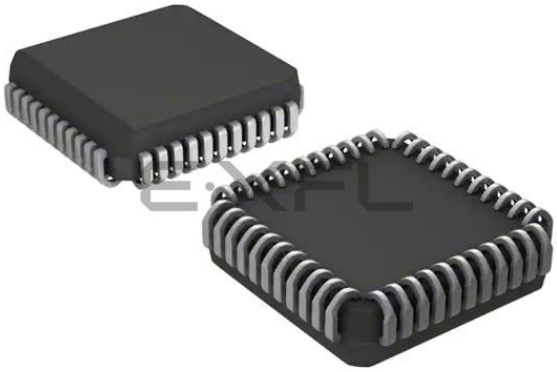

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	12MHz
Connectivity	CANbus, EBI/EMI, I²C, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p87c591vfa-00-512

12.2 PeliCAN structure

A 80C51 CPU Interface connects the PeliCAN to the internal bus of the P8xC591 microcontroller. Via five Special Function Registers CANADR, CANDAT, CANMOD, CANSTA and CANCON the CPU has access to the PeliCAN. The SFR will described later on.

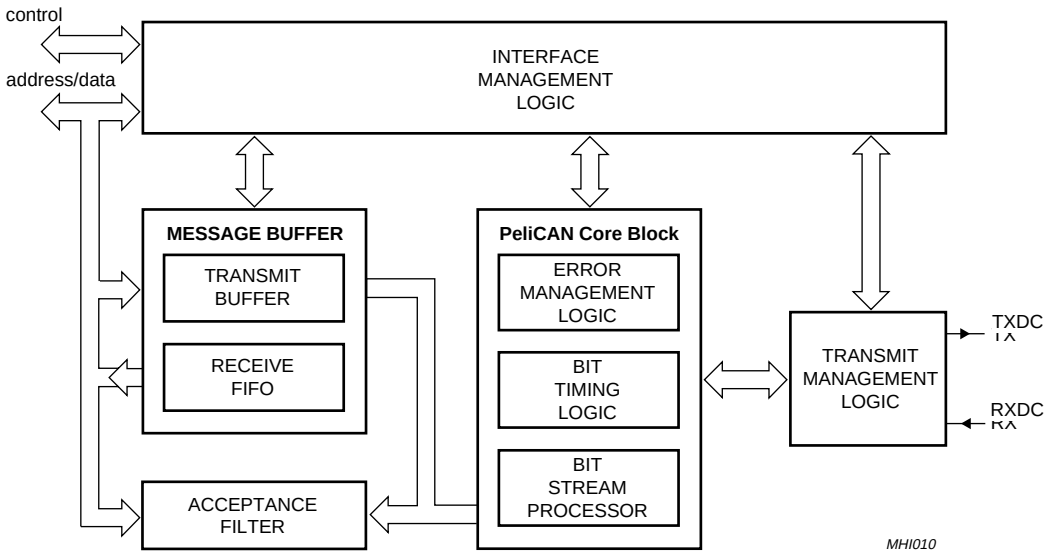

Fig.10 Block Diagram of the PeliCAN.

12.2.1 Interface Management Logic (IML)

The Interface Management Logic interprets commands from the CPU, controls addressing of the CAN Registers and provides interrupts and status information to the CPU. Additionally it drives the universal interface of the PeliCAN.

12.2.2 Transmit Buffer (TXB)

The Transmit Buffer is an interface between the CPU and the Bit Stream Processor (BSP) and is able to store a complete CAN message which should be transmitted over the CAN network. The buffer is 13 bytes long, written by the CPU and read out by the BSP or the CPU itself.

12.2.3 Receive Buffer (RXB, RXFIFO)

The Receive Buffer is an interface between the Acceptance Filter and the CPU and stores the received and accepted messages from the CAN Bus line. The Receive Buffer (RXB) represents a CPU-accessible 13-byte-window of the Receive FIFO (RXFIFO), which has a total length of 64 bytes. With the help of this FIFO the CPU is able to process one message while other messages are being received.

12.2.4 Acceptance Filter (ACF)

The Acceptance Filter compares the received identifier with the Acceptance Filter Table contents and decides whether this message should be accepted or not. In case of a positive acceptance test, the complete message is stored in the RXFIFO. The ACF contains 4 independent Acceptance Filter banks supporting extended and standard CAN frames with “change on the fly” feature.

12.2.5 Bit Stream Processor (BSP)

The Bit Stream Processor is a sequencer, controlling the data stream between the Transmit Buffer, RXFIFO and the CAN-Bus. It also performs the error detection, arbitration, stuffing and error handling on the CAN bus.

12.2.6 Error Management Logic (EML)

The EML is responsible for the error confinement of the transfer-layer modules. It gets error announcements from the BSP and then informs the BSP and IML about error statistics.

12.2.7 Bit Timing Logic (BTL)

The Bit Timing Logic monitors the serial CAN bus line and handles the Bus line-related bit timing. It synchronizes to the bit stream on the CAN Bus on a “recessive” to “dominant” Bus line transition at the beginning of a message (hard synchronization) and resynchronizes on further transitions during the reception of a message (soft synchronization). The BTL also provides programmable time segments to compensate for the propagation delay times and phase shifts (e.g., due to oscillator drifts) and to define the sampling time and the number of samples to be taken within a bit time.

12.2.8 Transmit Management Logic (TML)

The Transmit Management Logic provides the driver signals for the push-pull CAN TX transistor stage. Depending on the programmable output driver configuration the external transistors are switched on or off. Additionally a short circuit protection and the asynchronous float on hardware reset is performed here.

