NXP USA Inc. - PXAS37KBBE,557 Datasheet

What is "Embedded - Microcontrollers"?

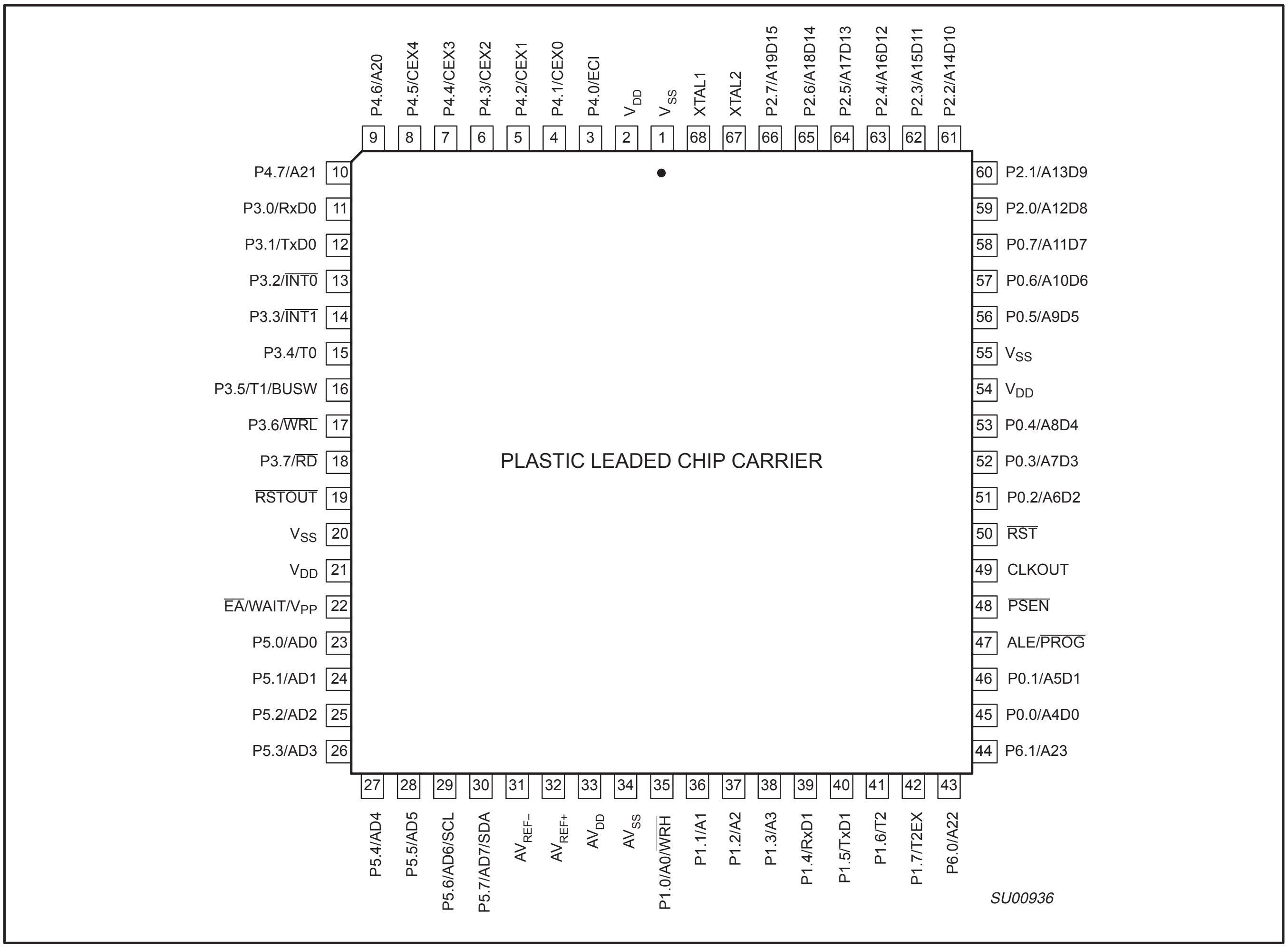
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

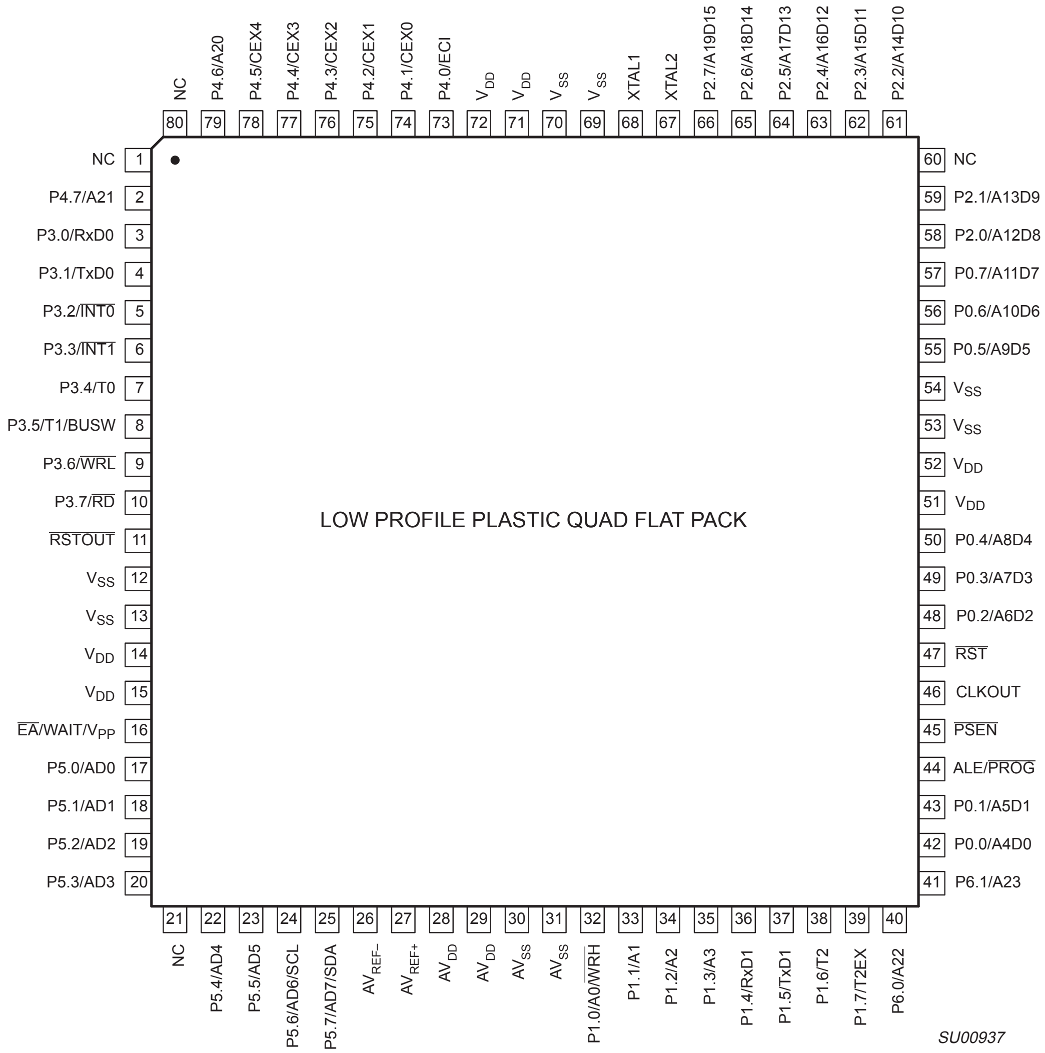
Details	
Product Status	Obsolete
Core Processor	XA
Core Size	16-Bit
Speed	30MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	PWM, WDT
Number of I/O	50
Program Memory Size	32KB (32K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/pxas37kbbe-557

PIN CONFIGURATIONS

68-pin PLCC package



80-pin LQFP package



PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER		TYPE	NAME AND FUNCTION
	PLCC	LQFP		
V_{SS}	1, 20, 55	12, 13, 53, 54, 69, 70	I	**Ground:** 0 V reference.
V_{DD}	2, 21, 54	14, 15, 51, 52, 71, 72	I	**Power Supply:** This is the power supply voltage for normal, idle, and power down operation.
$\overline{\text{RST}}$	50	47	I	**Reset:** A low on this pin resets the microcontroller, causing I/O ports and peripherals to take on their default states, and the processor to begin execution at the address contained in the reset vector.
$\overline{\text{RSTOUT}}$	19	11	O	**Reset Output:** This pin outputs a low whenever the XA-S3 processor is reset for any reason. This includes an external reset via the $\overline{\text{RST}}$ pin, watchdog reset, and the $\overline{\text{RESET}}$ instruction.
ALE/$\overline{\text{PROG}}$	47	44	I/O	**Address Latch Enable/Program Pulse:** A high output on the ALE pin signals external circuitry to latch the address portion of the multiplexed address/data bus. A pulse on ALE occurs only when it is needed in order to process a bus cycle.
$\overline{\text{PSEN}}$	48	45	O	**Program Store Enable:** The read strobe for external program memory. When the microcontroller accesses external program memory, $\overline{\text{PSEN}}$ is driven low in order to enable memory devices. $\overline{\text{PSEN}}$ is only active when external code accesses are performed.
$\overline{\text{EA}}$/WAIT/V_{PP}	22	16	I	**External Access/Bus Wait:** The $\overline{\text{EA}}$ input determines whether the internal program memory of the microcontroller is used for code execution. The value on the $\overline{\text{EA}}$ pin is latched as the external reset input is released and applies during later execution. When latched as a 0, external program memory is used exclusively. When latched as a 1, internal program memory will be used up to its limit, and external program memory used above that point. After reset is released, this pin takes on the function of bus WAIT input. If WAIT is asserted high during an external bus access, that cycle will be extended until WAIT is released.
XTAL1	68	68	I	**Crystal 1:** Input to the inverting amplifier used in the oscillator circuit and input to the internal clock generator circuits.
XTAL2	67	67	I	**Crystal 2:** Output from the oscillator amplifier.
CLKOUT	49	46	O	**Clock Output:** This pin outputs a buffered version of the internal CPU clock. The clock output may be used in conjunction with the external bus to synchronize WAIT state generators, etc. The clock output may be disabled by software.
AV_{DD}	33	28, 29	I	**Analog Power Supply:** Positive power supply input for the A/D converter.
AV_{SS}	34	30, 31	I	**Analog Ground.**
AV_{REF+}	32	27	I	**A/D Positive Reference Voltage:** High end reference for the A/D converter.
AV_{REF-}	31	26	I	**A/D Negative Reference Voltage:** Low end reference for the A/D converter.
P0.0 – P0.7	45, 46, 51–53, 56–58	42, 43, 48–50, 55–57	I/O	**Port 0:** Port 0 is an 8-bit I/O port with a user-configurable output type. Port 0 latches have 1s written to them and are configured in the quasi-bidirectional mode during reset. The operation of port 0 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. When the external program/data bus is used, Port 0 becomes the multiplexed low data/instruction byte and address lines 4 through 11.

