DATA SHEET

XA-G49

XA 16-bit microcontroller family

64K FLASH/2K RAM, watchdog, 2 UARTs

Preliminary specification
Supersedes data of 2000 Apr 03
IC28 Data Handbook

2000 Jun 21

Philips
Semiconductors

PHILIPS

XA 16-bit microcontroller family 64K Flash/2K RAM, watchdog, 2 UARTs

XA-G49

GENERAL DESCRIPTION

The XA-G49 is a member of Philips' 80C51 XA (eXtended Architecture) family of high performance 16-bit single-chip microcontrollers.

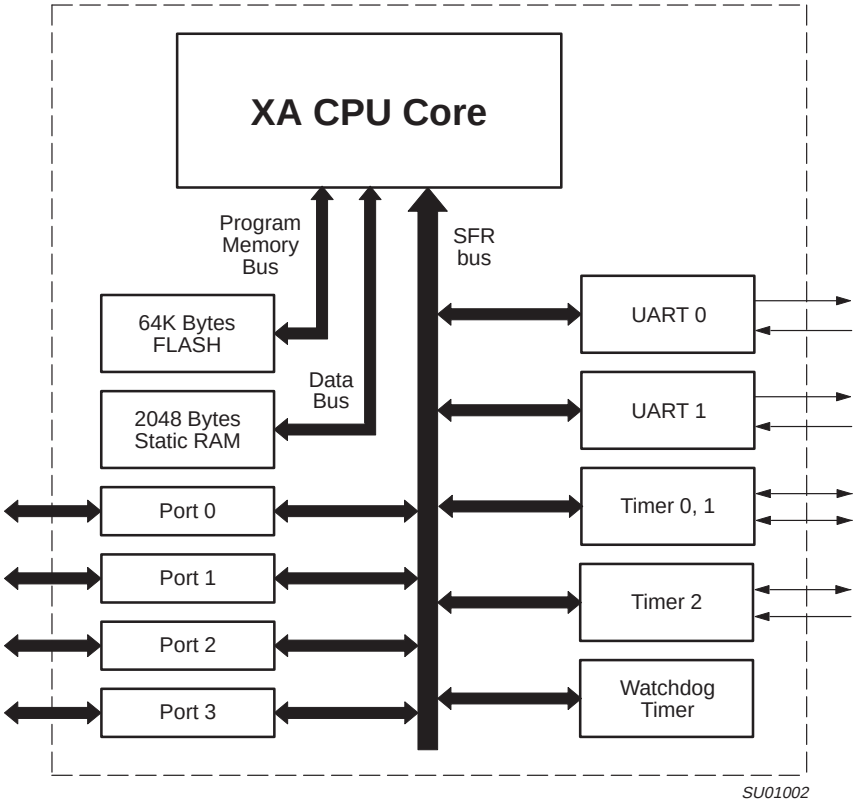
The XA-G49 contains 64 kbytes of Flash program memory, and provides three general purpose timers/counters, a watchdog timer, dual UARTs, and four general purpose I/O ports with programmable output configurations.

A default serial loader program in the Boot ROM allows In-System Programming (ISP) of the Flash memory without the need for a loader in the Flash code. User programs may erase and reprogram the Flash memory at will through the use of standard routines contained in the Boot ROM (In-Application Programming).

FEATURES

- 4.5 V to 5.5 V operation. For low voltage operation, consult factory.
- 64 kbytes of on-chip Flash program memory with In-System Programming capability
- Five Flash blocks = two 8 kbyte blocks and three 16 kbyte blocks
- Nearly identical to XA-G3, except for double the program and RAM memories
- Single supply voltage In-System Programming (ISP) of the Flash memory ($V_{PP} = V_{DD}$, or V_{PP} = 12 V if desired)
- Boot ROM contains low level Flash programming routines for In-Application Programming and a default serial loader using the UART
- 2048 bytes of on-chip data RAM
- Supports off-chip program and data addressing up to 1 megabyte (20 address lines)
- Three standard counter/timers with enhanced features (same as XA-G3 T0, T1, and T2). All timers have a toggle output capability
- Watchdog timer
- Two enhanced UARTs with independent baud rates
- Seven software interrupts
- Four 8-bit I/O ports, with 4 programmable output configurations for each pin
- 30 MHz operating frequency at 5 V
- Power saving operating modes: Idle and Power-Down. Wake-Up from power-down via an external interrupt is supported.
- 44-pin PLCC and 44-pin LQFP packages

