	0	off	on	"0"
	1	1	0	1	on	off	"1"
	1	1	1	0	on	off	"1"
	1	1	1	1	off	on	"0"

7. Group message register (GMR0: 3Ehex, GMR1: 3Fhex)
   These are registers to set the group message mode.
   Two messages can be set to the group message mode.
   At reset, all bits are set to "0".
   The group message mode is valid when the EGM0/EGM1 bit is "1".
   Using GMR03 to GMR00 and GMR13 to GMR10, set the message numbers of messages that are to be set to the group message mode.
   The bit configuration is as follows:

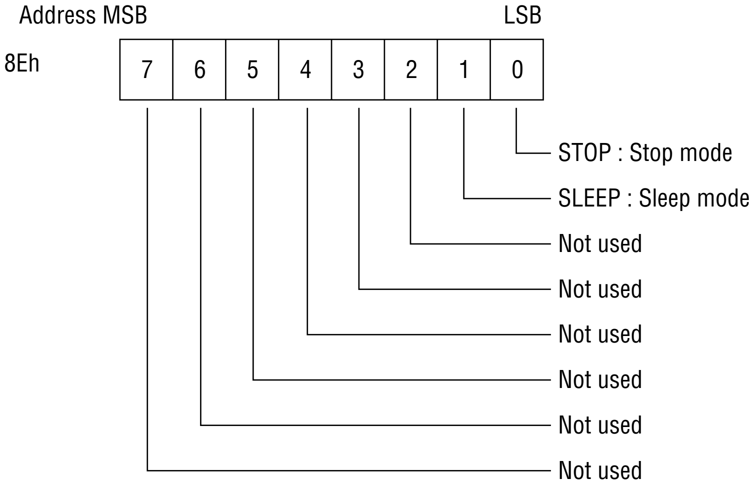
Address	MSB							LSB	
3Eh	EGM0	0	0	0	GMR03	GMR02	GMR01	GMR00	GMR0
3Fh	EGM1	0	0	0	GMR11	GMR12	GMR11	GMR10	GMR1

8. Group message mask register (GMSK)
   This is a register to judge identifiers when a message with a message number specified by the group message mode GMR is received.
   Using MiID28 to MiID0, set the bits to mask the identifier of a message set by the GMR bit.
   Setting "1" masks the bit, setting "0" does not mask the bit.
   (M0ID28 to M0ID0 are for GMR0, and M1ID28 to M1ID0 are for GMR1.)
   At reset, all bits are set to "0".
   The bit configuration is as follows:

Address	MSB							LSB	
4Eh	M0ID28	M0ID27	M0ID26	M0ID25	M0ID24	M0ID23	M0ID22	M0ID21	GMSK00
4Fh	M0ID20	M0ID19	M0ID18	M0ID17	M0ID16	M0ID15	M0ID14	M0ID13	GMSK01
5Eh	M0ID12	M0ID11	M0ID10	M0ID9	M0ID8	M0ID7	M0ID6	M0ID5	GMSK02
5Fh	M0ID4	M0ID3	M0ID2	M0ID1	M0ID0	0	0	0	GMSK03

Address	MSB							LSB	
6Eh	M1ID28	M1ID27	M1ID26	M1ID25	M1ID24	M1ID23	M1ID22	M1ID21	GMSK10
6Fh	M1ID20	M1ID19	M1ID18	M1ID17	M1ID16	M1ID15	M1ID14	M1ID13	GMSK11
7Eh	M1ID12	M1ID11	M1ID10	M1ID9	M1ID8	M1ID7	M1ID6	M1ID5	GMSK12
7Fh	M1ID4	M1ID3	M1ID2	M1ID1	M1ID0	0	0	0	GMSK13

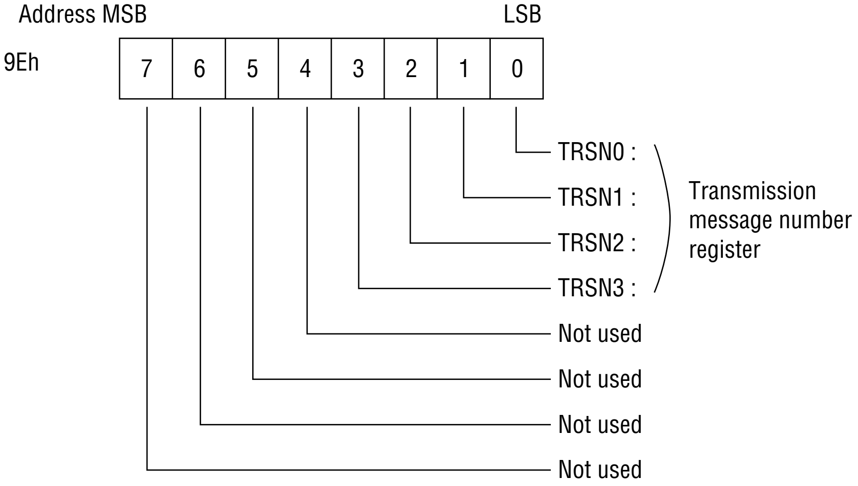
9. Standby control register (STBY: 8Ehex)
This register sets various modes, such as stop mode.
The bit configuration is as follows:



(1) Stop mode: STOP
If STOP is set to "1", the MSM9225 will enter the stop mode when the CAN bus is idle. In stop mode, the content of data memory is held but the oscillator and all circuits stop to save power consumption. Access to/from external units is therefore disabled.
Stop mode is cleared by a reset signal input from the $\overline{\text{RESET}}$ pin or $\overline{\text{CS}}$ pin = "0".
At reset, STOP is set to "0".

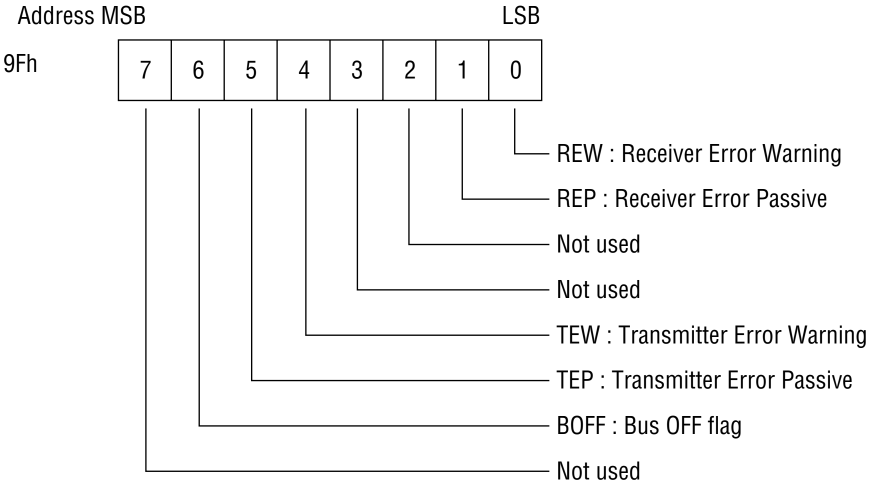
(2) Sleep mode: SLEEP
If SLEEP is set to "1", the MSM9225 will enter the sleep mode when the CAN bus is idle. In sleep mode, the content of data memory is held and the differential input of Rx0 and Rx1 operates, but the oscillator and other circuits stop operation. Access to/from external units is therefore disabled.
Sleep mode is cleared by a reset signal input from the $\overline{\text{RESET}}$ pin or $\overline{\text{CS}}$ pin = "0", or by the differential input of Rx0 and Rx1.
When both stop mode and sleep mode are set at the same time, the MSM9225 enters stop mode.
At reset, SLEEP is set to "0".

10. Communication message number register (TMN: 9Ehex)
    The communication message number is recorded in this register.
    The bit configuration is as follows:



(1) Transmission message number register: TRSN3 to TRSN0
This is a register to store the message number when a message is transmitted/received. When transmission completes, the transmitted message number is stored. When receiving completes, the received message number is stored. And when an error occurs, the message number of the message being transmitted/received at that time is stored. This is a read-only register and is set to "0000" at reset.

11. CAN status register (CANS: 9Fhex)
This is a status register to indicate the status of CAN.
Bit6 to bit4 are flags for the transmitter and bit1 and bit0 are for the receiver, and this register is read only.
The bit configuration is shown below.



(1) Receiver Error Warning: REW
When the Receiver Error Counter (REC) ≥ 96, REW becomes "1". If REW = "1", the bus may be seriously damaged. The bus must be tested for this condition.
At reset or when REC < 96, REW becomes "0".

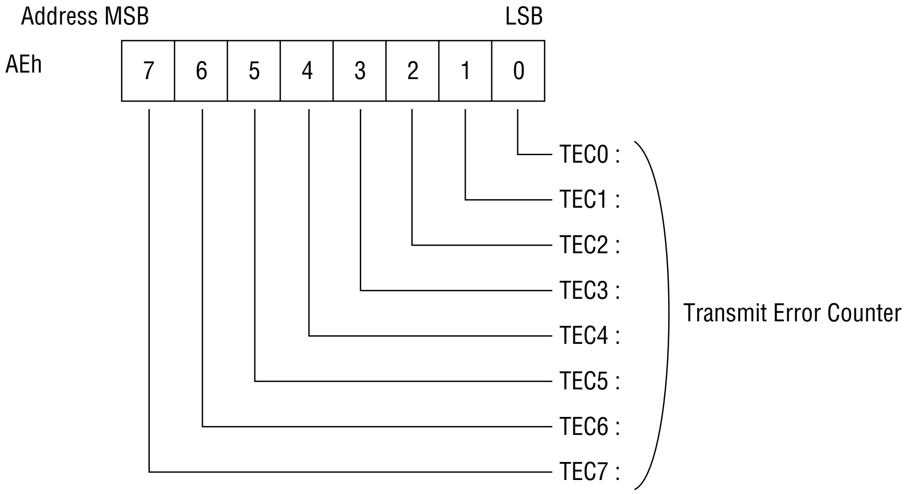
(2) Receiver Error Passive: REP
When the Receive Error Counter (REC) ≥ 128, REP becomes "1".
At reset or when REC < 128, REP becomes "0" (error active)

(3) Transmitter Error Warning: TEW
When the Transmit Error Counter (TEC) ≥ 96, TEW becomes "1".
If TEW = "1", the bus may be seriously damaged. The bus must be tested for this condition.
At reset or when TEC < 96, TEW becomes "0".

(4) Transmitter Error Passive: TEP
When the Transmit Error Counter (TEC) > 128, TEP becomes "1".
At reset or when TEP < 128, TEP becomes "0".

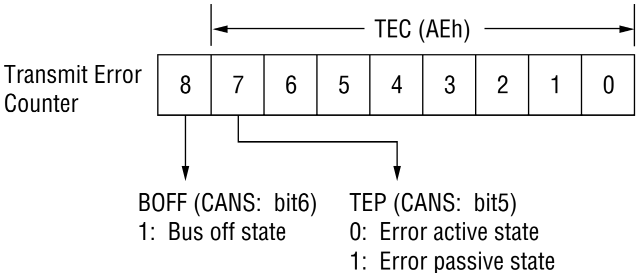
(5) Bus OFF: BOFF
This flag indicates the CAN bus status.
When the Transmit Error Counter (TEC) > 256 BOFF becomes "1" and the CAN bus is in the BUS OFF state.
At reset or when TEP < 256, BOFF becomes "0".

12. Transmit Error Counter (TEC: AEhex)
    TEC indicates the lower 8 bits of the 9-bit Transmit Error Counter.
    The bit configuration is shown below.

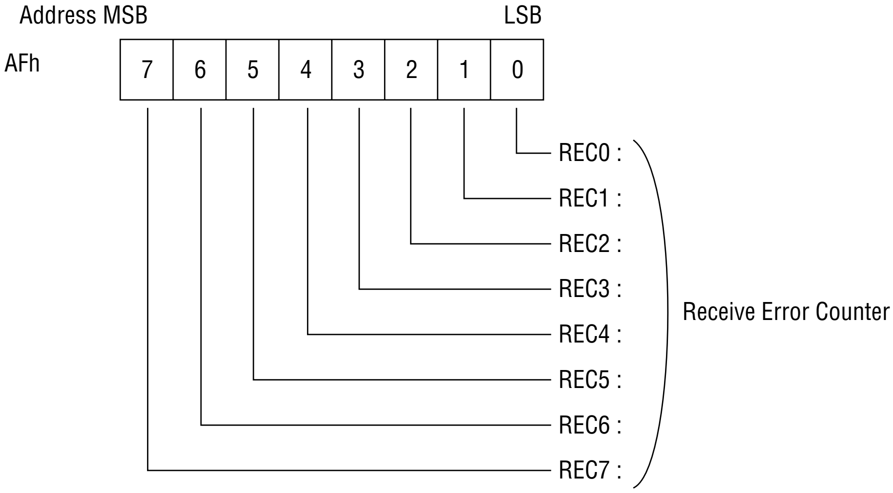


At reset, TEC is set to "0000 0000".

The relation between the Transmit Error Counter and TEC is shown below.

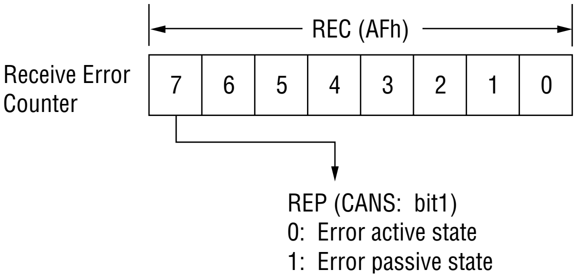


13. Receive Error Counter (REC: AFhex)
    The Receive Error Counter is read-only.
    The bit configuration is shown below.



At reset, REC is set to "0000 0000".

The relation between the Receive Error Counter and each register is shown below.



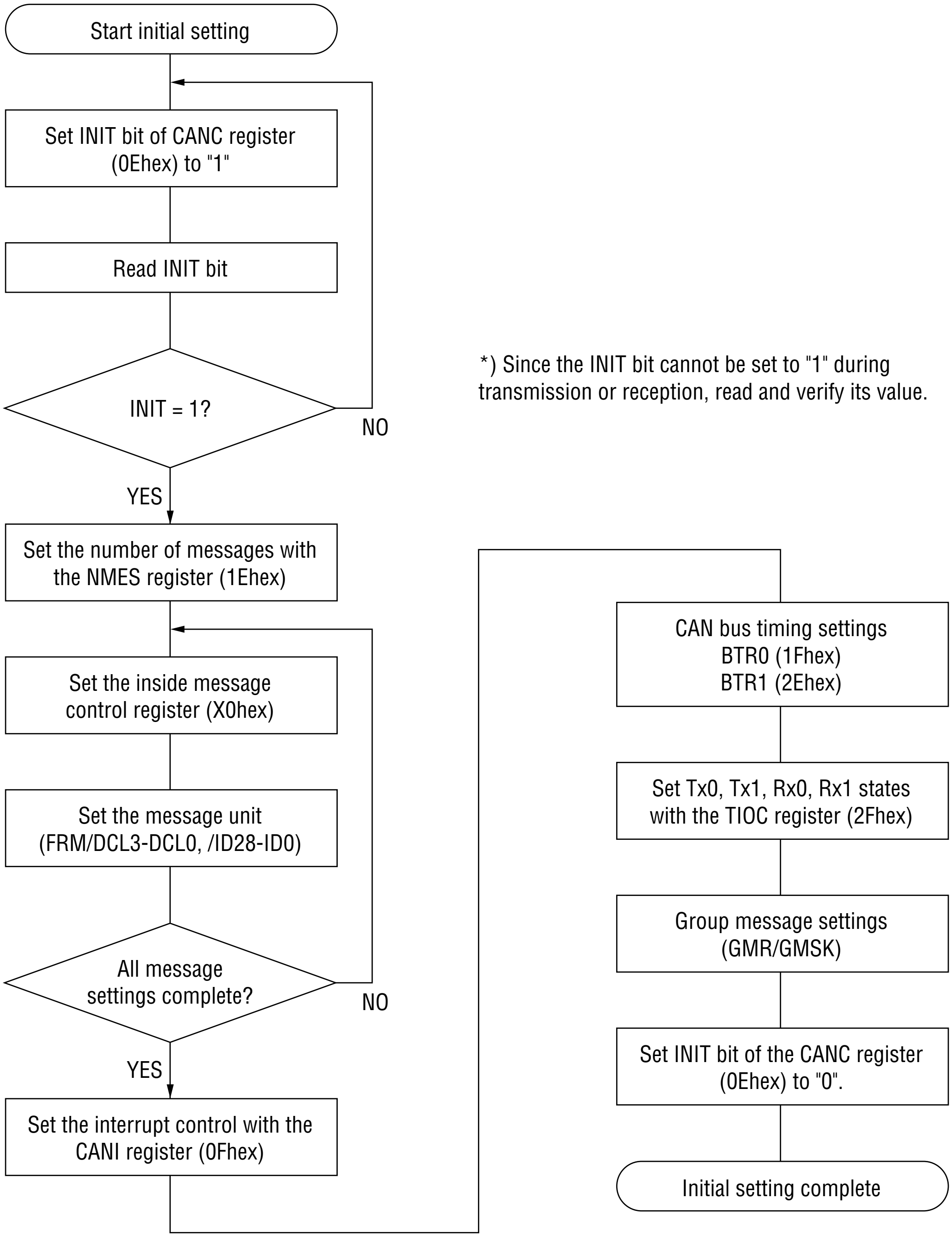
OPERATIONAL DESCRIPTION

MSM9225 operation is described below.