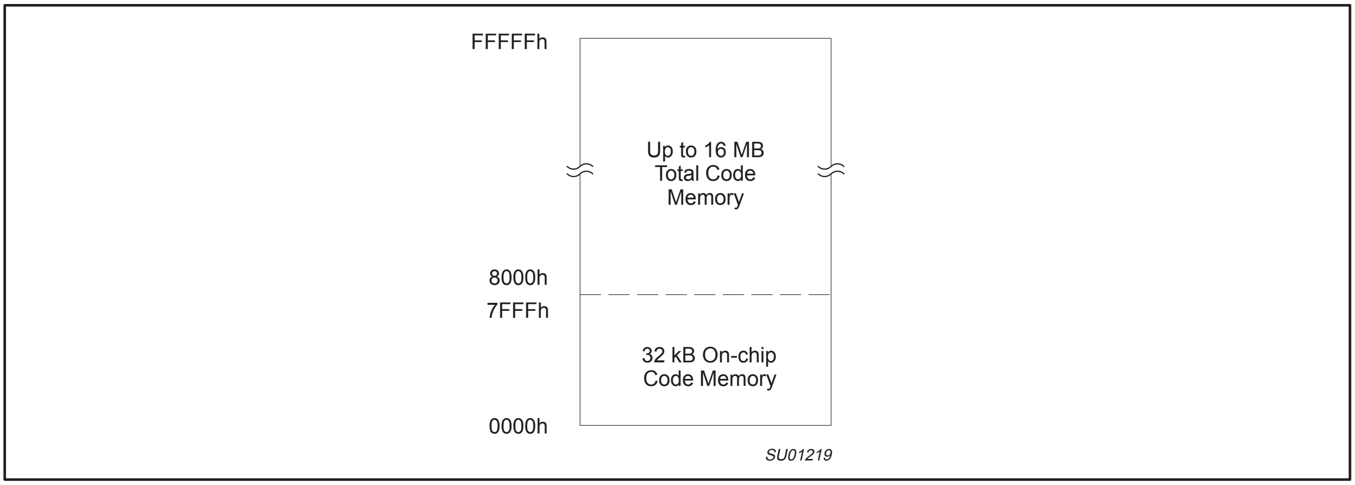


Figure 1. XA-S3 program memory map

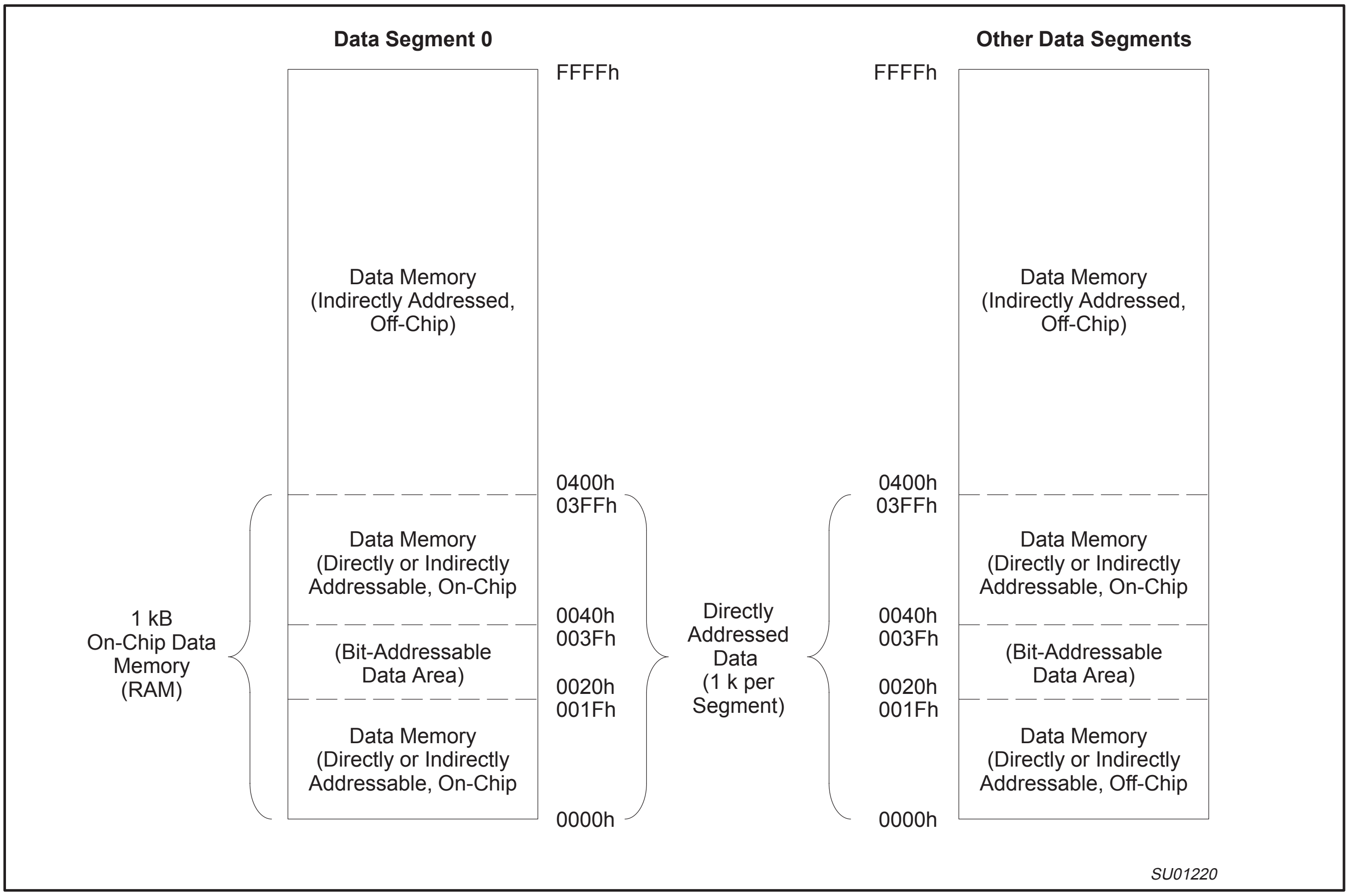


Figure 2. XA-S3 data memory map

FUNCTIONAL DESCRIPTION

Details of XA-S3 functions will be described in the following sections.

Analog to Digital converter

The XA-S3 has an 8-channel, 8-bit A/D converter with 8 sets of result registers, single scan and multiple scan operating modes. The A/D also has a 10-bit conversion mode that provides greater result resolution. The A/D input range is limited to 0 to AV_{DD} (3.3 V max.). The A/D inputs are on Port 5. Analog Power and Ground as well as AV_{REF+} and AV_{REF-} must be supplied in order for the A/D converter to be used. Prior to enabling the A/D converter or driving analog signals into the A/D inputs, the port configurations for the pins being used as A/D inputs must be set to the “off” (high impedance, input only) mode.

A/D timing can be adapted to the application clock frequency in order to provide the fastest possible conversion.

A/D converter operation is controlled through the ADCON (A/D Control) register, see Figure 1. Bits in ADCON start and stop the A/D, flag conversion completion, and select the converter operating modes. When 10-bit resolution is needed, the A/D mode may be set to give 10 result bits by setting the ADRES bit to 1. In this mode, the A/D takes longer to complete a conversion, and the timing must be set differently in ADCFG.

A/D Conversion Modes

The A/D converter supports a single scan mode and a continuous scan mode. In either mode, one or more A/D channels may be converted. The ADCS register determines which channels are converted. If the corresponding bit in the ADCS register is set, that channel is selected for conversions, otherwise that channel is skipped. The ADCS register is detailed in Figure 2.

For any A/D conversion, the results are stored in ADRSHn, corresponding to the A/D channel just converted. For a 10-bit conversion, the two least significant bits are read from the upper end of register ADRSL. These bits must be read before another conversion is begun.

