LIST OF FIGURES

LIST OF TABLES

GENERAL DESCRIPTION

The XA–C3 is a member of the Philips XA (eXtended Architecture) family of high–performance 16–bit single–chip microcontrollers. The XA–C3 combines an array of standard peripherals together with a PeliCAN CAN 2.0B engine and unique "Message Management" hardware to provide integrated support for most CAN Transport Layer (CTL) protocols such as DeviceNet, CANopen and OSEK. For additional details, refer to *Appendix A: XA-C3 CAN Functional Specification*.

The XA architecture supports:

- Upwards compatibility with 80C51 architecture
- 16–bit fully static CPU with 24–bit addressed PROGRAM and DATA spaces
- Twenty-one 16–bit CPU core registers capable of all arithmetic and logic operations while serving as memory pointers.
- An enhanced orthogonal instruction set tailored for high–level support of the C language
- Multi–tasking and direct real–time executive support
- Low–power operation intrinsic to the XA architecture includes Power–Down and Idle modes.

FEATURES IN COMMON WITH XA–G3

- Pin–compatibility (CAN RxD and CAN TxD use the XA-G3 NC pins).
- 32K bytes of on–chip EPROM/ROM PROGRAM memory (see Table 1)
- 44–pin PLCC (Figure 1 and Table 2) and 44–pin LQFP (Figure 2 and Table 3) packages.
- Commercial (0 to 70^{o}C) and Industrial (-40 to 85^{o}C) ranges
- Supports off–chip addressing of PROGRAM and DATA memory up to 1 megabyte each (20 address lines).
- Three standard counter/timers (T0, T1, and T2) with enhancements such as Auto Reload for PWM outputs
- UART-0 with enhancements such as separate Rx and Tx interrupts, Break Detection, and Automatic Address Recognition
- Watchdog with a secure WFEED1 / WFEED2 sequence
- Four 8–bit I/O ports with 4 programmable output configurations per pin.

XA–C3 SPECIFIC FEATURES

- 32 MHz operating frequency at 4.5 to 5.5V operation.
- One Serial Port Interface (SPI)
- 1024 bytes of on–chip DATA RAM.
- 42 vectored interrupts. These include 13 maskable Events, 7 Software interrupts, 6 Exceptions, 16 software Traps, segmented DATA memory, multiple User stacks, and banked registers to support rapid context switching

XA–C3 CAN AND CTL FEATURES

- A PeliCAN CAN 2.0B engine from the SJA1000 Stand-alone CAN controller which supports 11– and 29–bit IDentifiers and the maximum CAN data rate (1 Mbps) and CAN Diagnostics.
- Hardware “Message Management” support for all major CTL protocols: DeviceNet, CANopen, OSEK.
- Automatic (hardware) assembly of Fragmented Messages. Concurrent assembly of up to 32 separate interleaved Fragmented Messages
- 32 CAN Transport Layer (CTL) Message Objects are modelled as a FullCAN Object Superset.
- 32 separate filters/screeners (one per Message Object), each allowing a 30–bit ID Match and full 29–bit Mask (i.e., each filter/screener represents a unique Group address).
- Each Message Object can be configured as Receive or Transmit.
- A separate message buffer is associated with each CTL Message Object. 32 message buffers are located in XRAM and

managed by 32 DMA channels. Message buffer size for each Message Object is independently configurable in length (from 2 to 256 bytes).

- For single–chip systems there is a 512-byte (on–chip) XRAM message buffer, independent of the 1K on–chip DATA RAM, which is extendable (off–chip) to 8K bytes (i.e., 32 Message Objects that can be up to 256 bytes each).

LOGIC SYMBOL AND BLOCK DIAGRAM

Refer to Figure 3 for the logic symbol for the XA-C3 and to Figure 4 for a simplified block diagram representation.

UPGRADING XA–G3 DESIGNS TO CAN

- XA-G3 NC pins are XA-C3 CAN RxD and CAN TxD pins.
- XA-G3 UART-1 is replaced by a Serial Port Interface (SPI)
- XA-C3 software must never write to the BCR register
- XA-C3 software must initialize BTRH and BTRL with 00h

ORDERING INFORMATION

Table 1. Ordering Information

XA–C3 Type & Part Number	Temperature Range (degrees C.)	Package Description	Operating Frequency (MHz)	Drawing Number
OTP				
PXAC37 KB BD	0 to +70	Low Profile PQFP [LQFP44]	32	SOT389–1
PXAC37 KB A	0 to +70	PLCC [PLCC44]	32	SOT187–2
PXAC37 KF BD	–40 to +85	Low Profile PQFP [LQFP44]	32	SOT389–1
PXAC37 KF A	–40 to +85	PLCC [PLCC44]	32	SOT187–2
ROMless				
XAC30 KB BD	0 to +70	Low Profile PQFP [LQFP44]	32	SOT389–1
XAC30 KB A	0 to +70	PLCC [PLCC44]	32	SOT187–2
XAC30 KF BD	–40 to +85	Low Profile PQFP [LQFP44]	32	SOT389–1
XAC30 KF A	–40 to +85	PLCC [PLCC44]	32	SOT187–2
ROM				
XAC33 KB BD	0 to +70	Low Profile PQFP [LQFP44]	32	SOT389–1
XAC33 KB A	0 to +70	PLCC [PLCC44]	32	SOT187–2
XAC33 KF BD	–40 to +85	Low Profile PQFP [LQFP44]	32	SOT389–1
XAC33 KF A	–40 to +85	PLCC [PLCC44]	32	SOT187–2

PIN CONFIGURATIONS

44–PIN PLCC PACKAGE

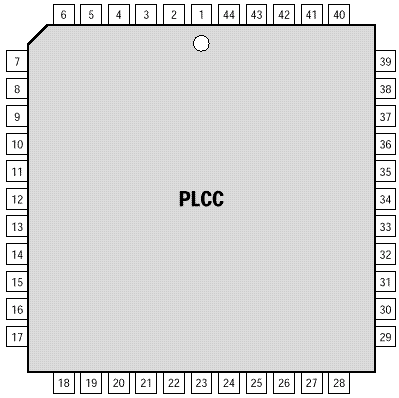

Figure 1. 44–pin PLCC Package

Table 2. 44-pin PLCC Package Pin Functions

Pin	Function (see Note)	Pin	Function (see Note)
1	VSS	23	VDD
2	P1.0 ; A0 ; WRH/	4	P2.0 ; A12D8
3	P1.1 ; A1	25	P2.1 ; A13D9
4	P1.2 ; A2	26	P2.2 ; A14D10
5	P1.3 ; A3	27	P2.3 ; A15D11
6	P1.4 ; SPIRx	28	P2.4 ; A16D12
7	P1.5 ; SPITx	29	P2.5 ; A17D13
8	P1.6 ; T2 ; SPICLK	30	P2.6 ; A18D14
9	P1.7 ; T2EX	31	P2.7 ; A19D15
10	RST/	32	PSEN/
11	P3.0 ; RxD0	33	ALE ; PROG/
12	CAN RxD	34	CAN TxD
13	P3.1 ; TxD0	35	EA/ ; Vpp ; WAIT
14	P3.2 ; INT0/	36	P0.7 ; A11D7
15	P3.3 ; INT1/	37	P0.6 ; A10D6
16	P3.4 ; T0	38	P0.5 ; A9D5
17	P3.5 ; T1 ; BUSW	39	P0.4 ; A8D4
18	P3.6 ; WRL/	40	P0.3 ; A7D3
19	P3.7 ; RD/	41	P0.2 ; A6D2
20	XTAL2	42	P0.1 ; A5D1
21	XTAL1	43	P0.0 ; A4D0
22	VSS	44	VDD

Note: All active–low signals are indicated by a "/" symbol

44–PIN LQFP PACKAGE

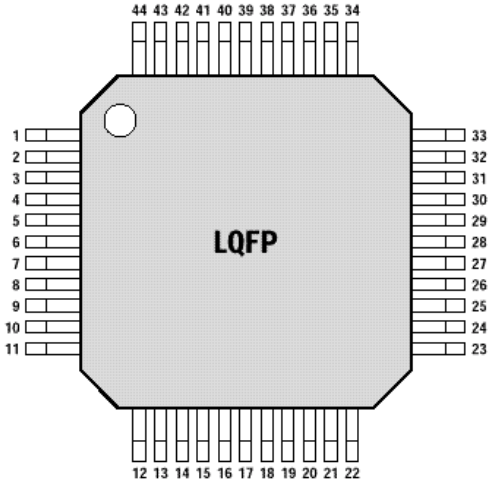

Figure 2. 44–pin LQFP Package

Table 3. 44-pin LQFP Package Pin Functions

Pin	Function (see Note)	Pin	Function (see Note)
1	P1.5 ; SPITx	23	P2.5 ; A17D13
2	P1.6 ; T2 ; SPICLK	4	P2.6 ; A18D14
3	P1.7 ; T2EX	25	P2.7 ; A19D15
4	RST**/**	26	PSEN**/**
5	P3.0 ; RxD0	27	ALE ; PROG**/**
6	CAN RxD	28	CAN TxD
7	P3.1 ; TxD0	29	EA**/** ; Vpp ; WAIT
8	P3.2 ; INT0**/**	30	P0.7 ; A11D7
9	P3.3 ; INT1**/**	31	P0.6 ; A10D6
10	P3.4 ; T0	32	P0.5 ; A9D5
11	P3.5 ; T1 ; BUSW	33	P0.4 ; A8D4
12	P3.6 ; WRL**/**	34	P0.3 ; A7D3
13	P3.7 ; RD**/**	35	P0.2 ; A6D2
14	XTAL2	36	P0.1 ; A5D1
15	XTAL1	37	P0.0 ; A4D0
16	VSS	38	VDD
17	VDD	39	VSS
18	P2.0 ; A12D8	40	P1.0 ; A0 ; WRH**/**
19	P2.1 ; A13D9	41	P1.1 ; A1
20	P2.2 ; A14D10	42	P1.2 ; A2
21	P2.3 ; A15D11	43	P1.3 ; A3
22	P2.4 ; A16D12	44	P1.4 ; SPIRx