Operational Procedure

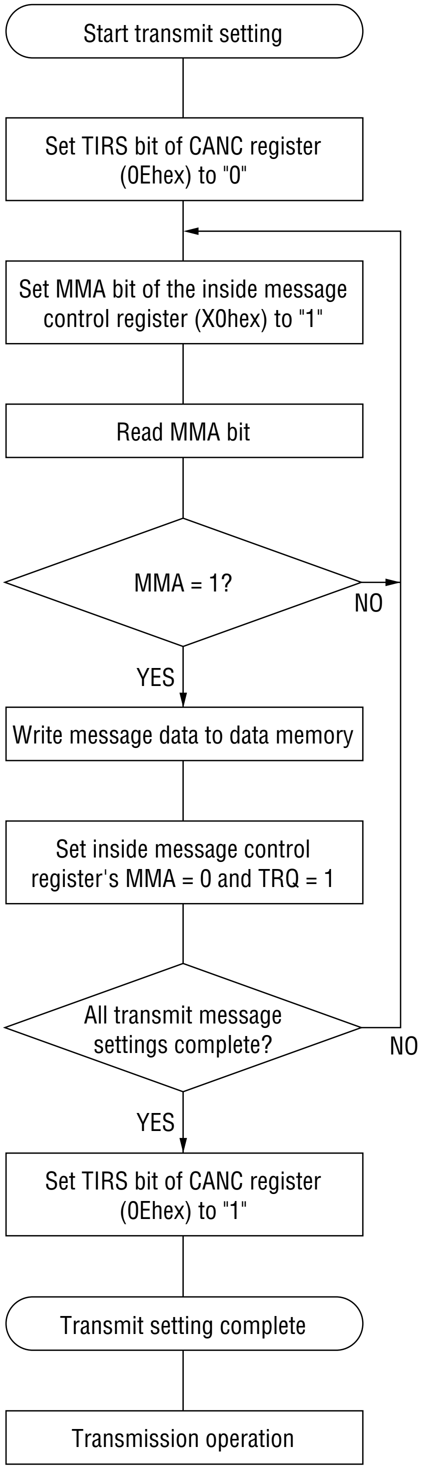
Procedures to set and operate various communication protocols are indicated below.

1. Initial setting
   The initial setting procedure is indicated below.



*) Since the INIT bit cannot be set to "1" during transmission or reception, read and verify its value.

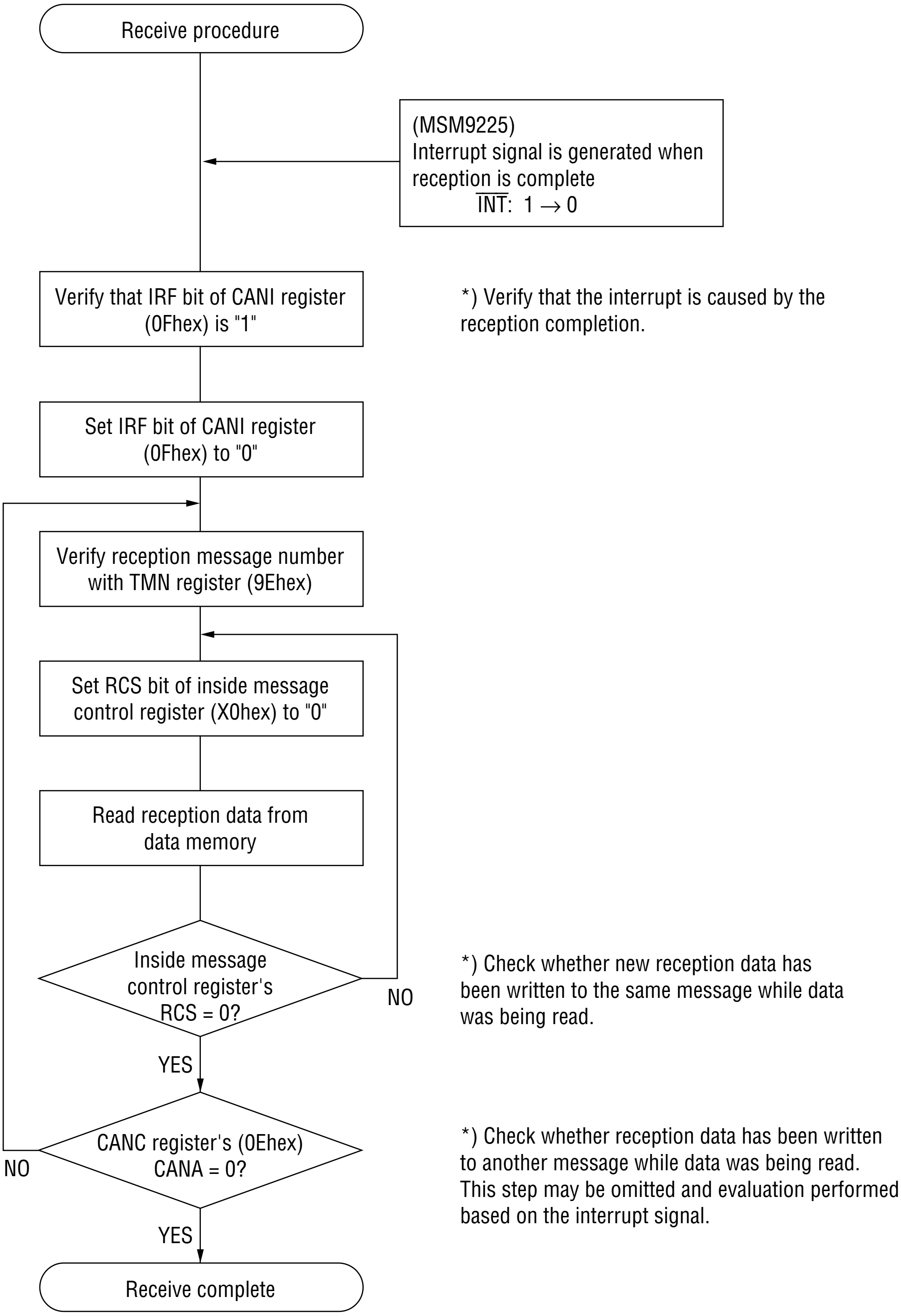
2. Transmit Procedure
   The transmit procedure is indicated below.



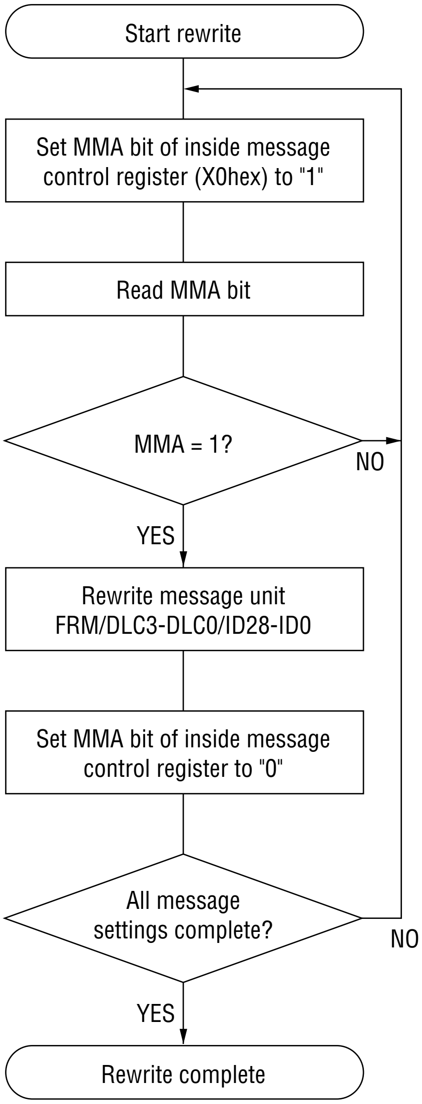
*) Since the MMA bit cannot be set to "1" while the message is being accessed, read and verify its value.

3. Receive Procedure

   The receive procedure is indicated below.



4. Message unit rewrites during operation
   The procedure to rewrite the IDentifier (ID) and Data Length Code (DLC) during operation is indicated below. The number of messages set in the NMES register at the initial setting is the number of (valid) messages that may be rewritten.



5. Remote Frame Operation
The following two methods are available for transmission after remote frame reception.
(1) Automatically transmit message data that has been previously set
(2) Set message data and then transmit

5-1. Automatic response
After remote frame reception, this method automatically transmits previously set message data.
Settings of the inside message control register are listed in the table below.

	Bit	**Symbol**	**Value**	**Comments**
Inside message control register	5	TRQ	0*	When reception is complete, TRQ bit changes from 0 → 1
	3	EIR	—	
	2	EIT	1	Set transmit interrupt to verify the end of transmission.
	1	FRM	0	Set the remote frame.
	0	ARES	1	Set automatic response.

A flow chart of the operation is shown on the following page.

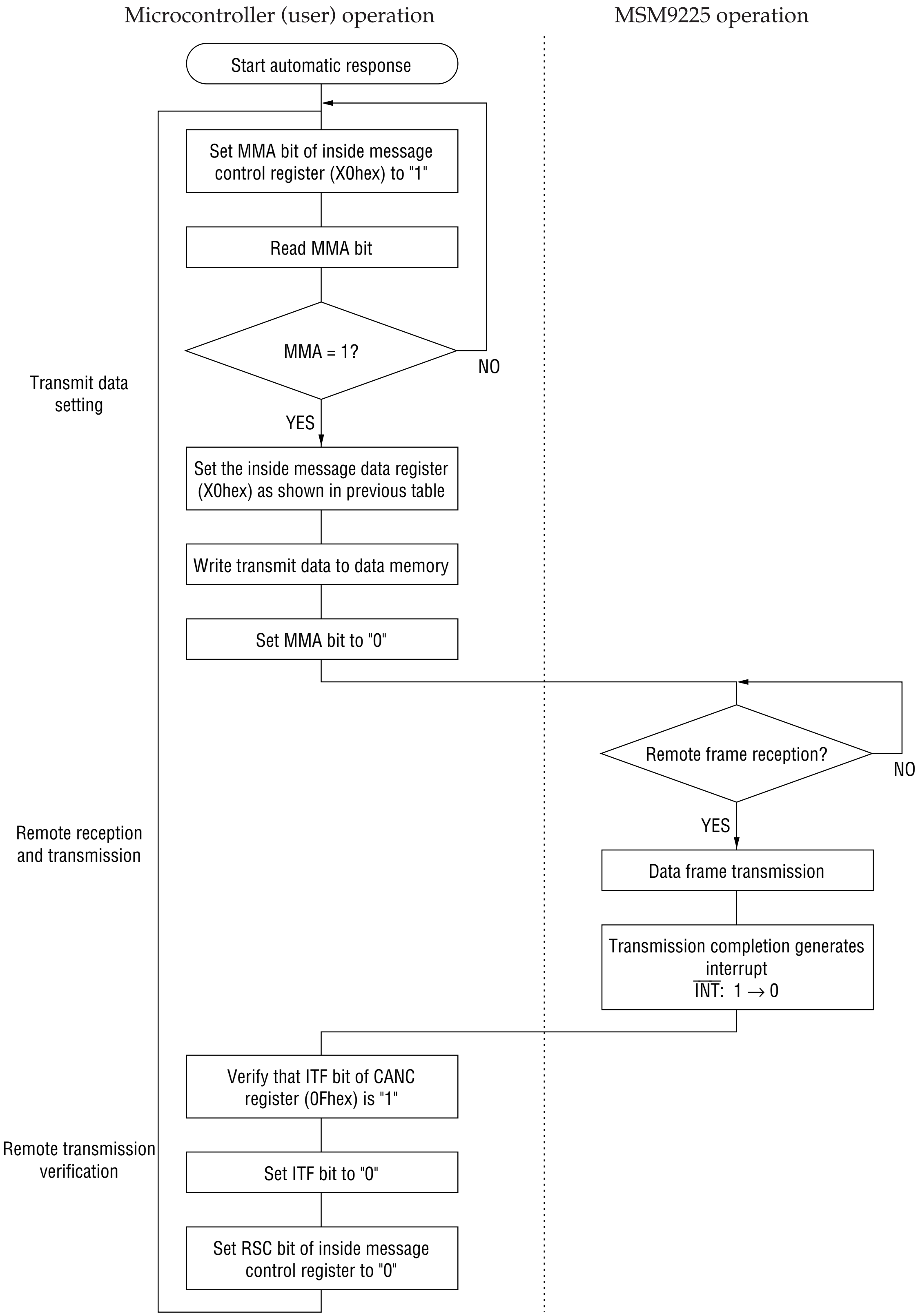


Figure: Automatic Response Operation Flow Chart

5-2. Manual response

In this method, after remote frame reception, the transmit data is set and then transmission begins.

Settings of the inside message control register are listed in the table below.

	Bit	**Symbol**	**Value**	**Comments**
Inside message control register	5	TRQ	0	Set to receive message.
	3	EIR	1	Set interrupt to verify (remote frame) reception.
	2	EIT	1	Set interrupt to verify the end of transmission.
	1	FRM	0	Set the remote frame.
	0	ARES	0	Specify that there will be no automatic response.

A flow chart of the operation is shown on the following page.

The basic operation is a combination of receive and transmit procedures.

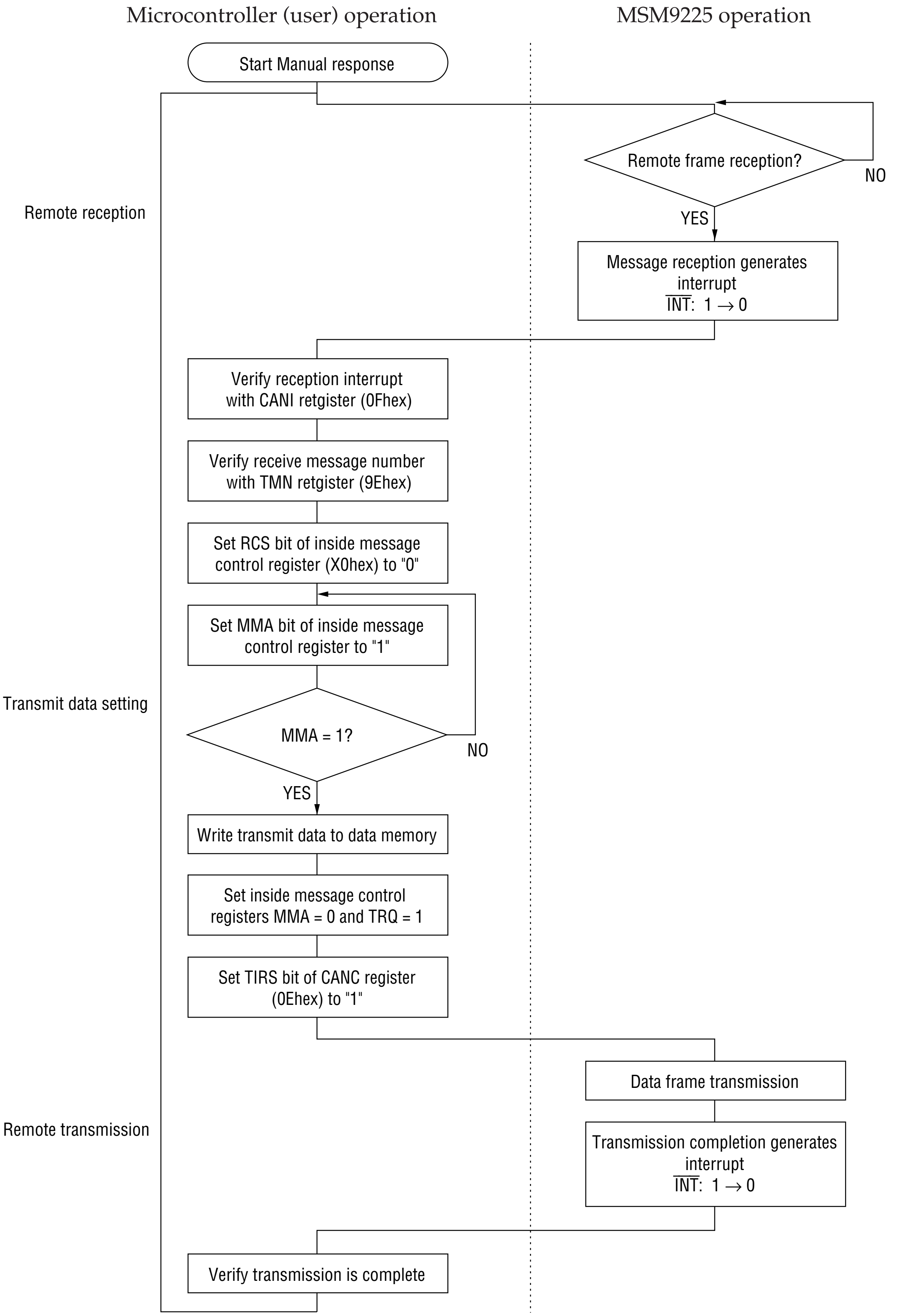


Figure: Manual Response Operation Flow Chart

Operation at Receiving Message

1. Priority of message

A message has the priority determined by the identifier setting. To determine priority, identifiers of messages are compared from the higher bit, and the identifier (set to "0") detected first has the higher priority. (see the example below)

Identifier (example)											Priority
0	0	0	0	1	1	1	0	1	0	1	Second
0	0	0	0	1	0	1	0	1	0	1	First
1	0	0	0	0	0	0	0	1	0	1	Fourth
0	0	1	0	0	1	0	1	1	0	1	Third

In this example, priority is determined at the shaded bits.