A/D conversions are begun by setting the A/D Start and STatus bit in ADCON. In the single scan mode, all of the channels selected by bits in the ADCS register will be converted once. The ADINT flag is set when the last channel is converted. In the continuous scan mode, the A/D converter continuously converts all A/D channels selected by bits in the ADCS register. The ADINT flag is set when all channels have been converted once.

The A/D converter can generate an interrupt when the ADINT flag is set. This will occur if the A/D interrupt is enabled (via the EAD bit in IEL), the interrupt system is enabled (via the EA bit in IEL), and the A/D interrupt priority (specified in IPA3 bits 3 to 0) is higher than the currently running code (PSW bits IM3 through IM0) and any other pending interrupt. ADINT must be cleared by software.

A/D Timing Configuration

The A/D sampling and conversion timing may be optimized for the particular oscillator frequency and input drive characteristics of the application. Because A/D operation is mostly dependent on real-time effects (charging time of sampling capacitors, settling time of the comparator, etc.), A/D conversion times are not necessarily much longer at slower clock frequencies. The A/D timing is controlled by the ADCFG register, as shown in Figure 3, Table 2 and Table 3.

The primary effect of ADCFG settings is to adjust the A/D sample and hold time to be relatively constant over various clock frequencies. Two settings (value 6 and B) are provided to allow fast conversions with a lower external source driving the A/D inputs. These settings provide double the sample time at the same frequency. Of course, settings intended for lower frequencies may also be used at higher frequencies in order to increase the A/D sampling time, but this method has the side effect of significantly increasing A/D conversion times.

ADCON Address:43Eh
Bit Addressable
Reset Value: 00h

MSB … LSB

—	—	—	—	ADRES	ADMOD	ADSST	ADINT

BIT	SYMBOL	FUNCTION
ADCON.7	—	Reserved for future use. Should not be set to 1 by user programs.
ADCON.6	—	Reserved for future use. Should not be set to 1 by user programs.
ADCON.5	—	Reserved for future use. Should not be set to 1 by user programs.
ADCON.4	—	Reserved for future use. Should not be set to 1 by user programs.
ADCON.3	ADRES	Selects 8-bit (0) or 10-bit (1) conversion mode.
ADCON.2	ADMOD	A/D mode select. 1 = continuous scan of selected inputs after a start of the A/D. 0 = single scan of selected inputs after a start of the A/D.
ADCON.1	ADSST	A/D start and status. Setting this bit by software starts the A/D conversion of the selected A/D inputs. ADSST remains set as long as the A/D is in operation. In continuous conversion mode, ADSST will remain set unless the A/D is stopped by software. While ADSST is set, new start commands are ignored. An A/D conversion is progress may be aborted by software clearing ADSST.
ADCON.0	ADINT	A/D conversion complete/interrupt flag. This flag is set when all selected A/D channels are converted in either the single scan or continuous scan modes. Must be cleared by software.

SU01229

Figure 1. A/D Control Register (ADCON)

ADCS Address:43Fh
Bit Addressable
Reset Value: 00h

MSB							LSB
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0

BIT	SYMBOL	FUNCTION
ADCS.7	ADCS7	A/D channel 7 select bit.
ADCS.6	ADCS6	A/D channel 6 select bit.
ADCS.5	ADCS5	A/D channel 5 select bit.
ADCS.4	ADCS4	A/D channel 4 select bit.
ADCS.3	ADCS3	A/D channel 3 select bit.
ADCS.2	ADCS2	A/D channel 2 select bit.
ADCS.1	ADCS1	A/D channel 1 select bit.
ADCS.0	ADCS0	A/D channel 0 select bit.

SU00939

Figure 2. A/D Channel Select Register (ADCS)

ADCFG Address:4B9h
Not bit Addressable
Reset Value: 00h

MSB							LSB
—	—	—	—	A/D Timing Configuration			

BIT	SYMBOL	FUNCTION
ADCFG.7	—	Reserved for future use. Should not be set to 1 by user programs.
ADCFG.6	—	Reserved for future use. Should not be set to 1 by user programs.
ADCFG.5	—	Reserved for future use. Should not be set to 1 by user programs.
ADCFG.4	—	Reserved for future use. Should not be set to 1 by user programs.
ADCFG.3–0	ADCFG	A/D timing configuration (see text and table).

SU00940

Figure 3. A/D Timing Configuration Register (ADCFG)

Table 2. A/D Timing Configuration

ADCFG.3–0	Max. Oscillator Frequency (MHz)	Conversion Time		Sampling Time (Osc. Clocks)
		Osc. Clocks	μsec at max. Osc.	
0h (0000)	6.66	72	10.81	4
1h (0001)	10	76	7.6	6
2h (0010)	11.11	80	7.2	8
3h (0011)	13.33	96	7.2	8
4h (0100)	16.66	100	6.0	10
5h (0101)	20	104	5.2	12
6h (0110)[1]	20	116	5.8	24
7h (0111)	22.2	108	4.86	14
8h (1000)	23.3	124	5.32	14
9h (1001)	26.6	128	4.81	16
Ah (1010)	30	132	4.4	18
Bh (1011)[1]	30	146	4.87	32
Ch (1100)	–	136	4.25	20
Dh (1101)	–	152	4.56	20
Eh (1110)	–	172	4.7	22
Fh (1111)	–	176	4.4	24

NOTE:
1. These settings provide additional A/D input sampling time, in order to allow accurate readings with a higher external sourceimpedance.

I^{2}C Interface

The I^{2}C interface on the XA-S3 is identical to the standard byte-style I^{2}C interface found on devices such as the 8xC552 except for the rate selection. ***The I^{2}C interface conforms to the 100 kHz I^{2}C specification, but may be used at rates up to 400 kHz (non-conforming).***

Important: Before the I^{2}C interface may be used, the port pins P5.6 and 5.7, which correspond to the I^{2}C functions SCL and SDA respectively, must be set to the open drain mode.

The processor interfaces to the I^{2}C logic via the following four special function registers: I2CON (I^{2}C control register), I2STA (I^{2}C status register), I2DAT (I^{2}C data register), and I2ADR (I^{2}C slave address register). The I^{2}C control logic interfaces to the external I^{2}C bus via two port 5 pins: P5.6/SCL (serial clock line) and P5.7/SDA (serial data line).

The Control Register, I2CON

This register is shown in Figure 6. Two bits are affected by the I^{2}C hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^{2}C bus. The STO bit is also cleared when ENA = “0”.

ENA, the I^{2}C Enable Bit

ENA = 0: When ENA is “0”, the SDA and SCL outputs are not driven. SDA and SCL input signals are ignored, SIO1 is in the “not addressed” slave state, and the STO bit in I2CON is forced to “0”. No other bits are affected. P5.6 and P5.7 may be used as open drain I/O ports.

ENA = 1: When ENA is “1”, SIO1 is enabled. The P5.6 and P5.7 port latches must be set to logic 1.

ENA should not be used to temporarily release the I^{2}C-bus since, when ENA is reset, the I^{2}C-bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed the ENA = “1”.

STA, the START flag

STA = 1: When the STA bit is set to enter a master mode, the I^{2}C hardware checks the status of the I^{2}C bus and generates a START condition if the bus is free. If the bus is not free, the I^{2}C interface waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while the I^{2}C interface is already in a master mode and one or more bytes are transmitted or received, the hardware transmits a repeated START condition. STA may be set at any time. STA may also be set when the I^{2}C interface is an addressed slave.

STA = 0: When the STA bit is reset, no START condition or repeated START condition will be generated.

STO, the STOP flag

STO = 1: When the STO bit is set while the I^{2}C interface is in a master mode, a STOP condition is transmitted to the I^{2}C bus. When the STOP condition is detected on the bus, the hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I^{2}C bus. However, the hardware behaves as if a STOP condition has been received and switches to the defined “not addressed” slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, then a STOP condition is transmitted to the I^{2}C bus if the interface is in a master mode (in a slave mode, the hardware generates an internal STOP condition which is not transmitted). The I^{2}C interface then transmits a START condition.

STO = 0: When the STO bit is reset, no STOP condition will be generated.

SI, the Serial Interrupt flag

SI = 1: When the SI flag is set, and the EA (interrupt system enable) and EI2 (I^{2}C interrupt enable) bits are also set, an I^{2}C interrupt is requested. SI is set by hardware when one of 25 of the 26 possible I^{2}C interface 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.

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 I2ADR is set.
- A data byte has been received while the I^{2}C interface is in the master receiver mode.
- A data byte has been received while the I^{2}C interface 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 the SCL line when:

- A data byte has been received while the I^{2}C interface is in the master receiver mode.
- A data byte has been received while the I^{2}C interface is in the addressed slave receiver mode.

When the I^{2}C interface is in the addressed slave transmitter mode, state C8H will be entered after the last serial data byte is transmitted. When SI is cleared, the I^{2}C interface 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 the I^{2}C interface 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, the hardware can be temporarily released from the I^{2}C bus while the bus status is monitored. While the hardware 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 part’s own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

XA-S3 Timer/Counters

The XA-S3 has three general purpose counter/timers, two of which may also be used as baud rate generators for either or both of the UARTs.

Timer 0 and 1
These are identical to the standard XA-G3 timer 0 and 1.

Timer 2
This is identical to the standard XA-G3 timer 2.

Programmable Counter Array (PCA)

The Programmable Counter Array available on the XA-S3 is a special 16-bit Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P4.1(CEX0), module 1 to P4.2(CEX1), etc. The basic PCA configuration is shown in Figure 7.