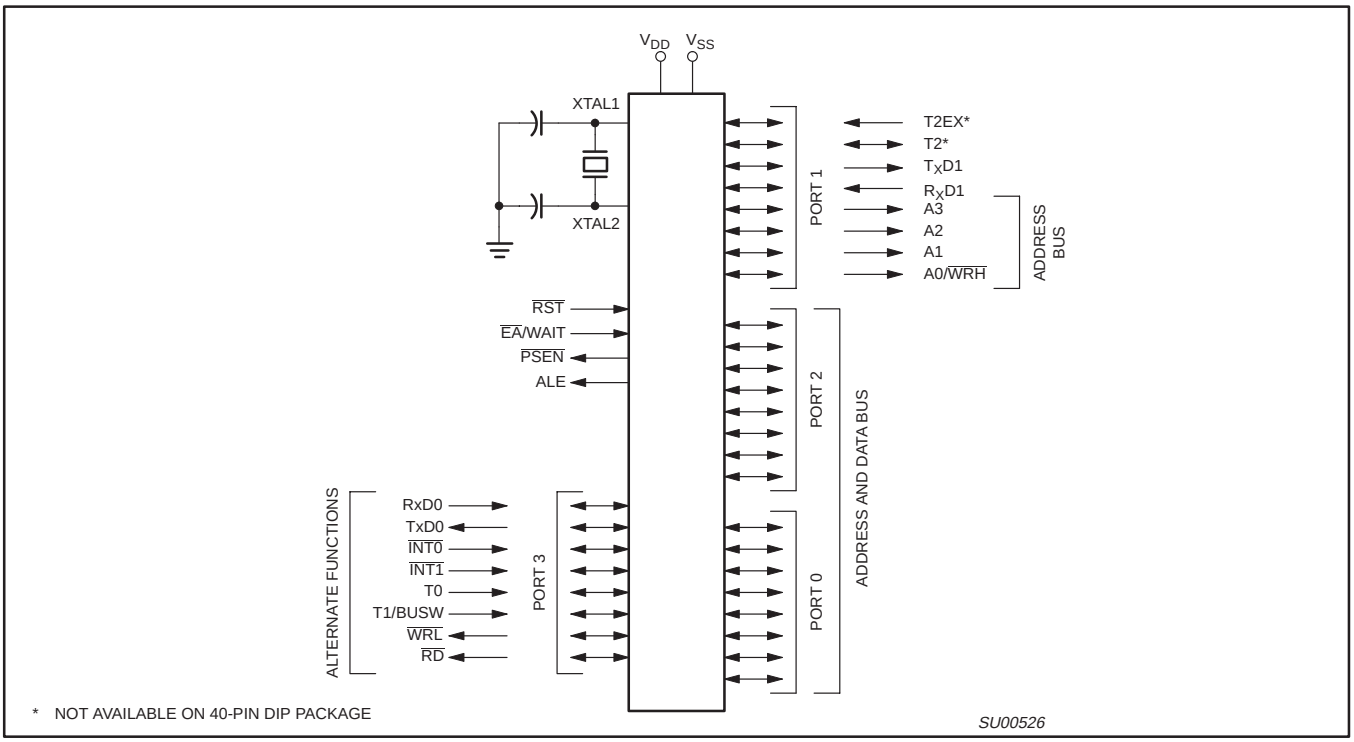
BLOCK DIAGRAM



ORDERING INFORMATION

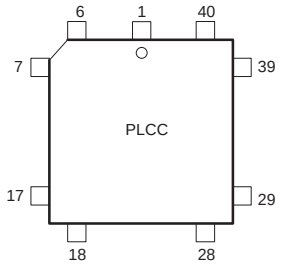
FLASH	TEMPERATURE RANGE (°C) AND PACKAGE	FREQ. (MHz)	DRAWING NUMBER
PXAG49KBA	0 to +70 44-pin Plastic Leaded Chip Carrier	30	SOT187-2
PXAG49KFA	–40 to +85 44-pin Plastic Leaded Chip Carrier	30	SOT187-2
PXAG49KB BD	0 to 70 44-pin Plastic Low Profile Quad Flat Package	30	SOT389-1