2. Data length code

When the received data length code (hereafter DLC) matches the DLC being set to the message memory, the number of bytes of data indicated by the received DLC is received and written to the message memory. When the received DLC does not match with the DLC being set to message memory, the MSM9225 operates as follows:

(1) Received DLC > DLC on message memory
The number of bytes of data indicated by the DLC on the message memory is received and written to the message memory.
The data exceeding the number of bytes indicated by DLC on the memory is not written to message memory.

(2) DLC on message memory > received DLC
The number of bytes of data indicated by the received DLC is received and written to the message memory.

3. Group message function

If the group message function is used, a part of an identifier can be masked. This can increase the number of receivable identifiers.

To use the group message function, set the message number of the target message to set the group message function at the GMR register. Then set the bits to be masked at the GMSK register. Depending on the location of bits to be masked, an another identifier being set at the message memory may be received.

In this case, the priority of identifiers being set on the message memory is calculated and the identifier having the highest priority is received. The received data is written to the message memory indicated by the message for which the identifier with the highest priority is set.

When same identifiers are set to multiple messages on message memory

When same identifiers are set to multiple messages on the message memory, operations are as follows.

1. Transmit operation
   Messages are transmitted sequentially from the smaller message number.

2. Receive operation
   The message is always written to the smallest message number.
   For example, the same identifier is set at message numbers 1 to 4, as shown below.

Message number	Identifier (example)										
0	0	0	0	0	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	0	0	1
2	1	1	1	0	0	1	1	1	0	0	1
3	1	1	1	0	0	1	1	1	0	0	1
4	1	1	1	0	0	1	1	1	0	0	1
5	0	0	0	0	0	0	0	0	1	1	1
6	1	0	0	0	0	0	0	0	0	1	1

The range in which the same identifier is set.

- Transmit operation
  If every message above is a transmit message, messages are transmitted sequentially in the order of message number 5 → 0 → 6 → 1 → 2 → 3 → 4.

- Receive operation
  When the identifier "11100111001" is received from the CAN bus, received data is always written to the message memory which is indicated by the message number 1.

MICROCONTROLLER INTERFACE

There are basically two methods of interfacing to the microcontroller.
(1) Synchronous serial interface (serial mode)
(2) Parallel bus interface (parallel mode)
Each interface is selected with the Mode0 and Mode1 pins. Refer to the section, PIN DESCRIPTIONS, "PIN DESCRIPTIONS" for the relation between pin values and interface selection.

Serial Interface

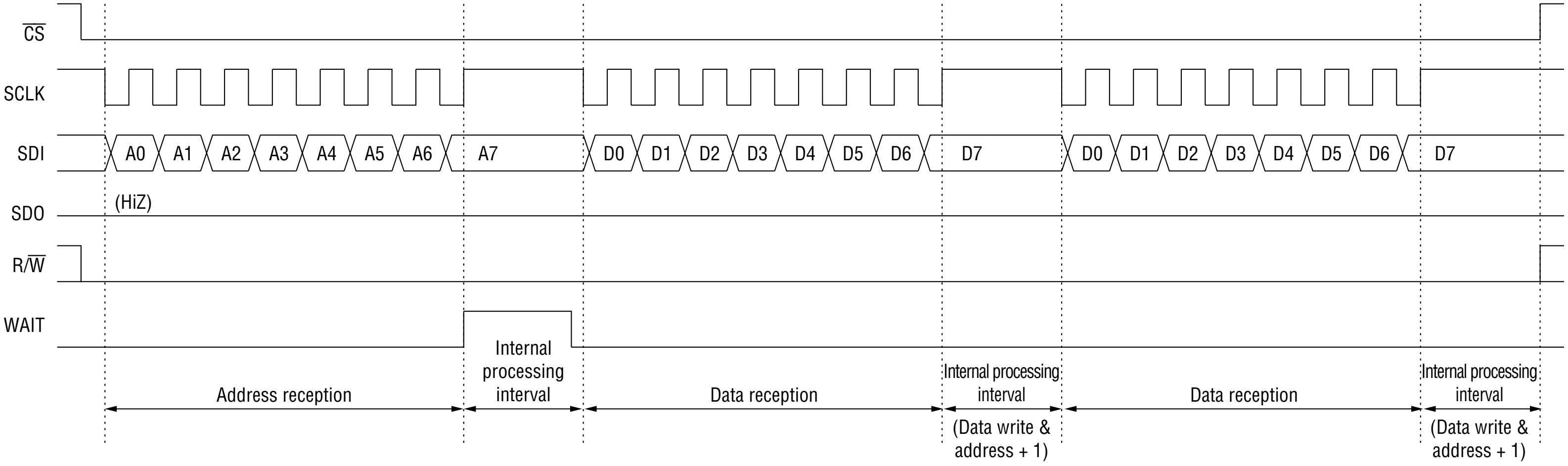
The transfer timing is indicated in the figure.
Address/data transfers begin when the $\overline{\text{CS}}$ pin is at a "L" level and end when it changes to a "H" level. Because the MSM9225 has an address increment function, the basic transfer consists of "1 address + multiple data." Therefore, to access a nonconsecutive address, the $\overline{\text{CS}}$ must be first pulled to a "H" level, and then the address reset.
Perform address/data transfers LSB first, in 8-byte units. During a transfer, an interval (WAIT) is necessary between address and data and between consecutive data transfers. (Refer to the section, ELECTRICAL CHARACTERISTICS, for interval values.) Note that the WAIT signal is only generated during the interval between address and data transfers.

(1) Data write
Data write operations are performed with the follwing procedure.
After setting the $\overline{\text{CS}}$ pin and $\overline{\text{PRD}}$/SR$\overline{\text{W}}$ pin to "L" levels, input an address to the SDI pin. Synchronized to the rising edge of synchronous clock SCLK, the MSM9225 captures the address in an internal register. When 8 SCLK clocks are received, the MSM9225 loads the address into the internal address counter and waits for data reception.
Next, input data to the SDI pin. An internal register captures data in a similar manner to the address capture, at the rising edge of SCLK. When 8 bits of data have been captured, the MSM9225 writes the data to the internal memory or register specified by the address that was received previously, and then increments the counter by 1. If data is to be written to consecutive addresses, continue the data transfer. After all data has been transferred, set the $\overline{\text{CS}}$ pin to a "H" level.

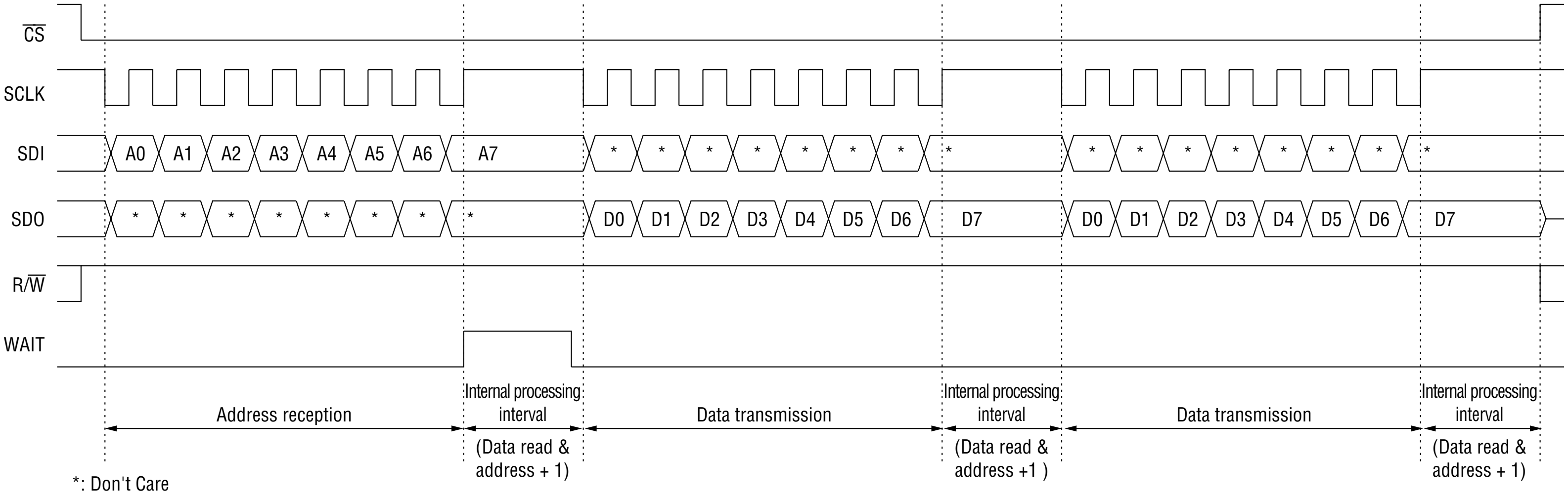
(2) Data read
Data read operations are performed with the following procedure.
After setting the $\overline{\text{CS}}$ pin to a "L" level and the $\overline{\text{PRD}}$ SR$\overline{\text{W}}$ pin to a "H" level, in the same manner as for the data write operation, input an address to the SDI pin. When 8 SCLK clocks are received, the MSM9225 loads the address into the internal address counter, reads data from the internal memory or register specified by the address, latches data into a shift register for data output and increments the address counter. Then, when SCLK is input, latched data is output from the SDO pin synchronized to the falling edge of SCLK. At this time, the contents of the data input from the SDI pin does not matter. If there exists remaining data to be read, input another 8 SCLK clocks. After all the data (at consecutive addresses) has been read, set the $\overline{\text{CS}}$ pin to a "H" level.

If the count value overflows (exceeds XFh), without changing the upper 4 bits of the address, the address increment function will reset the count value of the lower 4 bits to 0, and will continue counting.

(1) Data write timing



(2) Data read timing



*: Don't Care

Figure: Serial Interface Transfer Timing

Parallel Interface

The following three types of parallel interfaces are available.

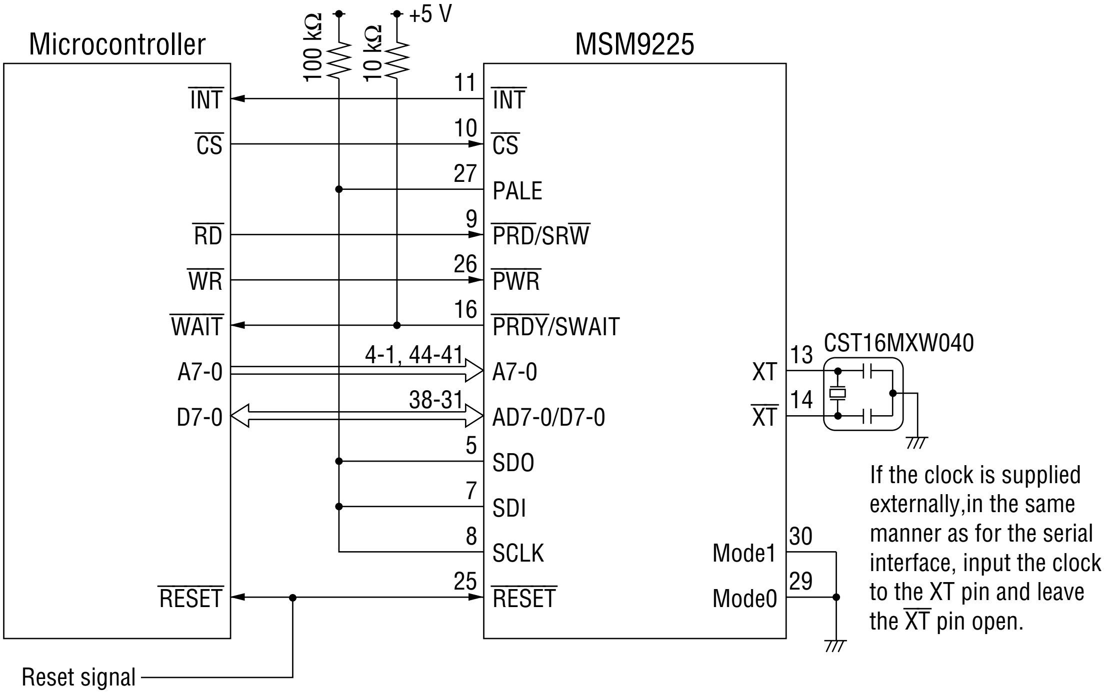
(1) Address/data separate bus type, no address latch signal
(2) Address/data separate bus type, with address latch signal
(3) Multiplexed bus type

For transfer timings, refer to the timing diagrams for electrical characteristics.