Note: All active–low signals are indicated by a "**/**" symbol

LOGIC SYMBOL

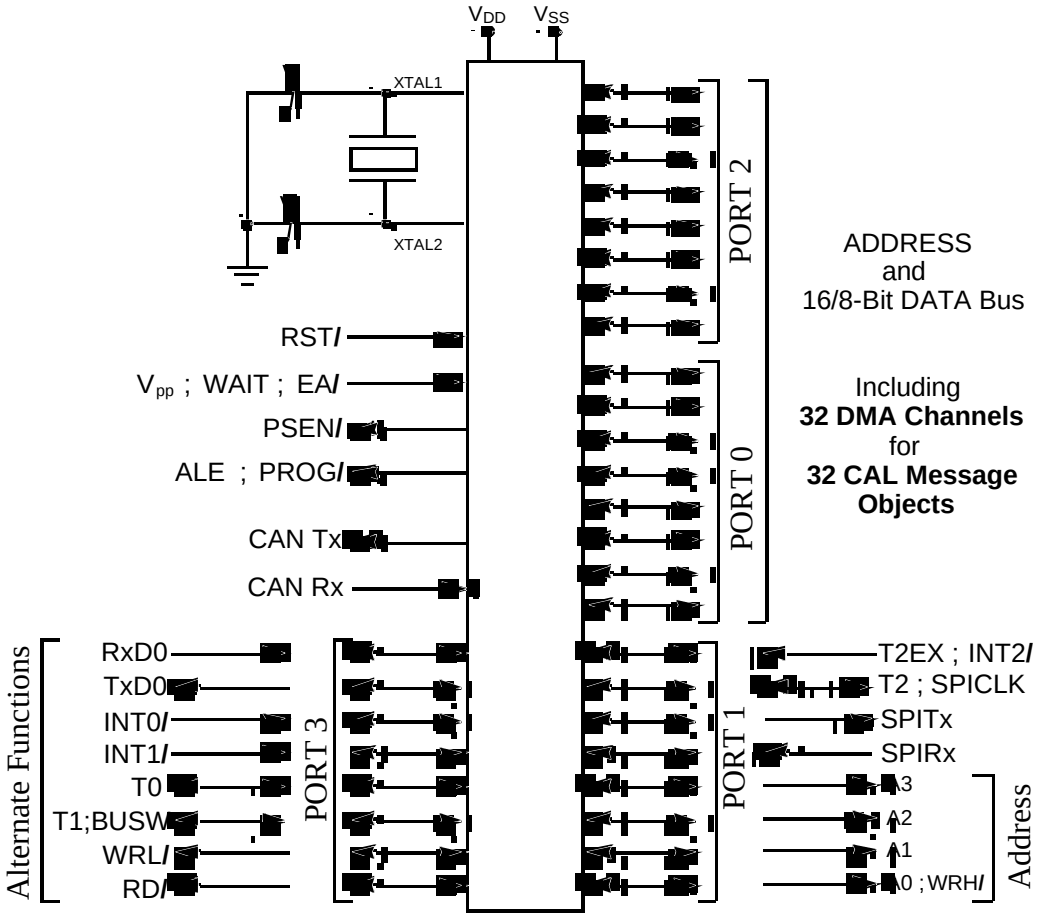

Figure 3. Logic Symbol

BLOCK DIAGRAM

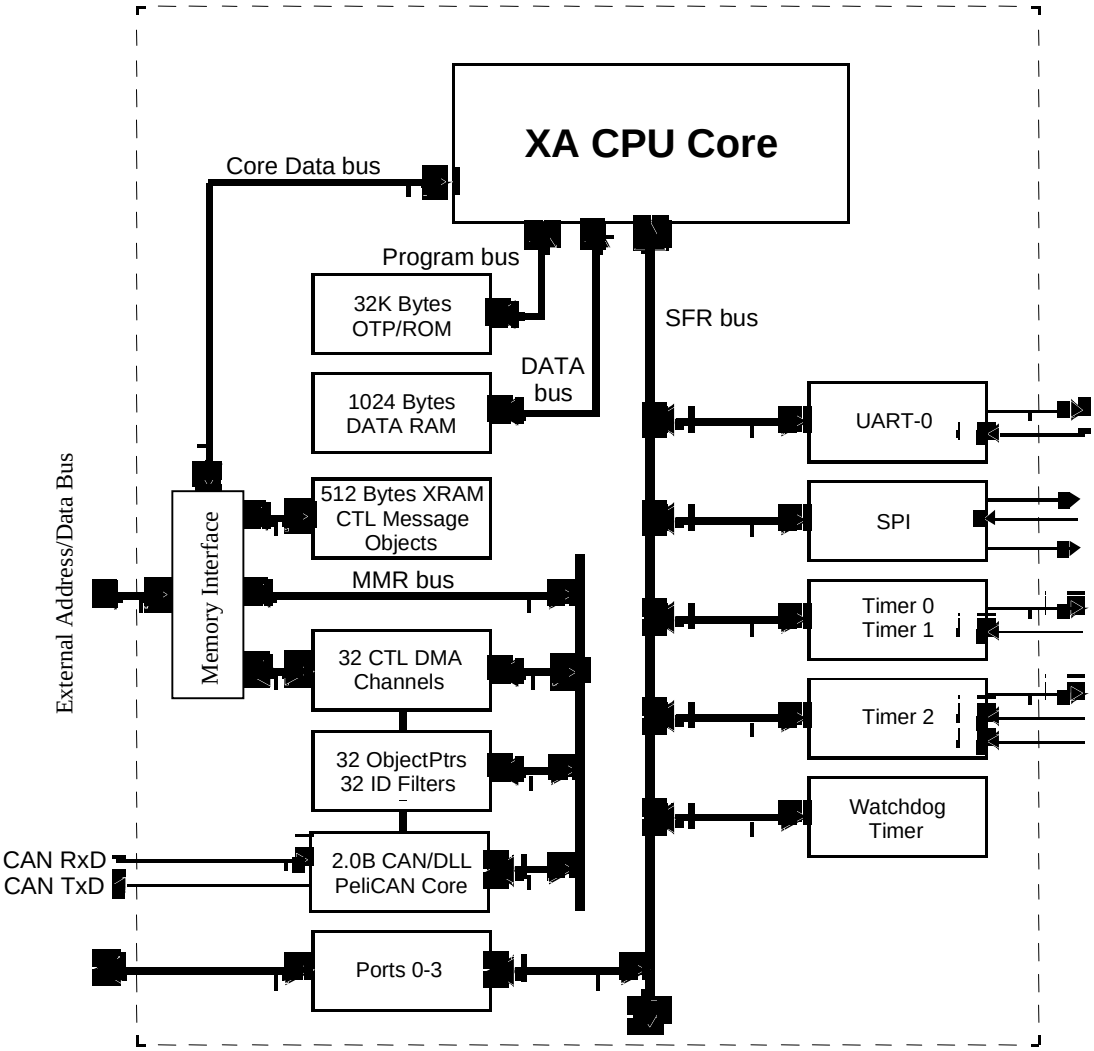

Figure 4. XA–C3 Simplified Block Diagram

PIN DESCRIPTIONS

Table 4. Pin Descriptions

MNEMONIC	PIN NUMBERS		TYPE	NAME AND FUNCTION
	PLCC	LQFP		Note: 1) All active–low signals are indicated by a "*/*" symbol.
V_{SS}	1, 22	16, 39	I	Ground: 0V Reference.
V_{DD}	23, 44	17, 38	I	Power Supply: This is the power supply voltage for normal, Idle and Power–Down operation.
P0.0 – P0.7	43 – 36	37 – 30	I/O	Port 0: Port 0 is an 8–bit I/O Port with user –configurable pins. Port 0 latches have 1's written to them and are configured in the Quasi–Bidirectional mode during Reset. The operation of Port 0 pins as inputs or outputs depends upon the Port configuration selected. Each Port pin is configured independently. *Refer to the sections on I/O Port configuration and DC Electrical Characteristics for details.* Note: 2) When the External PROGRAM/DATA bus is used, Port 0 becomes the multiplexed low DATA/Instruction Byte and Address lines 4 through 11.
P1.0 – P1.7	2 – 9	40 – 44 1 – 3	I/O	Port 1: Port 1 is an 8–bit I/O Port with user –configurable pins. Port 1 latches have 1's written to them and are configured in the Quasi–Bidirectional mode during Reset. The operation of Port 1 pins as inputs or outputs depends upon the Port configuration selected. Each Port pin is configured independently. *Refer to the sections on I/O Port configuration and DC Electrical Characteristics for details.*
P1.0	2	40	O	A0 ; WRH*/*: Address bit 0 of the External Address bus when the External DATA bus is configured for 8–bit width. When the External DATA bus is configured for 16–bit width, this pin becomes the High Byte Write Strobe (WRH).
P1.1	3	41	O	A1: Address bit 1 of the External Address bus.
P1.2	4	42	O	A2: Address bit 2 of the External Address bus.
P1.3	5	43	O	A3: Address bit 3 of the External Address bus.
P1.4	6	44	I	SPIRx: Receiver serial input of SPI.
P1.5	7	1	O	SPITx: Transmitter serial output of SPI.
P1.6	8	2	I	T2 ; SPICLK: Timer/counter 2 external clock input or Timer/counter 2 Clock–Out mode output, or SPI Clock output. Notes: 3) SPICLK must be configured to idle in the logic '1' state in order to use either the T2 or P1.6 output functions, even if the SPI Port is not in use! 4) The default state from Reset of the SPICLK polarity is "inverted" which yields an SPICLK idle state of logic '1'. 5) If the SPI Clock polarity is changed by the user during SPI Port usage, it must be restored to "inverted" polarity before using either the P1.6 or Timer/counter 2 output functions.
P1.7	9	3	O	T2EX: Timer/counter 2 reload/capture/direction control.
P2.0 – P2.7	24 – 31	18 – 25	I/O	Port 2: Port 2 is an 8–bit I/O port with user–configurable pins. Port 2 latches have 1's written to them and are configured in the Quasi–Bidirectional mode during Reset. The operation of Port 2 pins as inputs or outputs depends upon the Port configuration selected. Each Port pin is configured independently. *Refer to the sections on I/O port configuration and DC Electrical Characteristics for details.* Notes: 6) When the External PROGRAM/DATA bus is used in 16–bit mode, Port 2 becomes the multiplexed High DATA/Instruction Byte and Address lines 12 through 19. 7) When the External PROGRAM/DATA bus is used in 8–bit mode, the number of Address lines that appear on Port 2 is user–programmable.
P3.0 – P3.7	11, 13 – 19	5, 7 –12	I/O	Port 3: Port 3 is an 8–bit I/O Port with user–configurable pins. Notes: 8) Port 3 latches have 1's written to them and are configured in the Quasi–Bidirectional mode during Reset. 9) The operation of Port 3 pins as inputs or outputs depends upon the Port configuration selected. 10) Each Port pin is configured independently. *Refer to the sections on I/O Port configuration and DC Electrical Characteristics for details.*
P3.0	11	5	I	RxD0: Receiver serial input of UART 0.
P3.1	13	7	O	TxD0: Transmitter serial output of UART 0.
P3.2	14	8	I	INT0*/*: External interrupt 0 input.
P3.3	15	9	I	INT1*/*: External interrupt 1 input.
P3.4	16	10	I/O	T0: Timer 0 External count input or Timer 0 Overflow output.
P3.5	17	11	I/O	T1 ; BUSW: Timer 1 External count input or Timer 1 Overflow output. Note: 11) The value on this pin is latched as the (External) Reset input is released and defines the default External DATA bus width (1 is 16–Bits).
P3.6	18	12	O	WRL*/*: External DATA memory Low Byte Write Strobe.
P3.7	19	13	O	RD*/*: External DATA memory Read Strobe.
RST*/*	10	4	I	RESET*/*: Note: 12) A low on this pin resets the XA–C3, 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. *Refer to the Reset section for details.*
ALE ; PROG*/*	33	27	I/O	Address Latch Enable ; Program Pulse*/*: Notes: 13) A high output on the ALE pin signals External circuitry to latch the address portion of the multiplexed Address/DATA bus. 14) A pulse on ALE occurs only when needed to process an External bus cycle. During EPROM programming, this pin is used as the Program pulse input.

PSEN/	32	26	O	Program Store Enable/: This is the Read Strobe for External PROGRAM memory. Notes: 15) When the microcontroller accesses External PROGRAM memory, PSEN/ is driven low in order to enable memory devices. 16) PSEN/ is only active when External code accesses are performed.
EA/ ; WAIT ; V_{PP}	35	29	I	External Access/ ; WAIT ; Programming Supply Voltage: Notes: 17) The EA/ input determines whether the internal PROGRAM memory of the XA–C3 is used for code execution. 18) The EA/ pin is latched as the (External) Reset input is released and its value applied 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 is used above that point. 19) After Reset is released, this pin takes on the function of a Bus WAIT input. If WAIT is asserted High during any External bus access, that cycle will be extended until WAIT is released. 20) During EPROM programming, this pin is also the programming supply voltage input.
CAN RxD	12	6	I	CAN Receive Data input: CAN serial receiver input to the SJA1000 PeliCAN core.
CAN TxD	34	28	O	CAN Transmit Data output: CAN serial transmitter output from the SJA1000 PeliCAN core.
XTAL1	21	15	I	Crystal 1: Input to the inverting amplifier used in the oscillator circuit and input to the internal clock generator circuits.
XTAL2	20	14	O	Crystal 2: Output from the oscillator amplifier.