CAN ADDR.	OPERATING MODE				RESET MODE			
	READ		WRITE		READ		WRITE	
48	BANK 3	Acceptance Code 0	Acceptance Code 0		Acceptance Code 0		Acceptance Code 0	
49		Acceptance Code 1	Acceptance Code 1		Acceptance Code 1		Acceptance Code 1	
50		Acceptance Code 2	Acceptance Code 2		Acceptance Code 2		Acceptance Code 2	
51		Acceptance Code 3	Acceptance Code 3		Acceptance Code 3		Acceptance Code 3	
52		Acceptance Mask 0	Acceptance Mask 0		Acceptance Mask 0		Acceptance Mask 0	
53		Acceptance Mask 1	Acceptance Mask 1		Acceptance Mask 1		Acceptance Mask 1	
54		Acceptance Mask 2	Acceptance Mask 2		Acceptance Mask 2		Acceptance Mask 2	
55		Acceptance Mask 3	Acceptance Mask 3		Acceptance Mask 3		Acceptance Mask 3	
56	BANK 4	Acceptance Code 0	Acceptance Code 0		Acceptance Code 0		Acceptance Code 0	
57		Acceptance Code 1	Acceptance Code 1		Acceptance Code 1		Acceptance Code 1	
58		Acceptance Code 2	Acceptance Code 2		Acceptance Code 2		Acceptance Code 2	
59		Acceptance Code 3	Acceptance Code 3		Acceptance Code 3		Acceptance Code 3	
60		Acceptance Mask 0	Acceptance Mask 0		Acceptance Mask 0		Acceptance Mask 0	
61		Acceptance Mask 1	Acceptance Mask 1		Acceptance Mask 1		Acceptance Mask 1	
62		Acceptance Mask 2	Acceptance Mask 2		Acceptance Mask 2		Acceptance Mask 2	
63		Acceptance Mask 3	Acceptance Mask 3		Acceptance Mask 3		Acceptance Mask 3	
64 to 95	reserved (00)		-		reserved (00)		-	
96	(SFF) Rx Frame Info	(EFF) Rx Frame Info	-		(SFF) Rx Frame Info	(EFF) Rx Frame Info	(SFF) Rx Frame Info	(EFF) Rx Frame Info
97	Rx Identifier 1	Rx Identifier 1	-		Rx Identifier 1	Rx Identifier 1	Rx Identifier 1	Rx Identifier 1
98	Rx Identifier 2	Rx Identifier 2	-		Rx Identifier 2	Rx Identifier 2	Rx Identifier 2	Rx Identifier 2
99	Rx Data 1	Rx Identifier 3	-		Rx Data 1	Rx Identifier 3	Rx Data 1	Rx Identifier 3
100	Rx Data 2	Rx Identifier 4	-		Rx Data 2	Rx Identifier 4	Rx Data 2	Rx Identifier 4
101	Rx Data 3	Rx Data 1	-		Rx Data 3	Rx Data 1	Rx Data 3	Rx Data 1
102	Rx Data 4	Rx Data 2	-		Rx Data 4	Rx Data 2	Rx Data 4	Rx Data 2
103	Rx Data 5	Rx Data 3	-		Rx Data 5	Rx Data 3	Rx Data 5	Rx Data 3
104	Rx Data 6	Rx Data 4	-		Rx Data 6	Rx Data 4	Rx Data 6	Rx Data 4
105	Rx Data 7	Rx Data 5	-		Rx Data 7	Rx Data 5	Rx Data 7	Rx Data 5
106	Rx Data 8	Rx Data 6	-		Rx Data 8	Rx Data 6	Rx Data 8	Rx Data 6
107	(FIFO RAM) (1)	Rx Data 7	-		(FIFO RAM) (1)	Rx Data 7	(FIFO RAM) (1)	Rx Data 7
108	(FIFO RAM) (1)	Rx Data 8	-		(FIFO RAM) (1)	Rx Data 8	(FIFO RAM) (1)	Rx Data 8
109 to 111	reserved (00)		-		reserved (00)		-	
112	(SFF) Tx Frame Info	(EFF) Tx Frame Info	Tx Frame Info	Tx Frame Info	(SFF) Tx Frame Info	(EFF) Tx Frame Info	(SFF) Tx Frame Info	(EFF) Tx Frame Info
113	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1	Tx Identifier 1
114	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2	Tx Identifier 2

12.5.2 MODE REGISTER (MOD)

The contents of the Mode Register are used to change the behaviour of the CAN controller. Bits may be set or reset by the CPU that uses the Mode Register as a read / write memory. Reserved Bits are read as “0”.

Table 13 Mode Register (MOD) CAN Addr. 0 bit interpretation

BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD.7	TM	Test Mode; Note 1	1 (activated)	The TXDC pin will reflect the bit, detected on RXDC pin, with the next positive edge of the system clock. The RPM bit has no influence within this mode.
			0 (disabled)	
MOD.6	RIPM	Reserved.	–	–
MOD.5	RPM	Receive Polarity Mode	1 (high active)	RXD inputs are active high (dominant = 1).
			0 (low active)	RXD inputs are active low (dominant = 0).
MOD.4	SM	Sleep Mode; Note 2	1 (high active))	The CAN controller enters Sleep Mode if no CAN interrupt is pending and there is no bus activity.
			0 (low active)	
MOD.3	–	reserved	–	–
MOD.2	STM	Self Test Mode; Note 1	1 (self test)	In this mode a full node test is possible without any other active node on the bus using the Self Reception Request command. The CAN controller will perform a successful transmission, even if there is no acknowledge received.
			0 (normal)	An acknowledge is required for successful transmission.
MOD.1	LOM	Listen Only Mode; Notes 1 and 3	1 (reset)	In this mode the CAN would give no acknowledge to the CAN bus, even if a message is received successfully. No active error flags are driven to the bus. The error counters are stopped at the current value.
			0 (normal)	Normal communication.
MOD.0	RM	Reset Mode; Note 4	1 (reset)	Setting the Reset Mode bit results in aborting the current transmission/reception of a message and entering the Reset Mode.
			0 (normal)	On the’1’-to-’0’ transition of the Reset Mode bit, the CAN controller returns to the Operating Mode.