The PCA timer is a common time base for all five modules and can be programmed to run at: the TCLK rate (Osc/4, Osc/16, or Osc/64), the Timer 0 overflow, or the input on the ECI pin (P4.0). When the ECI input is used, the falling edge clocks the PCA counter. The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 10):

CPS1	CPS0	PCA Timer Count Source
0	X	TCLK (Osc/4, Osc/16, or Osc/64)
1	0	Timer 0 overflow
1	1	ECI (PCA External Clock Input (max rate = Osc/4)

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows. These functions are shown in Figure 8. In addition, each PCA module may generate a separate interrupt.

The watchdog timer function is implemented in module 4 (see Figure 17).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 11). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set, The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 9.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 12). The registers contain the bits that control the mode that each module will operate in. The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode. The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function. Figure 13 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

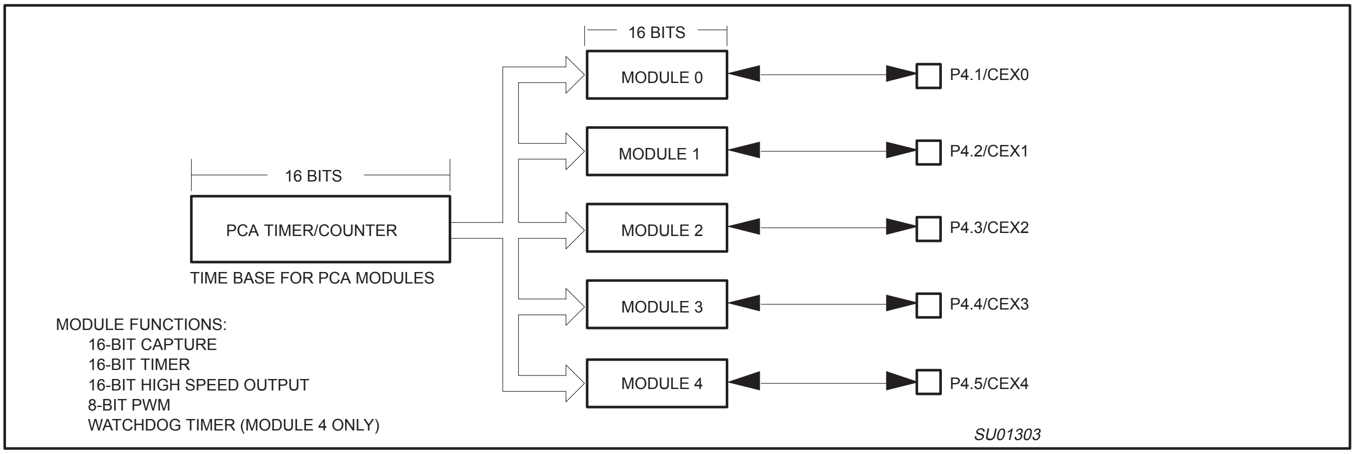


Figure 7. Programmable Counter Array (PCA)

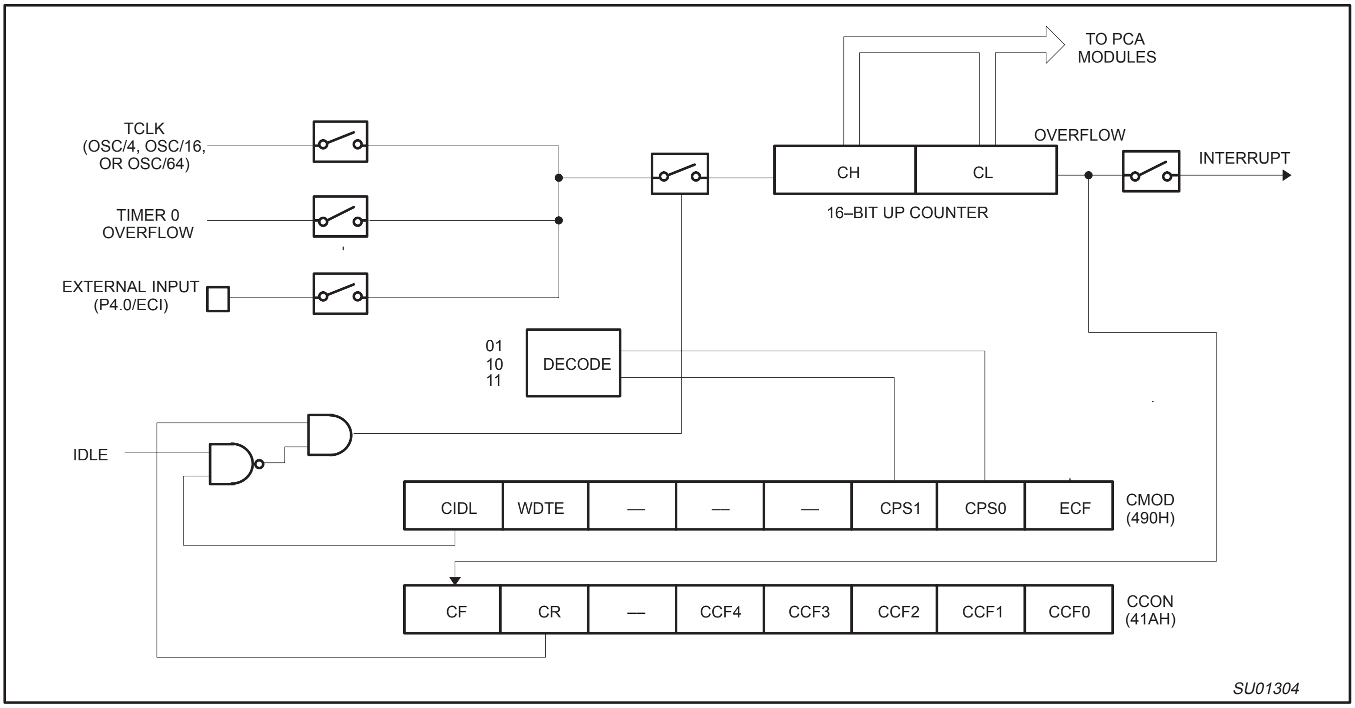


Figure 8. PCA Timer/Counter

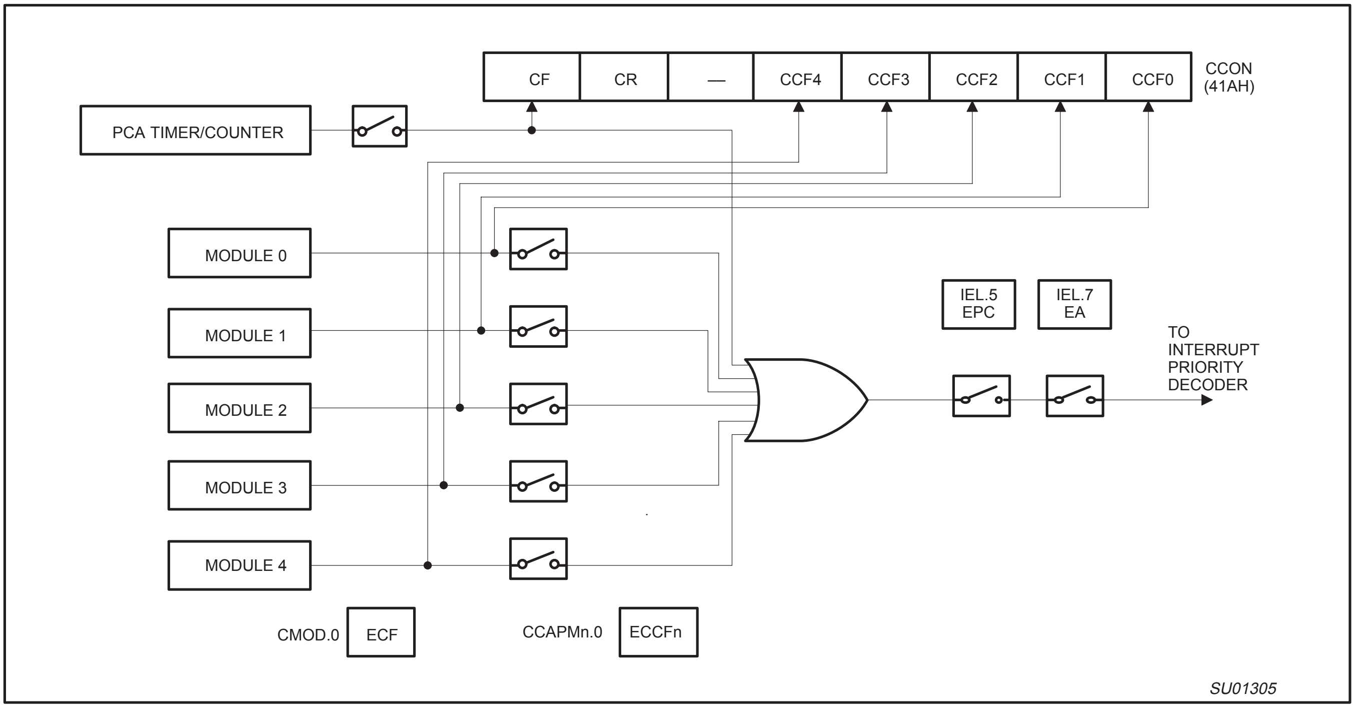


Figure 9. PCA Interrupt System

Serial Port Control Register

The serial port control and status register is the Special Function Register SnCON, shown in Figure 21. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8_n and RB8_n), and the serial port interrupt bits (TI_n and RI_n).

TI Flag

In order to allow easy use of the double buffered UART transmitter feature, the TI_n flag is set by the UART hardware under two conditions. The first condition is the completion of any byte transmission. This occurs at the end of the stop bit in modes 1, 2, or 3, or at the end of the eighth data bit in mode 0. The second condition is when SnBUF is written while the UART transmitter is idle. In this case, the TI_n flag is set in order to indicate that the second UART transmitter buffer is still available.