SPECIAL FUNCTION REGISTERS

Table 5. Special Function Registers

NAME	DESCRIPTION	SFR ADDRESS	BIT FUNCTIONS AND BIT ADDRESSES								RESET VALUE
			7	6	5	4	3	2	1	0	
BCR	Bus Configuration Register	46Ah	–	–	–	WAITD	BUSD	BC2	BC1	BC0	07h (Note 1)
BTRH	Bus Timing Register High	469h	DW1	DW0	DWA1	DWA0	DR1	DR0	DRA1	DRA0	FFh (Note 2)
BTRL	Bus Timing Register Low	468h	WM1	WM0	ALEW	–	CR1	CR0	CRA1	CRA0	EFh (Note 2)
MIFCNTL	MIF Control Register	495h	–	–	–	WDSBL	BUSD	8/ or 16	BC1	BC0	Note 3
MRBL	MMR Base address Low	496h	MA15	MA14	MA13	MA12	–	–	–	MRBE	F0h
MRBH	MMR Base address High	497h	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16	0Fh
DS	Data Segment	441h									00h
ES	Extra Segment	442h									00h
CS	Code Segment	443h									00h
			33F	33E	33D	33C	33B	33A	339	338	
IEH*	Interrupt Enable High	427h	EMRI	EMTI	EMER	ECER	ESPI	-	ETI0	ERI0	00h
			337	336	335	334	333	332	331	330	
IEL*	Interrupt Enable Low	426h	EA	–	EBUFF	ET2	ET1	EX1	ET0	EX0	00h
IPA0	Interrupt Priority Assignment 0	4A0h	–	PT0			–	PX0			00h
IPA1	Interrupt Priority Assignment 1	4A1h	–	PT1			–	PX1			00h
IPA2	Interrupt Priority Assignment 2	4A2h	–	PBUFF			–	PT2			00h
IPA4	Interrupt Priority Assignment 4	4A4h	–	PTI0			–	PRI0			00h
IPA5	Interrupt Priority Assignment 5	4A5h	–	PSPI			–	-			00h
IPA6	Interrupt Priority Assignment 6	4A6h	–	PMER			–	PCER			00h
IPA7	Interrupt Priority Assignment 7	4A7h	–	PMRI			–	PMTI			00h
			387	386	385	384	383	382	381	380	
P0*	Port 0	430h	A11D7	A10D6	A9D5	A8D4	A7D3	A6D2	A5D1	A4D0	FFh
			38F	38E	38D	38C	38B	38A	389	388	
P1*	Port 1	431h	T2EX	T2 ; SPICLK	SPITx	SPIRx	A3	A2	A1	A0 ; WRH/	FFh
			397	396	395	394	393	392	391	390	
P2*	Port 2	432h	A19D15	A18D14	A17D13	A16D12	A15D11	A14D10	A13D9	A12D8	FFh
			39F	39E	39D	39C	39B	39A	399	398	
P3*	Port 3	433h	RD/	WRL/	T1	T0	INT1/	INT0/	TxD0	RxD0	FFh
P0CFGA	Port 0 Configuration A	470h									Note 4
P1CFGA	Port 1 Configuration A	471h									Note 4
P2CFGA	Port 2 Configuration A	472h									Note 4
P3CFGA	Port 3 Configuration A	473h									Note 4
P0CFGB	Port 0 Configuration B	4F0h									Note 4
P1CFGB	Port 1 Configuration B	4F1h									Note 4
P2CFGB	Port 2 Configuration B	4F2h									Note 4
P3CFGB	Port 3 Configuration B	4F3h									Note 4
			227	226	225	224	223	222	221	220	
PCON*	Power Control Reg	404h	–	–	–	–	–	–	PD	IDL	00h
			20F	20E	20D	20C	20B	20A	209	208	
PSWH*	Program Status Word High	401h	SM	TM	RS1	RS0	IM3	IM2	IM1	IM0	Note 5
			207	206	205	204	203	202	201	200	
PSWL*	Program Status Word Low	400h	C	AC	–	–	–	V	N	Z	Note 5
			217	216	215	214	213	212	211	210	
PSW51*	80C51-compatible PSW	402h	C	AC	F0	RS1	RS0	V	F1	P	Note 6
RTH0	Timer 0 extended reload, high byte	455h									00h
RTH1	Timer 1 extended reload, high byte	457h									00h
RTL0	Timer 0 extended reload, low byte	454h									00h
RTL1	Timer 1 extended reload, low byte	456h									00h
			307	306	305	304	303	302	301	300	
S0CON*	Serial port 0 control register	420h	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00h
			30F	30E	30D	30C	30B	30A	309	308	
S0STAT*	Serial port 0 extended status	421h	–	–	–	–	FE0	BR0	OE0	STINT0	00h
S0BUF	Serial port 0 buffer register	460h									xxh
S0ADDR	Serial port 0 address register	461h									00h
S0ADEN	Serial port 0 address enable register	462h									00h
SCR	System configuration register	440h	–	–	–	–	PT1	PT0	CM	PZ	00h
			21F	21E	21D	21C	21B	21A	219	218	
SSEL*	Segment selection register	403h	ESWEN	R6SEG	R5SEG	R4SEG	R3SEG	R2SEG	R1SEG	R0SEG	00h
SWE	Software Interrupt Enable	47Ah	–	SWE7	SWE6	SWE5	SWE4	SWE3	SWE2	SWE1	00h
			357	356	355	354	353	352	351	350	
SWR*	Software Interrupt Request	42Ah	–	SWR7	SWR6	SWR5	SWR4	SWR3	SWR2	SWR1	00h
			2C7	2C6	2C5	2C4	2C3	2C2	2C1	2C0	
T2CON*	Timer 2 control register	418h	TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C2 or T2/	CP or RL2/	00h
			2CF	2CE	2CD	2CC	2CB	2CA	2C9	2C8	
T2MOD*	Timer 2 mode control	419h	–	–	–	–	–	–	T2OE	DCEN	00h
TH2	Timer 2 high byte	459h									00h
TL2	Timer 2 low byte	458h									00h
T2CAPH	Timer 2 capture register, high byte	45Bh									00h
T2CAPL	Timer 2 capture register, low byte	45Ah									00h
			287	286	285	284	283	282	281	280	

TCON*	Timer 0 and 1 control register	410h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h
TH0	Timer 0 high byte	451h									00h
TH1	Timer 1 high byte	453h									00h
TL0	Timer 0 low byte	450h									00h
TL1	Timer 1 low byte	452h									00h
TMOD	Timer 0 and 1 mode control	45Ch	GATE1	C1 or T1/	M1	M0	GATE0	C0 or T0/	M1	M0	00h
			28F	28E	28D	28C	28B	28A	289	288	
TSTAT*	Timer 0 and 1 extended status	411h	–	–	–	–	–	T1OE	–	T0OE	00h
			2FF	2FE	2FD	2FC	2FB	2FA	2F9	2F8	
WDCON*	Watchdog control register	41Fh	PRE2	PRE1	PRE0	–	–	WDRUN	WDTOF	–	Note 7
WDL	Watchdog timer reload	45Fh									00h
WFEED1	Watchdog feed 1	45Dh									xxh
WFEED2	Watchdog feed 2	45Eh									xxh

NOTES:

1. Users should never write to the BCR register.
2. Users must ALWAYS INITIALIZE (Write) 00h to this register.
3. At Reset, the MIFCNTL register is loaded with the binary value 0000 0a11, where “a” is the value (8/ or 16) on the BUSW pin. The address bus defaults to 20 bits.
4. Port configurations default to Quasi–Bidirectional when the XA begins execution from Internal code memory after Reset, based on the condition found on the EA/ pin. Thus, all PnCFGA registers will contain FFh and PnCFGB registers will contain 00h. When the XA begins execution using External code memory, the default configuration for pins that are associated with the External bus will be Push-Pull. The PnCFGA and PnCFGB register contents will reflect this difference.
5. SFR is loaded from the Reset vector.
6. All bits except F1, F0, and P are loaded from the Reset vector. Those bits are all 0.
7. The WDCON Reset value is E6h for a Watchdog Reset, E4h for all other Reset causes. The Watchdog is always turned ON as one consequence of RST/. Therefore, the user should turn OFF the Watchdog if immediate Watchdog operation is not desired: See the Watchdog Timer section in this Data Sheet for a recommended code example.

GENERAL NOTES:

- SFRs marked with an asterisk (*) are bit-addressable.
- The XA–C3 implements an 8–bit SFR bus, as stated in Chapter 8 of the *XA User Guide*. All SFR accesses must be 8–bit operations. Attempts to write 16 bits to an SFR will actually write only the lower 8 bits. Sixteen–bit SFR reads will return undefined data in the upper byte.
- Unimplemented bits in SFRs (indicated by "-"} are unknown at all times. Ones should not be written to these bits since they may be used for other purposes in future XA derivatives. In general, the Reset value shown for these unimplemented bits is 00h.
- The XA guards writes to all SFR bits that can be modified by hardware, including all SFR resident interrupt flags, as well as the WDTOF bit in WDCON. This mechanism, called Read–Modify–Write Lockout, prevents loss of an interrupt (or other status) flag if a bit is written to directly by hardware between the read and write of an instruction that performs a read–modify–write operation.

MEMORY–MAPPED REGISTERS

Table 6. Memory-Mapped Registers

Name	Description	Address Offset	Operation	ACCESS	Reset Value
MESSAGE OBJECT REGISTERS (n = 0 – 31)					
MnMIDH	Message n Match ID High	$000n_4n_3n_2n_1n_0$0000b (n0h)	R/W	Word only	xxxxh
MnMIDL	Message n Match ID Low	$000n_4n_3n_2n_1n_0$0010b (n2h)	R/W	Word only	x…x00b
MnMSKH	Message n Mask High	$000n_4n_3n_2n_1n_0$0100b (n4h)	R/W	Word only	xxxxh
MnMSKL	Message n Mask Low	$000n_4n_3n_2n_1n_0$0110b (n6h)	R/W	Word only	x…x000b
MnCTL	Message n Control	$000n_4n_3n_2n_1n_0$1000b (n8h)	R/W	Byte	00000xxxb
MnBLR	Message n Buffer Location	$000n_4n_3n_2n_1n_0$1010b (nAh)	R/W	Word only	xxxxh
MnBSZ	Message n Buffer Size	$000n_4n_3n_2n_1n_0$1100b (nCh)	R/W	Byte	00000xxxb
MnFCR	Message n Fragmentation Count	$000n_4n_3n_2n_1n_0$1110b (nEh)	R/W	Byte	00xxxxxxb
CAN/CTL INTERRUPT COMPLETE (CIC) REGISTERS					
MCPLH	Message Complete Status Flags High	226h	RC	Word	0000h
MCPLL	Message Complete Status Flags Low	224h	RC	Word	0000h
CANINTFLG	CAN Interrupt Flag Register	228h	RC	Byte	00h
MCIR	Message Complete Information	229h	RO	Byte	00h
MEIR	Message Error Information	22Ah	RO	Byte	00h
FEENR	Frame Error Enable	22Eh	R/W	Byte	00h
FESTR	Frame Error Status	22Ch	RC	Byte	00h
SPI REGISTERS					
SPICFG	SPI Configuration	260h	R/W	Byte	00h
SPIDATA	SPI Data	262h	R/W	Byte	00h
SPICS	SPI Control and Status	263h	R/W	Byte	00h
CAN CORE BLOCK (CCB) REGISTERS					
CANCMR	CAN Core Command	270h	R/W*	Byte	01h (Note 1)
CANSTR	CAN Core Status	271h	RO	Byte	00h
CANBTR	CAN Core Bus Timing	272h	R/W*	Word	0000h
TxERC	Tx Error Counter	274h	R/W*	Byte	00h
RxERC	Rx Error Counter	275h	R/W*	Byte	00h
EWLR	Error Warning Limit	276h	R/W	Byte	96h
ECCR	Error Code Capture	278h	RO	Byte	00h
ALCR	Arbitration Lost Capture	27Ah	RO	Byte	00h
GCTL	Global Control	27Eh	R/W	Byte	00h
MEMORY INTERFACE (MIF) REGISTERS					
MIFBTRH	MIF Bus Timing Register High	293h	R/W	Byte	FFh
MIFBTRL	MIF Bus Timing Register Low	292h	R/W	Byte	EFh
MBXSR	Message Buffer and XRAM Segment Register	291h	R/W	Byte	FFh
XRAMB	XRAM Base Address	290h	R/W	Byte	FEh

Possible Operations: R/W = Read & Write, RO = Read Only, RC = Read then Clear via a service routine, W* = Writeable only while the CAN Core is in Reset mode, x = Undefined after Reset

NOTES:

1. SLPEN (Sleep Enable), CANCMR[3], is writeable only when the CAN Core is in Normal mode.

XA–C3 TIMER/COUNTERS

The XA has two standard 16–bit enhanced Timer/Counters: Timer 0 and Timer 1. Additionally, it has a third 16–bit Up/Down timer/counter, T2. A central timing generator in the XA core provides the time–base for all XA Timers and Counters. The timer/event counters can perform the following functions:

– Measure time intervals and pulse duration

– Count External events

– Generate interrupt requests

– Generate PWM or timed output waveforms

All timer/counters (Timer 0, Timer 1 and Timer 2) can be independently programmed to operate either as timers or event counters. Timer 0 and Timer 1 are selectable via TMOD[6] and TMOD[2], respectively. Timer 2 is selectable via T2CON[1]. All timers may be dynamically read during program execution. All timers count up unless otherwise stated.