MSM9225 CONNECTION EXAMPLES

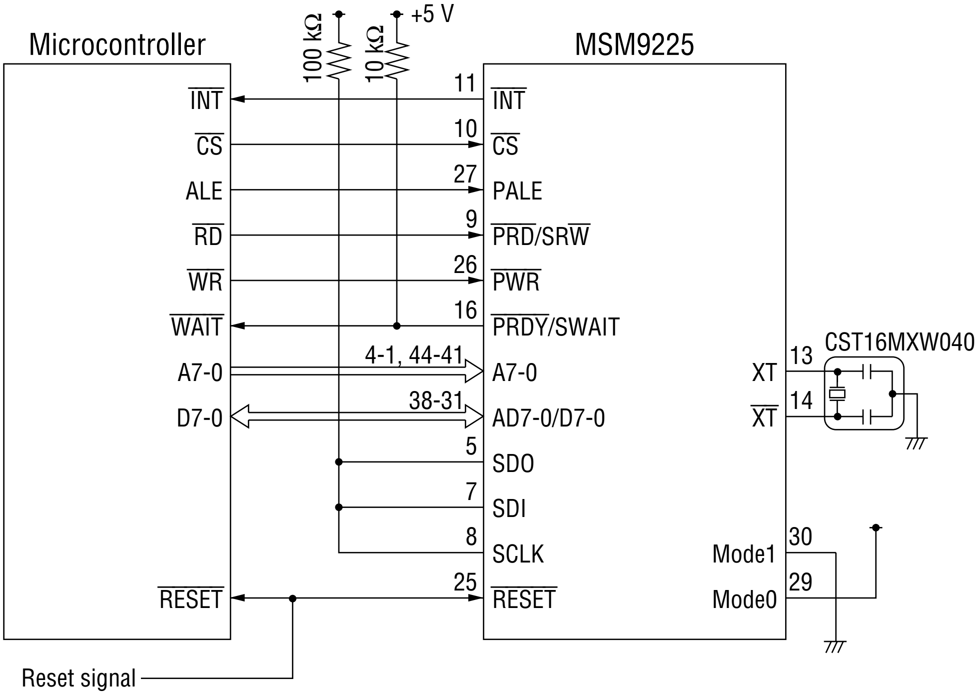
Microcontroller Interface

(1) Address/data separate bus (no address latch signal)

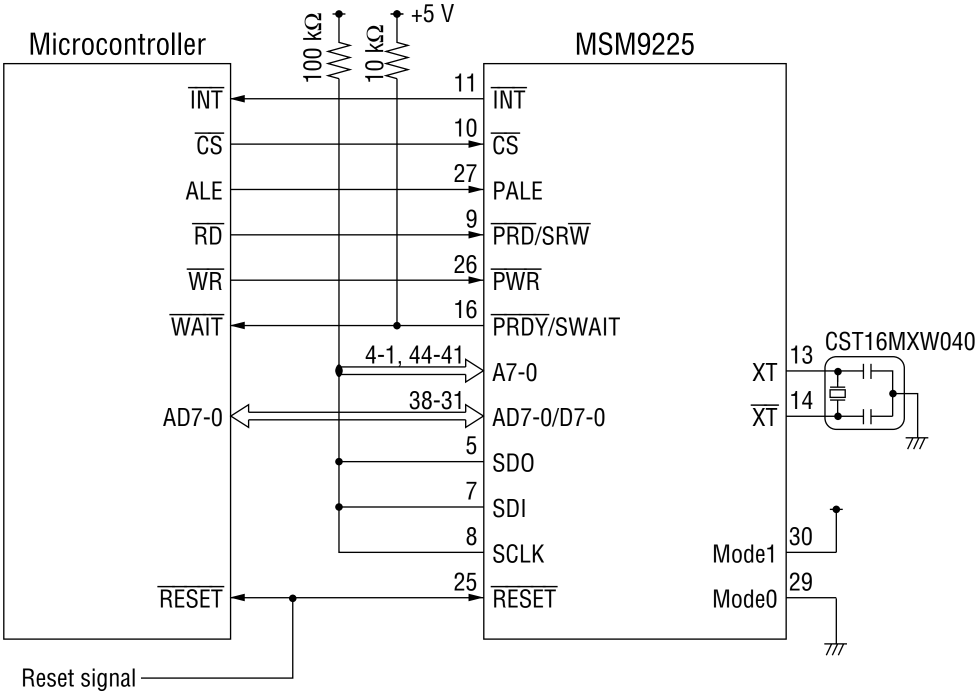


If the clock is supplied externally,in the same manner as for the serial interface, input the clock to the XT pin and leave the $\overline{\text{XT}}$ pin open.

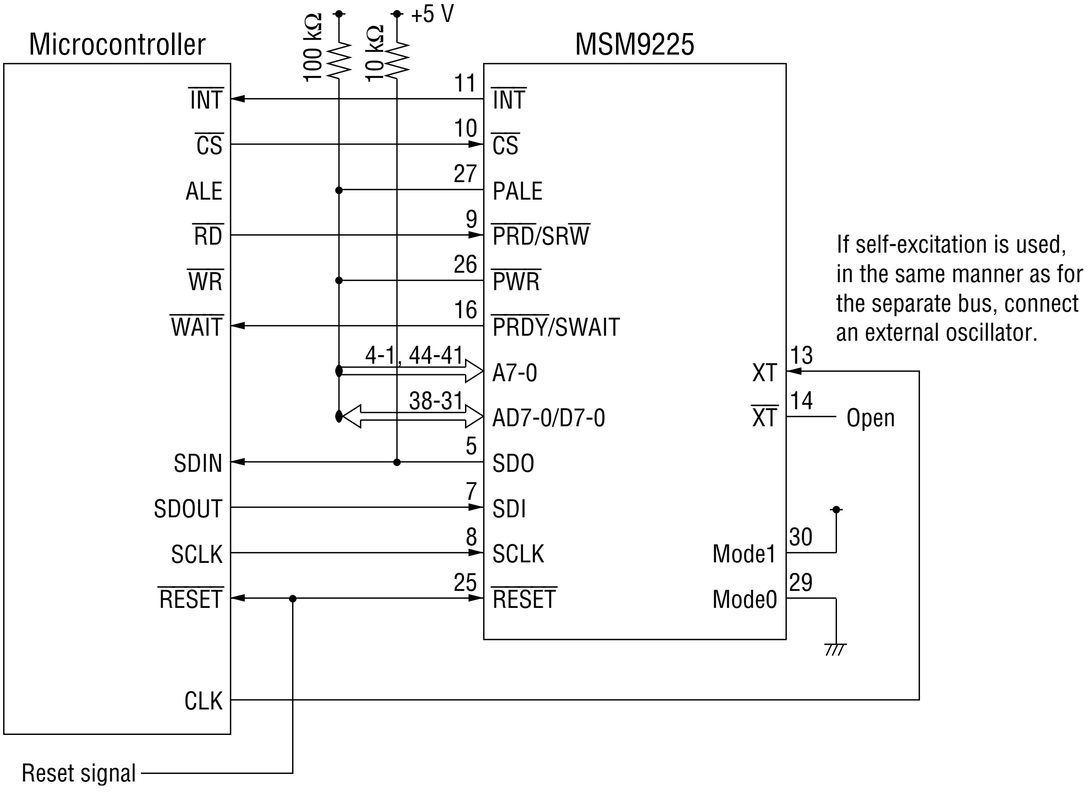
(2) Address/data separate bus (with address latch signal)



(3) Address/data multiplexed bus

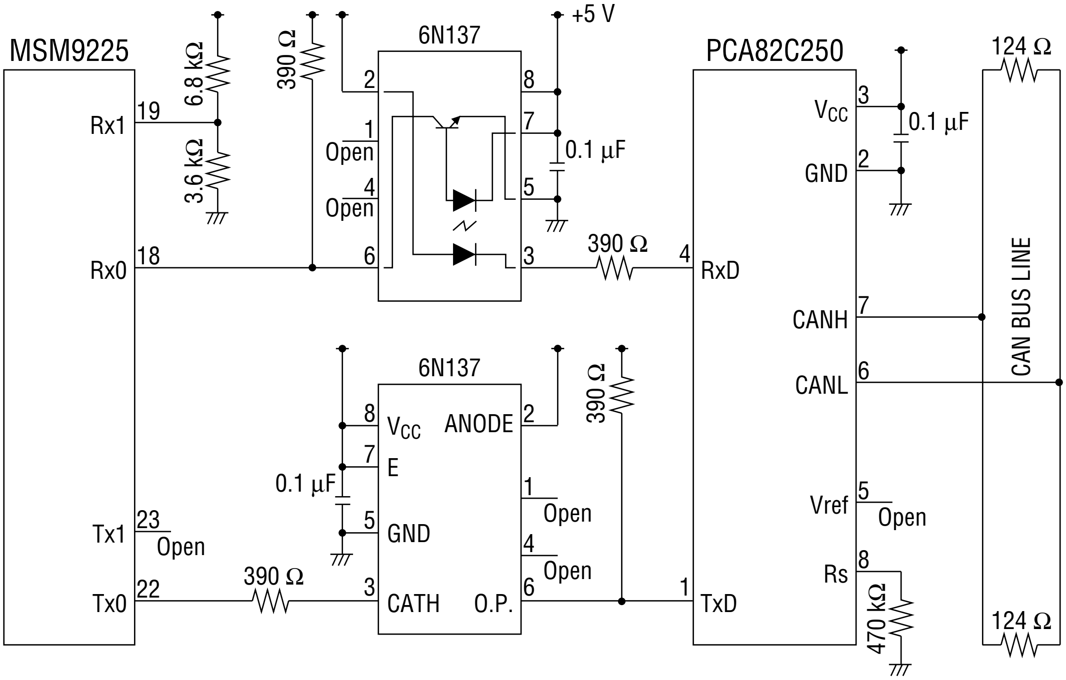


(4) Serial interface

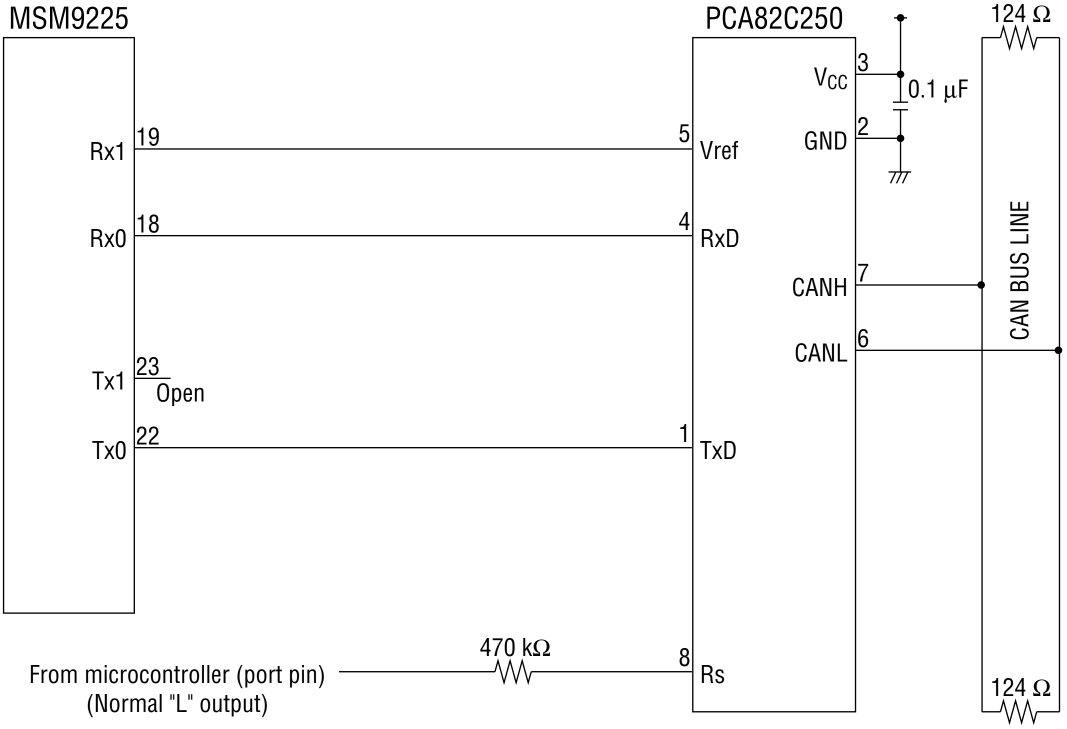


CAN Bus Interface

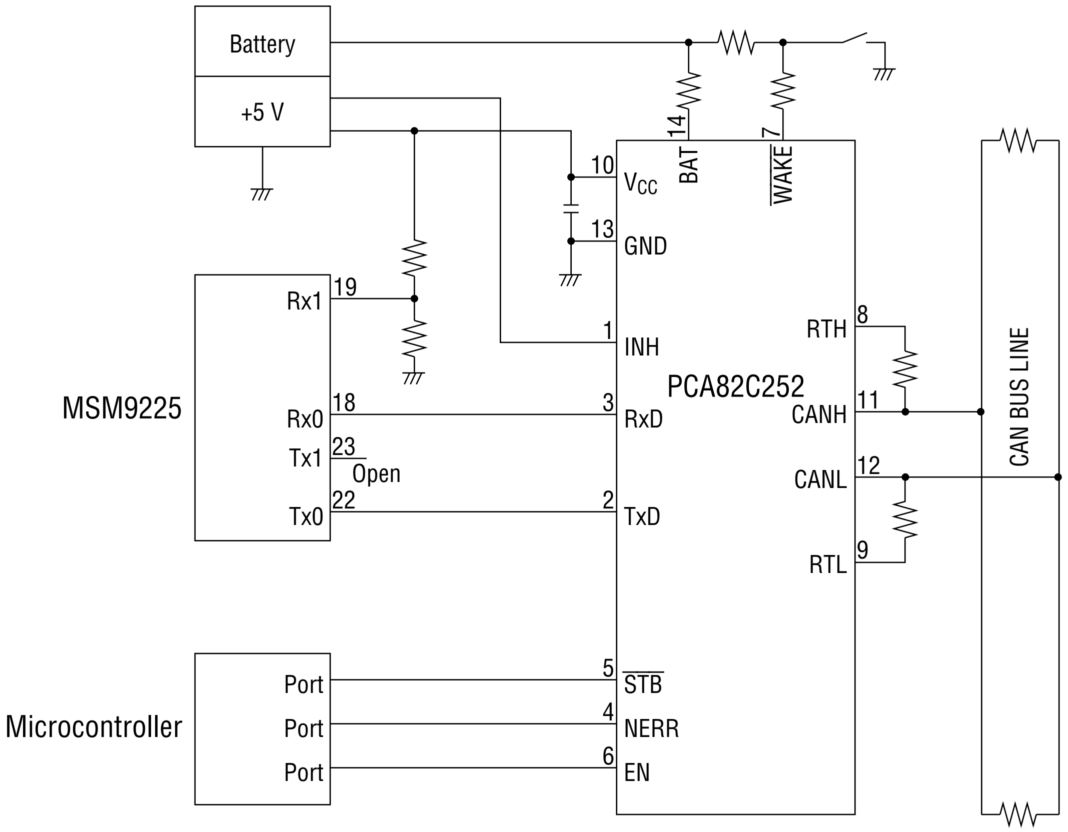
(1) Electrically isolated from bus transceiver (PCA82C250)



(2) Directly connected to bus transceiver (PCA82C250)



(3) Monitoring the CAN bus



PROTOCOL

The CAN (Controller Area Network) is a high-speed multiplexed communication protocol designed to perform real-time communication inside an automobile. CAN specifications are broadly classified into two layers, the physical layer and the data link layer. The data link layer consists of logical link control and medium access control.
The configuration of each layer is listed below.

Upper	Application layer (not including object)
↑	Data link layer
	• Logical link control (LLC): message and status handling
	• Medium access control (MAC): as per protocol
Lower	Physical layer: signal level and bit representation

Protocol Mode Function

(1) Standard format mode
2032 types of identifiers can be set in this mode.
Since the identifier is 11 bits, 2032 types of messages can be handled.

(2) Extended format mode
2032×2^{18} types of identifiers can be set in this mode.
In the standard format mode, the identifier is 11 bits. However, in the extended format mode, the identifier is extended to 29 bits (11 + 18).
If the SRR and IDE bits of the arbitration field are both "recessive", the mode changes to the extended format mode.
If remote frames for an extended format mode message and a standard format message are transmit simultaneously, the node that transmit the extended format message will change to the receive state.

Message Format

CAN protocol messages have the following 4 types of frames.

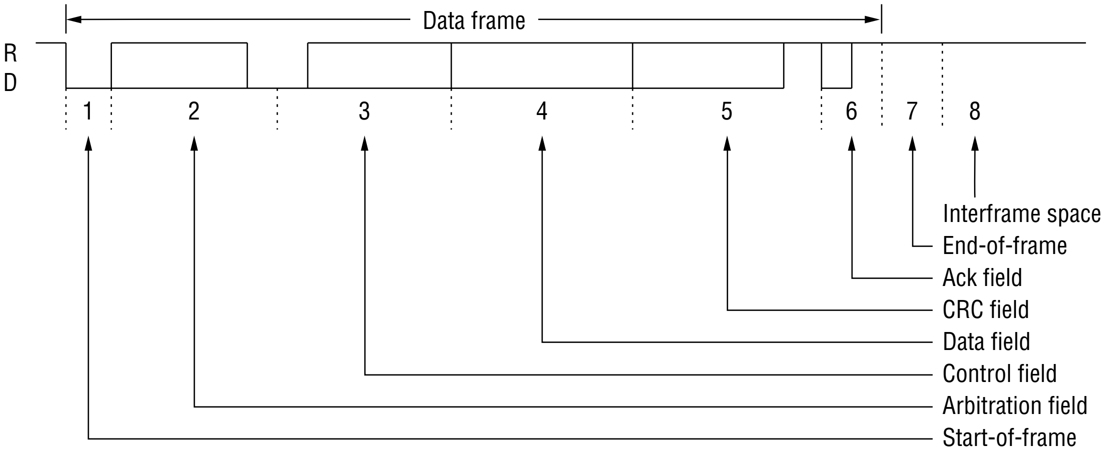
(1) Data frame	: transmit data frame
(2) Remote frame	: transmit request frame from the receive side
(3) Error frame	: frame that is output when an error is detected
(4) Overload frame	: frame that is output when the receive side has not completed preparing for reception

* In a wired-OR logic circuit, the stronger value is defined as "dominant" and the weaker value as "recessive". In figures hereafter, dominant (abbreviation: D) = 0, and recessive (abbreviation: R) = 1.

