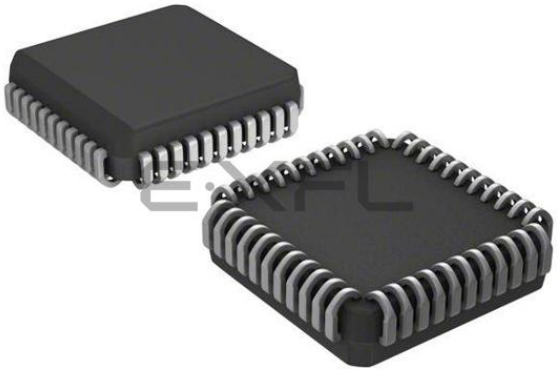
Microchip Technology - AT89C51CC03UA-SLSUM Datasheet



Welcome to E-XFL.COM

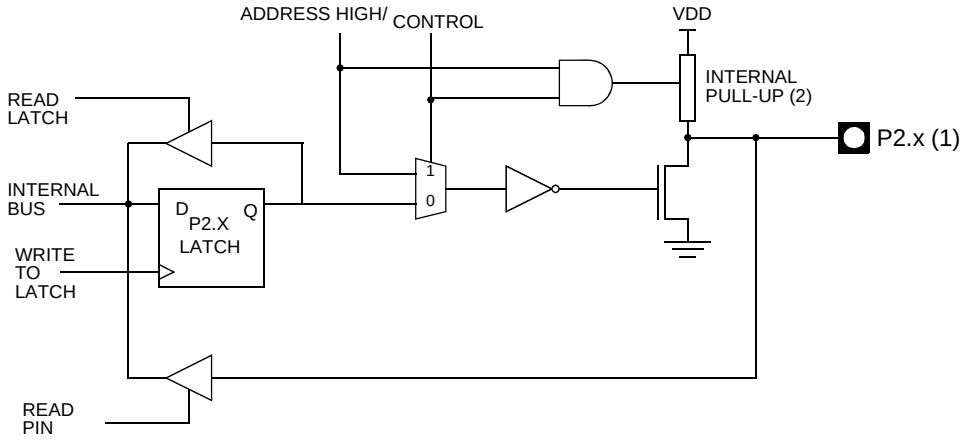
What is "Embedded - Microcontrollers"?

"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	Active
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.6x16.6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc03ua-slsum

Figure 3. Port 2 Structure



Notes: 1. Port 2 is precluded from use as general-purpose I/O Ports when as address/data bus drivers.
2. Port 2 internal strong pull-ups FET (P1 in FiGURE) assist the logic-one output for memory bus cycle.

When Port 0 and Port 2 are used for an external memory cycle, an internal control signal switches the output-driver input from the latch output to the internal address/data line.

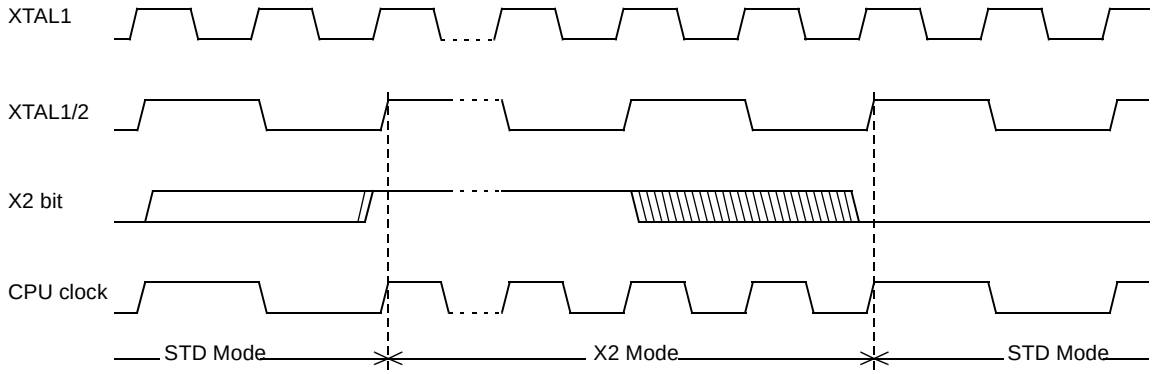
Read-Modify-Write Instructions

Some instructions read the latch data rather than the pin data. The latch based instructions read the data, modify the data and then rewrite the latch. These are called "Read-Modify-Write" instructions. Below is a complete list of these special instructions (see Table). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Instruction	Description	Example
ANL	logical AND	ANL P1, A
ORL	logical OR	ORL P2, A
XRL	logical EX-OR	XRL P3, A
JBC	jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	complement bit	CPL P3.0
INC	increment	INC P2
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