LOGIC SYMBOL



PIN CONFIGURATIONS

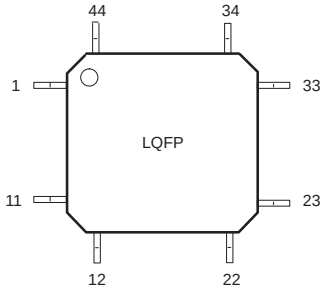
44-Pin PLCC Package



Pin	Function	Pin	Function
1	V_{SS}	23	V_{DD}
2	P1.0/A0/$\overline{WRH}$	24	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/RxD1	28	P2.4/A16D12
7	P1.5/TxD1	29	P2.5/A17D13
8	P1.6/T2	30	P2.6/A18D14
9	P1.7/T2EX	31	P2.7/A19D15
10	$\overline{RST}$	32	$\overline{PSEN}$
11	P3.0/RxD0	33	ALE
12	NC	34	NC
13	P3.1/TxD0	35	$\overline{EA}$/V_{PP}/WAIT
14	P3.2/$\overline{INT0}$	36	P0.7/A11D7
15	P3.3/$\overline{INT1}$	37	P0.6/A10D6
16	P3.4/T0	38	P0.5/A9D5
17	P3.5/T1/BUSW	39	P0.4/A8D4
18	P3.6/$\overline{WRL}$	40	P0.3/A7D3
19	P3.7/$\overline{RD}$	41	P0.2/A6D2
20	XTAL2	42	P0.1/A5D1
21	XTAL1	43	P0.0/A4D0
22	V_{SS}	44	V_{DD}

SU01035

44-Pin LQFP Package



Pin	Function	Pin	Function
1	P1.5/TxD1	23	P2.5/A17D13
2	P1.6/T2	24	P2.6/A18D14
3	P1.7/T2EX	25	P2.7/A19D15
4	$\overline{RST}$	26	$\overline{PSEN}$
5	P3.0/RxD0	27	ALE
6	NC	28	NC
7	P3.1/TxD0	29	$\overline{EA}$/V_{PP}/WAIT
8	P3.2/$\overline{INT0}$	30	P0.7/A11D7
9	P3.3/$\overline{INT1}$	31	P0.6/A10D6
10	P3.4/T0	32	P0.5/A9D5
11	P3.5/T1/BUSW	33	P0.4/A8D4
12	P3.6/$\overline{WRL}$	34	P0.3/A7D3
13	P3.7/$\overline{RD}$	35	P0.2/A6D2
14	XTAL2	36	P0.1/A5D1
15	XTAL1	37	P0.0/A4D0
16	V_{SS}	38	V_{DD}
17	V_{DD}	39	V_{SS}
18	P2.0/A12D8	40	P1.0/A0/$\overline{WRH}$
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/A16/D12	44	P1.4/RxD1

SU01036

PIN DESCRIPTIONS

MNEMONIC	PIN. NO. LCC	PIN. NO. LQFP	TYPE	NAME AND FUNCTION
V_{SS}	1, 22	16	I	**Ground:** 0 V reference.
V_{DD}	23, 44	17	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 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.
P1.0 – P1.7	2–9	40–44, 1–3	I/O	**Port 1:** Port 1 is an 8-bit I/O port with a user-configurable output type. Port 1 latches have 1s written to them and are configured in the quasi-bidirectional mode during reset. The operation of port 1 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. Port 1 also provides special functions as described below.
	2	40	O	**A0/$\overline{\text{WRH}}$:** Address bit 0 of the external address bus when the external data bus is configured for an 8 bit width. When the external data bus is configured for a 16 bit width, this pin becomes the high byte write strobe.
	3	41	O	**A1:** Address bit 1 of the external address bus.
	4	42	O	**A2:** Address bit 2 of the external address bus.
	5	43	O	**A3:** Address bit 3 of the external address bus.
	6	44	I	**RxD1 (P1.4):** Receiver input for serial port 1.
	7	1	O	**TxD1 (P1.5):** Transmitter output for serial port 1.
	8	2	I/O	**T2 (P1.6):** Timer/counter 2 external count input/clockout.
	9	3	I	**T2EX (P1.7):** 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 a user-configurable output type. Port 2 latches have 1s written to them and are configured in the quasi-bidirectional mode during reset. The operation of port 2 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 in 16-bit mode, Port 2 becomes the multiplexed high data/instruction byte and address lines 12 through 19. 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, 7–13	I/O	**Port 3:** Port 3 is an 8-bit I/O port with a user configurable output type. Port 3 latches have 1s written to them and are configured in the quasi-bidirectional mode during reset. the operation of port 3 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. Port 3 also provides various special functions as described below.
	11	5	I	**RxD0 (P3.0):** Receiver input for serial port 0.
	13	7	O	**TxD0 (P3.1):** Transmitter output for serial port 0.
	14	8	I	**$\overline{\text{INT0}}$ (P3.2):** External interrupt 0 input.
	15	9	I	**$\overline{\text{INT1}}$ (P3.3):** External interrupt 1 input.
	16	10	I/O	**T0 (P3.4):** Timer 0 external input, or timer 0 overflow output.
	17	11	I/O	**T1/BUSW (P3.5):** Timer 1 external input, or timer 1 overflow output. The value on this pin is latched as the external reset input is released and defines the default external data bus width (BUSW). 0 = 8-bit bus and 1 = 16-bit bus.
	18	12	O	**$\overline{\text{WRL}}$ (P3.6):** External data memory low byte write strobe.
	19	13	O	**$\overline{\text{RD}}$ (P3.7):** External data memory read strobe.
$\overline{\text{RST}}$	10	4	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. Refer to the section on Reset for details.
ALE	33	27	I/O	**Address Latch Enable:** 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.