1. Data frame and remote frame

(1) Data frame

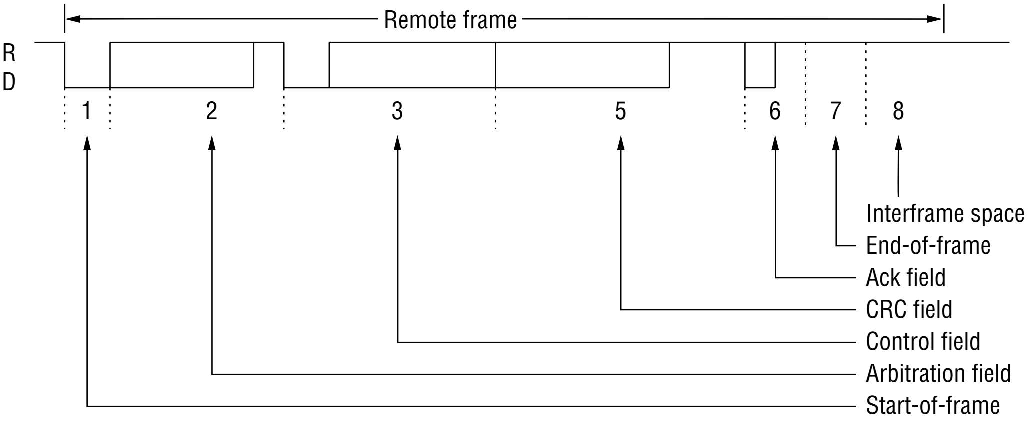
The data frame is for data transmission and consists of 8 fields.



(2) Remote frame

This frame is transmit when the receive node requests transmission.

The data field is deleted from the data frame and the RTR bit of the arbitration field is made "recessive".

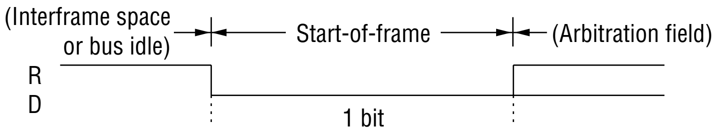


* Even when the data length code of the control field is nonzero, there will be no data frame transfer.

(3) Description of each frame

(a) Start-of-frame

Start-of-frame indicates the beginning of a data frame or remote frame and is one dominant bit.

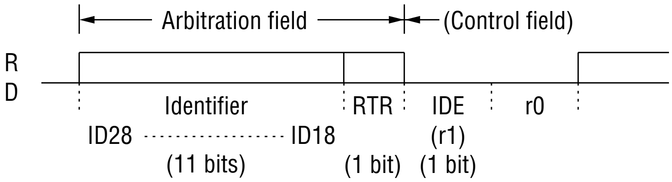


The start-of-frame begins when the bus line level changes.
If "dominant" is detected at the sample point, reception continues.
If "recessive" is detected at the sample point, the bus becomes idle.

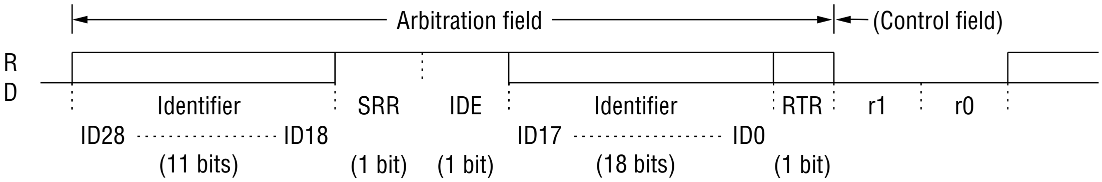
(b) Arbitration field

This field sets priority and data frame/remote frame protocol modes.
The arbitration field consists of an identifier, RTR bit, and extended format setting bits.

Standard format mode



Extended format mode



* Notes:

ID28 to ID0 is the identifier.
The identifier is transmitted MSB first.
It is prohibited to set the identifier = 1111111XXXXX.

Number of Identifier Bits

Protocol mode	No. of bits
Standard format mode	11 bits
Extended format mode	29 bits

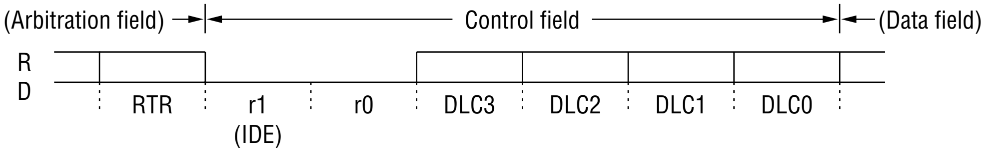
RTR Bit Setting

RTR bit	Frame type
Dominant	Data frame
Recessive	Remote frame

Mode Setting

Protocol mode	SRR bit	IDE bit
Standard format mode	None	Dominant
Extended format mode	Recessive	Recessive

(c) Control field
The control field sets the number of data bytes (N) in the data field. (N: 0 to 8)
r1 and r0 are fixed as "dominant". The number of bytes is set with DLC3 to DLC0.



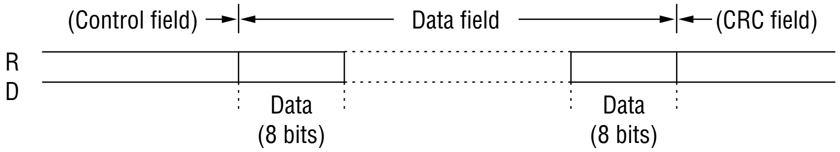
During the standard format mode, the r1 bit and IDE bit of the arbitration field are the same bit.

Data Length Code Setting

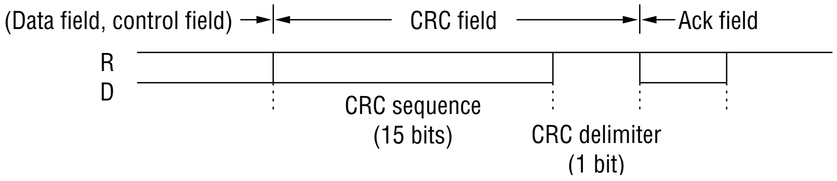
Data length code				No. of data bytes
DLC3	**DLC2**	**DLC1**	**DLC0**	
0	0	0	0	0
0	0	0	1	1
• •	• •	• •	• •	• •
0	1	1	1	7
1	0	0	0	8

* In the case of a remote frame, even when the data length code ≠ 0, there is no data field.

(d) Data field
The data field contains the number of data groups set by the control field. A maximum of 8 data groups can be set.
8 bits form 1 data group. (MSB first)



(e) CRC field
A 15-bit CRC sequence checks for transmission errors.
The CRC field consists of a 15-bit CRC sequence and a 1-bit CRC delimiter.

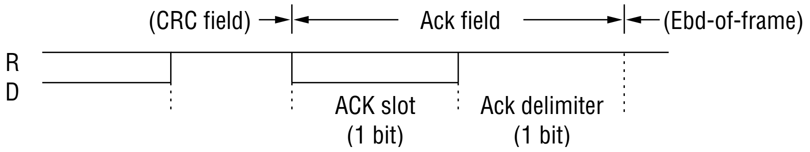


- The polynominal P(X) that generates the 15-bit CRC is expressed as follows.
  $\underline{P(X) = X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1}$
- The transmit node transmits a CRC sequence computed from all basic data bits of the start-of-frame, arbitration field, control field, and data field, without bit stuffing.
- The receive node, compares the CRC sequence computed from data bits of the received data (excluding stuff bits) with the CRC sequence in the CRC field. If they do not match, the node switches to an error frame.

(f) Ack field

The field verifies correct reception.

The Ack field consists of a 1-bit Ack slot and a 1-bit Ack delimiter.



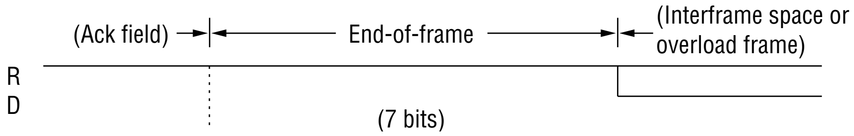
If the receive node detects an error between the start-of-frame and the CRC field, Ack slot = "recessive" is output. If an error is not detected, Ack slot = "dominant" is output.

The transmit node outputs 2 "recessive" bits, and verifies the reception status of the receive node.

(g) End-of-frame

This frame indicates the completion of transmission or reception.

The end-of-frame consists of 7 "recessive" bits.

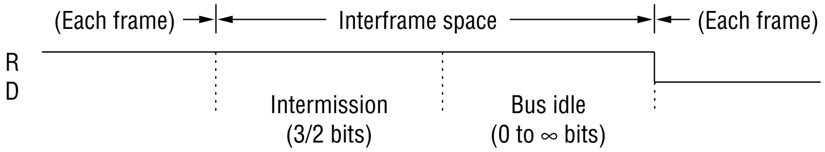


(h) Interframe space

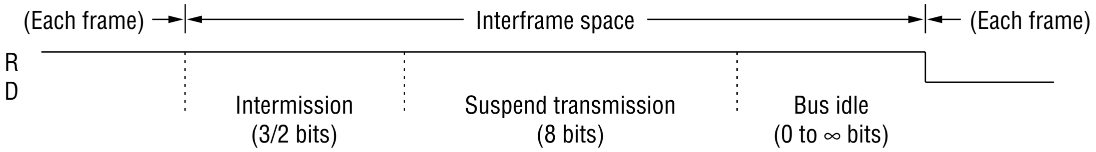
The interframe space is inserted between the data frame, remote frame, error frame, and overload frame and the next frame. The interframe space indicates the separation between frames.

Output is prohibited during intermission.

- Error active: The interframe space consists of a 3- or 2-bit intermission and bus idle.



- Error passive: The interframe space consists of intermission, suspend transmission, and bus idle.



Intermission Bit Length

Protocol mode	Bit length
Standard format mode	3 bits

Error Status and Operation

Error status	Operation
Error active	When the bus becomes idle, each node is able to transmit. The node with a transmit request begins to transmit.
Errpr passive	After bus idle has continued for 8 bits, transmission becomes possible. If another node begins transmission while the bus is idle, the node changes to reception.

Operation when the 3rd Intermission Bit is "Dominant"

Transmit status	Operation
No transmit hold	Evaluated as a start-of-frame output from another node. Reception is performed.
Transmit hold	Evaluated as a start-of-frame from own node. The identifier is transmit.

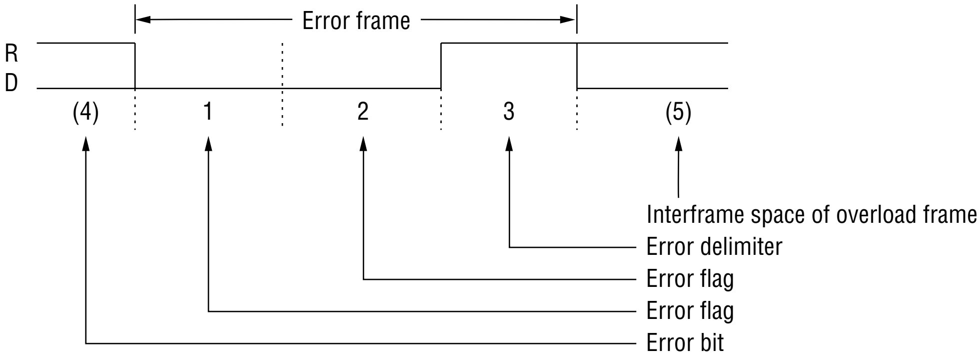
Bus idle: State where bus is not being used by any node.

2. Error frame

When an error occurs, the node that detected the error will output this frame.

While a passive error flag is being output, if another node outputs "dominant", the passive error flag will not end until 6 consecutive bits at the same level are detected.

If 6 consecutive bits are "recessive" but the 7th bit is "dominant", the error flag will end after the bit level changes to "recessive".



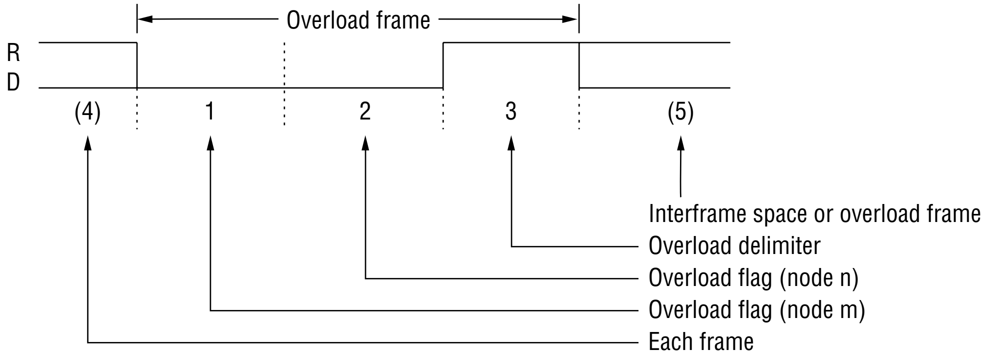
Field Definitions

No.	Name	No. of bits	Difinition
1	Error flag	6	Error active node: Outputs 6 consecutive "dominant" bits. Error passive node: Outputs 6 consecutive "recessive bits".
2	Error flag	0 to 6	The node that has received an "error flag" detects a bit stuff error and outputs an "error flag" again.
3	Error delimiter	8	Outputs 8 consecutive "receive" bits. If the 8th bit is observed to be "dominant", an overload frame is transmit biginning at the next bit.
4	Error bit	—	Output following the bit in which an error occurred. (In the case of a CRC error, this field is output following the Ack delimiter.)
5	Interframe space/ overload frame	3/10 20 Max	"Interframe space" or "overload frame" continues.

3. Overload frame

When reception preparations are not complete, the receive node outputs this frame from the 1st intermission bit.

If a bit error is detected during intermission, this frame is output from the next bit after a bit error is detected.



Field Definitions

No.	Name	No. of bits	Difinition
1	Overload flag from node m	6	Outputs 6 consecutive "dominant" bits. The overload flag is output because node m has not finished reception preparations.
2	Overload flag from node n	0 to 6	Having received an "overload flag" during an "interframe space", node n outputs an overload flag.
3	Overload delimiter	8	Outputs 8 consecutive "recessive" bits. If the 8th bit is observed to be "dominant", an overload frame is transmit biginning at the next bit.
4	Each frame	—	Output following end-of-frame, error delimiter, and overload delimiter.
5	Interframe space/ overload frame	3/10 20 Max	"Interframe space" or "overload frame" continues.

FUNCTIONS

1. Bus priority decisions

(1) When a single node has started transmission
While the bus is idle, the node that outputs data first will transmit.

(2) When multiple nodes have started transmission
Beginning from the 1st bit of the arbitration field, the node that outputs the longest consecutive string of "dominant" bits will have priority. (Since the bus has a wired-OR configuration, "dominant" is strong.)
The transmit node compares the arbitration field that it has output with the data levels on the bus.

Matching levels	Transmission continues.
Non-matching levels	Data output is terminated from the next bit after non-matching is detedted. The operation changes to reception.

(3) Data frame and remote frame priority
If a data frame and remote frame contend for control of the bus, the data frame whose last bit, RTR, is "dominant" will be given priority.

2. Bit stuffing
If 5 or more consecutive bits have the same level, bit stuffing prevents a burst error by appending 1 bit of inverted data, and then re-synchronizing.

Transmission	When transmitting a data frame or remote frame, if there are 5 consecutive bits with the same level between the start-of-frame and the CRC field, 1-bit of data at the inverted level of the previous 5 bits is inserted before the next bit.
Reception	When receiving a data frame or a remote frame, if there are 5 consecutive bits with the same level between the start-of-frame and the CRC field, the next bit is deleted and the data received

3. Multi-master
So that bus priority can be determined by the identifier, any node may become the bus master.

4. Multi-cast
There is one transmit node, however since multiple nodes can be set with the same identifier, multiple nodes can simultaneously receive the same data.

5. Sleep and stop mode functions
These modes are low-power consuming standby modes.
Setting the SLEEP bit of the STBY register to "1" sets the sleep mode.
(after bus idle)
Setting the STOP bit of the STBY register to "1" sets the stop mode.
(after bus idle)
The sleep mode is released when the Rx0 and Rx1 differential inputs, the $\overline{\text{RESET}}$ pin input, or the $\overline{\text{CS}}$ pin input is at a "L" level.
The stop mode is released when the $\overline{\text{RESET}}$ pin input or the $\overline{\text{CS}}$ pin input is at a "L" level.