Notes

1. A write access to the bits MOD.1, MOD.2, MOD.5, MOD.6 and MOD.7 is possible only, if the Reset Mode is entered previously.
2. The PeliCAN Block will enter Sleep Mode, if the Sleep Mode bit is set ‘1’ (sleep), there is no bus activity and no interrupt is pending. Setting of SM with at least one of the previously mentioned exceptions valid will result in a wake-up interrupt. The CAN controller will wake up if SM is set LOW (wake-up) or there is bus activity. On wake-up, a Wake-up Interrupt is generated. A sleeping CAN controller which wakes up due to bus activity will not be able to receive this message until it detects 11 consecutive recessive bits (Bus-Free sequence). Note that setting of SM is not possible in Reset Mode. After clearing of Reset Mode, setting of SM is possible first, when Bus-Free is detected again.
3. This mode of operation forces the CAN controller to be error passive. Message Transmission is not possible. The Listen Only Mode can be used e.g. for software driven bit rate detection and “hot plugging”.

12.5.5 INTERRUPT REGISTER (IR)

The Interrupt Register allows the identification of an interrupt source. When one or more bits of this register are set, a CAN interrupt will be indicated to the CPU. After this register is read by the CPU all bits are reset except of the Receive Interrupt bit.
The Interrupt Register appears to the CPU as a read only memory.

Table 16 Interrupt Register (IR) CAN Addr. 3, bit interpretation

BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	BEI	Bus Error Interrupt	1 (set) 0 (reset)	This bit is set when the CAN controller detects an error on the CAN Bus and the BEIE bit is set within the Interrupt Enable Register. After a bus error interrupt event this interrupt is locked until the Error Code Capture Register is read out once.
IR.6	ALI	Arbitration Lost Interrupt	1 (set) 0 (reset)	This bit is set when the CAN controller has lost arbitration and becomes a receiver and the ALIE bit is set within the Interrupt Enable Register. After an arbitration lost interrupt event this interrupt is locked until the Arbitration Lost Capture Register is read out once.
IR.5	EPI	Error Passive Interrupt	1 (set) 0 (reset)	This bit is set whenever the CAN controller has reached the Error Passive Status (at least one error counter exceeds the CAN protocol defined level of 127) or if the CAN controller is in Error Passive Status and enters the Error Active Status again and the EPIE bit is set within the Interrupt Enable Register.
IR.4	WUI	Wake-Up Interrupt; Note 1	1 (set) 0 (reset)	This bit is set when the CAN controller is sleeping and bus activity is detected and the WUIE bit is set within the Interrupt Enable Register.
IR.3	DOI	Data Overrun Interrupt	1 (set) 0 (reset)	This bit is set on a 0-to-1 change of the Data Overrun Status bit, when the Data Overrun Interrupt Enable is set to ‘1’ (enabled).
IR.2	EI	Error Interrupt	1 (set) 0 (reset)	This bit is set on every change (set and clear) of either the Error Status or Bus Status bits if the Error Interrupt Enable is set to ‘1’ (enabled).
IR.1	TI	Transmit Interrupt; Note 2	1 (set) 0 (reset)	This bit is set whenever the Transmit Buffer Status changes from ‘0’ to ‘1’ (released) and Transmit Interrupt Enable is set to ‘1’ (enabled).
IR.0	RI	Receive Interrupt; Note 2	1 (set) 0 (reset)	This bit is set whenever the RXFIFO is filled with more bytes than specified in the Rx Interrupt Level register or a message has passed an acceptance filter which is set to “high priority” and the RIE bit is set within the Interrupt Enable Register.

Notes to Table 16**:**

1. A Wake-Up Interrupt is also generated, if the CPU tries to set the Sleep bit while the CAN controller is involved in bus activities or a CAN Interrupt is pending.
2. In order to support high priority messages, the Receive Interrupt is forced immediately upon a received message, which has passed successfully an acceptance filter with high priority (see acceptance filter section). As long as only messages are received via low priority acceptance filters, the receive interrupt is not forced until the FIFO is filled with more bytes than programmed in the Rx Interrupt Level Register.

   The Receive Interrupt Bit is not cleared upon a read access to the Interrupt Register. Giving the Command “Release Receive Buffer” will clear RI temporarily. If there is another message available within the FIFO after the release command, RI is set again. Otherwise RI keeps cleared.

12.5.6 Interrupt Enable Register (IER)

The register allows to enable different types of interrupt sources which are signalled to the CPU. The Interrupt Enable Register appears to the CPU as a read / write memory.

Table 17 Interrupt Enable Register (IER) CAN Addr. 4, bit interpretation

BIT	SYMBOL	NAME	VALUE	FUNCTION
IER.7	BEIE	Bus Error Interrupt Enable	1 (enabled) 0 (disabled)	If a bus error has been detected, the CAN controller requests the respective interrupt.
IER.6	ALIE	Arbitration Lost Interrupt Enable	1 (enabled) 0 (disabled)	If the CAN controller has lost arbitration, the respective interrupt is requested.
IER.5	EPIE	Error Passive Interrupt Enable	1 (enabled) 0 (disabled)	If the error status of the CAN controller changes from error active to error passive or vice versa, the respective interrupt is requested.
IER.4	WUIE	Wake-Up Interrupt Enable	1 (enabled) 0 (disabled)	If the sleeping CAN controller wakes up, the respective interrupt is requested.
IER.3	DOIE	Data Overrun Interrupt Enable	1 (enabled) 0 (disabled)	If the Data Overrun Status bit is set (see Status Register), the CAN controller requests the respective interrupt.
IER.2	EIE	Error Interrupt Enable	1 (enabled) 0 (disabled)	If the Error or Bus Status change (see Status Register), the CAN controller requests the respective interrupt.
IER.1	TIE	Transmit Interrupt Enable	1 (enabled) 0 (disabled)	When a message has been successfully transmitted or the Transmit Buffer is accessible again, (e.g. after an Abort Transmission command) the CAN controller requests the respective interrupt.
IER.0	RIE	Receive Interrupt Enable	1 (enabled) 0 (disabled)	When the Receive Buffer Status is ‘full’ the CAN controller requests the respective interrupt.

12.5.17.5 Dual Filter Configuration

In this filter configuration two short filters could be defined. A received message is compared with both filters to decide, whether this message should be copied into the Receive Buffer or not. If at least one of the filters signals an acceptance, the received message becomes valid. The bit correspondences between the filter bytes and the message bytes depends on the currently received Frame Format.