MNEMONIC	PIN. NO.		TYPE	NAME AND FUNCTION
	LCC	LQFP		
$\overline{PSEN}$	32	26	O	**Program Store Enable:** The read strobe for external program memory. When the microcontroller accesses external program memory, $\overline{PSEN}$ is driven low in order to enable memory devices. $\overline{PSEN}$ is only active when external code accesses are performed.
$\overline{EA}$/WAIT/ V_{PP}	35	29	I	**External Access/Wait/Programming Supply Voltage:** The $\overline{EA}$ input determines whether the internal program memory of the microcontroller is used for code execution. The value on the $\overline{EA}$ 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 any external bus access, that cycle will be extended until Wait is released. During EPROM programming, this pin is also the programming supply voltage input.
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

NAME	DESCRIPTION	SFR ADDRESS	BIT FUNCTIONS AND ADDRESSES MSB							LSB	RESET VALUE
AUXR	Auxiliary function register	44C	ENBOOT	FMIDLE	PWR_VLD	—	—	—	—	—	
BCR	Bus configuration register	46A	—	—	—	WAITD	BUSD	BC2	BC1	BC0	Note 1
BTRH	Bus timing register high byte	469	DW1	DW0	DWA1	DWA0	DR1	DR0	DRA1	DRA0	FF
BTRL	Bus timing register low byte	468	WM1	WM0	ALEW	—	CR1	CR0	CRA1	CRA0	EF
CS	Code segment	443									00
DS	Data segment	441									00
ES	Extra segment	442									00
			33F	33E	33D	33C	33B	33A	339	338	
IEH*	Interrupt enable high byte	427	—	—	—	—	ETI1	ERI1	ETI0	ERI0	00
			337	336	335	334	333	332	331	330	
IEL*	Interrupt enable low byte	426	EA	—	—	ET2	ET1	EX1	ET0	EX0	00
IPA0	Interrupt priority 0	4A0	—	PT0			—	PX0			00
IPA1	Interrupt priority 1	4A1	—	PT1			—	PX1			00
IPA2	Interrupt priority 2	4A2	—	—			—	PT2			00
IPA4	Interrupt priority 4	4A4	—	PTI0			—	PRI0			00
IPA5	Interrupt priority 5	4A5	—	PTI1			—	PRI1			00
			387	386	385	384	383	382	381	380	
P0*	Port 0	430	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FF
			38F	38E	38D	38C	38B	38A	389	388	
P1*	Port 1	431	T2EX	T2	TxD1	RxD1	A3	A2	A1	WRH	FF
			397	396	395	394	393	392	391	390	
P2*	Port 2	432	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FF
			39F	39E	39D	39C	39B	39A	399	398	
P3*	Port 3	433	RD	WR	T1	T0	INT1	INT0	TxD0	RxD0	FF