When running in timer mode (as opposed to counter mode) the base clock rate of all timers, including the Watchdog timer, is user–programmable. The clock driving the timers is called TCLK and is determined by the setting of two bits (PT1, PT0) (SCR[3:2]) in the System Configuration Register – See Table 5. The frequency of TCLK may be selected to be the oscillator input divided by 4 (f_c/4), the oscillator input divided by 16 (f_c/16), or the oscillator input divided by 64 (f_c/64). This gives a range of possibilities for the XA timer functions, including baud rate generation and Timer 2 capture. Note: Tthis single SCR rate setting applies to all timers.

When timers T0, T1, or T2 are used in the counter mode, the timers will increment whenever a falling edge (high–to–low transition) is detected on an External clock pin. These inputs are sampled once every two oscillator cycles, so it can take as many as four oscillator cycles to detect a transition. Thus, the maximum count rate that can be supported is f_c /4. In general, the duty cycle of the timer clock inputs is not important. However, any high or low state on the timer clock input pins must be present for two oscillator cycles before it is guaranteed to be “seen” by the timer logic.

TIMER 0 AND TIMER 1

These two Timer/Counters have four operating modes, which are selected by bit–pairs (M1, M0) in the TMOD register. Timer modes 1, 2, and 3 in XA are kept identical to the 80C51 timer modes for code compatibility. Only the mode 0 is replaced in the XA by a more powerful 16–bit auto–reload mode. This gives the XA timers a much larger range when used as time bases.
The recommended M1, M0 settings for the different modes are shown in Figure 6.

Figure SU0589 (XA-G3 Data Sheet Figure 1)

goes here.

Figure 5. System Configuration Register (SCR)

Figure SU00605 (Figure 2 in XA-G3 Data Sheet) goes here.

Figure 6. Timer/Counter Mode Control (TMOD) Register

NEW ENHANCED MODE 0

For timers T0 or T1 the 13–bit count mode on the 80C51 (current Mode 0) has been replaced in the XA with a 16–bit auto–reload mode. Four additional 8–bit data registers (two per timer: RTHn and RTLn) are created to hold the auto–reload values. In this mode, the TH overflow will set the TF flag in the TCON register (see Figure 7) and cause both the TL and TH counters to be loaded from the RTL and RTH registers respectively.

These new SFRs will also be used to hold the TL reload data in the 8–bit auto–reload mode (Mode 2) instead of TH.

The overflow rate for Timer 0 or Timer 1 in Mode 0 may be calculated as follows:

$$Timer\_Rate = f_{osc} / (N * (65536 - Timer\_Reload\_Value))$$

where N = the TCLK prescaler value: 4 (default), 16, or 64.

MODE 1

Mode 1 is the 16–bit non–auto reload mode.

MODE 2

Mode 2 configures the Timer register as an 8–bit Counter (TLn) with automatic reload. Overflow from TLn not only sets TFn, but also reloads TLn with the contents of RTLn, which is preset by software. The reload leaves THn unchanged.

Mode 2 operation is the same for Timer/Counter 0.

The overflow rate for Timer 0 or Timer 1 in Mode 2 may be calculated as follows:

$$Timer\_Rate = f_{osc} / (N * (256 - Timer\_Reload\_Value))$$

where N = the TCLK prescaler value: 4, 16, or 64.

MODE 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. TL0 uses the Timer 0 control bits: C0 ; T0/, GATE0, TR0, INT0/ and TF0. TH0 is locked into a timer function and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the “Timer 1” interrupt.

Mode 3 is provided for applications requiring an extra 8–bit timer. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

Figure SU00604C (Figure 3 in XA-G3 Data Sheet) goes here.

Figure 7. Timer/Counter Control (TCON) Register

Figure SU00606A (Figure 4 in XA-G3 Data Sheet) goes here.

Figure 8. Timer/Counter 2 Control (T2CON) Register

NEW TIMER–OVERFLOW TOGGLE OUTPUT

In the XA, the timer module now has two outputs, which toggle on overflow from the individual timers. The same device pins that are used for the T0 and T1 count inputs are also used for the new overflow outputs. An SFR bit (TnOE in the TSTAT register – see Figure 9 -- is associated with each counter and indicates whether Port–SFR data or the overflow signal is output to the pin. These outputs could be used in applications for generating variable duty cycle PWM outputs (changing the auto–reload register values). Also, variable frequency ($f_{osc}/8$ to $f_{osc}/8{,}388{,}608$) outputs could be achieved by adjusting the prescaler along with the auto–reload register values.

TIMER T2

Timer 2 in the XA is a 16–bit Timer/Counter which can operate as either a timer or as an event counter. This is selected by {C2 or T2*I*} (T2CON[1]) (see Figure 8). Upon timer T2 overflow/underflow, the TF2 flag is set, which may be used to generate an interrupt. It can be operated in one of three operating modes: auto–reload (up or down counting), capture, or as the baud rate generator (for the UART via SFRs T2CON and T2MOD – see Figure 10.
These modes are shown in Table 7.

CAPTURE MODE

In the capture mode there are two options which are selected by bit EXEN2 (T2CON[3]). If EXEN2 = 0, then timer 2 is a 16–bit timer or counter, which upon overflowing sets bit TF2 (T2CON[7]), the timer 2 overflow bit. This will cause an interrupt when the timer 2 interrupt is enabled.
If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1–to–0 transition at External input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 (T2CON[6]) to be set. This will cause an interrupt in the same fashion as TF2 when the Timer 2 interrupt is enabled. The capture mode is illustrated in Figure 11.

AUTO–RELOAD MODE (UP OR DOWN COUNTER)

In the auto–reload mode, the timer registers are loaded with the 16–bit value in T2CAPH and T2CAPL when the count overflows. T2CAPH and T2CAPL are initialized by software. If the EXEN2 bit (T2CON[3]) is set, the timer registers will also be reloaded and the EXF2 flag T2CON[6] set when a 1–to–0 transition occurs at input T2EX. The auto–reload mode is shown in Figure 12.
In this mode, Timer 2 can be configured to count up or down. This is done by setting or clearing the DCEN (Down Counter Enable) bit T2MOD[0] (see Table 7). The T2EX pin then controls the count direction. When T2EX is high, the count is in the up direction, when T2EX is low, the count is in the down direction.

Figure 12 shows Timer 2, which will count up automatically, since DCEN = 0. In this mode there are two options selected by bit EXEN2 in the T2CON register. If EXEN2 bit = 0, then Timer 2 counts up to FFFFh and sets the TF2 (Overflow Flag) bit T2CON[7] upon overflow. This causes the Timer 2 registers to be reloaded with the 16–bit value in T2CAPL and T2CAPH, whose values are preset by software. If EXEN2 bit T2CON[3] = 1, a 16–bit reload can be triggered either by an overflow or by a 1–to–0 transition at input T2EX. This transition also sets the EXF2 bit. If enabled, either TF2 bit or EXF2 bit can generate the Timer 2 interrupt.

In Figure 13 where the DCEN bit = 1; this enables the Timer 2 to count up or down. In this mode, the logic level of T2EX pin controls the direction of count. When a logic ‘1’ is applied at pin T2EX, the Timer 2 will count up. The Timer 2 will overflow at FFFFh and set the TF2 bit flag, which can then generate an interrupt if enabled. This timer overflow also causes the 16–bit value in T2CAPL and T2CAPH to be reloaded into timer registers TL2 and TH2, respectively.

A logic ‘0’ at pin T2EX causes Timer 2 to count down. When counting down, the timer value is compared to the 16–bit value contained in T2CAPH and T2CAPL. When the value is equal, the timer register is loaded with FFFF hex. The underflow also sets the TF2 flag, which can generate an interrupt if enabled.

The External flag EXF2 bit toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution, if needed. the EXF2 bit flag does not generate an interrupt in this mode. As the baud rate generator, timer T2 is incremented by TCLK.

BAUD RATE GENERATOR MODE

By setting the TCLK0 and/or RCLK0 in T2CON, Timer 2 can be chosen as the baud rate generator for the Transmitter and/or Receiver sides of UART-0.

PROGRAMMABLE CLOCK–OUT

A 50% duty cycle clock can be programmed to come out on P1.6. This pin, besides being a regular I/O pin, has two alternate functions. Either it can be programmed to input the External clock for Timer/Counter 2 or to output a 50% duty cycle clock.

To configure the Timer/Counter 2 as a clock generator, bit {C2 or T2*I*} (T2CON[1]) must be cleared and bit T2OE (T2MOD[1]) must be set. Bit TR2 (T2CON[2]) also must be set to start the timer.

The Clock–Out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (TCAP2H, TCAP2L) as shown in this equation:

$$\frac{TCLK}{2\times(65536-TCAP2H,TCAP2L)}$$

In the Clock–Out mode Timer 2 roll–overs will not generate an interrupt. This is similar to when it is used as a baud–rate generator. It is possible to use Timer 2 as a baud–rate generator and a clock generator simultaneously. Note, however, that the baud–rate will be 1/8 of the Clock–Out frequency.

Table 7. Timer 2 Operating Modes

Bits of Special Function Registers				
TR2 T2CON[2]	CP or RL2/ T2CON[0]	RCLK0 or TCLK0 T2CON[5] or T2CON[4]	DCEN T2MOD[0]	MODE
0	X	X	X	Timer off (stopped)
1	0	0	0	16–bit auto–reload, counting up
1	0	0	1	16–bit auto–reload, counting up or down depending on T2EX pin
1	1	0	X	16–bit capture
1	X	1	X	Baud rate generator

Figure SU00612B (XA-G3 Figure 5) goes here.

Figure 9. Timer 0 And 1 Extended Status (TSTAT)

Figure SU00610B (XA-G3 Figure 6) goes here.

Figure 10. Timer 2 Mode Control (T2MOD)

Figure SU00704 (XA-G3 Figure 7) goes here.

Figure 11. Timer 2 in Capture Mode

Figure SU00705 (XA-G3 Figure 8) goes here.

Figure 12. Timer 2 in Auto–Reload Mode (DCEN = 0)

Figure SU00706 (XA-G3 Figure 9) goes here.

Figure 13. Timer 2 Auto Reload Mode (DCEN = 1)

WATCHDOG TIMER

The watchdog timer subsystem protects the system from incorrect code execution by causing a system Reset when the watchdog timer underflows as a result of a failure of software to feed the timer prior to the timer reaching its terminal count. It is important to note that the watchdog timer is running after any type of Reset and must be turned off by user software if the application does not use the watchdog function.