It is not obvious the last three instructions in this list are Read-Modify-Write instructions. These instructions read the port (all 8 bits), modify the specifically addressed bit and

Figure 6. Mode Switching Waveforms



Note: In order to prevent any incorrect operation while operating in the X2 mode, users must be aware that all peripherals using the clock frequency as a time reference (UART, timers...) will have their time reference divided by two. For example a free running timer generating an interrupt every 20 ms will then generate an interrupt every 10 ms. A UART with a 4800 baud rate will have a 9600 baud rate.

Table 3. CKCON1 Register

CKCON1 (S:9Fh)
Clock Control Register 1

7	6	5	4	3	2	1	0
							SPIX2

Bit Number	Bit Mnemonic	Description
7-1	-	**Reserved** The value read from these bits is indeterminate. Do not set these bits.
0	SPIX2	**SPI clock** (1) Clear to select 6 clock periods per peripheral clock cycle. Set to select 12 clock periods per peripheral clock cycle.

Note: 1. This control bit is validated when the CPU clock bit X2 is set; when X2 is low, this bit has no effect.

Reset Value = 0000 0000b

Internal Space

Lower 128 Bytes RAM

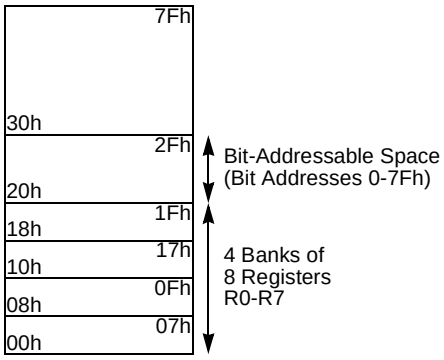
The lower 128 Bytes of RAM (see Figure 8) are accessible from address 00h to 7Fh using direct or indirect addressing modes. The lowest 32 Bytes are grouped into 4 banks of 8 registers (R0 to R7). Two bits RS0 and RS1 in PSW register (see Figure 6) select which bank is in use according to Table 4. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing, and can be used for context switching in interrupt service routines.

Table 4. Register Bank Selection

RS1	RS0	Description
0	0	Register bank 0 from 00h to 07h
0	1	Register bank 0 from 08h to 0Fh
1	0	Register bank 0 from 10h to 17h
1	1	Register bank 0 from 18h to 1Fh

The next 16 Bytes above the register banks form a block of bit-addressable memory space. The C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00h to 7Fh.

Figure 9. Lower 128 Bytes Internal RAM Organization



Upper 128 Bytes RAM

The upper 128 Bytes of RAM are accessible from address 80h to FFh using only indirect addressing mode.

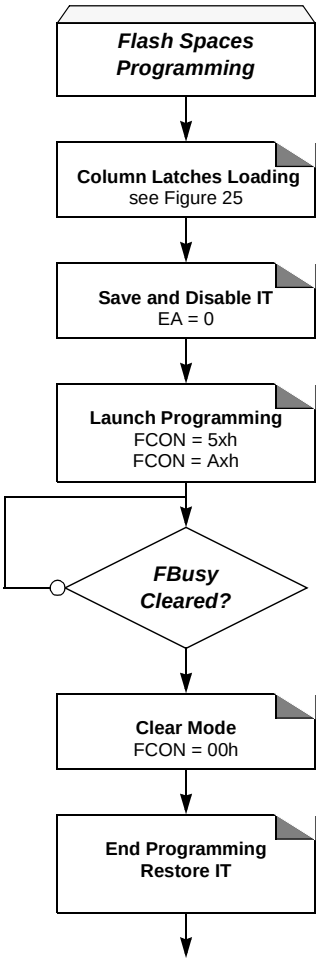
Expanded RAM

The on-chip 2048 Bytes of expanded RAM (ERAM) are accessible from address 0000h to 07FFh using indirect addressing mode through MOVX instructions. In this address range, the bit EXTRAM in AUXR register is used to select the ERAM (default) or the XRAM. As shown in Figure 8 when EXTRAM = 0, the ERAM is selected and when EXTRAM = 1, the XRAM is selected.