Dual Filter Standard Frame:

If the Standard Frame Format is selected, the two defined filters are different. The first filter compares the complete Standard Identifier including the RTR bit **and** the first Data Byte of the message. The second filter just compares the complete Standard Identifier including the RTR bit.

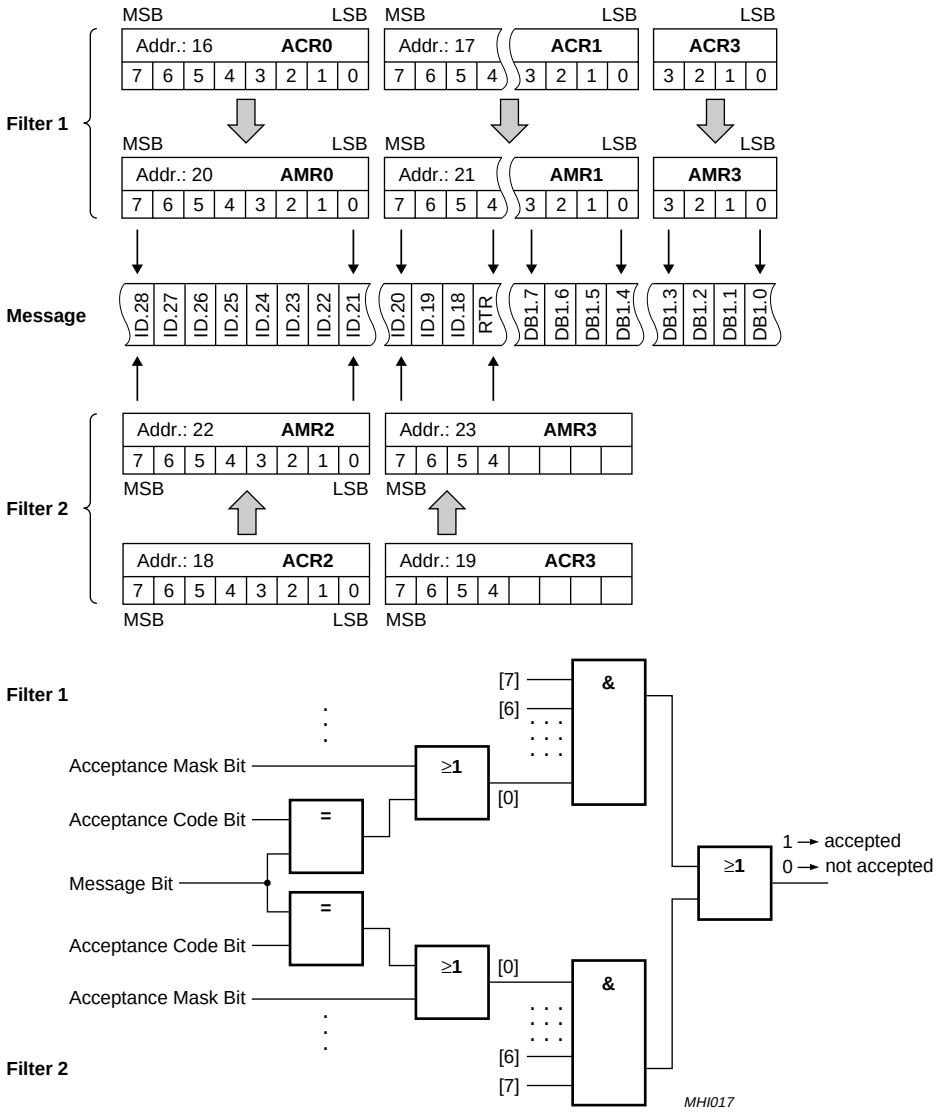

Fig.17 Dual Filter Configuration, receiving Standard Frame Messages.

12.5.19.1 Descriptor File of the Receive Buffer

Identifier, Frame Format, Remote Transmission Request bit and Data Length Code have the same meaning as described in the Transmit Buffer.

Standard Frame Format (SFF)

Addr. 96	RX Frame Information						
7	6	5	4	3	2	1	0
FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 97	RX Identifier 1						
7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 98	RX Identifier 2						
7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	RTR	0	0	0	0

Meaning of the Receive Buffer Bits:

ID.x	Identifier bit x
FF	Frame Format
RTR	Remote Transmission Request
DLC.x	Data Length Code bit x

Extended Frame Format (EFF)

Addr. 96	RX Frame Information						
7	6	5	4	3	2	1	0
FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0

Addr. 97	RX Identifier 1						
7	6	5	4	3	2	1	0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Addr. 98	RX Identifier 2						
7	6	5	4	3	2	1	0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Addr. 99	RX Identifier 3						
7	6	5	4	3	2	1	0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Addr. 100	RX Identifier 4						
7	6	5	4	3	2	1	0
ID.4	ID.3	ID.2	ID.1	ID.0	RTR	0	0

Fig.22 Bit Layout Receive Buffer.

Note:

The received Data Length Code located in the Frame Information Byte represents the real sent Data Length Code, which may be greater than 8 (depends on transmitting CAN node). Nevertheless, the maximum number of received data bytes is 8. This should be taken into account by reading a message from the Receive Buffer.

It depends on the data length how many CAN messages can fit in the RXFIFO at one time. If there is not enough space for a new message within the RXFIFO, the CAN controller generates a Data Overrun condition the moment this message becomes valid and the acceptance test was positive. A message that is partly written into the RXFIFO, when the Data Overrun situation occurs, is deleted. This situation is signalled to the CPU via the Status Register and the Data Overrun Interrupt, if enabled.