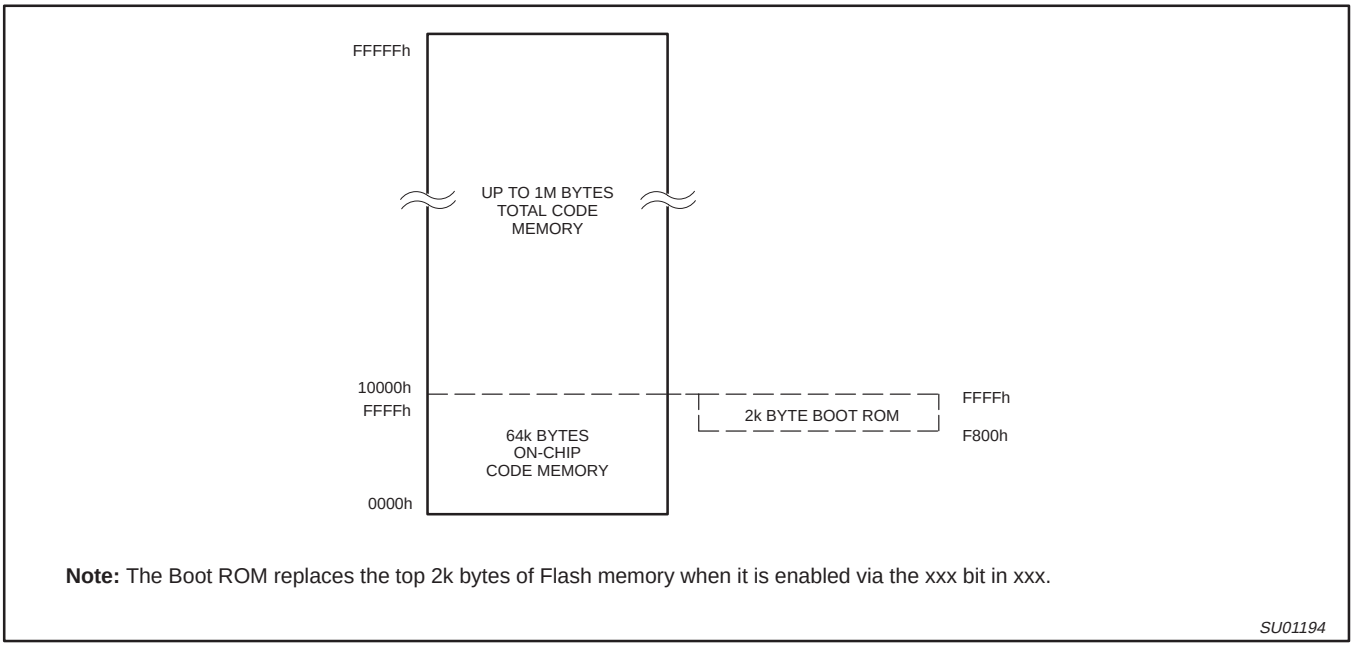
NAME	DESCRIPTION	SFR ADDRESS	BIT FUNCTIONS AND ADDRESSES MSB							LSB	RESET VALUE
P0CFGA	Port 0 configuration A	470									Note 5
P1CFGA	Port 1 configuration A	471									Note 5
P2CFGA	Port 2 configuration A	472									Note 5
P3CFGA	Port 3 configuration A	473									Note 5
P0CFGB	Port 0 configuration B	4F0									Note 5
P1CFGB	Port 1 configuration B	4F1									Note 5
P2CFGB	Port 2 configuration B	4F2									Note 5
P3CFGB	Port 3 configuration B	4F3									Note 5
			227	226	225	224	223	222	221	220	
PCON*	Power control register	404	—	—	—	—	—	—	PD	IDL	00
			20F	20E	20D	20C	20B	20A	209	208	
PSWH*	Program status word (high byte)	401	SM	TM	RS1	RS0	IM3	IM2	IM1	IM0	Note 2
			207	206	205	204	203	202	201	200	
PSWL*	Program status word (low byte)	400	C	AC	—	—	—	V	N	Z	Note 2
			217	216	215	214	213	212	211	210	
PSW51*	80C51 compatible PSW	402	C	AC	F0	RS1	RS0	V	F1	P	Note 3
RTH0	Timer 0 extended reload, high byte	455									00
RTH1	Timer 1 extended reload, high byte	457									00
RTL0	Timer 0 extended reload, low byte	454									00
RTL1	Timer 1 extended reload, low byte	456									00
			307	306	305	304	303	302	301	300	
S0CON*	Serial port 0 control register	420	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00
			30F	30E	30D	30C	30B	30A	309	308	
S0STAT*	Serial port 0 extended status	421	—	—	—	—	FE0	BR0	OE0	STINT0	00
S0BUF	Serial port 0 buffer register	460									x
S0ADDR	Serial port 0 address register	461									00
S0ADEN	Serial port 0 address enable register	462									00
			327	326	325	324	323	322	321	320	
S1CON*	Serial port 1 control register	424	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	00
			32F	32E	32D	32C	32B	32A	329	328	
S1STAT*	Serial port 1 extended status	425	—	—	—	—	FE1	BR1	OE1	STINT1	00
S1BUF	Serial port 1 buffer register	464									x
S1ADDR	Serial port 1 address register	465									00
S1ADEN	Serial port 1 address enable register	466									00
SCR	System configuration register	440	—	—	—	—	PT1	PT0	CM	PZ	00
			21F	21E	21D	21C	21B	21A	219	218	
SSEL*	Segment selection register	403	ESWEN	R6SEG	R5SEG	R4SEG	R3SEG	R2SEG	R1SEG	R0SEG	00
SWE	Software Interrupt Enable	47A	—	SWE7	SWE6	SWE5	SWE4	SWE3	SWE2	SWE1	00