The size of ERAM can be configured by XRS2-0 bit in AUXR register (default size is 2048 Bytes).

Note: Lower 128 Bytes RAM, Upper 128 Bytes RAM, and expanded RAM are made of volatile memory cells. This means that the RAM content is indeterminate after power-up and must then be initialized properly.

Figure 26. Flash and Extra Row Programming Procedure

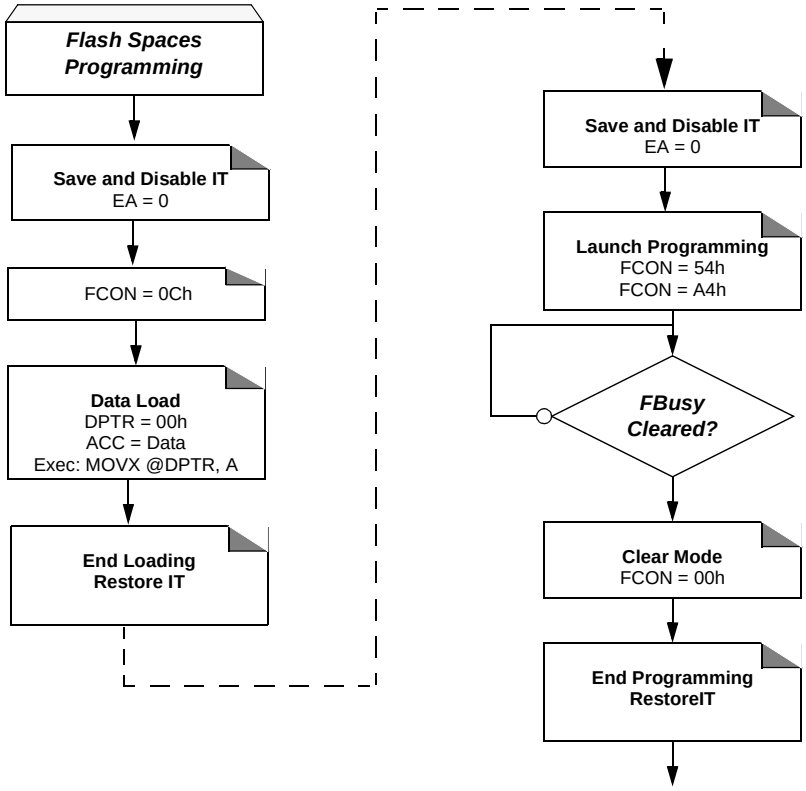


Hardware Security Byte

The following procedure is used to program the Hardware Security Byte space and is summarized in Figure 27:

- Set FPS and map Hardware byte (FCON = 0x0C)
- Save and disable the interrupts.
- Load DPTR at address 0000h.
- Load Accumulator register with the data to load.
- Execute the MOVX @DPTR, A instruction.
- Launch the programming by writing the data sequence 54h followed by A4h in FCON register (only from FM1).
  The end of the programming indicated by the FBusy flag cleared.
- Restore the interrupts.

Figure 27. Hardware Programming Procedure



Reset the Column Latches

An automatic reset of the column latches is performed after a successful Flash write sequence. User can also reset the column latches manually, for instance to reload the column latches before writing the Flash. The following procedure is summarized below.

- Save and disable the interrupts.
- Launch the reset by writing the data sequence 56h followed by A6h in FCON register (only from FM1).
- Restore the interrupts.