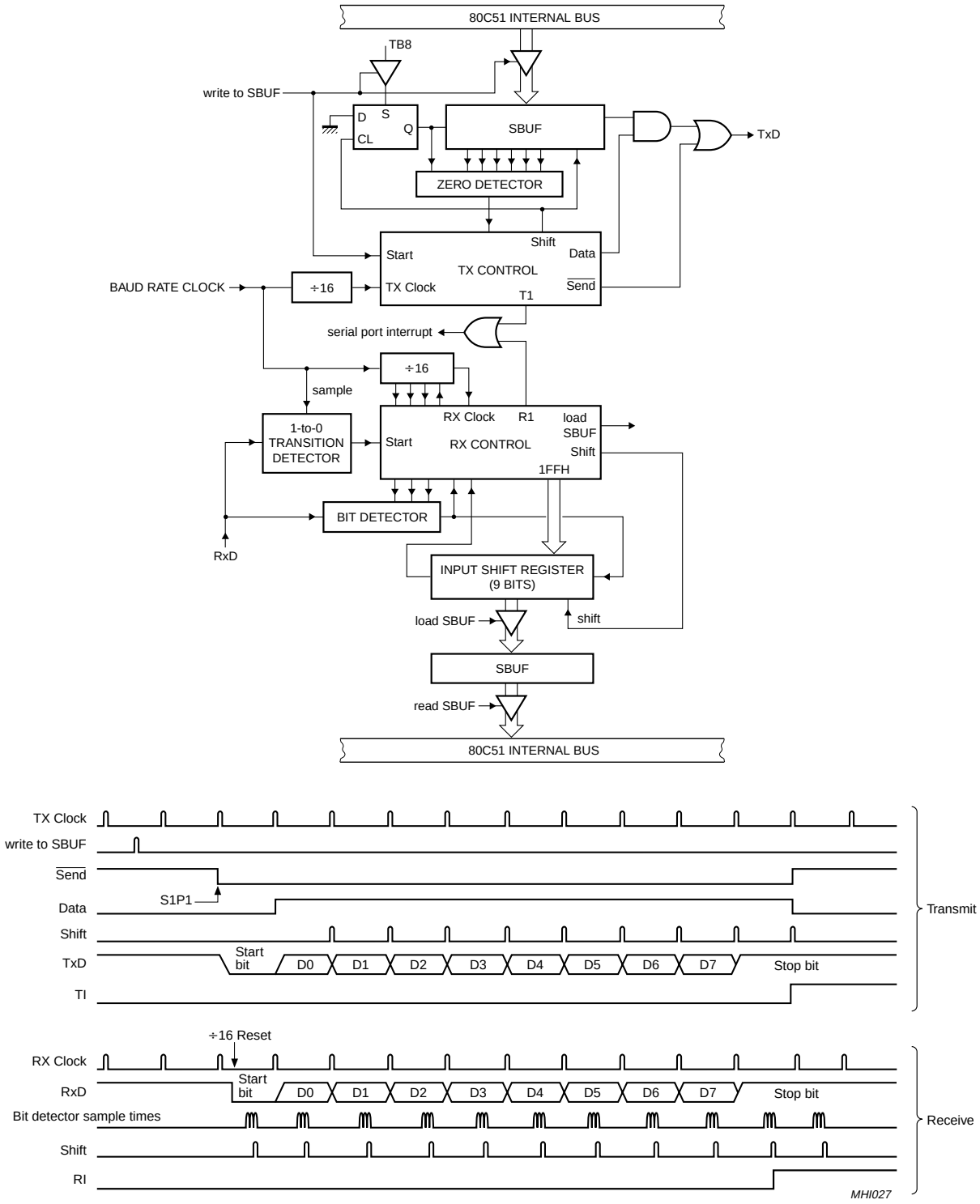

Fig.26 Serial Port Mode 1.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. All of the slaves may be contacted by using the Broadcast address. Two Special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR	=	1100 0000
	SADEN	=	1111 1101
	Given	=	1100 00X0
Slave 1	SADDR	=	1100 0000
	SADEN	=	1111 1110
	Given	=	1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two salves. Slave 0 requires as 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for Slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for Slave 0) and bit 1 = 0 (for Slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select Slaves 1 and 2 while excluding Slave 0:

Slave 0	SADDR	=	1100 0000
	SADEN	=	1111 1001
	Given	=	1100 0XX0
Slave 1	SADDR	=	1110 0000
	SADEN	=	1111 1010
	Given	=	1110 0X0X
Slave 2	SADDR	=	1110 0000
	SADEN	=	1111 1100
	Given	=	1110 00XX

In the above example the differentiation among the 3 Slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude Slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are trended as don't cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are leaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

15 SIO1, I²C SERIAL IO

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Bidirectional data transfer between masters and slaves
- Multimaster bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- The I²C bus may be used for test and diagnostic purposes

The I/O pins P1.6 and P1.7 must be set to Open Drain (SCL and SDA).

The 8xC591 on-chip I²C logic provides a serial interface that meets the I²C bus specification. The SIO1 logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I²C bus.

15.2.12.4 SI, the Serial Interrupt Flag

SI = “1”: When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = 0: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

15.2.12.5 AA, the Assert Acknowledge flag

AA = “1”: If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The “own slave address” has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 40). When SI is cleared, SIO1 leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I^2C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the parts own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

15.2.12.6 CR0, CR1, and CR2, the Clock Rate Bits

These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 57.

A 12.5 kHz bit rate may be used by devices that interface to the I^2C bus via standard I/O port lines which are software driven and slow. 100kHz is usually the maximum bit rate and can be derived from a 16 MHz, 12 MHz, or a 6 MHz oscillator. A variable bit rate (0.5 kHz to 62.5 kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 57 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100 kHz.