Typically, UART transmitters generate one interrupt per byte transmitted. In the case of the XA UART, one additional interrupt is generated as defined by the stated conditions for setting the TI_n flag. This additional interrupt does not occur if double buffering is bypassed as explained below. Note that if a character oriented approach is used to transmit data through the UART, there could be a second interrupt for each character transmitted, depending on the timing of the writes to SBUF. For this reason, it is generally better to bypass double buffering when the UART transmitter is used in character oriented mode. This is also true if the UART is polled rather than interrupt driven, and when transmission is character oriented rather than message or string oriented. The interrupt occurs at the end of the last byte transmitted when the UART becomes idle. Among other things, this allows a program to determine when a message has been transmitted completely. The interrupt service routine should handle this additional interrupt.

The recommended method of using the double buffering in the application program is to have the interrupt service routine handle a single byte for each interrupt occurrence. In this manner the program essentially does not require any special considerations for double buffering. Unless higher priority interrupts cause delays in the servicing of the UART transmitter interrupt, the double buffering will result in transmitted bytes being tightly packed with no intervening gaps.

9-bit Mode

Please note that the ninth data bit (TB8) is not double buffered. Care must be taken to insure that the TB8 bit contains the intended data at the point where it is transmitted. Double buffering of the UART transmitter may be bypassed as a simple means of synchronizing TB8 to the rest of the data stream.

Bypassing Double Buffering

The UART transmitter may be used as if it is single buffered. The recommended UART transmitter interrupt service routine (ISR) technique to bypass double buffering first clears the TI_n flag upon entry into the ISR, as in standard practice. This clears the interrupt that activated the ISR. Secondly, the TI_n flag is cleared immediately following each write to SnBUF. This clears the interrupt flag that would otherwise direct the program to write to the second transmitter buffer. If there is any possibility that a higher priority interrupt might become active between the write to SnBUF and the clearing of the TI_n flag, the interrupt system may have to be temporarily disabled during that sequence by clearing, then setting the EA bit in the IEL register.

CLOCKING SCHEME/BAUD RATE GENERATION

The XA UARTS clock rates are determined by either a fixed division (modes 0 and 2) of the oscillator clock or by the Timer 1 or Timer 2 overflow rate (modes 1 and 3).

The clock for the UARTs in XA runs at 16x the Baud rate. If the timers are used as the source for Baud Clock, since maximum speed of timers/Baud Clock is Osc/4, the maximum baud rate is timer overflow divided by 16 i.e. Osc/64.

In Mode 0, it is fixed at Osc/16. In Mode 2, however, the fixed rate is Osc/32.

Pre-scaler for all Timers T0,1,2 controlled by PT1, PT0 bits in SCR	00	Osc/4
	01	Osc/16
	10	Osc/64
	11	reserved

Baud Rate for UART Mode 0:
Baud_Rate = Osc/16

Baud Rate calculation for UART Mode 1 and 3:
Baud_Rate = Timer_Rate/16

Timer_Rate = Osc/(N*(Timer_Range– Timer_Reload_Value))

where N = the TCLK prescaler value: 4, 16, or 64.
and Timer_Range = 256 for timer 1 in mode 2.
65536 for timer 1 in mode 0 and timer 2 in count up mode.

The timer reload value may be calculated as follows:
Timer_Reload_Value = Timer_Range–(Osc/(Baud_Rate*N*16))

NOTES:

1.. The maximum baud rate for a UART in mode 1 or 3 is Osc/64.
2.. The lowest possible baud rate (for a given oscillator frequency and N value) may be found by using a timer reload value of 0.
3.. The timer reload value may never be larger than the timer range.
4.. If a timer reload value calculation gives a negative or fractional result, the baud rate requested is not possible at the given oscillator frequency and N value.

Baud Rate for UART Mode 2:
Baud_Rate = Osc/32

Using Timer 2 to Generate Baud Rates

Timer T2 is a 16-bit up/down counter in XA. As a baud rate generator, timer 2 is selected as a clock source for either/both UART0 and UART1 transmitters and/or receivers by setting TCLKn and/or RCLKn in T2CON and T2MOD. As the baud rate generator, T2 is incremented as Osc/N where N = 4, 16 or 64 depending on TCLK as programmed in the SCR bits PT1, and PTO. So, if T2 is the source of one UART, the other UART could be clocked by either T1 overflow or fixed clock, and the UARTs could run independently with different baud rates.

T2CON 0x418		bit5	bit4	
		RCLK0	TCLK0	

T2MOD 0x419		bit5	bit4	
		RCLK1	TCLK1	

Prescaler Select for Timer Clock (TCLK)

SCR 0x440		bit3	bit2	
		PT1	PT0	

SnSTAT Address: S0STAT 421
S1STAT 425
Bit Addressable
Reset Value: 00H

MSB							LSB
—	—	—	—	FEn	BRn	OEn	STINTn

BIT	SYMBOL	FUNCTION
SnSTAT.3	FEn	Framing Error flag is set when the receiver fails to see a valid STOP bit at the end of the frame. Cleared by software.
SnSTAT.2	BRn	Break Detect flag is set if a character is received with all bits (including STOP bit) being logic ‘0’. Thus it gives a “Start of Break Detect” on bit 8 for Mode 1 and bit 9 for Modes 2 and 3. The break detect feature operates independently of the UARTs and provides the START of Break Detect status bit that a user program may poll. Cleared by software.
SnSTAT.1	OEn	Overrun Error flag is set if a new character is received in the receiver buffer while it is still full (before the software has read the previous character from the buffer), i.e., when bit 8 of a new byte is received while RI in SnCON is still set. Cleared by software.
SnSTAT.0	STINTn	This flag must be set to enable any of the above status flags to generate a receive interrupt (RIn). The only way it can be cleared is by a software write to this register.

SU00607B

Figure 20. Serial Port Extended Status (SnSTAT) Register (See also Figure 22 regarding Framing Error flag)

UART INTERRUPT SCHEME

There are separate interrupt vectors for each UART’s transmit and receive functions.

Table 5. Interrupt Vector Locations for UARTs

Vector Address	Interrupt Source	Arbitration
A0H – A3H	UART 0 Receiver	9
A4H – A7H	UART 0 Transmitter	10
A8H – ABH	UART 1 Receiver	11
ACH – AFH	UART 1 Transmitter	12

NOTE:
The transmit and receive vectors could contain the same ISR address to work like a 8051 interrupt scheme

Error Handling, Status Flags and Break Detect

XA UARTs have several error flags as described in Figures 20 and 22.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren’t being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit although this is better done with the Framing Error (FE) flag. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the “Given” address or the “Broadcast” address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 23.

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. 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 slaves. Slave 0 requires a 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

RSTSRC Address:463h
Not bit Addressable
Reset Value: see below

MSB							LSB
—	—	—	—	—	R_WD	R_CMD	R_EXT

BIT	SYMBOL	FUNCTION
RSTSRC.7	—	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.6	—	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.5	—	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.4	—	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.3	—	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.2	R_WD	Indicates that the last reset was caused by a watchdog timer overflow.
RSTSRC.1	R_CMD	Indicates that the last reset was caused by execution of the RESET instruction.
RSTSRC.0	R_EXT	Indicates that the last reset was caused by the external $\overline{\text{RST}}$ input.

SU00942

Figure 24. Reset source register (RSTSRC)

INTERRUPTS

XA-S3 interrupt sources include the following:

- External interrupts 0 and 1 (2)
- Timer 0, 1, and 2 interrupts (3)
- PCA: 1 global and 5 channel interrupts (6)
- A/D interrupt (1)
- UART 0 transmitter and receiver interrupts (2)
- UART 1 transmitter and receiver interrupts (2)
- I²C interrupt (1)
- Software interrupts (7)