Error Reports

Flash Programming Sequence Errors

When a wrong sequence is detected, the SEQERR bit in FSTA register is set. Possible wrong sequence are :

- MOV FCON, 5xh instruction not immediately followed by a MOV FCON, Ax instruction.
- A write Flash sequence is launched while no data were loaded in the column latches

The SEQERR bit can be cleared

- By software
- By hardware when a correct programming sequence is completed

When multiple pages are written into the Flash, the user should check FSTA for errors after each write page sequences, not only at the end of the multiple write pages.

Operation Cross Memory Access

Space addressable in read and write are:

- RAM
- ERAM (Expanded RAM access by movx)
- XRAM (eXternal RAM)
- EEPROM DATA
- FM0 (user flash)
- Hardware byte
- XROW
- Boot Flash
- Flash Column latch

The table below provide the different kind of memory which can be accessed from different code location.

Table 18. Cross Memory Access

	Action	RAM	XRAM ERAM	Boot FLASH	FM0	E² Data	Hardware Byte	XROW
boot FLASH	Read			OK	OK	OK	OK	-
	Write			-	OK(1)	OK(1)	OK(1)	OK(1)
FM0	Read			OK	OK	OK	OK	-
	Write			-	OK (idle)	OK(1)	-	OK
External memory EA = 0 or Code Roll Over	Read			-	-	OK	-	-
	Write			-	-	OK(1)	-	-

Note: 1. RWW: Read While Write

Timers/Counters

The AT89C51CC03 implements two general-purpose, 16-bit Timers/Counters. Such are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a Timer or an event Counter. When operating as a Timer, the Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, the Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request. The various operating modes of each Timer/Counter are described in the following sections.

Timer/Counter Operations

A basic operation is Timer registers THx and TLx (x = 0, 1) connected in cascade to form a 16-bit Timer. Setting the run control bit (TRx) in TCON register (see Figure 30) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the Timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the Timer/Counter is unpredictable.

The C/Tx# control bit selects Timer operation or Counter operation by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the Timer/Counter is unpredictable.

For Timer operation (C/Tx# = 0), the Timer register counts the divided-down peripheral clock. The Timer register is incremented once every peripheral cycle (6 peripheral clock periods). The Timer clock rate is $F_{PER}/6$, i.e. $F_{OSC}/12$ in standard mode or $F_{OSC}/6$ in X2 mode.

For Counter operation (C/Tx# = 1), the Timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycles. When the sample is high in one cycle and low in the next one, the Counter is incremented. Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is $F_{PER}/12$, i.e. $F_{OSC}/24$ in standard mode or $F_{OSC}/12$ in X2 mode. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle.

Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 35 to Figure 38 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (see Figure 31) and bits 0, 1, 4 and 5 of TCON register (see Figure 30). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) and mode of operation (M10 and M00). TCON register provides Timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).

For normal Timer operation (GATE0 = 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control Timer operation.

Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag generating an interrupt request.

It is important to stop Timer/Counter before changing mode.

Table 31. TMOD Register

TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	**C/T1#**	**M11**	**M01**	**GATE0**	**C/T0#**	**M10**	**M00**

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control Bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select Bit** Clear for Timer operation: Timer 1 counts the divided-down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select Bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL1) (1) 1 1 Mode 3: Timer 1 halted. Retains count
4	M01	
3	GATE0	**Timer 0 Gating Control Bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select Bit** Clear for Timer operation: Timer 0 counts the divided-down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select Bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL0) (2) 1 1 Mode 3: TL0 is an 8-bit Timer/Counter TH0 is an 8-bit Timer using Timer 1's TR0 and TF0 bits.
0	M00	