SI01 EQUATE LIST

```
LOC     OBJ     SOURCE
                !*************************************************************************************************************
                ! LOCATIONS OF THE SI01 SPECIAL FUNCTION REGISTERS!
                !*************************************************************************************************************
00D8            S1CON           -0xd8
00D9            S1STA           -0xd9
00DA            S1DAT           -0xda
00DB            S1ADR           -0xdb

00A8            IEN0            -0xa8
00B8            IP0             02b8
                !*************************************************************************************************************
                ! BIT LOCATIONS
                !*************************************************************************************************************
00DD            STA             -0xdd           ! STA bit in S1CON
00BD            SI01HP          -0xbd           ! IP0, SI01 Priority bit
                !*************************************************************************************************************
                ! IMMEDIATE DATA TO WRITE INTO REGISTER S1CON
                !*************************************************************************************************************
00D5            ENS1_NOTSTA_STO_NOTSI_AA_CR0            -0xd5   ! Generates STOP
                                                                ! (CR0 = 100kHz @ fOSC =
                                                                !  6 MHz)
00C5            ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0         -0xc5   ! Releases BUS and ACK
                                                                !
00C1            ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0      -0xc1   ! Releases BUS and
                                                                ! NOT ACK
00E5            ENS1_STA_NOTSTO_NOTSI_AA_CR0            -0xe5   ! Releases BUS and set
                                                                ! STA
                !*************************************************************************************************************
                ! GENERAL IMMEDIATE DATA
                !*************************************************************************************************************
0031            OWNSLA          -0x31           ! Own SLA+General Call
                                                ! must be written into S1ADR
00A0            ENSI01          -0xa0           ! EA+ES1, enable SIO1 interrupt
                                                ! must be written into IEN0
0001            PAG1            -0x01           ! select PAG1 as HADD
00C0            SLAW            -0xc0           ! SLA+W to be transmitted
00C1            SLAR            -0xc1           ! SLA+R to be transmitted
0018            SELRB3          -0x18           ! Select Register Bank 3
```

```
LOC     OBJ       SOURCE
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 48, SLA+R have been transmitted, NOT ACK received.
                  ! ACTION : STOP condition will be generated.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     mts48
                  .base     0x148
0148    75D8D5    STOP:               mov       S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                          ! set STO, clr SI
014B    D0D0                          pop       psw
014D     32                           reti
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 50, DATA have been received, ACK returned.
                  ! ACTION : Read DATA of S1DAT. DATA will be received, if it is last DATA then NOT ACK will be returned
                  !              else ACK will be returned.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     mrs50
                  .base     0x150
0150    75D018                        mov       psw,#SELRB3
0153    A6DA                          mov       @r0,S1DAT                 ! Read received DATA
0155    01C0                          ajmp      REC1
                  .sect     mrs50s
                  .base     0xc0

00C0    D55205    REC1:               djnz      NUMBYTMST,NOTLDAT2
00C3    75D8C1                        mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                                          ! clr SI,AA
00C6    8003                          sjmp      RETmr
00C8    75D8C5    NOTLDAT2:           mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                          ! clr SI, set AA
00CB    08        RETmr:              inc       r0
00CC    D0D0                          pop       psw
00CE    32                            reti
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 58, DATA have been received, NOT ACK returned.
                  ! ACTION : Read DATA of MASTER STATE SERVICE ROUTINESS1DAT and generate a STOP condition.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     mrs58
                  .base     0x158
0158    75D018                        mov       psw,#SELRB3
015B    A6DA                          mov       @R0,S1DAT
015D    80E9                          sjmp      STOP
```

```
LOC     OBJ       SOURCE
                  !*************************************************************************************************************
                  ! SLAVE RECEIVER STATE SERVICE ROUTINES
                  !*************************************************************************************************************

                  !-------------------------------------------------------------
                  ! STATE   : 60, Own SLA+W have been received, ACK returned.
                  ! ACTION : DATA will be recMASTER STATE SERVICE ROUTINESeived and ACK returned.
                  !-------------------------------------------------------------
                  .sect     srs60
                  .base     0x160
0160    75D8C5                        mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                        ! clr SI, set AA
0163    75D018                        mov       psw,#SELRB3
0166    01D0                          ajmp      INITSRD
                  .sect     insrd
                  .base     0xd0

00D0    7840      INITSRD:            mov       r0,#SRD
00D2    7908                          mov       r1,#8
00D4    D0D0                          pop       psw
00D6    32                            reti

                  !-------------------------------------------------------------
                  ! STATE   : 68, Arbitration lost in SLA and R/W as MST Own SLA+W have been received, ACK returned
                  ! ACTION : DATA will be received and ACK returned. STA is set to restart MST mode after the bus is free again.
                  !-------------------------------------------------------------
                  .sect     srs68
                  .base     0x168
0168    75D8E5                        mov       S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
016B    75D018                        mov       psw,#SELRB3
016E    01D0                          ajmp      INITSRD

                  !-------------------------------------------------------------
                  ! STATE   : 70, General call has been received, ACK returned.
                  ! ACTION : DATA will be received and ACK returned.
                  !-------------------------------------------------------------
                  .sect     srs70
                  .base     0x170
0170    75D8C5                        mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                        ! clr SI, set AA
0173    75D018                        mov       psw,#SELRB3           ! Initialize SRD counter
0176    01D0                          ajmp      initsrd

                  !-------------------------------------------------------------
                  ! STATE   : 78, Arbitration lost in SLA+R/W as MST. General call has been received, ACK returned.
                  ! ACTION : DATA will be received and ACK returned. STA is set to restart MST mode after the bus is free again.
                  !-------------------------------------------------------------
                  .sect     srs78
                  .base     0x178
0178    75D8E5                        mov       S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
017B    75D018                        mov       psw,#SELRB3           ! Initialize SRD counter
017E    01D0                          ajmp      INITSRD
```

```
LOC     OBJ       SOURCE
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 80, Previously addressed with own SLA. DATA received, ACK returned.
                  ! ACTION : Read DATA.
                  !          IF received DATA was the last
                  !          THEN superfluous DATA will be received and NOT ACK returned
                  !          ELSE next DATA will be received and ACK returned.!
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs80
                  .base     0x180
0180    75D018                      mov       psw,#SELRB3
0183    A6DA                        mov       @r0,S1DAT                 ! Read received DATA
0185    01D8                        ajmp      REC2

                  .sect     srs80s
                  .base      0xd8

00D8    D906      REC2:             djnz       r1,NOTLDAT3
00DA    75D8C1    LDAT:             mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                                        ! clr SI,AA
00DD    D0D0                        pop       psw
00DF    32                          reti
00E0    75D8C5    NOTLDAT3:         mov        S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                        ! clr SI, set AA
00E3    08                          inc        r0
00E4    D0D0      RETsr:            pop       psw
00E6    32                          reti
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 88, Previously addressed with own SLA. DATA received NOT ACK returned.
                  ! ACTION : No save of DATA, Enter NOT addressed SLV mode.
                  !          Recognition of own SLA. General call recognized, if S1ADR. 01.!
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs88
                  .base     0x188
0188    75D8C5                       mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                        ! clr SI, set AA
018B    01E4                        ajmp      RETsr
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 90, Previously addressed with general call. DATA has been received, ACK has been returned.
                  ! ACTION : Read DATA.
                  !          After General call only one byte will be received with ACK the second DATA
                  !          will be received with NOT ACK.
                  !          DATA will be received and NOT ACK returned.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs90
                  .base     0x190
0190    76D018                      mov       psw,#SELRB3
0193    A6DA                        mov       @r0,S1DAT                 ! Read received DATA
0195    01DA                        ajmp      LDAT
```

16.1.7 Timer T2 Interrupt Flag Register TM2IR

Seven of the eight Timer T2 interrupt flags are located in special function register TM2IR (see Section 16.1.7.1). The eights flag is TM2CON.4.