WATCHDOG FUNCTION

The watchdog consists of a programmable prescaler and the main timer. The prescaler derives its clock from the TCLK source that also drives timers 0, 1, and 2. The watchdog timer subsystem consists of a programmable 13–bit prescaler, and an 8–bit main timer. The main timer is clocked (decremented) by a tap taken from one of the top 8–bits of the prescaler as shown in Figure 14. The clock source for the prescaler is the same as TCLK (same as the clock source for the timers). Thus the main counter can be clocked as often as once every 32 TCLKs (seeTable 8). The watchdog generates an underflow signal (and is autoloaded from WDL) when the watchdog is at count 0 and the clock to decrement the watchdog occurs. The watchdog is 8 bits wide and the autoload value can range from 0 to FFh. (The autoload value of 0 is permissible since the prescaler is cleared upon autoload).

This leads to the following user design equations:

$t_{MIN} = t_{OSC} \times 4 \times 32$ (W = 0, N = 4)

$t_{MAX} = t_{OSC} \times 64 \times 4096 \times 256$ (W = 255, N = 64)

$t_D = t_{OSC} \times N \times P \times (W + 1)$

where

t_{OSC} is the oscillator period

N is the selected prescaler tap value

W is the main counter autoload value

P is the prescaler value from Table 8

t_{MIN} is the minimum watchdog time–out value (when the autoload value is 0)

t_{MAX} is the maximum time–out value (when the autoload value is FFh)

t_D is the design time–out value.

The watchdog timer is not directly loadable by the user. Instead, the value to be loaded into the main timer is held in an autoload register. In order to cause the main timer to be loaded with the appropriate value, a special sequence of software action must take place. This operation is referred to as feeding the watchdog timer.

To feed the watchdog, two instructions must be sequentially executed successfully. No intervening SFR accesses are allowed, so interrupts should be disabled before feeding the watchdog. The instructions should move A5h to the WFEED1 register and then 5Ah to the WFEED2 register. If WFEED1 is correctly loaded and WFEED2 is not correctly loaded, then an immediate watchdog Reset will occur. The program sequence to feed the watchdog timer or cause new WDCON settings to take effect is as follows:

```
clr     ea              ; disable global interrupts.
mov.b   wfeed1,#A5h     ; do watchdog feed part 1
```

```
mov.b   wfeed2,#5Ah   ; do watchdog feed part 2
setb    ea            ; re–enable global interrupts.
```

This sequence assumes that the XA interrupt system is enabled and there is a possibility of an interrupt request occurring during the feed sequence. If an interrupt was allowed to be serviced and the service routine contained any SFR access, it would trigger a watchdog Reset. If it is known that no interrupt could occur during the feed sequence, the instructions to disable and re–enable interrupts may be removed.

The software must be written so that a feed operation takes place every t_D seconds from the last feed operation. Some tradeoffs may need to be made. It is not advisable to include feed operations in minor loops or in subroutines unless the feed operation is a specific subroutine.

To turn the watchdog timer completely off, the following code sequence should be used:

```
mov.b   wdcon,#0      ; set WD control register to clear WDRUN.
mov.b   wfeed1,#A5h   ; do watchdog feed part 1
mov.b   wfeed2,#5Ah   ; do watchdog feed part 2
```

This sequence assumes that the watchdog timer is being turned off at the beginning of the User's initialization code and that the XA interrupt system has not yet been enabled. If the watchdog timer is to be turned off at a point when interrupts may be enabled, instructions to disable and re–enable interrupts should be added to this sequence.

WATCHDOG CONTROL REGISTER (WDCON)

The Reset values of the WDCON and WDL registers will be such that the watchdog timer has a timeout period of $4 \times 4096 \times t_{OSC}$ and the watchdog is running. WDCON can be written by software but the changes only take effect after executing a valid watchdog feed sequence.

Table 8. Prescaler Select Values in WDCON

PRE2	PRE1	PRE0	DIVISOR
0	0	0	32
0	0	1	64
0	1	0	128
0	1	1	256
1	0	0	512
1	0	1	1024
1	1	0	2048
1	1	1	4096

WATCHDOG DETAILED OPERATION

When External Reset is applied, the following takes place:

- Watchdog run control bit set to ON (1).
- Autoload register WDL set to 00 (min. count).
- Watchdog time–out flag cleared.
- Prescaler is cleared.
- Prescaler tap set to the highest divide.
- Autoload takes place.

When coming out of a hardware Reset, the software should load the autoload register and then feed the watchdog (i.e., cause an autoload).

If the watchdog is running and happens to underflow at the time the External Reset is applied, the watchdog time–out flag will be cleared.

Figure SU00581A (XA-G3 Figure 10) goes here.

Figure 14. Watchdog Timer in XA–C3

When the watchdog underflows, the following action takes place (see Figure 14):

- Autoload takes place.
- Watchdog time–out flag is set
- Watchdog run bit unchanged.
- Autoload (WDL) register unchanged.
- Prescaler tap unchanged.
- All other device action same as External Reset.

Note that if the watchdog underflows, the Program counter will be loaded from the Reset vector as in the case of an internal Reset. The watchdog time–out flag can be examined to determine if the watchdog has caused the Reset condition. The watchdog time–out flag bit can be cleared by software.

WDCON REGISTER BIT DEFINITIONS

WDCON[7]	PRE2	Prescaler Select 2, Reset to 1
WDCON[6]	PRE1	Prescaler Select 1, Reset to 1
WDCON[5]	PRE0	Prescaler Select 0, Reset to 1
WDCON[2]	WDRUN	Watchdog Run Control bit, Reset to 1
WDCON[1]	WDTOF	Timeout flag

UART

The XA–C3 includes 1 UART port (UART-0) that is compatible with the enhanced UART used on the 8xC51FB. Baud rate selection is somewhat different due to the clocking scheme used for the XA timers.

Four other enhancements have been made to UART operation: First, there are separate interrupt vectors for UART transmit and receive functions. Second, the UART-0 transmitter has been double–buffered, allowing packed transmission of data with no gaps between bytes and less critical interrupt service routine timing. Third, a break detect function has been added to UART-0. This operates independently of the UART and provides a start–of–break status bit that the User program may use to test BR0 (S0STAT[2]). Fourth, an Overrun Error flag has been added to detect missed characters in the received data stream.

The UART baud rate is determined by either a fixed division of the oscillator (in UART-0 Modes 0 and 2) or by the Timer 1 or Timer 2 overflow rate (in UART-0 Modes 1 and 3).

Timer 1 defaults to clock UART-0. Timer 2 can clock UART-0 through T2CON via bits RCLK0 (T2CON[5]) and/or TCLK0 (T2CON[4]).

The serial port receive and transmit registers are both accessed at Special Function Register S0BUF. Writing to S0BUF loads the transmit register, and reading S0BUF accesses the physically separate receive register.

The serial port can operate in 4 modes:

Mode 0: Serial I/O expansion mode. Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). (The baud rate is fixed at 1/16 the oscillator frequency.)

Mode 1: Standard 8–bit UART mode. 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into bit RB8_0 (S0CON[2]). The baud rate is variable via Timer 1 or Timer 2 overflow rates.

Mode 2: Fixed rate 9–bit UART mode. 11 bits are transmitted (through TxD) or received (through RxD): start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit TB8_0 (S0CON[3]) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8_0. On receive, the 9th data bit goes into bit RB8_0, while the stop bit is ignored. The baud rate is programmable to 1/32 of the oscillator frequency.

Mode 3: Standard 9–bit UART mode. 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable via Timer 1 or Timer 2 overflow rates.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. Reception is initiated in Mode 0 by the condition RI_0 (S0CON[0]) = 0 AND REN_0 (S0CON[4]) = 1. Reception is initiated in Mode 1, 2, or 3 by the incoming start bit if REN_0 = 1.

SERIAL PORT CONTROL REGISTER

The serial port control and status register is the Special Function Register S0CON, shown in Figure 16. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive TB8_0 (S0CON[3]) and RB8_0 (S0CON[2]), and the serial port interrupt bits Transmit Interrupt flag TI_0 (S0CON[1]) and Receive Interrupt flag RI_0 (S0CON[0]) .

TRANSMIT INTERRUPT FLAG

In order to allow easy use of the double–buffered UART-0 transmitter feature, the TI_0 flag is set by the UART-0 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 S0BUF is written while the UART-0 transmitter is idle.

Generally, UART transmitters generate one interrupt per byte transmitted. However, UART-0 generates one additional interrupt (as defined by the stated conditions for setting the TI_0 flag). This additional interrupt does not occur if double–buffering is bypassed as explained below. Note: If character–oriented transmission is used (not block–transmission of characters), there could be a second interrupt for each character transmitted, depending on the timing of the writes to S0BUF. For this reason, it is generally better to bypass double–buffering when UART-0 is used in character–oriented mode. This is also true if UART-0 is polled rather than interrupt–driven. 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 way to use transmit double–buffering in an application program is to have the UART interrupt service routine handle a single byte for each interrupt occurrence. Thus, the program will not require any special considerations for double–buffering. Transmitted bytes will then be tightly packed with no intervening gaps. Note: Be aware that higher priority interrupts may cause delays in servicing a transmitter interrupt, and this would defeat double–buffering.

9–BIT MODE

Because the ninth data bit TB8_0 (S0CON[3]) is not double–buffered, you must insure S0CON[3] contains the intended ninth data bit whenever it is transmitted. Alternatively, to synchronize the ninth data bit with the rest of the data stream, you could bypass double-buffering.

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_0 flag (S0CON[1]) upon entry into the ISR, as in standard practice. This clears the interrupt that activated the ISR. Secondly, the TI_0 flag is cleared immediately following each write to S0BUF. 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 S0BUF and the clearing of the TI_0 flag, the interrupt system may have to be temporarily disabled during that sequence by clearing, then setting the EA bit (IEL[7]).

CLOCKING SCHEME AND BAUD RATE GENERATION

CLOCK RATES FOR ALL UART MODES

For UART Modes 0 and 2 the UART clock rate is determined by a fixed division of the oscillator clock. For Modes 1 and 3 the UART clock rate is determined by the overflow rates of either T1 or T2.

BAUD RATES FOR UART MODES 0 AND 2

In UART Mode 0, the baud rate is fixed at $f_{osc}/16$. In Mode 2, however, it is fixed rate at $f_{osc}/32$.

BAUD RATE CALCULATIONS FOR UART MODES 0 AND 2

Baud Rate for UART Mode 0:

Baud_Rate = $f_{osc}/16$

Baud Rate for UART Mode 2:

Baud_Rate = $f_{osc}/32$

BAUD RATES FOR UART MODES 1 AND 3

Table 9 shows the relationship of TCLK to pre-scalar settings for all Timers T0, T1, and T2.

Table 9. TCLK Frequencies

Pre-scalar Value	PT1 ; SCR[3]	PT0 ; SCR[2]	TCLK
4	0	0	$f_{osc}/4$
16	0	1	$f_{osc}/16$
64	1	0	$f_{osc}/64$
-	1	1	reserved

Thus, when Timers T0, T1, and T2 are used to establish the baud rate for Baud Clock, the maximum speed of timers/(Baud Clock) is $f_{osc}/4$ (since the minimum pre-scalar value N is equal to 4). Consequently, the maximum Baud_Rate equals Timer_Rate (timer overflow) divided by 16, i.e., $f_{osc}/64$.

BAUD RATE CALCULATIONS FOR UART MODES 1 AND 3

Baud Rate calculations for UART Mode 1 and 3:

Baud_Rate = Timer_Rate/16

Timer_Rate = $f_{osc}/(N \times (Timer\_Range - Timer\_Reload\_Value))$

where N = the TCLK prescaler value (4, 16, or 64).
and Timer_Range = 256 for Timer 1 in Mode 2.
and Timer_Range = 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 – (f_{OSC}/(Baud_Rate*N*16))

NOTES:

1. The maximum baud rate for UART-0 in Mode 1 or 3 is f_{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.

USING TIMER 2 TO GENERATE BAUD RATES

Timer T2 is a 16–bit up/down counter. As a baud rate generator, Timer 2 is selected as a clock source for UART-0 transmitter and/or receiver by setting TCLK0 and/or RCLK0 in T2CON (see Table 10). As the baud rate generator, T2 is incremented as f_{osc}/N where N = 4, 16, or 64 depending on TCLK as programmed in SCR bits PT1 (SCR[3]) and PTO (SCR[2]). See Table 11).