1. Reloaded from TH1 at overflow.
2. Reloaded from TH0 at overflow.

Reset Value = 0000 0000b

Registers

Table 36. T2CON Register

T2CON (S:C8h)
Timer 2 Control Register

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 Overflow Flag** TF2 is not set if RCLK=1 or TCLK = 1. Must be cleared by software. Set by hardware on timer 2 overflow.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. Set to cause the CPU to vector to timer 2 interrupt routine when timer 2 interrupt is enabled. Must be cleared by software.
5	RCLK	**Receive Clock bit** Clear to use timer 1 overflow as receive clock for serial port in mode 1 or 3. Set to use timer 2 overflow as receive clock for serial port in mode 1 or 3.
4	TCLK	**Transmit Clock bit** Clear to use timer 1 overflow as transmit clock for serial port in mode 1 or 3. Set to use timer 2 overflow as transmit clock for serial port in mode 1 or 3.
3	EXEN2	**Timer 2 External Enable bit** Clear to ignore events on T2EX pin for timer 2 operation. Set to cause a capture or reload when a negative transition on T2EX pin is detected, if timer 2 is not used to clock the serial port.
2	TR2	**Timer 2 Run Control bit** Clear to turn off timer 2. Set to turn on timer 2.
1	C/T2#	**Timer/Counter 2 Select bit** Clear for timer operation (input from internal clock system: F_{OSC}). Set for counter operation (input from T2 input pin).
0	CP/RL2#	**Timer 2 Capture/Reload bit** If RCLK=1 or TCLK=1, CP/RL2# is ignored and timer is forced to auto-reload on timer 2 overflow. Clear to auto-reload on timer 2 overflows or negative transitions on T2EX pin if EXEN2=1. Set to capture on negative transitions on T2EX pin if EXEN2=1.

Reset Value = 0000 0000b
Bit addressable

Watchdog Timer During Power-down Mode and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are 2 methods of exiting Power-down mode: by a hardware reset or via a level activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, the Watchdog is disabled. Exiting Power-down with an interrupt is significantly different. The interrupt shall be held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down.

To ensure that the WDT does not overflow within a few states of exiting powerdown, it is best to reset the WDT just before entering powerdown.

In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting AT89C51CC03 while in Idle mode, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle mode.

Register

Table 44. WDTPRG Register

WDTPRG (S:A7h)
Watchdog Timer Duration Programming Register

7	6	5	4	3	2	1	0
–	–	–	–	–	S2	S1	S0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	S2	**Watchdog Timer Duration selection bit 2** Work in conjunction with bit 1 and bit 0.
1	S1	**Watchdog Timer Duration selection bit 1** Work in conjunction with bit 2 and bit 0.
0	S0	**Watchdog Timer Duration selection bit 0** Work in conjunction with bit 1 and bit 2.

Reset Value = XXXX X000b

Bit Shortening

If, on the other hand, the transmitter oscillator is faster than the receiver one, the next falling edge used for resynchronization may be too early. So Phase Segment 2 in bit N is shortened in order to adjust the sample point for bit N+1 and the end of the bit time

Synchronization Jump Width

The limit to the amount of lengthening or shortening of the Phase Segments is set by the Resynchronization Jump Width.

This segment may not be longer than Phase Segment 2.

Programming the Sample Point

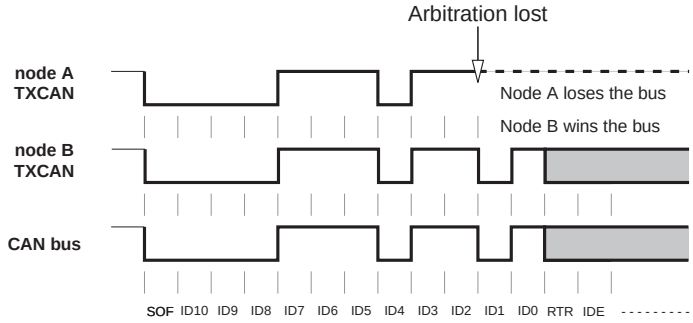
Programming of the sample point allows "tuning" of the characteristics to suit the bus.

Early sampling allows more Time Quanta in the Phase Segment 2 so the Synchronization Jump Width can be programmed to its maximum. This maximum capacity to shorten or lengthen the bit time decreases the sensitivity to node oscillator tolerances, so that lower cost oscillators such as ceramic resonators may be used.

Late sampling allows more Time Quanta in the Propagation Time Segment which allows a poorer bus topology and maximum bus length.

Arbitration

Figure 46. Bus Arbitration



The CAN protocol handles bus accesses according to the concept called “Carrier Sense Multiple Access with Arbitration on Message Priority”.