The CT0I and CT1I flags are set during S4 of the cycle in which the contents of Timer T2 are captured. CT0I is scanned by the interrupt logic during S2, and CT1I is scanned during S3. CT2I and CT3I are set during S6 and are scanned during S4 and S5. The associated interrupt requests are recognized during the following cycle. If these flags are polled, a transition at CT0I or CT1I will be recognized one cycle before a transition on CT2I or CT3I since registers are read during S5. The CMI0, CMI1 and CMI2 flags are set during S6 of the cycle following a match. CMI0 is scanned by the interrupt logic during S2; CMI1 and CMI2 are scanned during S3 and S4. A match of CMI0 and CMI1 will be recognized by the interrupt logic (or by polling the flags) two cycles after the match takes place. A match of CMI2 will cause no interrupt, this flag can be polled only.

The 16-bit overflow flag (T2OV) and the byte overflow flag (T2BO) are set during S6 of the cycle in which the overflow occurs. These flags are recognized by the interrupt logic during the next cycle. Special function register IP1 (see Section 16.1.7.2) is used to determine the Timer T2 interrupt priority. Setting a bit high gives that function a high priority, and setting a bit low gives the function a low priority. The functions controlled by the various bits of the IP1 register are shown in Section 16.1.6.2.

16.1.7.1 Interrupt Flag Register (TM2IR)

Table 78 Interrupt flag register (address C8H)

7	6	5	4	3	2	1	0
T2OV	CMI2/CAN	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0

Table 79 Description of TM2IR bits

BIT	SYMBOL	DESCRIPTION
7	T2OV	T2: 16-bit overflow interrupt flag.
6	CMI2/CAN	CM2: flag (for polling only). CAN: CAN interrupt flag (polling only).
5	CMI1	CM1: interrupt flag.
4	CMI0	CM0: interrupt flag.
3	CTI3	CT3: interrupt flag.
2	CTI2	CT2: interrupt flag.
1	CTI1	CT1: interrupt flag.
0	CTI0	CT0: interrupt flag.

16.1.7.2 Interrupt Priority Register 1 (IP1)

Table 80 Interrupt Priority Register 1 (address F8H)

7	6	5	4	3	2	1	0
PT2	PCAN	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

Table 81 Description of IP1 bits

BIT	SYMBOL	DESCRIPTION
7	PT2	T2 overflow interrupt(s) priority level.
6	PCAN	CAN interrupt priority level.
5	PCM1	T2 comparator 1 priority interrupt level.
4	PCM0	T2 comparator 0 priority interrupt level.
3	PCT3	T2 capture register 3 priority interrupt level.
2	PCT2	T2 capture register 2 priority interrupt level.
1	PCT1	T2 capture register 1 priority interrupt level.
0	PCT0	T2 capture register 0 priority interrupt level.

Table 110 Instruction set description: Boolean variable manipulation, Program and machine control

MNEMONIC		DESCRIPTION	BYTES	CYCLES	OPCODE (HEX)
Boolean variable manipulation					
CLR	C	Clear carry flag	1	1	C3
CLR	bit	Clear direct bit	2	1	C2
SETB	C	Set carry flag	1	1	D3
SETB	bit	Set direct bit	2	1	D2
CPL	C	Complement carry flag	1	1	B3
CPL	bit	Complement direct bit	2	1	B2
ANL	C,bit	AND direct bit to carry flag	2	2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2	2	B0
ORL	C,bit	OR direct bit to carry flag	2	2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2	2	A0
MOV	C,bit	Move direct bit to carry flag	2	1	A2
MOV	bit,C	Move carry flag to direct bit	2	2	92
Program and machine control					
ACALL	addr11	Absolute subroutine call	2	2	•1
LCALL	addr16	Long subroutine call	3	2	12
RET		Return from subroutine	1	2	22
RETI		Return from interrupt	1	2	32
AJMP	addr11	Absolute jump	2	2	♦1
LJMP	addr16	Long jump	3	2	02
SJMP	rel	Short jump (relative address)	2	2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2	73
JZ	rel	Jump if A is zero	2	2	60
JNZ	rel	Jump if A is not zero	2	2	70
JC	rel	Jump if carry flag is set	2	2	40
JNC	rel	Jump if carry flag is not set	2	2	50
JB	bit,rel	Jump if direct bit is set	3	2	20
JNB	bit,rel	Jump if direct bit is not set	3	2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3	2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3	2	B4
CJNE	Rr,#data,rel	Compare immediate to register and jump if not equal	3	2	B*
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2	2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2	D5
NOP		No operation	1	1	00

23 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134); Note 1

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	Voltage on V_{DD} to V_{SS} and SCL, SDA to V_{SS}	–0.5	+6.5	V
V_I	Input voltage on any other pin to V_{SS}	–0.5	V_{DD} + 0.5	V
I_I, I_O	Input/output current on any I/O pin	–	5	mA
P_{tot}	Total power dissipation (Note 2)	–	1.0	W
T_{stg}	Storage temperature range	–65	+150	°C
T_{amb}	Operating ambient temperature range: P8xC591VFx	 –40	 +85	 °C
V_{PP}	Voltage on $\overline{EA}/V_{PP}$ to V_{SS}	–0.5	+13	V