Note: Pin T2EX [P1.7] acts as an additional External interrupt “INT2/” whenever Timer T2 is used as a baud rate generator.

Table 10. T2CON Settings

T2CON 0x418		T2CON[5]	T2CON[4]	
		RCLK0	TCLK0	

Table 11. Prescaler Select for Timer Clock (TCLK)

SCR 0x440		SCR[3]	SCR[2]	
		PT1	PT0	

An EDITED Figure SU00607B (XA-G3 Figure 11) goes here.

NOTE: Be sure to make the following editing changes:

Remove “S1STAT 425” ; replace “RIn” with “RI_0” ; and

Replace all 13 “n” characters with numeral “0” (e.g., FEn becomes FE0).

Figure 15. Serial Port Extended Status (S0STAT) Register

Note: See also Figure 17 regarding Framing Error flag

UART INTERRUPT SCHEME

There are separate interrupt vectors for UART-0 transmit and receive functions (see Table 12 below).

Table 12. Vector Locations for UART in XA

Vector Address	Interrupt Source	Arbitration
00A0h – 00A3h	UART 0 Receiver	10
00A4h – 00A7h	UART 0 Transmitter	11

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

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 bit RB_8 (S0CON[2]). Then comes a stop bit. UART-0 can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB_8 = 1. This feature is enabled by setting bit SM2_0 (S0CON[5]). 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_0 = 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_0 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2_0 bits set and go on about their business, ignoring the incoming data bytes.

SM2_0 has no effect in UART Mode 0, and in UART Mode 1 can be used to check the validity of the stop bit although this is better done with the Framing Error flag (FE0) {S0STAT[3]}. In a Mode 1 reception, if SM2_0 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

ERROR HANDLING, STATUS FLAGS AND BREAK DETECT

UART-0 has the four error flags as described in Figure 15.

AUTOMATIC ADDRESS RECOGNITION

Automatic Address Recognition is a feature which allows UART-0 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_0 bit. In the 9–bit UART Modes (Mode 2 and Mode 3) the Receive Interrupt flag (RI_0) (S0CON[0]) 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 16.

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, S0ADDR, and the address mask, S0ADEN. S0ADEN is used to define which bits in the S0ADDR are to be used and which bits are "don't care". The S0ADEN mask can be logically ANDed with the S0ADDR 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	S0ADDR	=	1100 0000
	S0ADEN	=	1111 1101
	Given	=	1100 00X0
Slave 1	S0ADDR	=	1100 0000
	S0ADEN	=	1111 1110
	Given	=	1100 000X

In the above example S0ADDR is the same and the S0ADEN 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 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	S0ADDR	=	1100 0000
	S0ADEN	=	1111 1001
	Given	=	1100 0XX0
Slave 1	S0ADDR	=	1110 0000
	S0ADEN	=	1111 1010
	Given	=	1110 0X0X
Slave 2	S0ADDR	=	1110 0000
	S0ADEN	=	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 S0ADDR and S0ADEN. Zeros in this result are treated as don't–cares. In most cases, interpreting the don't–cares as ones, the broadcast address will be FF hexadecimal.

Upon Reset, S0ADDR and S0ADEN are loaded 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 UART drivers which do not make use of this feature.

Figure SU00597C (XA-G3 Figure 12) goes here.

Figure 16. Serial Port Control (S0CON) Register

Figure SU00598 (XA-G3 Figure 13) goes here.

Figure 17. UART Framing Error Detection

Figure SU00613 (XA-G3 Figure 14) goes here.

Figure 18. UART Multiprocessor Communication, Automatic Address Recognition

INPUT/OUTPUT PORT PIN CONFIGURATION

Each I/O port pin can be user–configured to one of four modes: Quasi–Bidirectional (essentially the same as standard 80C51 family I/O ports), Open–Drain, Push–Pull, and Off (High Impedance). After Reset, the default configuration is Quasi–Bidirectional. However, in ROMless operation (established if the EA*I* pin is low at Reset) the port pins that comprise the External DATA bus will default to Push–Pull mode.

I/O port pin configurations are determined by the settings in port configuration SFRs. There are two SFRs for each port, called PnCFGA and PnCFGB, where “n” is the port number. One bit in each of the two SFRs relates to the setting for the corresponding port pin, allowing any combination of the four modes to be mixed on any port pins. For instance, the mode of port 1 pin 3 (P1.3) is controlled by setting bit 3 (P1CFGA[3] and P1CFGB[3]).

Table 13 shows the configuration register settings for the four port pin modes. The DC electrical characteristics of each mode may be found in Table 19.

Table 13. Port Configuration Register Settings

PnCFGB	PnCFGA	Port Pin Mode
0	0	Open–Drain
0	1	Quasi–Bidirectional
1	0	Off (High Impedance)
1	1	Push–Pull

Note: Mode changes may cause glitches to occur during transitions. When modifying both registers, WRITE instructions should be carried out consecutively.

EXTERNAL BUS

The External PROGRAM/DATA bus allows for 8–bit or 16–bit bus width, and address sizes from 12 to 20 bits. The bus width is selected by the BUSW pin at Reset (see Reset Options below), while address size is selected in the MIFCNTL SFR register. If off–chip code is selected (through the use of the EA*I* pin), the initial code fetches will be done with the maximum address size (20 bits).

RESET

Refer to Figure 19 for a recommended Reset circuit example.

Figure SU00702 (XA-G3 Figure 15) goes here.

Figure 19. Recommended Reset Circuit

RST*/* PIN PROPERTIES AND REQUIREMENTS

- Active LOW for improved noise immunity
- Schmitt Trigger with Threshold = 0.7 Vdd
- RST*/* must be low for the longer of 10 µs or 10 clocks
- If EA*/* = 1, all Port pins are set to Quasi-Bidirectional mode
- If EA*/* = 0, all External Bus pins are set to Push-Pull mode
- BUSW mode (8- or 16-bit) is latched

POWER-ON RESET

- Must be > 10 msec to allow the on-chip oscillator to stabilize

OTHER RESET EFFECTS

- Register File is zeroed except [R7] USP/SSP is set to 100h
- Internal DATA RAM is not affected
- All maskable interrupts are disabled
- DS, ES, CS, SSEL, PZ, CM, PT0 and PT1 are zeroed
- The Watchdog Timer is turned ON

RESET TIMING

The EA*/* pin is sampled on the rising edge of the Reset (RST*/*) pulse. The result of this sampling determines whether the device is to begin execution from internal or External PROGRAM memory. Specifically, if EA*/* is pulled high, the XA starts in Single–Chip mode. Alternatively, if EA*/* is driven low, the device enters ROMless mode. Lastly, after RST*/* is released, the {WAIT ; V_{pp} ; EA*/*} pin becomes a bus WAIT signal for External bus transactions.

The BUSW pin is weakly pulled high while RST*/* is asserted, allowing simple biasing of the pin with a resistor to ground to select the alternate bus width. If the BUSW pin is not driven at Reset, the weak pull–up will cause a 1 to be loaded for the bus width, giving a 16–bit External bus. BUSW may be pulled low with a 2.7KΩ or smaller value resistor, giving an 8–bit External bus. Once the User program is running, the bus width setting from the BUSW pin may be overridden by software.

Note: Both EA*/* and BUSW must be held for eight equivalent oscillator clock periods after RST*/* is deasserted (i.e., after RST*/* returns to ONE) to guarantee that the EA*/* and BUSW values are latched correctly.

The relationship of EA*/* and BUSW timing with respect to both RST*/* and ALE signals is shown in Figure 20.

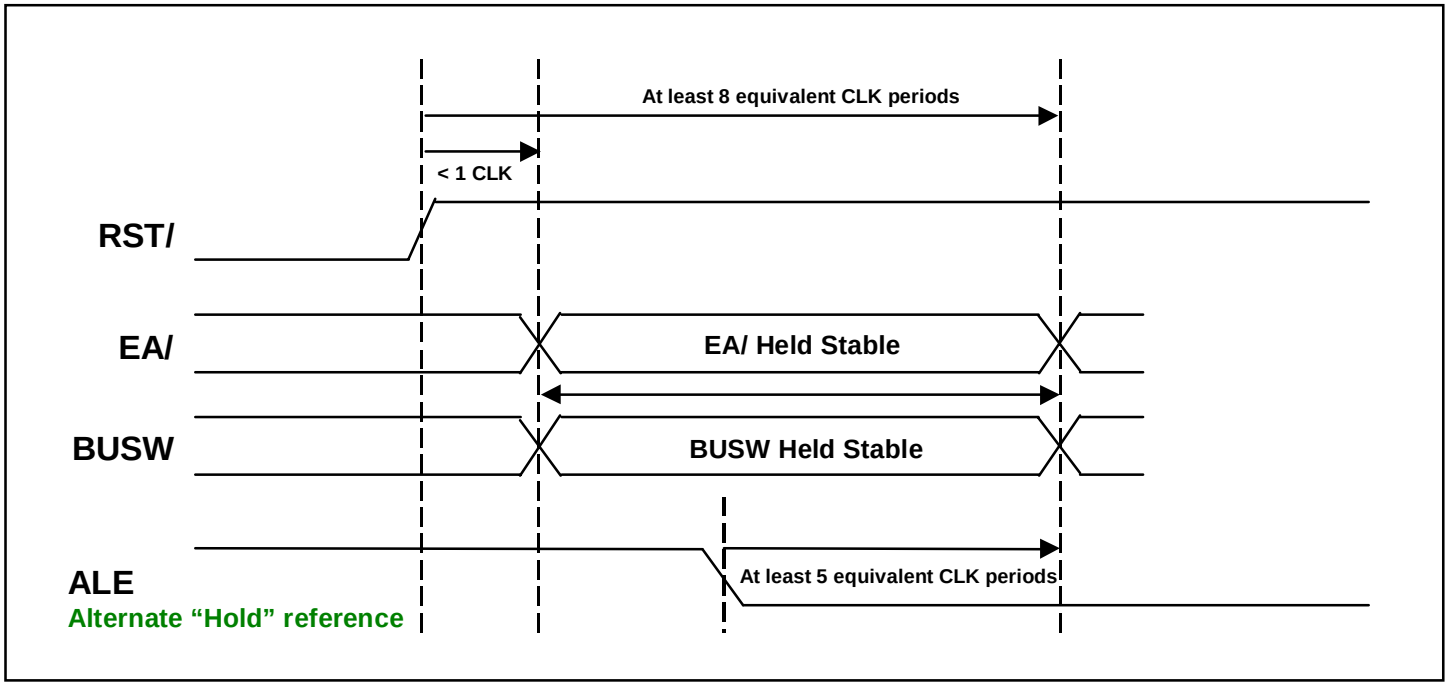

Figure 20. EA/ and BUSW Timing Diagram

POWER REDUCTION MODES

The XA–C3 supports Idle and Power–Down modes of power reduction. The Idle mode leaves some peripherals running to allow them to wake up the processor when an interrupt is generated. The Power–Down mode stops the oscillator in order to minimize power. The processor can be made to exit Power–Down mode via Reset or one of the External interrupt inputs. In order to use an External interrupt to re–activate the XA while in Power–Down mode, the External interrupt must be enabled and be configured to level–sensitive mode. In Power–Down mode, the power supply voltage may be reduced to the RAM keep–alive voltage (2V), retaining the RAM, register, and SFR values at the point where the Power–Down mode was entered.

INTERRUPTS

INTERRUPT TYPES

There are four types of interrupts:

- **Event Interrupts** – service peripherals such as UARTs and timers.
- **Software Interrupts** – demote the priority level of a running Event Interrupt below the lowest Event priority level (i.e., 9), thereby permitting lower priority Event Interrupts to run.
- **Trap Interrupts** –accomplish multi–tasking services, such as RTOS, via non–maskable interrupts.
- **Exception Interrupts** – process non–maskable events, such as Reset, Stack Overflow, and Divide–by–zero.