During transmission, arbitration on the CAN bus can be lost to a competing device with a higher priority CAN Identifier. This arbitration concept avoids collisions of messages whose transmission was started by more than one node simultaneously and makes sure the most important message is sent first without time loss.

The bus access conflict is resolved during the arbitration field mostly over the identifier value. If a data frame and a remote frame with the same identifier are initiated at the same time, the data frame prevails over the remote frame (c.f. RTR bit).

Errors

The CAN protocol signals any errors immediately as they occur. Three error detection mechanisms are implemented at the message level and two at the bit level:

Error at Message Level

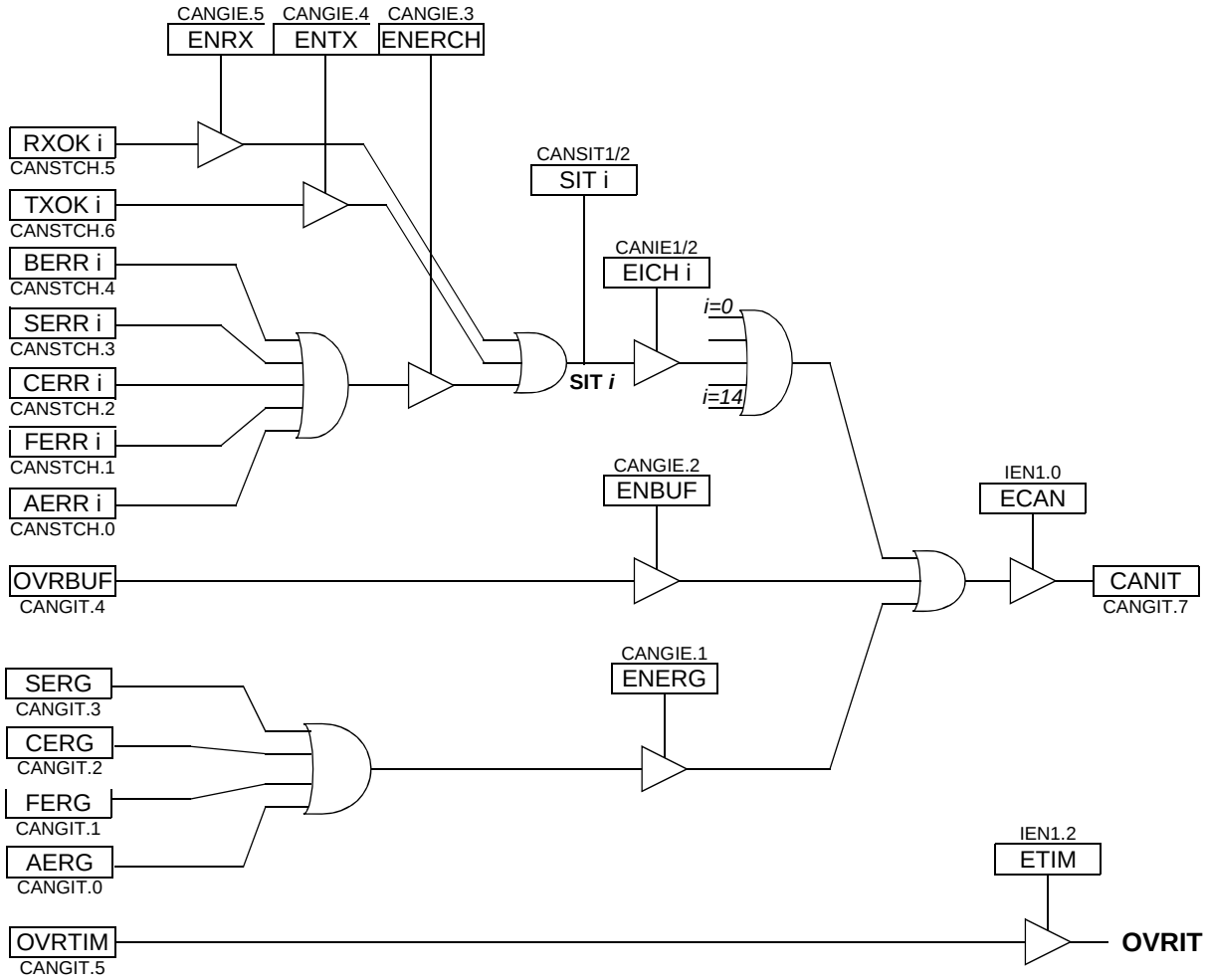
- Cyclic Redundancy Check (CRC)
  The CRC safeguards the information in the frame by adding redundant check bits at the transmission end. At the receiver these bits are re-computed and tested against the received bits. If they do not agree there has been a CRC error.
- Frame Check
  This mechanism verifies the structure of the transmitted frame by checking the bit