6. Error control functions

(1) Types of errors

Type of error	**Error description**		**Detection state**	
	Detection method	**Detection condition**	**Transmit/Receive**	**Field/Frame**
Bit error	Comparison of output level and bus level (excluding stuff bits)	Both levels do not match	Transmit/Receive node	Bits that output data onto the bus, start-of-frame to end-of-frame, error frame, and overload frame
Stuff error	Verify received data with the stuff bit	Same level of data for 6 consecutive bits	Transmit/Receive node	Start-of-frame to CRC saquence
CRC error	CRC generated from received data compared to received CRC sequence	CRC's do not match	Receive node	Start-of-frame to data field
Form error	Verify fixed format field/frame	Detection of fixed format violation	Receive node	CRC delimiter • Ack field • End-of-frame • Error frame • Overload frame
Ack error	Verify Ack slot by transmit node	Detection of a "recessive" bit during Ack slot	Transmit node	Ack slot

(2) Error frame output timing

Type of error	**Output timing**
Bit error, stuff error, form error, Ack error	Error frame is output at the next bit after the error is detected.
CRC error	Error frame is output at the next bit after the Ack delimiter.

(3) Procedure when an error is generated

After the error frame, the transmit node retransmits a data frame or a remote frame.

(4) Error states

(a) Types of error states

- There are three types of error states: error active, error passive, and bus OFF.
- Error states are managed by the transmit error counter and the receive error counter.
- Each error state is classified according to the error counter value.
- The error flag that is output differs depending upon whether the error state is a transmit or receive operation
- If the value of the error counter is 96 or greater, the bus may be heavily damaged. The bus must be tested for this condition.
- If only one node is active at startup, even if data is transmit an Ack will not be returned. Therefore, error frame and data retransmission are repeated. In this case, the bus OFF state will not be entered. Even if an error state is repeated at the node that transmits messages, the bus OFF state will not be entered.
- After reset and after the sleep mode wakes up, the error passive state continues until Ack is received. Regardless of the number of errors that occur, the transmit error counter will be 255.
- Reception can be performed even if transmission is in the bus OFF state.

Type of error state	Operation	Error counter value	Type of error flag to be output
Error active	Transmit/Receive	from 0 to 127	Active error flag (6 consecutive "dominant" bits)
Error passive	Transmit	from 128 to 255	Passive error flag (6 consecutive "recessive" bits)
	Receive	128 or greater	
Bus OFF	Transmit	256 or greater	Communication not possible. If 11 consecutive "recessive" bits occur 128 times, then when the error counter = 0, the state can return to error active.
	Receive	—	No bus OFF

(b) Error counter

The error counter is incremented when errors occur and is decremented when transmission or reception is performed correctly. Timing of the increment or decrement occurs at the 1st bit of the error flag.

State	Transmit error counter	Receive error counter
Receive node has detected an error (excluding bit errors within the active error flag or overload flag)	No change	+1
Receive node detects "dominant" after error flag output of error frame	No change	+8
Transmit node transmits error flag [when error counter = 0] (1) Error passive state and Ack error detected, but "dominant" not detected in passive error flag output (2) Stuff error occurred during arbitration field	+8	No change
Bit error detected in output of active error flag, overload flag (error active transmit node)	+8	No change
Bit error detected in output of active error flag, overload flag (error active receive node)	No change	+8
Each node detects 14 consecutive "recessive" bits from the beginning of the active error flag or overload flag, and 8 consecutive "dominant" bits detected thereafter Each node detects 8 consecutive "dominant" bits after the passive error flag	+8	+8
Transmit node completes transmission without errors	–1 (±0 when error counter = 0)	No change
Receive node completes reception without errors	No change	(1) –1 (1 ≤ REC ≤ 127) (2) ±0 (REC = 0) (3) Set to 127

* REC: Receive Error Counter

(c) Bit error occurring during intermission
Overload frame is generated.

Note) When an error has occured, error control is performed by the error counter at that time. After an error flag is output, the indicated values are added to the error counter.

7. Baud rate control function

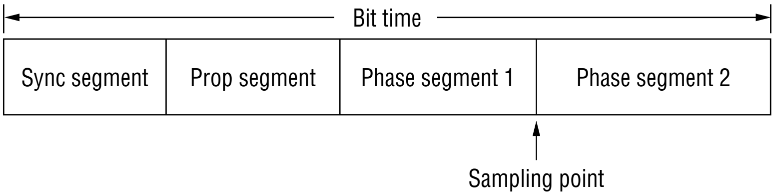
(1) Prescaler

The MSM9225 has a prescaler that divides the frequency of the system clock. The prescaler divides the system clock frequency by a factor of 1 to 64 to generate clock CK_{BTL}. (BTL: Bit Time Logic)

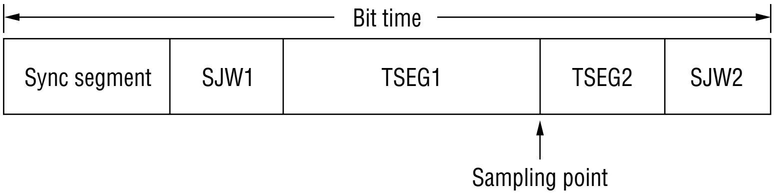
(2) Bit timing

The timing for 1 data bit is defined below.

Definition for CAN protocol



Definition for MSM9225



• Sync segment	: This is the first segment for bit synchronization.
• Prop segment	: This segment absorbs the delay of the output buffer, CAN bus and input buffer. Set the prop segment so that Ack will be returned by the start of phase segment 1. Prop segment time ≥ (output buffer delay) + (CAN bus delay) + (input buffer delay)
• Phase segments	: These segments compesate for deviations in the data bit timing. The larger these segments, the greater the allowable deviation, however communication speed will decrease.
• SJW	: Abbreviation of reSynchronization Jump Width. These bits set the bit synchronization range.

Segment name		Segment length (BTL)
CAN protocol	**MSM9225**	
Sync segment (Synchronization segment)	Sync segment (Synchronization segment)	1
Prop segment (Propagation segment)	SJW1	1 to 4, programmable
Phase segment 1 (Phase Buffer segment)	TSEG1 (Time segment)	1 to 16, programmable
Phase segment 2 (Phase buffer segment)	TSEG2 (Time segment)	1 to 8, programmable
	SJW2 protocol	1 to 4, programmable

(3) Data bit synchronization

Since there is no sync signal for the receive node, synchronization is obtained from level changes on the bus.

The transmit node transmits data is synchronization with the transmit node bit timing.

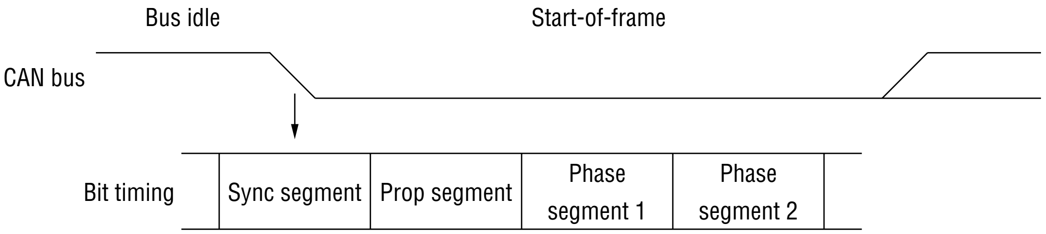
(a) Hardware synchronization

Hardware synchronizaion is the bit synchronization performed when a receive node in the bus idle state detects a start-of-frame.

If a falling edge is detected on the bus, that bit is the sync segment and is followed by the prop segment. In this case, syncronization is obtained without regard for SJW.

After reset and after wake up, it is necessary to obtain bit synchronization.

Therefore, hardware synchronizes to the first bus level change only.



(b) Bit synchronization

If a level change is detected on the bus during receprion, bit synchronization is obtained.

There are two methods of synchronization.

Normal operation: falling edge of level

Low-speed operation: falling edge and rising edge of level

During the bit timing interval specified by SJW, synchronization is obtained only if an edge is detected.

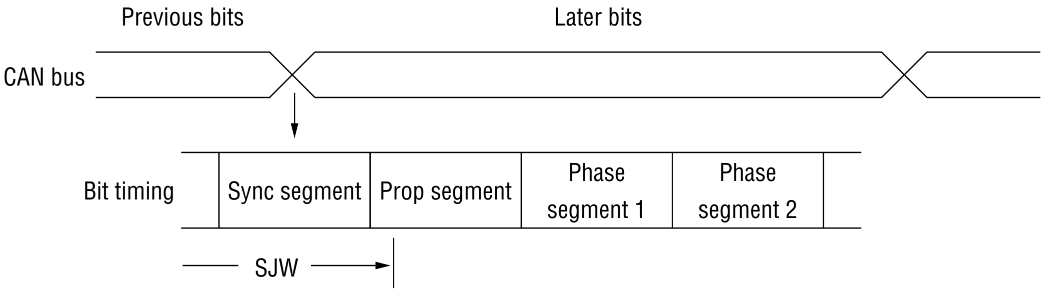
The data sampling point of the receive node will move in relation to the shift in baud rate between the transmit node and receive node.

The range of allowable "shift" is defined as "SJW". The SJW range is centered on the sync segment and extends both before and after that segment (+/– baud rate). If an edge occurs within the SJW range, synchronization is obtained.

If an edge occurs outside the SJW range, synchronization is not obtained.

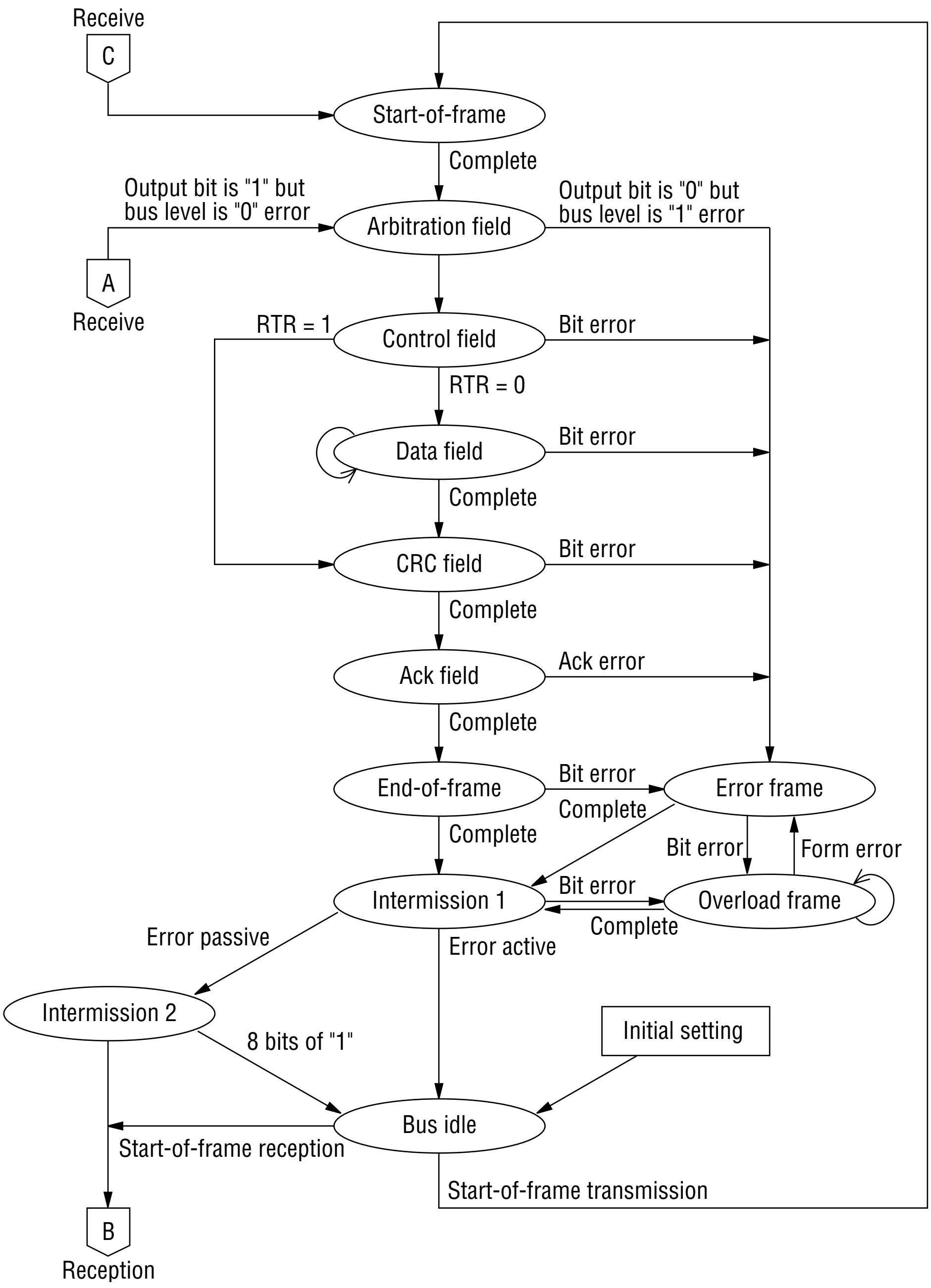
The bit detected at the edge forces the sync segment, and is followed by the prop segment.

The bit timing is restarted.