The XA–C3 supports 42 vectored interrupts. These include 13 maskable Event Interrupts, 7 Software Interrupts, 16 Trap interrupts, and 6 Exception Interrupts. The number of Event Interrupts is related to the number of on-chip peripherals. The XA–C3 supports 13 maskable Event Interrupts. However, Software, Trap, and Exception Interrupts are standardized within the XA core. For core details refer to the *XA User Guide.*

INTERRUPT STRUCTURES

Four tables provide details of the XA-C3 Interrupt structure.

- Table 14 defines the sixteen interrupt priority levels
- Table 15 describes the Exception and Trap Interrupts
- Table 16 explains the Event Interrupts
- Table 17 lists the Software Interrupts

EVENT INTERRUPT HANDLING

If a higher priority Event occurs while a lower priority Event is being serviced, the higher priority Event takes over.

When Events of different priorities occur simultaneously, the highest priority Event is serviced first.

When Events of equal priority occur simultaneously, Arbitration Ranking determines which Event is serviced first. See Table 15 and Table 16.

INTERRUPT PRIORITY DETAILS

Each Event interrupt has 8 priority levels. Event interrupts may be individually masked by bits in SFR Registers IEL and IEH (see Table 5). Event interrupts can also be globally disabled via the EA bit (IEL[7]).

Using 3-bit sub-groups, Interrupt Priority Assignment (IPA) registers (IPA0, IPA1, IPA2, IPA4, IPA5, IPA6, and IPA7) assign 1 of 8 priority levels per Event Interrupt. A zero value assigns interrupt priority 0, in effect disabling an interrupt. The remaining seven priority levels are defined in Table 14.

Table 14. Interrupt Priority Levels

Priority Level	Type of Interrupt
15	**Event Interrupt**
14	**Event Interrupt**
13	**Event Interrupt**
12	**Event Interrupt**
11	**Event Interrupt**
10	**Event Interrupt**
9	**Event Interrupt**
8	
7	**Software Interrupt**
6	**Software Interrupt**
5	**Software Interrupt**
4	**Software Interrupt**
3	**Software Interrupt**
2	**Software Interrupt**
1	**Software Interrupt**
0	**Interrupt Disable**

Note: Details of the priority scheme may be found in the XA User Guide.

Table 15. Exception and Trap Interrupt Vectors

DESCRIPTION	VECTOR ADDRESS	ARBITRATION RANKING
Reset (h/w, watchdog, s/w)	0000 – 0003	0 (High)
Breakpoint (h/w trap 1)	0004 – 0007	1
Trace (h/w trap 2)	0008 – 000B	1
Stack Overflow (h/w trap 3)	000C – 000F	1
Divide by 0 (h/w trap 4)	0010 – 0013	1
User RETI (h/w trap 5)	0014 – 0017	1
TRAP 0– 15 (software)	0040 – 007F	1

Table 16. Event Interrupt Vectors

DESCRIPTION	FLAG BIT	VECTOR ADDRESS	ENABLE BIT	INTERRUPT PRIORITY	ARBITRATION RANKING
External interrupt 0	IE0 ; TCON[1]	0080–0083	EX0 ; IEL[0]	PX0 ; IPA0[2:0]	2
Timer 0 interrupt	TF0 ; TCON[5]	0084–0087	ET0 ; IEL[1]	PT0 ; IPA0[6:4]	3
External interrupt 1	IE1 ; TCON[3]	0088–008B	EX1 ; IEL[2]	PX1 ; IPA1[2:0]	4
Timer 1 interrupt	TF1 ; TCON[7]	008C–008F	ET1 ; IEL[3]	PT1 ; IPA1[6:4]	5
Timer 2 interrupt	TF2 ; T2CON[7] or T2EX [P1.7][a] or EXF2 ; T2CON[6]	0090–0093	ET2 ; IEL[4]	PT2 ; IPA2[2:0]	6
(CAN) Rx buffer full	CANINTFLG[2]	0094–0097	EBUFF ; IEL[5]	PBUFF ; IPA2[6:4]	7
Serial port 0 Rx	RI_0 ; S0CON[0]	00A0–00A3	ERI0 ; IEH[0]	PRI0 ; IPA4[2:0]	10
Serial port 0 Tx	TI_0 ; S0CON[1]	00A4–00A7	ETI0 ; IEH[1]	PTI0 ; IPA4[6:4]	11
SPI Interrupt	SPFG ; SPICS[3]	00AC–00AF	ESPI ; IEH[3]	PSPI ; IPA5[6:4]	13
(CAN) Frame Error	CANINTFLG[4]	00B0–00B3	ECER ; IEH[4]	PCER ; IPA6[2:0]	14
(CAN) Message Error	CANINTFLG[3]	00B4–00B7	EMER ; IEH[5]	PMER ; IPA6[6:4]	15
(CAN) Tx message complete	CANINTFLG[1]	00B8–00BB	EMTI ; IEH[6]	PMTI ; IPA7[2:0]	16
(CAN) Rx message complete	CANINTFLG[0]	00BC–00BF	EMRI ; IEH[7]	PMRI ; IPA7[6:4]	17

[a] When Timer 2 is used as a baud rate generator, pin T2EX [P1.7] acts as an additional External interrupt.

Table 17. Software Interrupt Vectors

DESCRIPTION	REQUEST BIT	VECTOR ADDRESS	ENABLE BIT	INTERRUPT PRIORITY
Software interrupt 7	SWR7 ; SWR[6]	0118–011B	SWE7 ; SWE[6]	fixed at 7 (highest priority)
Software interrupt 6	SWR6 ; SWR[5]	0114–0117	SWE6 ; SWE[5]	fixed at 6
Software interrupt 5	SWR5 ; SWR[4]	0110–0113	SWE5 ; SWE[4]	fixed at 5
Software interrupt 4	SWR4 ; SWR[3]	010C–010F	SWE4 ; SWE[3]	fixed at 4
Software interrupt 3	SWR3 ; SWR[2]	0108–010B	SWE3 ; SWE[2]	fixed at 3
Software interrupt 2	SWR2 ; SWR[1]	0104–0107	SWE2 ; SWE[1]	fixed at 2
Software interrupt 1	SWR1 ; SWR[0]	0100–0103	SWE1 ; SWE[0]	fixed at 1 (lowest priority)

ABSOLUTE MAXIMUM RATINGS

Table 18. Absolute Maximum Ratings

PARAMETER	RATING	UNIT
Operating temperature under bias	–55 to +125	°C
Storage temperature range	–65 to +150	°C
Voltage on EA*/* ; V_{PP} pin to V_{SS}	0 to +13.0	V
Voltage on any other pin to V_{SS}	–0.5 to V_{DD}+0.5V	V
Maximum I_{OL} per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

DC ELECTRICAL CHARACTERISTICS

Table 19. DC Electrical Characteristics

V_{DD} = 4.5V to 5.5V unless otherwise specified;

$T_{ambient}$ = 0 to +70°C for commercial, –40°C to +85°C for industrial, unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
Supply Currents						
I_{DD}	Supply current, operating mode	32 MHz		54	80	mA
I_{ID}	Supply current, Idle mode	32 MHz		25	30	mA
I_{PD}	Power–Down mode current			5	100	µA
I_{PDI}	Power–Down mode current (–40°C to +85°C)				150	µA
Inputs						
V_{RAM}	RAM keep–alive voltage	RAM keep–alive voltage	1.5			V
V_{IL}	Input Low voltage		–0.5		$0.22V_{DD}$	V
V_{IH}	Input High voltage, except XTAL1, RST*/*	At 5.0V	2.2			V
V_{IH1}	Input High voltage to XTAL1, RST*/*	At 5.0V	$0.7V_{DD}$			V
V_{OL}	Output Low voltage all ports, ALE, PSEN*/*[3]	I_{OL} = 3.2mA, V_{DD} = 5.0V			0.5	V
V_{OH1}	Output High voltage all ports, ALE, PSEN*/*[1]	I_{OH} = –100mA,	2.4			V

		V_{DD} = 4.5V				
V_{OH2}	Output High voltage, ports P0–3, ALE, PSEN[2]	I_{OH} = 3.2mA, V_{DD} = 4.5V	2.4			V
C_{IO}	Input/Output pin capacitance				15	pF
I_{IL}	Logical 0 Input current, P0–3[6]	V_{IN} = 0.45V		–25	–75	µA
I_{LI}	Input Leakage current, P0–3[5]	V_{IN} = V_{IL} or V_{IH}			±10	µA
I_{TL}	Logical 1–to–0 Transition current -- all ports[4]	At 5.5V			–650	µA

NOTES:

1. Ports in Quasi–Bidirectional mode with weak pull–up (applies to ALE, PSEN/ only during Reset operations).
2. Ports in Push–Pull mode, both pull–up and pull–down are assumed to be of the same strength
3. In all output modes
4. Port pins source a transition current when used in Quasi–Bidirectional mode and externally driven from 1 to 0. This current is highest when V_{IN} is approximately 2V.
5. Measured with port in high-impedance output mode.
6. Measured with port in Quasi–Bidirectional output mode.
7. Load capacitance for all outputs=80pF.
8. Under steady state (non–transient) conditions, I_{OL} must be externally limited as follows:

   Maximum I_{OL} per port pin: 15mA (*NOTE: This is 85°C specification for VDD = 5V.)

   Maximum I_{OL} per 8–bit port: 26mA

   Maximum total I_{OL} for all outputs: 71mA

   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
9. See Figures 30, 31, 33, and 34 for I_{DD} test conditions, and Figure 32 for I_{CC} vs. Frequency.

AC ELECTRICAL CHARACTERISTICS

Table 20. AC Electrical Characteristics

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

SYMBOL	Figure	PARAMETER	VARIABLE CLOCK		UNIT
			MIN	MAX	
External Clock					
f_C		Oscillator frequency	0	32	MHz
t_C	22	Clock period and CPU timing cycle	$1/f_C$		ns
t_{CHCX}	22	Clock high time	$t_C * 0.5$		ns
t_{CLCX}	22	Clock low time	$t_C * 0.4$		ns
t_{CLCH}	22	Clock rise time		5	ns
t_{CHCL}	22	Clock fall time		5	ns
Address Cycle					
t_{CRAR}	21	Delay from clock rising edge to ALE rising edge	10	46	ns
t_{LHLL}	16	ALE pulse width (programmable)	$(V1 * t_C) – 6$		ns
t_{AVLL}	16	Address valid to ALE de–asserted (set–up)	$(V1 * t_C) – 12$		ns
t_{LLAX}	16	Address hold after ALE de–asserted	$(t_C/2) – 10$		ns
Code Read Cycle					
t_{PLPH}	16	PSEN/ pulse width	$(V2 * t_C) – 10$		ns
t_{LLPL}	16	ALE de–asserted to PSEN/ asserted	$(t_C/2) – 7$		ns
t_{AVIVA}	16	Address valid to instruction valid, ALE cycle (access time)		$(V3 * t_C) – 36$	ns
t_{AVIVB}	17	Address valid to instruction valid, non–ALE cycle (access time)		$(V4 * t_C) – 29$	ns
t_{PLIV}	16	PSEN/ asserted to instruction valid (enable time)		$(V2 * t_C) – 29$	ns
t_{PXIX}	16	Instruction hold after PSEN/ de–asserted	0		ns
t_{PXIZ}	16	Bus 3–State after PSEN/ de–asserted (disable time)		$t_C – 8$	ns
t_{IXUA}	16	Hold time of unlatched part of address after instruction latched	0		ns
Data Read Cycle					
t_{RLRH}	18	RD/ pulse width	$(V7 * t_C) – 10$		ns
t_{LLRL}	18	ALE de–asserted to RD/ asserted	$(t_C/2) – 7$		ns
t_{AVDVA}	18	Address valid to data input valid, ALE cycle (access time)		$(V6 * t_C) – 36$	ns
t_{AVDVB}	19	Address valid to data input valid, non–ALE cycle (access time)		$(V5 * t_C) – 29$	ns
t_{RLDV}	18	RD/ low to valid data in, enable time		$(V7 * t_C) – 29$	ns
t_{RHDX}	18	Data hold time after RD/ de–asserted	0		ns
t_{RHDZ}	18	Bus 3–State after RD/ de–asserted (disable time)		$t_C – 8$	ns
t_{DXUA}	18	Hold time of unlatched part of address after data latched	0		ns
Data Write Cycle					
t_{WLWH}	20	WR/ pulse width	$(V8 * t_C) – 10$		ns
t_{LLWL}	20	ALE falling edge to WR/ asserted	$(V12 * t_C) – 10$		ns
t_{QVWX}	20	Data valid before WR/ asserted (data setup time)	$(V13 * t_C) – 22$		ns
t_{WHQX}	20	Data hold time after WR/ de–asserted (Note 6)	$(V11 * t_C) – 5$		ns
t_{AVWL}	20	Address valid to WR/ asserted (address setup time) (Note 5)	$(V9 * t_C) – 22$		ns
t_{UAWH}	20	Hold time of unlatched part of address after WR/ is de–asserted	$(V11 * t_C) – 7$		ns
WAIT Input					
t_{WTH}	21	WAIT stable after bus strobe (RD/, WR/, or PSEN/) asserted		$(V10 * t_C) – 30$	ns
t_{WTL}	21	WAIT hold after bus strobe (RD/, WR/, or PSEN/) assertion	$(V10 * t_C) – 5$		ns