IT CAN Management

The different interrupts are:

- Transmission interrupt,
- Reception interrupt,
- Interrupt on error (bit error, stuff error, crc error, form error, acknowledge error),
- Interrupt when Buffer receive is full,
- Interrupt on overrun of CAN Timer.

Figure 50. CAN Controller Interrupt Structure



To enable a transmission interrupt:

- Enable General CAN IT in the interrupt system register,
- Enable interrupt by message object, EICHi,
- Enable transmission interrupt, ENTX.

To enable a reception interrupt:

- Enable General CAN IT in the interrupt system register,
- Enable interrupt by message object, EICHi,
- Enable reception interrupt, ENRX.

To enable an interrupt on message object error:

Table 83. CANTCON Register

CANTCON (S:A1h)
CAN Timer ClockControl

7	6	5	4	3	2	1	0
TPRESC 7	**TPRESC 6**	**TPRESC 5**	**TPRESC 4**	**TPRESC 3**	**TPRESC 2**	**TPRESC 1**	**TPRESC 0**

Bit Number	Bit Mnemonic	Description
7-0	TPRESC7:0	**Timer Prescaler of CAN Timer** This register is a prescaler for the main timer upper counter range = 0 to 255. See Figure 55.

Reset Value = 00h

Table 84. CANTIMH Register

CANTIMH (S:ADh)
CAN Timer High

7	6	5	4	3	2	1	0
CANGTIM 15	**CANGTIM 14**	**CANGTIM 13**	**CANGTIM 12**	**CANGTIM 11**	**CANGTIM 10**	**CANGTIM 9**	**CANGTIM 8**

Bit Number	Bit Mnemonic	Description
7-0	CANGTIM15:8	**High byte of Message Timer** See Figure 55.

Reset Value = 0000 0000b

Table 85. CANTIML Register

CANTIML (S:ACh)
CAN Timer Low

7	6	5	4	3	2	1	0
CANGTIM 7	**CANGTIM 6**	**CANGTIM 5**	**CANGTIM 4**	**CANGTIM 3**	**CANGTIM 2**	**CANGTIM 1**	**CANGTIM 0**

Bit Number	Bit Mnemonic	Description
7-0	CANGTIM7:0	**Low byte of Message Timer** See Figure 55.

Reset Value = 0000 0000b

Table 89. CANTTCL Register

CANTTCL (S:A4h Read Only)
CAN TTC Timer Low

7	6	5	4	3	2	1	0
TIMTTC 7	TIMTTC 6	TIMTTC 5	TIMTTC 4	TIMTTC 3	TIMTTC 2	TIMTTC 1	TIMTTC 0

Bit Number	Bit Mnemonic	Description
7-0	TIMTTC7:0	**Low byte of TTC Timer** See Figure 55.

Reset Value = 0000 0000b

ADC Converter Operation

A start of single A/D conversion is triggered by setting bit ADSST (ADCON.3).

After completion of the A/D conversion, the ADSST bit is cleared by hardware.

The end-of-conversion flag ADEOC (ADCON.4) is set when the value of conversion is available in ADDH and ADDL, it must be cleared by software. If the bit EADC (IEN1.1) is set, an interrupt occur when flag ADEOC is set (see Figure 76). Clear this flag for re-arming the interrupt.

The bits SCH0 to SCH2 in ADCON register are used for the analog input channel selection.