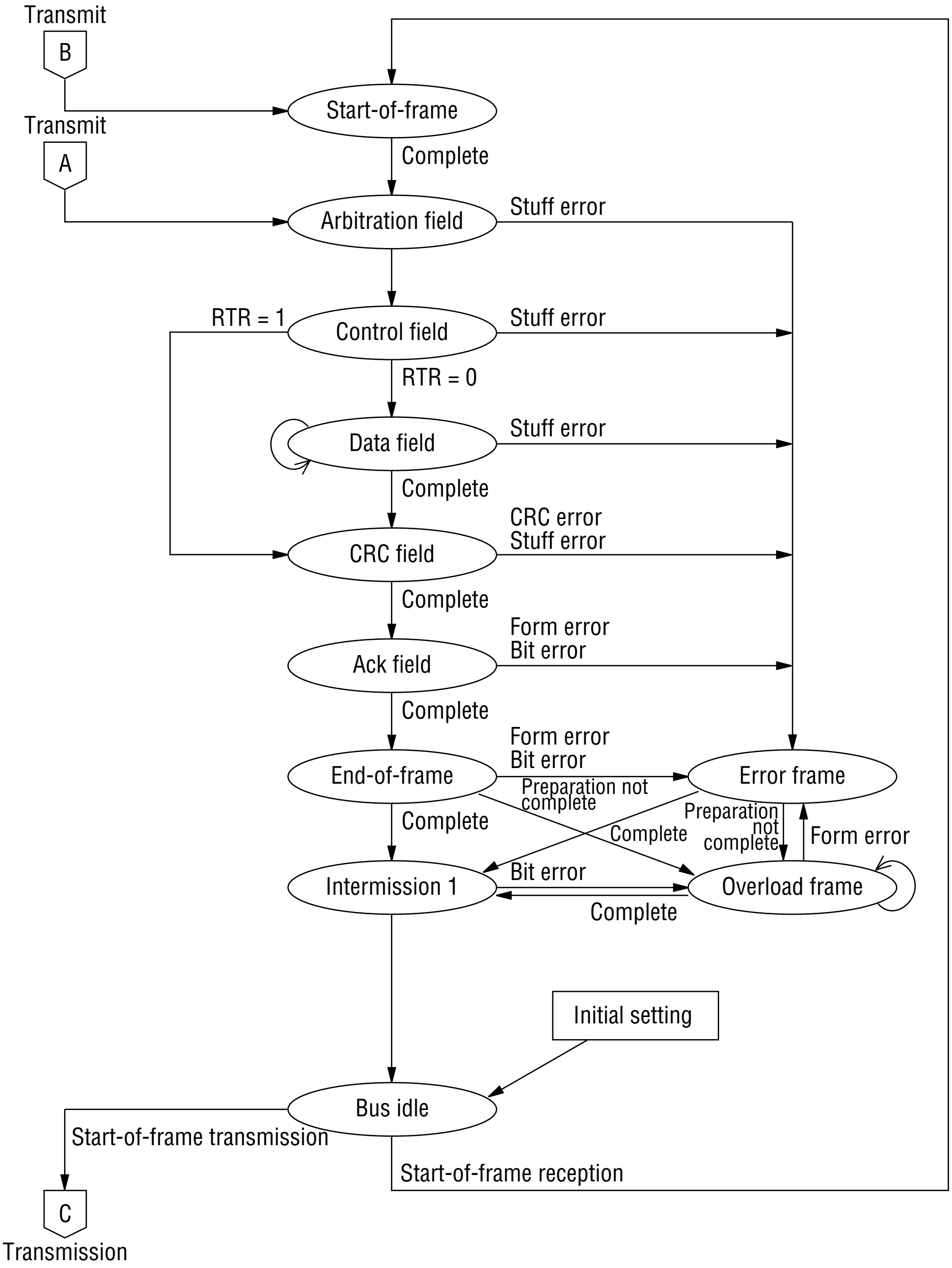


8. State transition diagrams

(1) Transmit state transition diagram

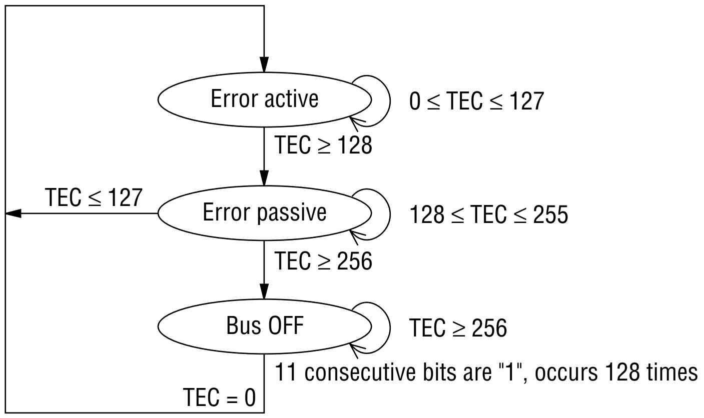


(2) Receive state transition diagram



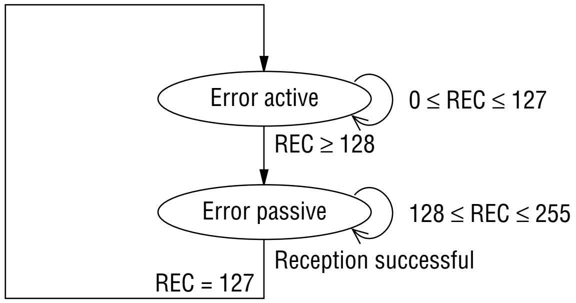
(3) Error state transition diagram

(a) Transmit



*TEC: Transmit Error Counter

(b) Receive



*REC: Receive Error Counter

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Condition	Rating	Unit
Power Supply Voltage	V_{DD}	Ta = 25°C	–0.3 to +7.0	V
	AV_{DD}		–0.3 to +7.0 (AV_{DD} = V_{DD})	V
Input Voltage	V_I	—	–0.3 to V_{DD} + 0.3	V
Output Voltage	V_O	—	–0.3 to V_{DD} + 0.3	V
Power Dissipation	P_D	Ta ≤ 25°C	615	mW
Operating Temperature	T_{OP}	—	–40 to +115	°C
Storage Temperature	T_{STG}	—	–65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Power Supply Voltage	V_{DD}	V_{DD} = AV_{DD}	4.5	5.0	5.5	V
Operating Temperature	T_{OP}	—	–40	+25	+115	°C

ELECTRICAL CHARACTERISTICS

DC Characteristics

(V_{DD} = AV_{DD} = 4.5 to 5.5 V, Ta = –40 to +115°C)

Parameter	Symbol	Applicable pin	Condition	Min.	Max.	Unit
"H" Input Voltage	V_{IH}	Applies to all inputs	—	$0.8V_{DD}$	$V_{DD} + 0.3$	V
"L" Input Voltage	V_{IL}	Applies to all inputs	—	–0.3	$+0.2V_{DD}$	V
"H" Input Current	I_{IH1}	XT	$V_I = V_{DD}$	3	25	µA
	I_{IH2}	Other inputs		–1.0	+1.0	µA
"L" Input Current	I_{IL1}	XT	V_I = 0 V	–25	–3	µA
	I_{IL2}	Other inputs		–1.0	+1.0	µA
"H" Output Voltage	V_{OH1}	$\overline{\text{INT}}$, $\overline{\text{PRDY}}$/SWAIT	I_{OH1} = –80 µA	$V_{DD} - 1.0$	—	V
	V_{OH2}	AD7-0/D7-0	I_{OH2} = –400 µA	$V_{DD} - 1.0$	—	V
"L" Output Voltage	V_{OL1}	$\overline{\text{INT}}$, $\overline{\text{PRDY}}$/SWAIT	I_{OL1} = 1.6 mA	—	0.4	V
	V_{OL2}	AD7-0/D7-0	I_{OL2} = 3.2 mA	—	0.4	V
Output Leakage Current	I_{IH1}	$\overline{\text{PRDY}}$/SWAIT, AD7-0/D7-0	$V_I = V_{DD}$/0 V	–1.0	+1.0	µA
Dynamic Supply Current	I_{DD}	—	f_{OSC} = 16 MHz, No Load	—	15	mA
Static Supply Current	I_{DDS}	—	SLEEP/STOP Mode	—	100	µA

Rx0, Rx1 Characteristics

(V_{DD} = AV_{DD} = 4.5 to 5.5 V, Ta = –40 to +115°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Input Voltage	VR_{XI}	—	0.5	$AV_{DD} - 1.5$	V
Input Offset Voltage	V_{OFF}	—	–20	+20	mV
Input Leakage Current	I_{LK}	—	–10	+10	µA
AV_{DD} Supply Current	AI_{DD}	—	—	4	mA

Tx0, Tx1 Characteristics

(V_{DD} = AV_{DD} = 4.5 to 5.5 V, Ta = –40 to +115°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
"H" Output Voltage	V_{OH}	I_{OH} = –3.0 mA	$AV_{DD} - 0.4$	—	V
	V_{OH}	I_{OH} = –6.0 mA	$AV_{DD} - 1.0$	—	V
"L" Output Voltage	V_{OL}	I_{OL} = 10.0 mA	—	0.4	V
	V_{OL}	I_{OL} = 20.0 mA	—	1.0	V

AC Characteristics

Parallel mode

($V_{DD} = AV_{DD} = 4.5$ to 5.5 V, Ta = –40 to +115°C, $f_{OSC} = 16$ MHz)

Parameter	Symbol	Condition	Min	Max	Unit
ALE Address Setup Time	t_{AS}	—	10	—	ns
ALE Address Hold Time	t_{AH}	—	10	—	ns
$\overline{\text{PRD}}$ Output Data Delay Time	t_{RDLY}	—	—	40	ns
$\overline{\text{PRD}}$ Output Data Hold Time	t_{RDH}	—	5	—	ns
ALE "H" Level Width	t_{WALEH}	—	20	—	ns
Access Cycle Time	t_{cyc}	—	4T	—	ns
Address Hold Time from $\overline{\text{PRD}}$	t_{RAH}	—	10	—	ns
ALE Delay Time from $\overline{\text{PRD}}$	t_{HRA}	—	20	—	ns
$\overline{\text{PRD}}$ "H" Level Width	t_{WRDH}	—	20	—	ns
$\overline{\text{PRDY}}$ "L" Delay Time	t_{ARLDLY}	—	—	35	ns
$\overline{\text{PRDY}}$ "H" Delay Time	t_{ARHDLY}	—	—	2.5T + 35	ns
Input Data Setup Time	t_{WDS}	—	30	—	ns
Input Data Hold Time	t_{WDH}	—	5	—	ns
$\overline{\text{PWR}}$ Delay Time	t_{WS}	—	10	—	ns
Address Hold Time from $\overline{\text{PWR}}$	t_{WAH}	—	20	—	ns
ALE Delay Time from $\overline{\text{PWR}}$	t_{HWA}	—	20	—	ns
$\overline{\text{PWR}}$ "H" Level Width	t_{WRH}	—	40	—	ns
$\overline{\text{PWR}}$ "L" Level Width	t_{WRL}	—	20	—	ns
$\overline{\text{CS}}$ Delay Time from $\overline{\text{PRD}}$	t_{HRC}	—	0	—	ns
$\overline{\text{CS}}$ Delay Time from $\overline{\text{PWR}}$	t_{HWC}	—	0	—	ns

Serial mode

($V_{DD} = AV_{DD} = 4.5$ to 5.5 V, Ta = –40 to +115°C, $f_{OSC} = 16$ MHz)

Parameter	Symbol	Condition	Min	Max	Unit
$\overline{\text{CS}}$ Setup Time	t_{CS}	—	10	—	ns
$\overline{\text{CS}}$ Hold Time	t_{CH}	—	8T	—	ns
SCLK Cycle	t_{CP}	—	167	—	ns
SCLK Pulse Width	t_{CW}	—	83	—	ns
SDI Setup Time	t_{DS}	—	30	—	ns
SDI Hold Time	t_{DH}	—	5	—	ns
SDO Output Enable Time	t_{CSODLY}	—	—	30	ns
SDO Output Disable Time	t_{CSZDLY}	—	—	30	ns
SDO Output Delay Time	t_{PD}	—	—	30	ns
SR$\overline{\text{W}}$ Setup Time	t_{RS}	—	10	—	ns
SR$\overline{\text{W}}$ Hold Time	t_{RH}	—	0	—	ns
SWAIT Output Delay Time	t_{SRDLY}	—	—	2T	ns
SWAIT "H" Level Width	t_{WRDY}	—	—	6T	ns
Byte Delay	t_{WAIT}	—	8T	—	ns

Other timing characteristics

(V_{DD} = AV_{DD} = 4.5 to 5.5 V, Ta = –40 to +115°C)

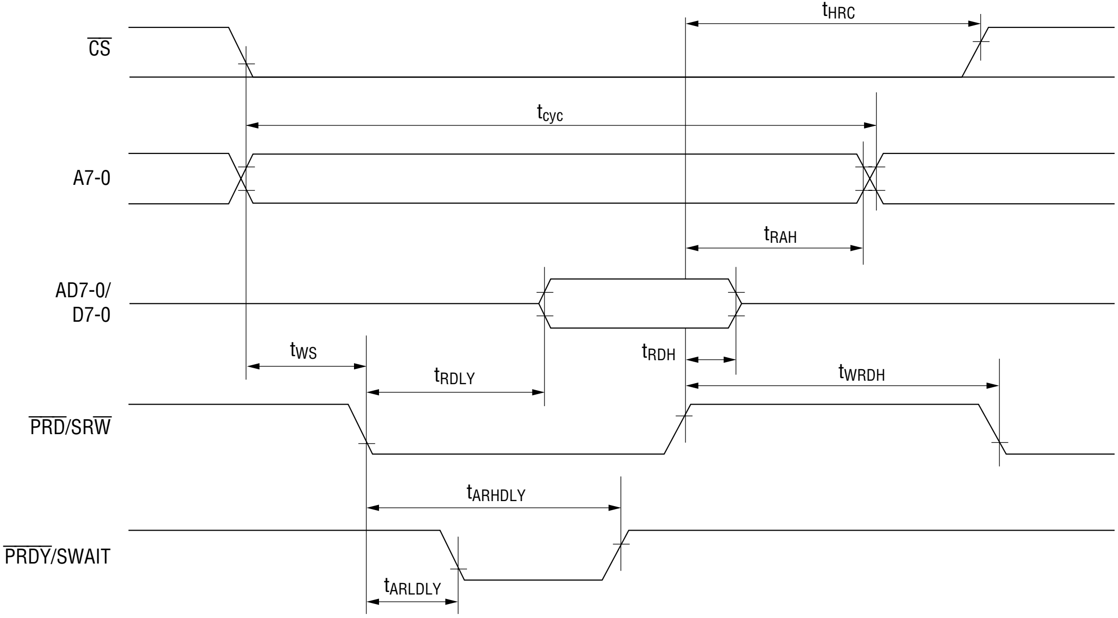
Parameter	Symbol	Condition	Min.	Max.	Unit
System Clock Cycle	t_{clkcy}	—	62	—	ns
$\overline{\text{RESET}}$ "H" Level Input Width	t_{WRSTH}	—	5	—	µs
$\overline{\text{RESET}}$ "L" Level Input Width	t_{WRSTL}	—	5	—	µs
$\overline{\text{INT}}$ "L" Level Output Width	t_{WINTL}	—	32T	—	ns