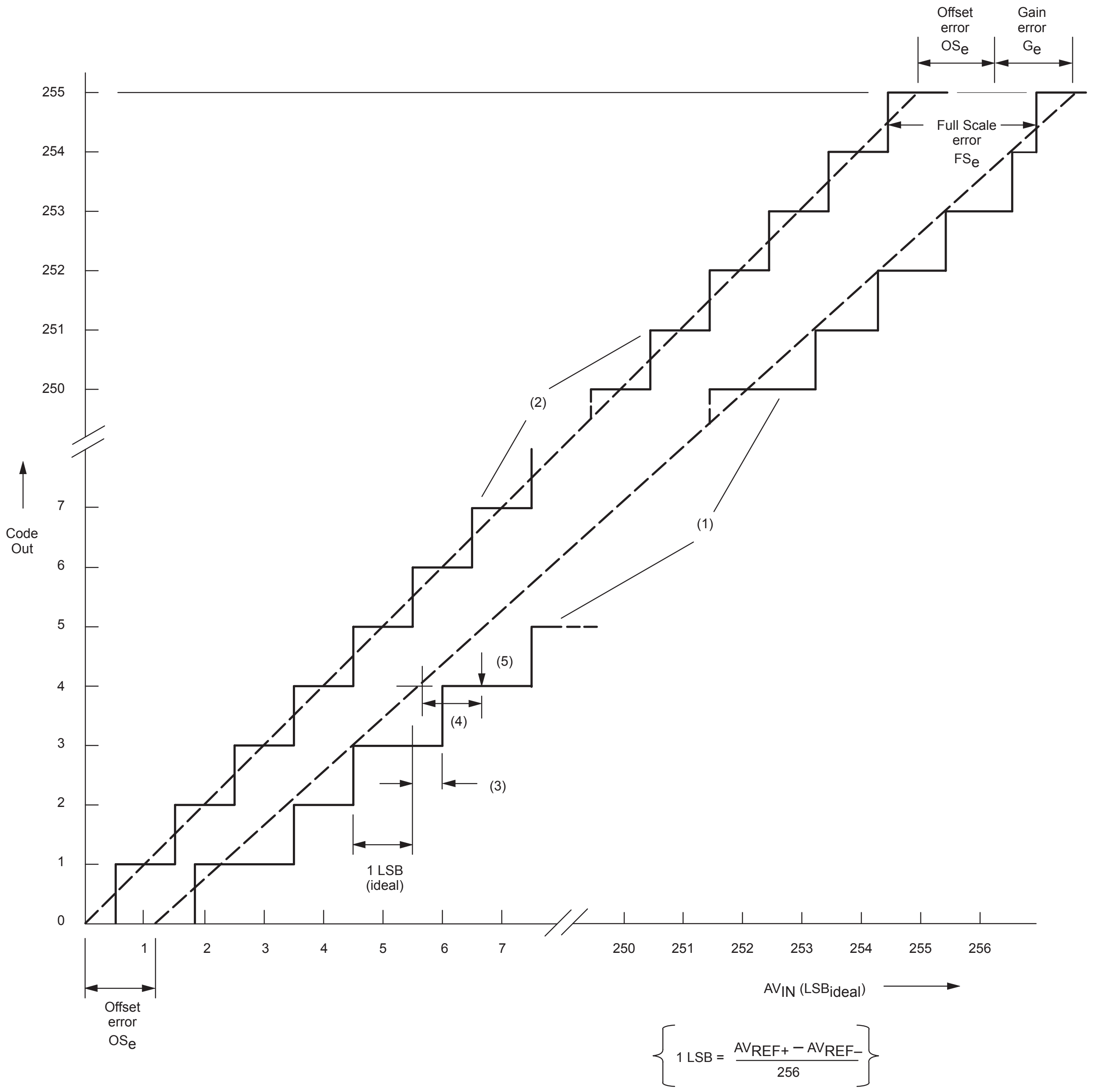
There are a total of 17 **hardware** interrupt sources, enable bits, priority bit sets, etc.

The XA-S3 supports a total of 17 maskable event interrupt sources (for the various XA peripherals), seven software interrupts, 5 exception interrupts (plus reset), and 16 traps. The maskable event interrupts share a global interrupt disable bit (the EA bit in the IEL register) and each also has a separate individual interrupt enable bit (in the IEL or IEH registers). Only three bits of the IPA register values are used on the XA-S3. Each event interrupt can be set to occur at one of 8 priority levels via bits in the Interrupt Priority (IP) registers, IPA0 through IPA5. The value 0 in the IPA field gives the interrupt priority 0, in effect disabling the interrupt. A value of 1 gives the interrupt a priority of 9, the value 2 gives priority 10, etc. The result is the same as if all four bits were used and the top bit set for all values except 0. Details of the priority scheme may be found in the *XA User Guide*.

The complete interrupt vector list for the XA-S3, including all 4 interrupt types, is shown in the following tables. The tables include the address of the vector for each interrupt, the related priority register bits (if any), and the arbitration ranking for that interrupt source. The arbitration ranking determines the order in which interrupts are processed if more than one interrupt of the same priority occurs simultaneously.

EXCEPTION/TRAPS PRECEDENCE

DESCRIPTION	VECTOR ADDRESS	ARBITRATION RANKING
Reset (h/w, watchdog, s/w)	0000–0003	0 (High)
Breakpoint	0004–0007	1
Trace	0008–000B	1
Stack Overflow	000C–000F	1
Divide by 0	0010–0013	1
User RETI	0014–0017	1
TRAP 0–15 (software)	0040–007F	1



(1) Example of an actual transfer curve.
(2) The ideal transfer curve.
(3) Differential non-linearity (DL_e).
(4) Integral non-linearity (IL_e).
(5) Center of a step of the actual transfer curve.

Figure 25. ADC Conversion Characteristic

AC ELECTRICAL CHARACTERISTICS (5 V)

V_{DD} = 4.5 V to 5.5 V; T_{amb} = 0 to +70°C for commercial, T_{amb} = –40°C to +85°C for industrial.

SYMBOL	FIGURE	PARAMETER	LIMITS		UNIT
			MIN	MAX	
External Clock					
f_C	32	Oscillator frequency	0	30	MHz
t_C	32	Clock period and CPU timing cycle	$1/f_C$		ns
t_{CHCX}	32	Clock high-time (Note 7)	t_C * 0.5		ns
t_{CLCX}	32	Clock low time (Note 7)	t_C * 0.4		ns
t_{CLCH}	32	Clock rise time (Note 7)		5	ns
t_{CHCL}	32	Clock fall time (Note 7)		5	ns
Address Cycle					
t_{LHLL}	26, 28, 30	ALE pulse width (programmable)	(V1 * t_C) – 6		ns
t_{AVLL}	26, 28, 30	Address valid to ALE de-asserted (set-up)	(V1 * t_C) – 12		ns
t_{LLAX}	26, 28, 30	Address hold after ALE de-asserted	(t_C/2) – 10		ns
Code Read Cycle					
t_{PLPH}	26	$\overline{\text{PSEN}}$ pulse width	(V2 * t_C) – 10		ns
t_{LLPL}	26	ALE de-asserted to $\overline{\text{PSEN}}$ asserted	(t_C/2) – 7		ns
t_{AVIVA}	26	Address valid to instruction valid, ALE cycle (access time)		(V3 * t_C) – 36	ns
t_{AVIVB}	27	Address valid to instruction valid, non-ALE cycle (access time)		(V4 * t_C) – 29	ns
t_{PLIV}	26	$\overline{\text{PSEN}}$ asserted to instruction valid (enable time)		(V2 * t_C) – 29	ns
t_{PHIX}	26	Instruction hold after $\overline{\text{PSEN}}$ de-asserted	0		ns
t_{PHIZ}	26	Bus 3-State after $\overline{\text{PSEN}}$ de-asserted		t_C – 8	ns
t_{IXUA}	26	Hold time of unlatched part of address after instruction latched	0		ns
Data Read Cycle					
t_{RLRH}	28	$\overline{\text{RD}}$ pulse width	(V7 * t_C) – 10		ns
t_{LLRL}	28	ALE de-asserted to $\overline{\text{RD}}$ asserted	(t_C/2) – 7		ns
t_{AVDVA}	28	Address valid to data input valid, ALE cycle (access time)		(V6 * t_C) – 36	ns
t_{AVDVB}	29	Address valid to data input valid, non-ALE cycle (access time)		(V5 * t_C) – 29	ns
t_{RLDV}	28	$\overline{\text{RD}}$ low to valid data in (enable time)		(V7 * t_C) – 29	ns
t_{RHDX}	28	Data hold time after $\overline{\text{RD}}$ de–asserted	0		ns
t_{RHDZ}	28	Bus 3-State after $\overline{\text{RD}}$ de-asserted (disable time)		t_C – 8	ns
t_{DXUA}	28	Hold time of unlatched part of address after data latched	0		ns
Data Write Cycle					
t_{WLWH}	30	$\overline{\text{WR}}$ pulse width	(V8 * t_C) – 10		ns
t_{LLWL}	30	ALE falling edge to $\overline{\text{WR}}$ asserted	(V12 * t_C) – 10		ns
t_{QVWX}	30	Data valid before $\overline{\text{WR}}$ asserted (data set-up time)	(V13 * t_C) – 22		ns
t_{WHQX}	30	Data hold time after $\overline{\text{WR}}$ de-asserted (Note 6)	(V11 * t_C) – 5		ns
t_{AVWL}	30	Address valid to $\overline{\text{WR}}$ asserted (address set-up time) (Note 5)	(V9 * t_C) – 22		ns
t_{UAWH}	30	Hold time of unlatched part of address after $\overline{\text{WR}}$ is de-asserted	(V11 * t_C) – 7		ns
Wait Input					
t_{WTH}	31	WAIT stable after bus strobe ($\overline{\text{RD}}$, $\overline{\text{WR}}$, or $\overline{\text{PSEN}}$) asserted		(V10 * t_C) – 30	ns
t_{WTL}	31	WAIT hold after bus strobe ($\overline{\text{RD}}$, $\overline{\text{WR}}$, or $\overline{\text{PSEN}}$) asserted	(V10 * t_C) – 5		ns

NOTES ON PAGE 41.

V7) This variable represents the programmed width of the $\overline{\text{RD}}$ pulse as determined by the DR1 and DR0 bits or the DRA1, DRA0 in the BTRH register, and the SLEW bit in the BTRL register. Note that during a 16-bit operation on an 8-bit external bus, $\overline{\text{RD}}$ remains low and does not exhibit a transition between the first and second byte bus cycles. V7 still applies for the purpose of determining peripheral timing requirements. The timing for the first byte is for a bus cycle with ALE, the timing for the second byte is for a bus cycle with no ALE.

- For a bus cycle with **no** ALE, V7 = 1 if DR1/0 = 00, 2 if DR1/0 = 01, 3 if DR1/0 = 10, and 4 if DR1/0 = 11.
- For a bus cycle **with** an ALE, V7 = the total bus cycle duration (2 if DRA1/0 = 00, 3 if DRA1/0 = 01, 4 if DRA1/0 = 10, and 5 if DRA1/0 = 11) minus the number of clocks used by ALE (V1 + 0.5).
  Example: if DRA1/0 = 00 and ALEW = 0, then V7 = 2 – (0.5 +0.5) = 1.