Notes

1. The following applies to the Absolute Maximum Ratings:
   a) Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the Chapters 24 and 25 of this specification is not implied.
   b) This product includes circuitry specifically designed for the protection of its internal devices from the damaging effect of excessive static charge. However, its suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
   c) Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.
2. This value is based on the maximum allowable die temperature and the thermal resistance of the package, not on device power consumption.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
Outputs					
V_{OL}	LOW level output voltage Ports 1, 2, 3 (except P1.0, P1.6, P1.7)	I_{OL} = 1.6 mA; see Note 8		0.4	V
V_{OL1}	LOW level output voltage Port 0, ALE, $\overline{PSEN}$, $\overline{RST}$, PWM0, PWM1	I_{OL} = 3.2; see Note 8		0.4	V
V_{OL2}	LOW level output voltage P1.6, P1.7	I_{OL} = 3.0 mA; see Note 8	–	0.4	V
V_{OL3}	LOW level output voltage P1.0 and P1.1	I_{OL} = 8.0 mA	–	0.3 V_{DD}	V
V_{OH}	HIGH level output voltage Ports 1, 2, 3 in pseudo-bidirectional output mode (except P1.1, P1.6 and P1.7	I_{OH} = -60 µA	2.4		V
V_{OH1}	HIGH level output voltage Port 0 and Port 2 in external bus mode, Port 2 in push-pull mode, ALE, $\overline{PSEN}$, PWM0, PWM1	I_{OH} = –3.2 mA; see Note 9	V_{DD} –0.7		V
V_{OH2}	HIGH level output voltage, P1.0 and P1.1	I_{OH} = –1.6 mA	0.7 V_{DD}		V
V_{OH3}	HIGH level output voltage, Ports 1, 2, 3 in push-pull output mode (except P1.0, P1.1, P1.6, P1.7)	I_{OH} = –1.6 mA	V_{DD} –0.7		V
$R_{\overline{RST}}$	$\overline{RST}$ pull-up resistor		40	225	kΩ
$C_{I/O}$	I/O pin capacitance	test frequency = 1 MHz; T_{amb} = 25 °C	–	15	pF
Analog inputs					
AV_{IN}	analog input voltage		AV_{SS} – 0.2	V_{DD} + 0.2	V
AV_{ref+}	reference voltage		–	V_{DD} + 0.2	V
R_{REF}	resistance between AV_{ref+} and AV_{SS}		10	50	kΩ
C_{IA}	analog input capacitance		–	15	pF
t_{ADS}	sampling time		–	5 tcy; Note 1 8 tcy	µs µs
t_{ADC}	conversion time (including sampling time)		–	24 tcy; Note 1 50 tcy	µs µs
DL_e	differential non-linearity	see Notes 10, 11, 12	–	±1	LSB
IL_{e8}	integral non-linearity (8-bit mode)		–	±1; Note 1	LSB
IL_e	integral non-linearity	see Notes 10, 13	–	±2	LSB
OS_{e8}	offset error (8-bit mode)		–	±1; Note 1	LSB
OS_e	offset error	see Notes 10, 15	–	±2	LSB
G_e	gain error		–	±0.4	%
A_e	absolute voltage error	see Notes 10, 16	–	±3	LSB
M_{ctc}	channel-to-channel matching		–	±1	LSB
C_t	crosstalk between analog inputs of Port 1	0 to 100 kHz; see Notes 17, 18	–	–60	dB

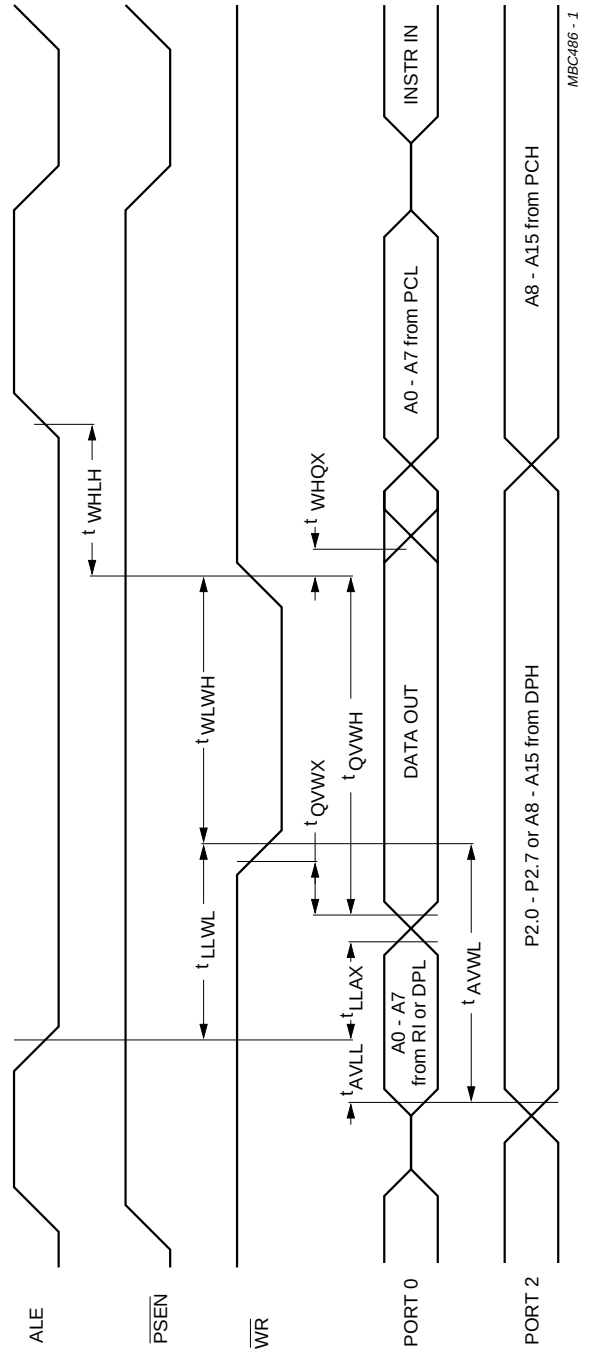

Fig.57 External data memory write cycle.