Table 102. Selected Analog input

SCH2	SCH1	SCH0	Selected Analog input
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Voltage Conversion

When the ADCIN is equals to VAREF the ADC converts the signal to 3FFh (full scale). If the input voltage equals VAGND, the ADC converts it to 000h. Input voltage between VAREF and VAGND are a straight-line linear conversion. All other voltages will result in 3FFh if greater than VAREF and 000h if less than VAGND.

Note that ADCIN should not exceed VAREF absolute maximum range! (See section “AC-DC”)

Clock Selection

The ADC clock is the same as CPU.

The maximum clock frequency is defined in the DC parameters for A/D converter. A prescaler is featured (ADCCLH) to generate the ADC clock from the oscillator frequency.

if PRS = 0 then $F_{ADC} = F_{periph} / 64$

if PRS > 0 then $F_{ADC} = F_{periph} / 2 \times PRS$

Bit Number	Bit Mnemonic	Description
7-2	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
1-0	ADAT1:0	**ADC result** bits 1-0

Reset Value = 00h

Table 123. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter	Units
T_{RLRH}	Min	6 T - x	3 T - x	20	ns
T_{WLWH}	Min	6 T - x	3 T - x	20	ns
T_{RLDV}	Max	5 T - x	2.5 T - x	25	ns
T_{RHDX}	Min	x	x	0	ns
T_{RHDZ}	Max	2 T - x	T - x	20	ns
T_{LLDV}	Max	8 T - x	4T -x	40	ns
T_{AVDV}	Max	9 T - x	4.5 T - x	60	ns
T_{LLWL}	Min	3 T - x	1.5 T - x	25	ns
T_{LLWL}	Max	3 T + x	1.5 T + x	25	ns
T_{AVWL}	Min	4 T - x	2 T - x	25	ns
T_{QVWX}	Min	T - x	0.5 T - x	15	ns
T_{QVWH}	Min	7 T - x	3.5 T - x	25	ns
T_{WHQX}	Min	T - x	0.5 T - x	10	ns
T_{RLAZ}	Max	x	x	0	ns
T_{WHLH}	Min	T - x	0.5 T - x	15	ns
T_{WHLH}	Max	T + x	0.5 T + x	15	ns

Timings

Test conditions: capacitive load on all pins= 60 pF.

Table 1. SPI Interface Master AC Timing
V_{DD} = 2.7 to 3.3 V, T_A = -40 to +85°C

Symbol	Parameter	Min	Max	Unit
Slave Mode				
T_{CHCH}	Clock Period	6(1)		T_{PER}
T_{CHCX}	Clock High Time	3(1)		T_{PER}
T_{CLCX}	Clock Low Time	3(1)		T_{PER}
T_{SLCH}, T_{SLCL}	$\overline{SS}$ Low to Clock edge	$4T_{PER}$-20ns(1)		ns
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	50		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	50		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		50	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{CLSH}, T_{CHSH}	$\overline{SS}$ High after Clock Edge	$4T_{PER}$+20ns(1)		ns
T_{SLOV}	$\overline{SS}$ Low to Output Data Valid		$4T_{PER}$+20ns(1)	ns
T_{SHOX}	Output Data Hold after $\overline{SS}$ High		$2T_{PER}$+100ns(1)	ns
T_{SHSL}	$\overline{SS}$ High to $\overline{SS}$ Low	$2T_{PER}$+120ns(1)		
T_{OLOH}	Output Rise time		100	ns
T_{OHOL}	Output Fall Time		100	ns
Master Mode				
T_{CHCH}	Clock Period	4(1)		T_{PER}
T_{CHCX}	Clock High Time	$2T_{PER}$-20ns(1)		T_{PER}
T_{CLCX}	Clock Low Time	$2T_{PER}$-20ns(1)		T_{PER}
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	50		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	0		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		20	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{CLCH}	Output Data Rise time		100	ns
T_{CHCL}	Output Data Fall Time		100	ns

Note: 1. Value of this parameter depends on prescacler ratio defined in bits 0,1 and 7 of SCON Register.In the above table, the ratio used is 4. As it can be set also to 8, 16, 32, 64 or 128, the factor of T_{PER} must be changed according to the new ratio.E.g. 2TPER-20ns(1) will be changed to 4TPER-20ns(1) if the prescaler ratio equals 8.