V8) This variable represents the programmed width of the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse as determined by the WM1 bit in the BTRL register. V8 = 1 if WM1 = 0, and 2 if WM1 = 1.

V9) This variable represents the programmed address setup time for a write as determined by the data write cycle duration (defined by DW1 and DW0 or the DWA1 and DWA0 bits in the BTRH register), the WM0 bit in the BTRL register, and the value of V8.

- For a bus cycle **with** an ALE, V9 = the total bus write cycle duration (2 if DWA1/0 = 00, 3 if DWA1/0 = 01, 4 if DWA1/0 = 10, and 5 if DWA1/0 = 11) minus the number of clocks used by the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse (V8) minus the number of clocks used by data hold time (0 if WM0 = 0 and 1 if WM0 = 1).
  Example: If DWA1/0 = 10, WM0 = 1, and WM1 = 1, then V9 = 4 – 1 – 2 = 1.
- For a bus cycle with **no** ALE, V9 = the total bus cycle duration (2 if DW1/0 = 00, 3 if DW1/0 = 01, 4 if DW1/0 = 10, and 5 if DW1/0 = 11) minus the number of clocks used by the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse (V8), minus the number of clocks used by data hold time (0 if WMo = 0 and 1 if WM0 = 1).
  Example: If DW1/0 = 11, WM0 = 1, and WM1 = 0, then V9 = 5 – 1 – 1 = 3.

V10) This variable represents the length of a bus strobe for calculation of WAIT set-up and hold times. The strobe may be $\overline{\text{RD}}$ (for data read cycles), $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ (for data write cycles), or $\overline{\text{PSEN}}$ (for code read cycles), depending on the type of bus cycle being widened by WAIT. V10 = 2 for WAIT associated with a code read cycle using $\overline{\text{PSEN}}$. V10 = V8 for a data write cycle using $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$. V10 = V7 – 1 for a data read cycle using $\overline{\text{RD}}$. This means that a single clock data read cycle cannot be stretched using WAIT. If WAIT is used to vary the duration of data read cycles, the $\overline{\text{RD}}$ strobe width must be set to be at least two clocks in duration. Also see Note 4.

V11) This variable represents the programmed write hold time as determined by the WM0 bit in the BTRL register. V11 0 if the WM0 bit = 0, and 1 if the WM0 bit = 1.

V12) this variable represents the programmed period between the end of the ALE pulse and the beginning of the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse as determined by the data write cycle duration (defined by the DWA1 and DWA0 bits in the BTRH register), the WM0 bit in the BTRL register, and the values of V1 and V8. V12 = the total bus cycle duration (2 if DWA1/0 = 00, 3 if DWA1/0 = 01, 4 if DWA1/0 = 10, and 5 if DWA1/0 = 11) minus the number of clocks used by the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse (V8), minus the number of clocks used by data hold time (0 if WM0 = 0 and 1 if WM0 = 1), minus the width of the ALE pulse (V1).
Example: If SWA1/0 = 11, WM0 = 1, WM1 = 0, and ALEW = 1, then V12 = 5 – 1 – 1 – 1.5 = 1.5.

V13) This variable represents the programmed data setup time for a write as determined by the data write cycle duration (defined by DW1 and DW0 or the DWA1 and DWA0 bits in the BTRH register), the WM0 bit in the BTRL register, and the values of V1 and V8.

- For a bus cycle **with** an ALE, V13 = the total bus cycle duration (2 if DWA1/0 = 00, 3 if DWA1/0 = 01, 4 if DWA1/0 = 10, and 5 if DWA1/0 = 11) minus the number of clocks used by the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse (V8), minus the number of clocks used by data hold time (0 if WM0 = 0 and 1 if WM0 = 1), minus the number of clocks used by ALE (V1 + 0.5).
  Example: If DWA1/0 = 11, WM0 = 1, WM1 = 1, and ALEW = 0, then V13 = 5 – 1 – 2 – 1 = 1.
- For a bus cycle with **no** ALE, V13 = the total bus cycle duration (2 if DW1/0 = 00, 3 if DW1/0 = 01, 4 if DW1/0 = 10, and 5 if DW1/0 = 11) minus the number of clocks used by the $\overline{\text{WRL}}$ and/or $\overline{\text{WRH}}$ pulse (V8), minus the number of clocks used by data hold time (0 if WM0 = 0 and 1 if WM0 = 1).
  Example: If DW1/0 = 01, WM0 = 1, and WM1 = 0, then V13 = 3 – 1 – 1 = 1.

3. Not all combinations of bus timing configuration values result in valid bus cycles. Please refer to the *XA User Guide* section on the External Bus for details.
4. When code is being fetched for execution on the external bus, a burst mode fetch is used that dows not have $\overline{\text{PSEN}}$ edges in every fetch cycle. This would be A3–A0 for an 8-bit bus, and A3–A1 for a 16-bit bus. Also, a 16-bit read operation conducted on an 8-bit wide bus similarly does not include two separate $\overline{\text{RD}}$ strobes. So, a rising edge on the low order address line (A0) must be used to trigger a WAIT in the second half of such a cycle.
5. This parameter is provided for peripherals that have the data clocked in on the falling edge of the $\overline{\text{WR}}$ strobe. This is not usually the case and in most applications this parameter is not used.
6. Please note that the XA-S3 requires that extended data bus hold time (WM0 = 1) to be used with external bus write cycles.
7. Applies only to an external clock source, not when a crystal is connected to the XTAL1 and XTAL2 pins.
8. WAIT should not change between these times.