(*) $T = 1/f_{OSC}$

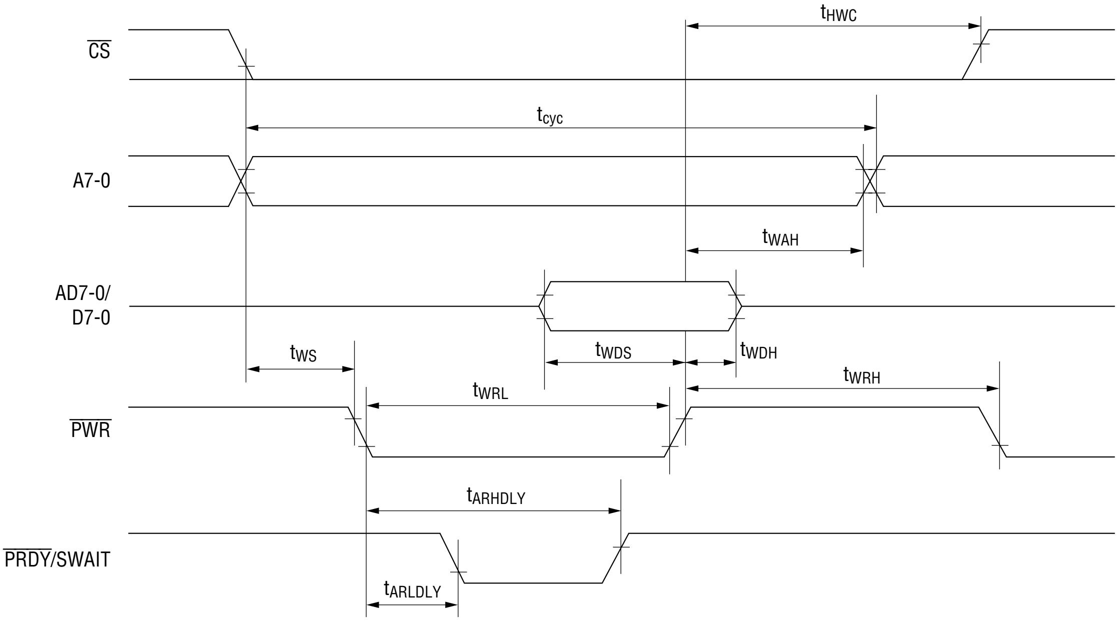
TIMING DIAGRAMS

Separate Bus Mode

Read access timing

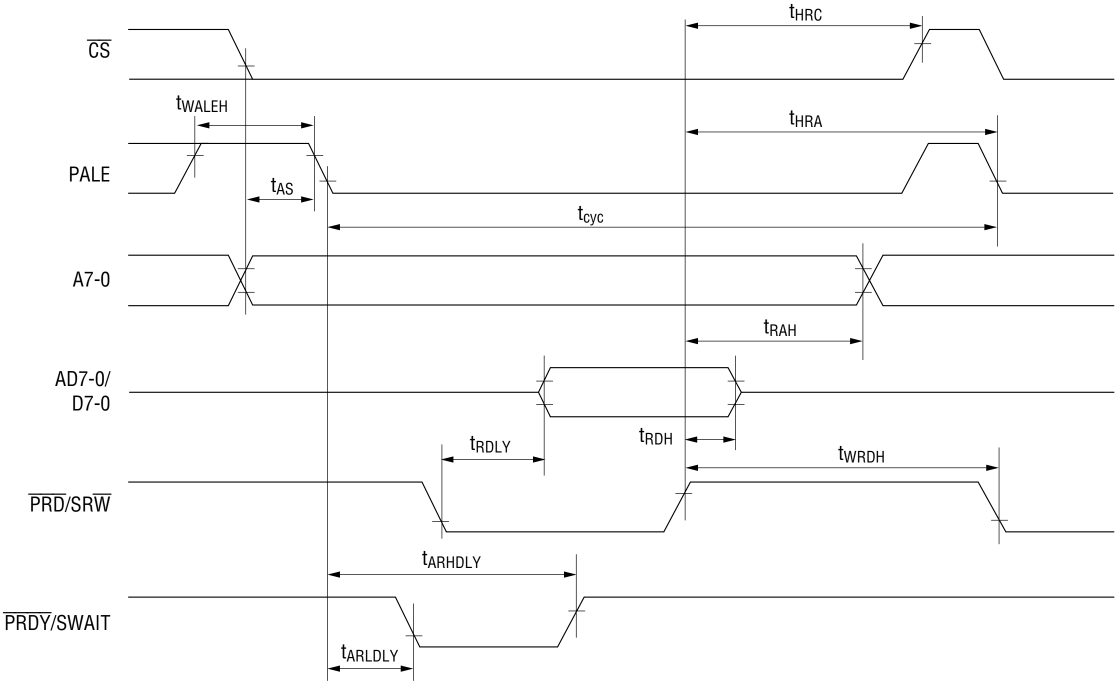


Write access timing

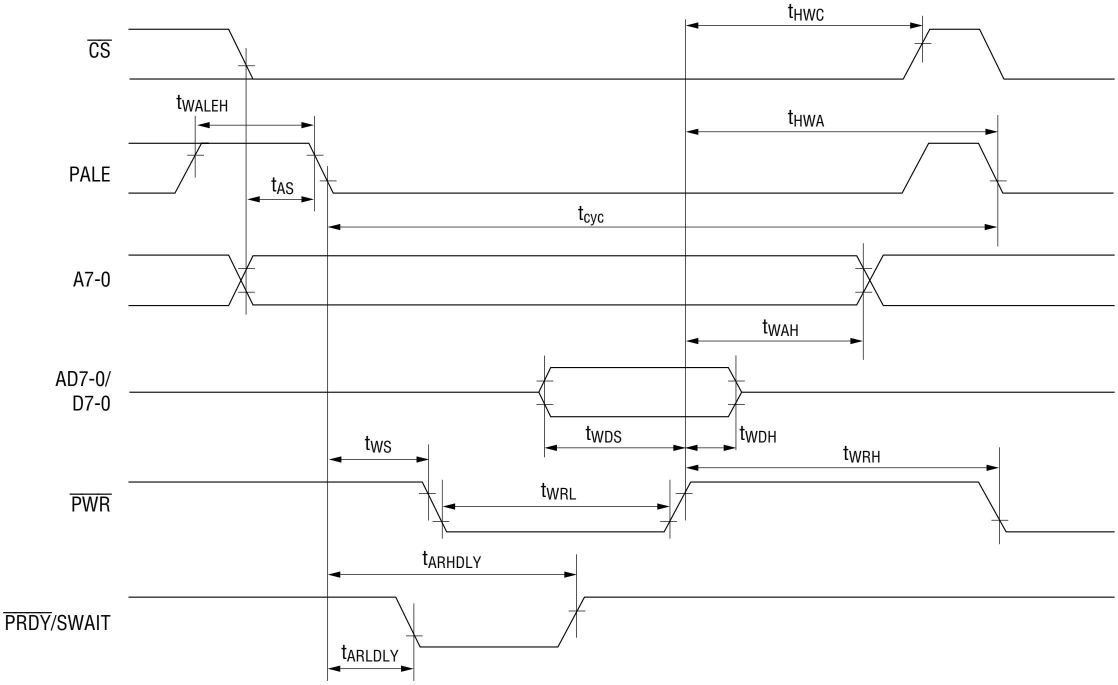


Separate Bus/Address Latch Mode

Read access timing

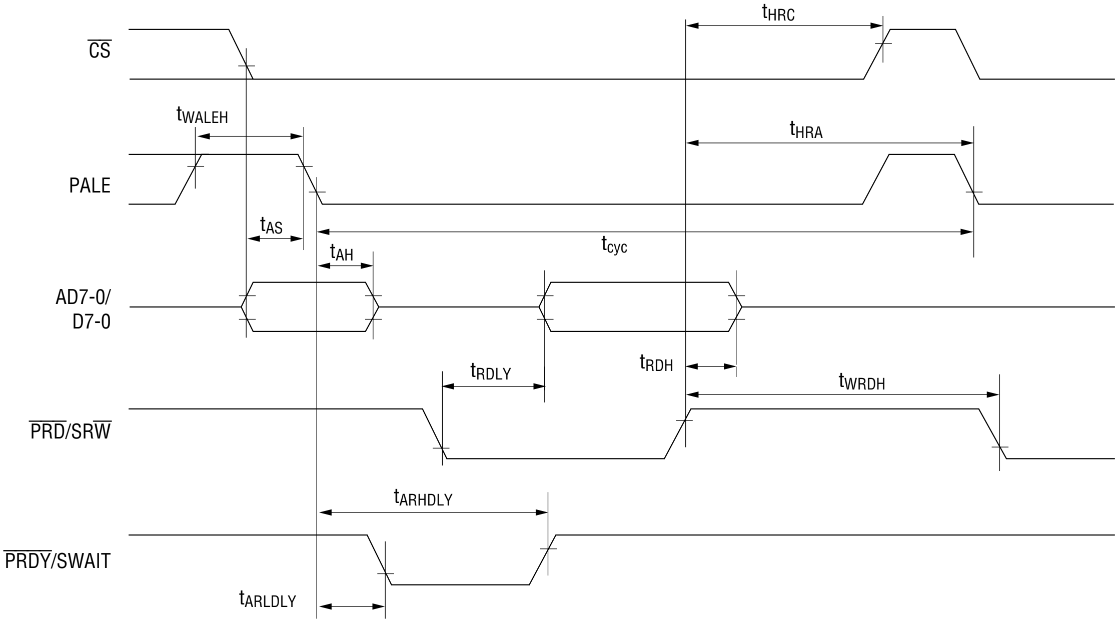


Write access timing

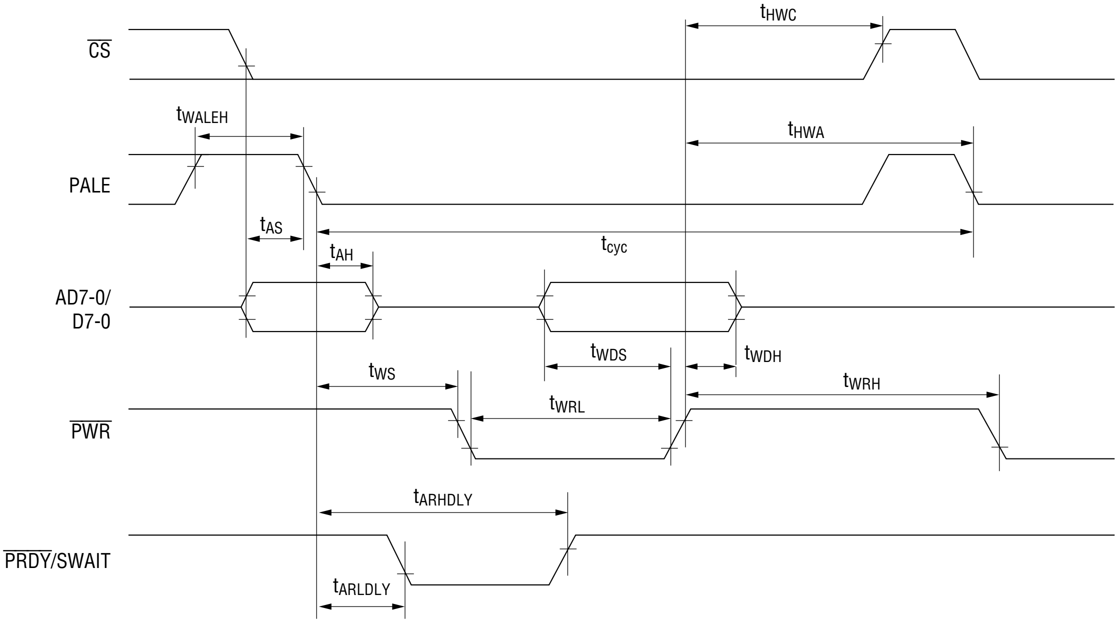


Multiplexed Bus Mode

Read access timing

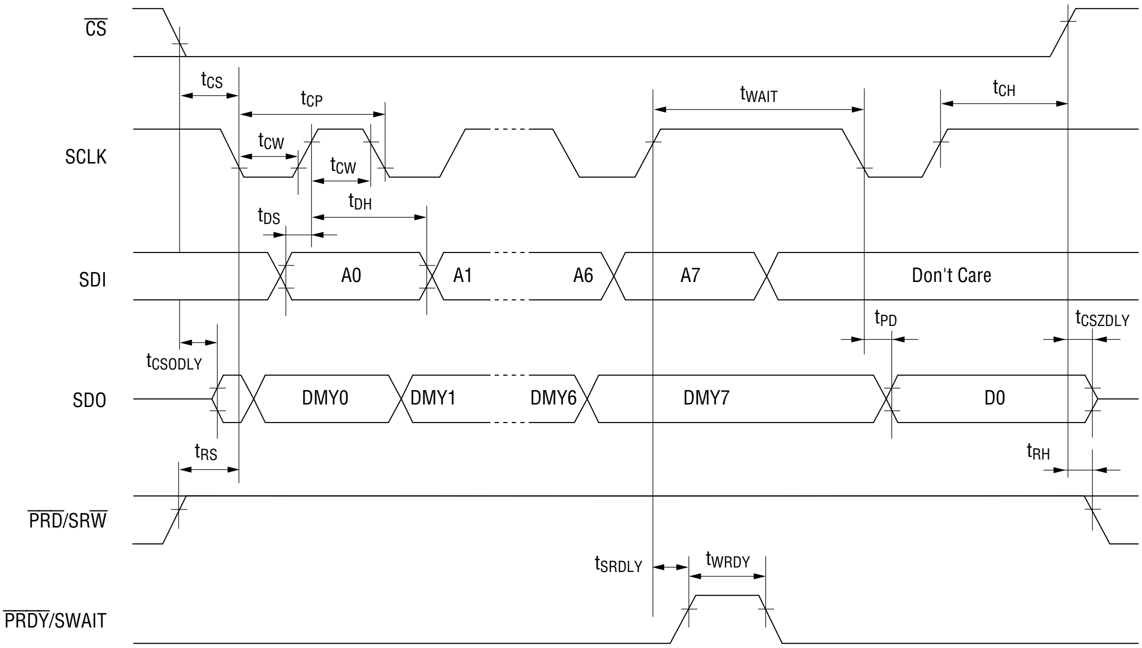


Write access timing

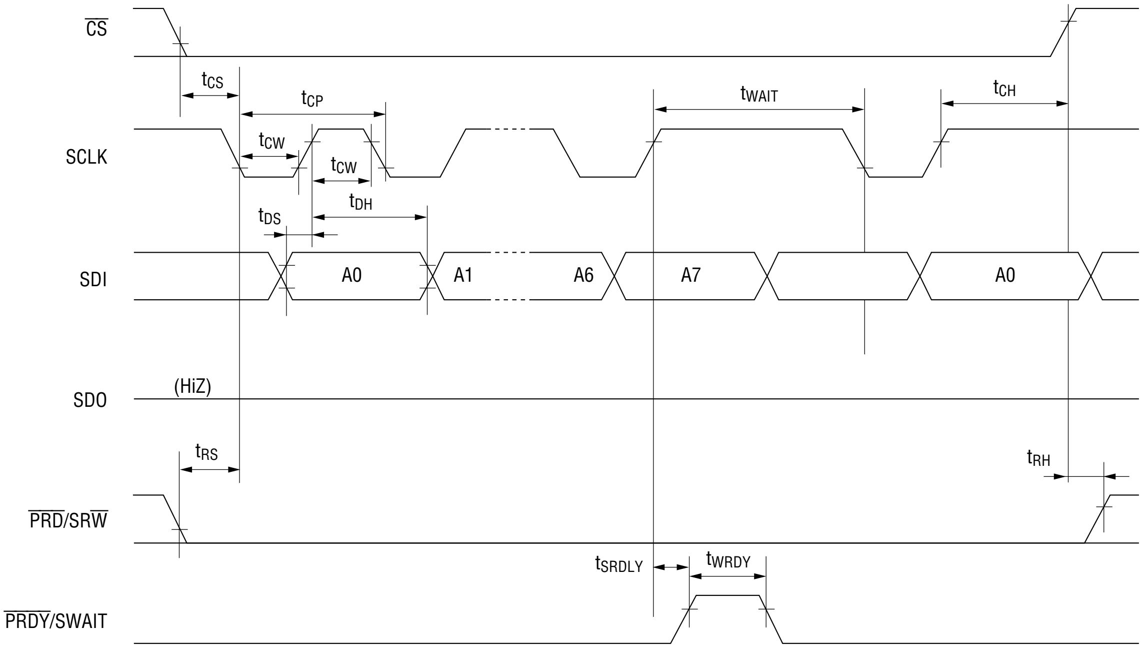


Serial Mode

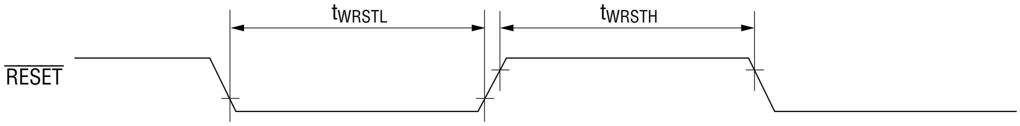
Read access timing

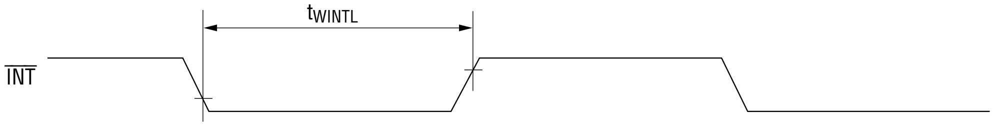


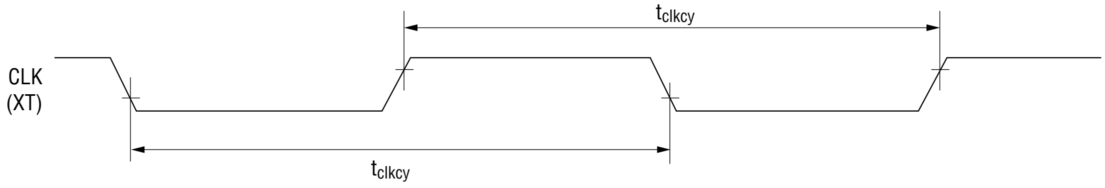
Write timing



Other Timings



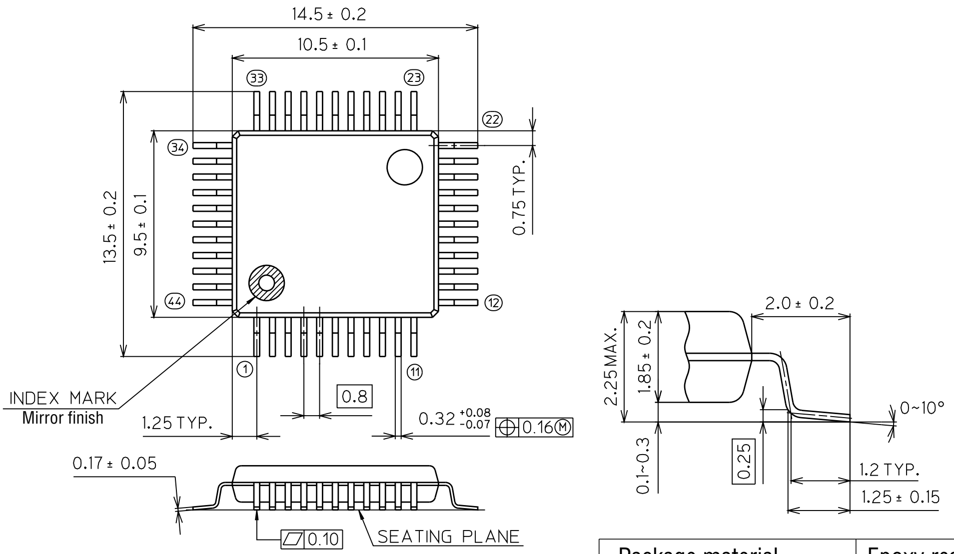




PACKAGE DIMENSIONS

(Unit : mm)

QFP44-P-910-0.80-2K



Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating
Solder plate thickness	5 μm or more
Package weight (g)	0.41 TYP.

Notes for Mounting the Surface Mount Type Package

The SOP, QFP, TSOP, TQFP, LQFP, SOJ, QFJ (PLCC), SHP, and BGA are surface mount type packages, which are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact Oki's responsible sales person on the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.
   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents cotained herein may be reprinted or reproduced without our prior permission.

9. MS-DOS is a registered trademark of Microsoft Corporation.

Printed in Japan