NOTES:

1. Load capacitance for all outputs = 80pF.
2. Variables V1 through V13 reflect programmable bus timing, which is programmed via the Bus Timing registers (BTRH and BTRL).
   Refer to the XA User Guide for details of the bus timing settings.
   - V1) This variable represents the programmed width of the ALE pulse as determined by the ALEW bit in the BTRL register. V1 = 0.5 if the ALEW bit = 0, and 1.5 if the ALEW bit = 1.
   - V2) This variable represents the programmed width of the PSEN**/** pulse as determined by the CR1 and CR0 bits or the CRA1, CRA0, and ALEW bits in the BTRL register.
     – For a bus cycle with **no** ALE, V2 = 1 if CR1/0 = 00, 2 if CR1/0 = 01, 3 if CR1/0 = 10, and 4 if CR1/0 = 11. Note that during burst mode code fetches, PSEN**/** does not exhibit transitions at the boundaries of bus cycles. V2 still applies for the purpose of determining peripheral timing requirements.
     – For a bus cycle **with** an ALE, V2 = the total bus cycle duration (2 if CRA1/0 = 00, 3 if CRA1/0 = 01, 4 if CRA1/0 = 10, and 5 if CRA1/0 = 11) minus the number of clocks used by ALE (V1 + 0.5).
       Example: If CRA1/0 = 10 and ALEW = 1, the V2 = 4 – (1.5 + 0.5) = 2.
   - V3) This variable represents the programmed length of an entire code read cycle **with** ALE. This time is determined by the CRA1 and CRA0 bits in the BTRL register. V3 = the total bus cycle duration (2 if CRA1/0 = 00, 3 if CRA1/0 = 01, 4 if CRA1/0 = 10, and 5 if CRA1/0 = 11).
   - V4) This variable represents the programmed length of an entire code read cycle with **no** ALE. This time is determined by the CR1 and CR0 bits in the BTRL register. V4 = 1 if CR1/0 = 00, 2 if CR1/0 = 01, 3 if CR1/0 = 10, and 4 if CR1/0 = 11.
   - V5) This variable represents the programmed length of an entire data read cycle with **no** ALE. this time is determined by the DR1 and DR0 bits in the BTRH register. V5 = 1 if DR1/0 = 00, 2 if DR1/0 = 01, 3 if DR1/0 = 10, and 4 if DR1/0 = 11.
   - V6) This variable represents the programmed length of an entire data read cycle **with** ALE. The time is determined by the DRA1 and DRA0 bits in the BTRH register. V6 = 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).
   - V7) This variable represents the programmed width of the RD**/** pulse as determined by the DR1 and DR0 bits or the DRA1, DRA0 in the BTRH register, and the ALEW bit in the BTRL register. Note that during a 16–bit operation on an 8–bit External bus, 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 WRL**/** and/or 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 WRL**/** and/or 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 WRL**/** and/or 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 = 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 setup and hold times. The strobe may be RD**/** (for data read cycles), WRL**/** and/or WRH**/** (for data write cycles), or PSEN**/** (for code read cycles), depending on the type of bus cycle being widened by WAIT. V10 = V2 for WAIT associated with a code read cycle using PSEN**/**. V10 = V8 for a data write cycle using WRL**/** and/or WRH**/**. V10 = V7–1 for a data read cycle using 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 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 WRL**/** and/or 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 WRL**/** and/or 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 DWA1/0 = 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 WRL**/** and/or 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 WRL**/** and/or 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. *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 does not have PSEN*/* edges in every fetch cycle. Thus, if WAIT is used to delay code fetch cycles, a change in the low-order address lines must be detected to locate the beginning of a cycle. This would be A3–A0 for an 8–bit bus, and A3–A1 for a 16–bit bus. Also, a 16–bit data read operation conducted on a 8–bit wide bus similarly does not include two separate 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 WR*/* strobe. This is not usually the case, and in most applications this parameter is not used.
6. Please note that the XA–C3 requires that extended data bus hold time (WM0 = 1) to be used with External bus write cycles.

Figure SU00946 (XA-G3 Figure 16) goes here.

Figure 21. External PROGRAM Memory Read Cycle (ALE Cycle)

Figure SU00707 (XA-G3 Figure 17) goes here.

Figure 22. External PROGRAM Memory Read Cycle (Non–ALE Cycle)

Figure SU00947 (XA-G3 Figure 18) goes here.

Figure 23. External DATA Memory Read Cycle (ALE Cycle)

Figure SU00708A (XA-G3 Figure 19) goes here.

Figure 24. External DATA Memory Read Cycle (Non–ALE Cycle) 8–Bit Bus Only

Figure SU00584C (XA-G3 Figure 20) goes here.

Figure 25. External DATA Memory Write Cycle

Figure SU00709A (XA-G3 Figure 21) goes here.

Figure 26. WAIT Signal Timing

Figure SU00842 (XA-G3 Figure 22) goes here.

Figure 27. External Clock Drive

Figure SU00703A (XA-G3 Figure 23) goes here.

Figure 28. AC Testing Input/Output

Figure SU00011B (XA-G3 Figure 24) goes here.

Figure 29. Float Waveform

Figure SU00591B (XA-G3 Figure 25) goes here.

Figure 30. I_{DD} Test Condition, Active Mode

Note: All other pins are disconnected

Figure SU00590B (XA-G3 Figure 26) goes here.

Figure 31. I_{DD} Test Condition, Idle Mode

Note: All other pins are disconnected

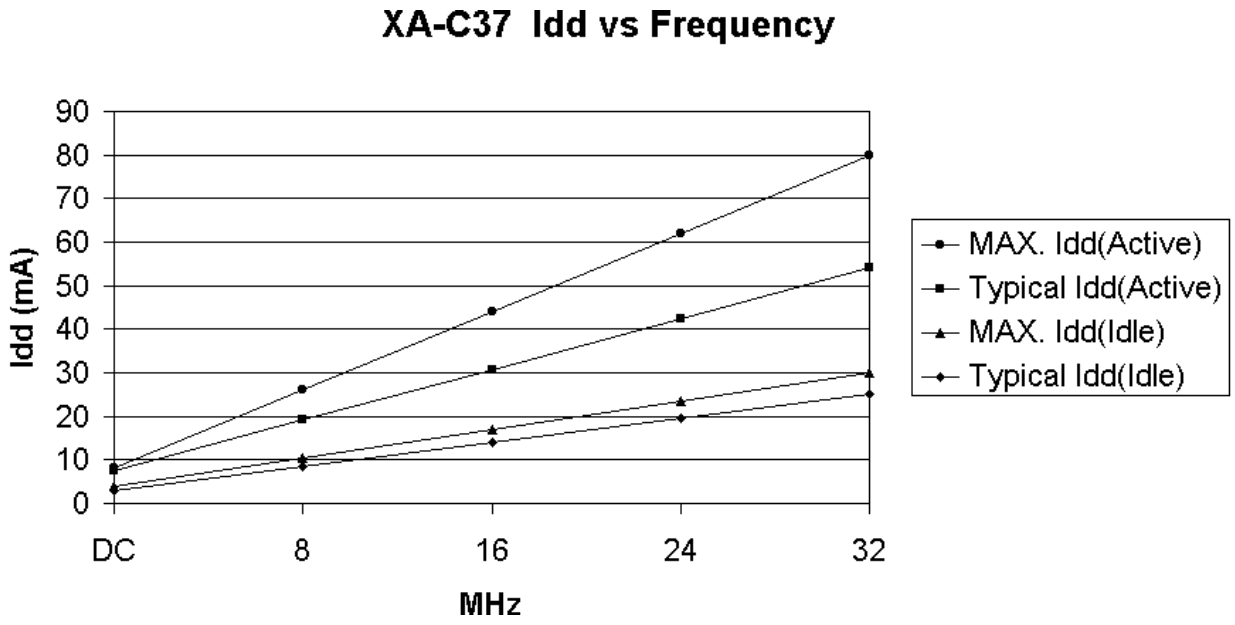

Figure 32. I_{DD} vs. Frequency at V_{DD} = 5.0V

Figure SU00608A (XA-G3 Figure 29) goes here.

Figure 33. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes

Note: $t_{CLCH} = t_{CHCL}$ = 5 ns

Figure SU00585A (XA-G3 Figure 30)goes here.

Figure 34. I_{DD} Test Condition, Power–Down Mode

Note: All other pins are disconnected. V_{DD}=2V to 5.5V

EPROM CHARACTERISTICS

The XA–C37 is programmed by using a modified Improved Quick–Pulse Programming™ algorithm. This algorithm is essentially the same as that used by the later 80C51 family EPROM parts. However, different pins are used for many programming functions.

Detailed EPROM programming information may be obtained from the internet at www.philipsmcu.com/ftp.html.

The XA–C3 contains three signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an XA–Gx manufactured by Philips.

SECURITY BITS

With none of the security bits programmed the code in the PROGRAM memory can be verified. When only security bit 1 (see Table 21) is programmed, MOVC instructions executed from External PROGRAM memory are disabled from fetching code bytes from the internal memory. All further programming of the EPROM is disabled. When, in addition to the above, security bits 1 and 2 are programmed, verify mode is disabled. When all three security bits are programmed, all of the conditions above apply and all External PROGRAM memory execution is disabled. (See Table 21).

Table 21. PROGRAM Security Bits

PROGRAM LOCK BITS				
	SB1	SB2	SB3	PROTECTION DESCRIPTION
1	U	U	U	No PROGRAM Security features enabled.
2	P	U	U	MOVC instructions executed from External PROGRAM memory are disabled from fetching code bytes from internal memory and further programming of the EPROM is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, External execution is disabled. Internal DATA RAM is not accessible.

NOTES:
1. P – programmed. U – unprogrammed.
2. Any other combination of the security bits is not defined.

ROM CODE SUBMISSION

When submitting ROM code for the XA–C33, the following must be specified:
1. 32k bytes user ROM DATA.
2. ROM security bits.

Table 22. ROM Code Submission

ADDRESS	CONTENT	BIT(S)	COMMENT
0000h to 7FFFh	DATA	7:0	User ROM DATA
8020h	SECURITY BIT	0	ROM Security Bit 1 0 = enable security 1 = disable security
8020h	SECURITY BIT	1	ROM Security Bit 2 0 = enable security 1 = disable security
8020h	SECURITY BIT	3	ROM Security Bit 3 0 = enable security 1 = disable security