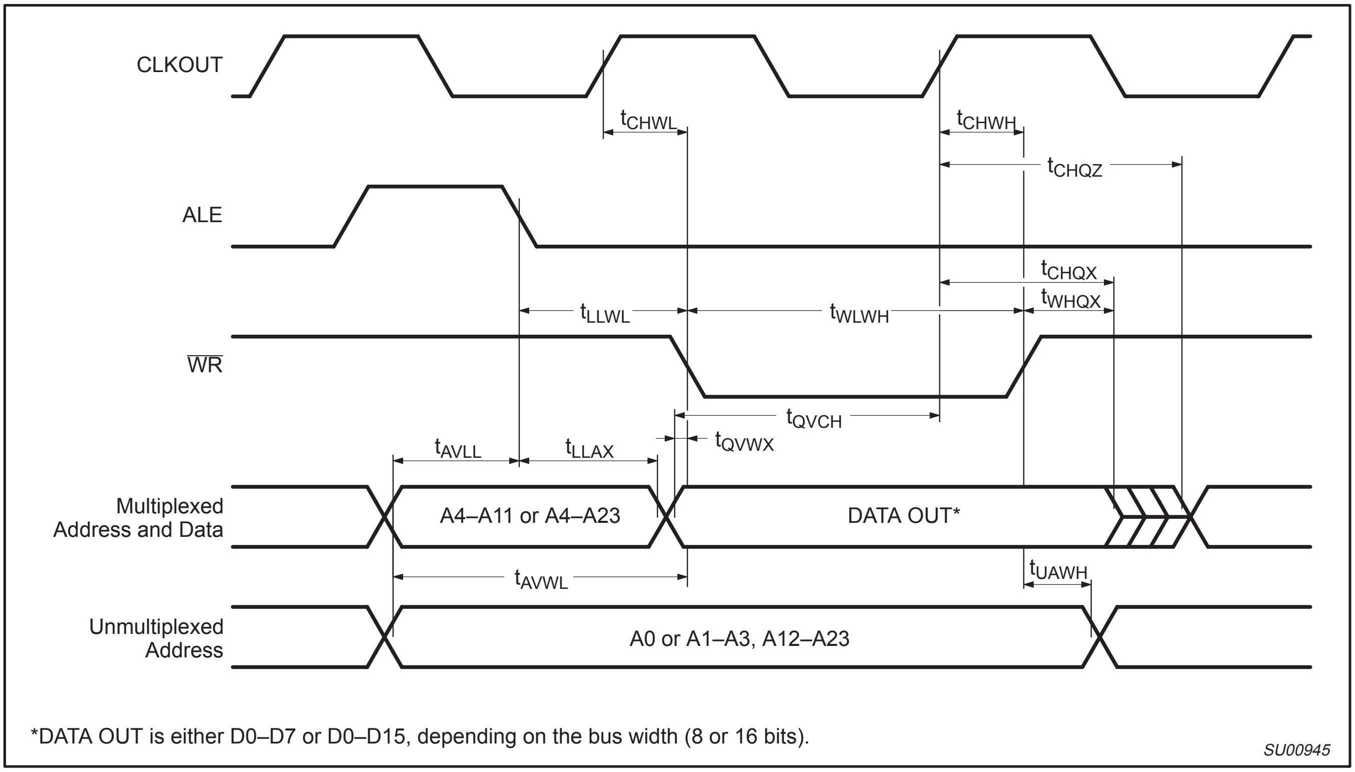


Figure 30. External Data Memory Write Cycle

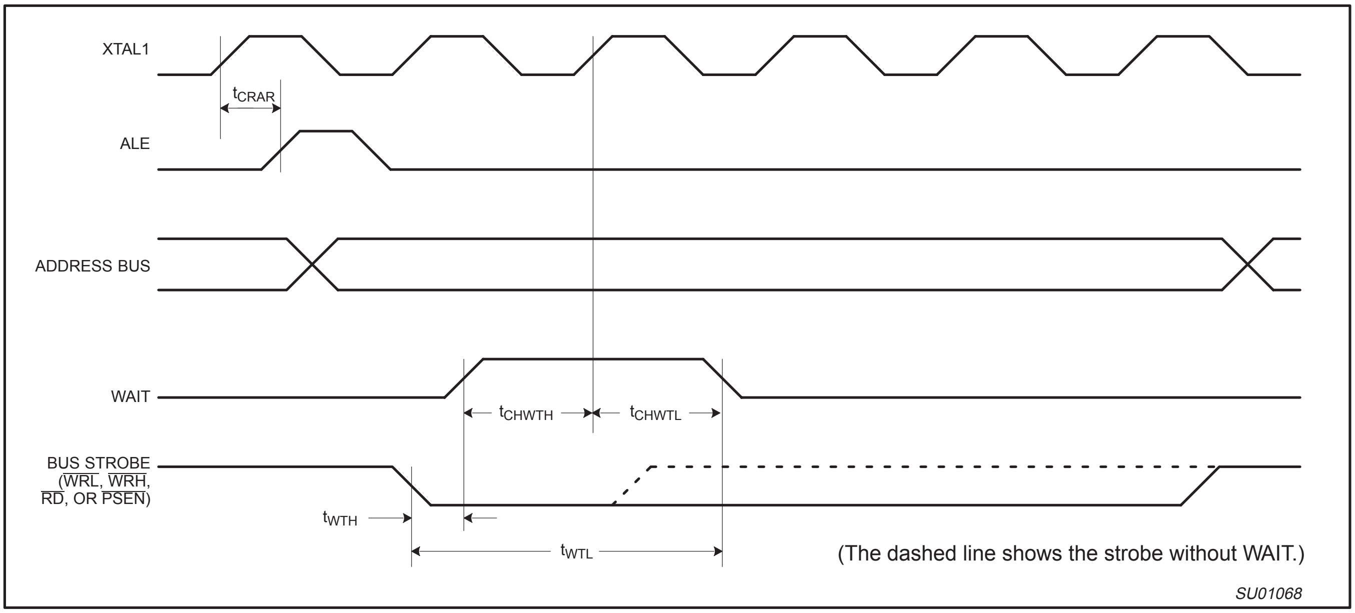

XTAL1
tCRAR
ALE
ADDRESS BUS
WAIT
tCHWTH
tCHWTL
BUS STROBE (WRL, WRH, RD, OR PSEN)
tWTH
tWTL
(The dashed line shows the strobe without WAIT.)
SU01068

Figure 31. WAIT Signal Timing

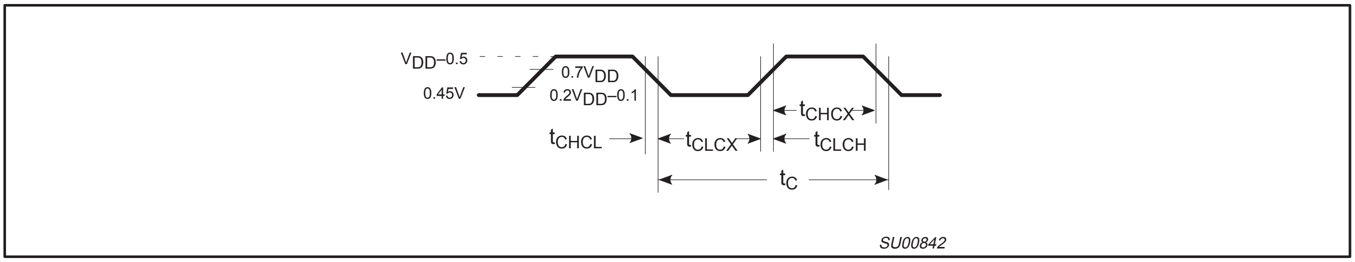
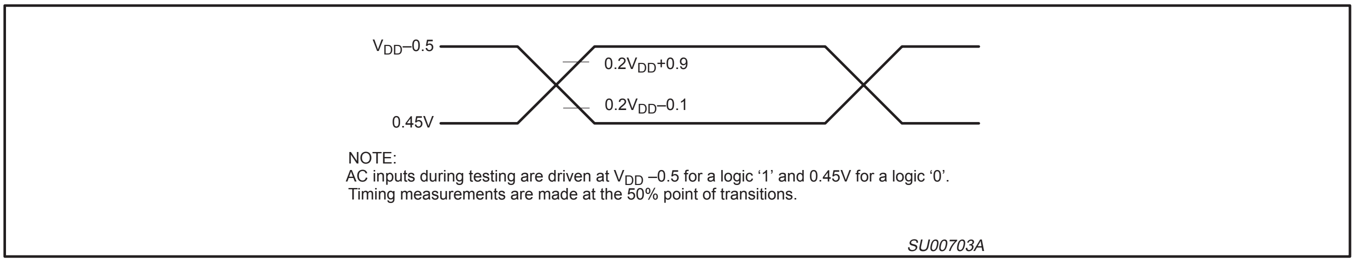
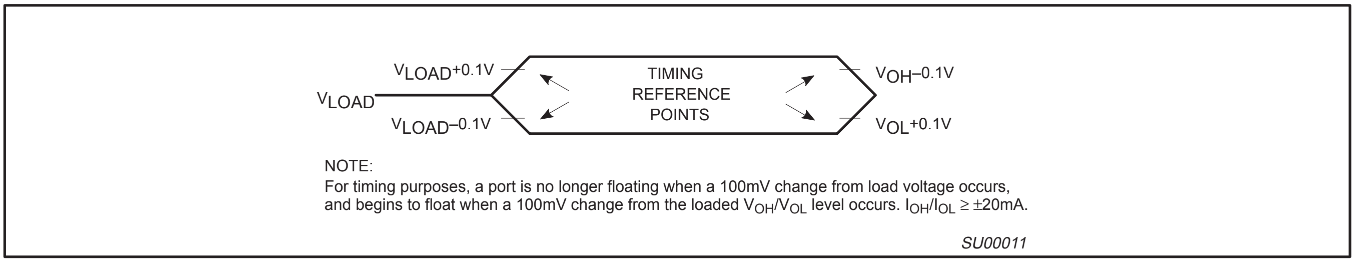


Figure 32. External Clock Drive



NOTE:
AC inputs during testing are driven at V_{DD} –0.5 for a logic ‘1’ and 0.45V for a logic ‘0’.
Timing measurements are made at the 50% point of transitions.

Figure 33. AC Testing Input/Output



NOTE:
For timing purposes, a port is no longer floating when a 100mV change from load voltage occurs, and begins to float when a 100mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OH}/I_{OL} \geq \pm 20$mA.

Figure 34. Float Waveform

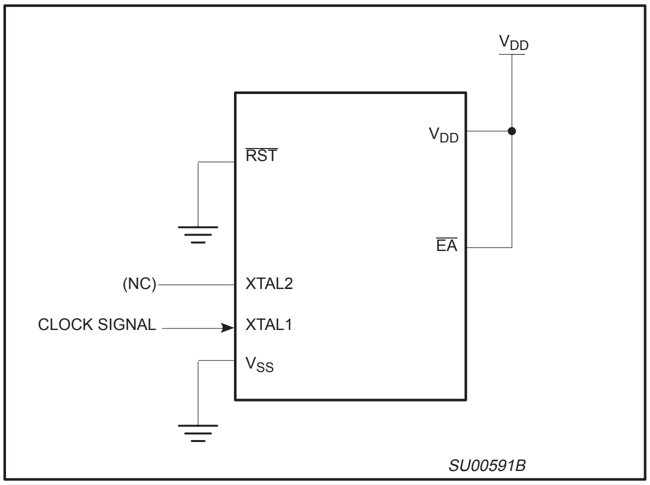


Figure 35. I_{DD} Test Condition, Active Mode All other pins are disconnected

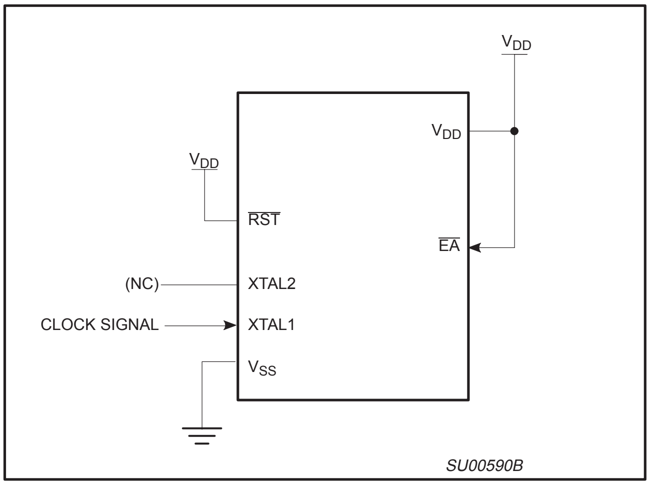


Figure 36. I_{DD} Test Condition, Idle Mode All other pins are disconnected

NOTES

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

TRADEMARKS

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I^{2}C-bus — logo is a trademark of NXP B.V.

NXP Semiconductors

Customer notification

This data sheet was changed to reflect the new company name NXP Semiconductors.
Changes to content include: Corrected SOT188-3 to SOT188-2; changed data sheet specification to Product; updated legal definitions and disclaimers.

Contact information

For additional information please visit: **http://www.nxp.com**
For sales offices addresses send e-mail to: **salesaddresses@nxp.com**

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands R05/10/pp54 Date of release: 2013 Sep 04 Document order number: 9397 750 07816