NAME	DESCRIPTION	SFR ADDRESS	BIT FUNCTIONS AND ADDRESSES MSB							LSB	RESET VALUE
			357	356	355	354	353	352	351	350	
SWR*	Software Interrupt Request	42A	—	SWR7	SWR6	SWR5	SWR4	SWR3	SWR2	SWR1	00
			2C7	2C6	2C5	2C4	2C3	2C2	2C1	2C0	
T2CON*	Timer 2 control register	418	TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2	00
			2CF	2CE	2CD	2CC	2CB	2CA	2C9	2C8	
T2MOD*	Timer 2 mode control	419	—	—	RCLK1	TCLK1	—	—	T2OE	DCEN	00
TH2	Timer 2 high byte	459									00
TL2	Timer 2 low byte	458									00
T2CAPH	Timer 2 capture register, high byte	45B									00
T2CAPL	Timer 2 capture register, low byte	45A									00
			287	286	285	284	283	282	281	280	
TCON*	Timer 0 and 1 control register	410	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00
TH0	Timer 0 high byte	451									00
TH1	Timer 1 high byte	453									00
TL0	Timer 0 low byte	450									00
TL1	Timer 1 low byte	452									00
TMOD	Timer 0 and 1 mode control	45C	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00
			28F	28E	28D	28C	28B	28A	289	288	
TSTAT*	Timer 0 and 1 extended status	411	—	—	—	—	—	T1OE	—	T0OE	00
			2FF	2FE	2FD	2FC	2FB	2FA	2F9	2F8	
WDCON*	Watchdog control register	41F	PRE2	PRE1	PRE0	—	—	WDRUN	WDTOF	—	Note 6
WDL	Watchdog timer reload	45F									00
WFEED1	Watchdog feed 1	45D									x
WFEED2	Watchdog feed 2	45E									x

NOTES:

* SFRs are bit addressable.

1. At reset, the BCR register is loaded with the binary value 0000 0a11, where “a” is the value on the BUSW pin. This defaults the address bus size to 20 bits since the XA-G49 has only 20 address lines.
2. SFR is loaded from the reset vector.
3. All bits except F1, F0, and P are loaded from the reset vector. Those bits are all 0.
4. Unimplemented bits in SFRs are X (unknown) at all times. Ones should not be written to these bits since they may be used for other purposes in future XA derivatives. The reset value shown for these bits is 0.
5. 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 FF and PnCFGB registers will contain 00. 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.
6. The WDCON reset value is E6 for a Watchdog reset, E4 for all other reset causes.
7. The XA-G49 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.
8. The AUXR reset value is typically 00h. If the Boot Loader is activated at reset because the Flash status byte is non-zero or because the Boot Vector has been forced (by $\overline{PSEN}$ = 0, ALE = 1, $\overline{EA}$ = 1 at reset), the AUXR reset value will be 1x00 0000b. Bit 6 will be a 1 if the on-chip V_{PP} generator is running and ready, otherwise it will be a 0.



Note: The Boot ROM replaces the top 2k bytes of Flash memory when it is enabled via the xxx bit in xxx.

Figure 1. XA-G49 Program Memory Map

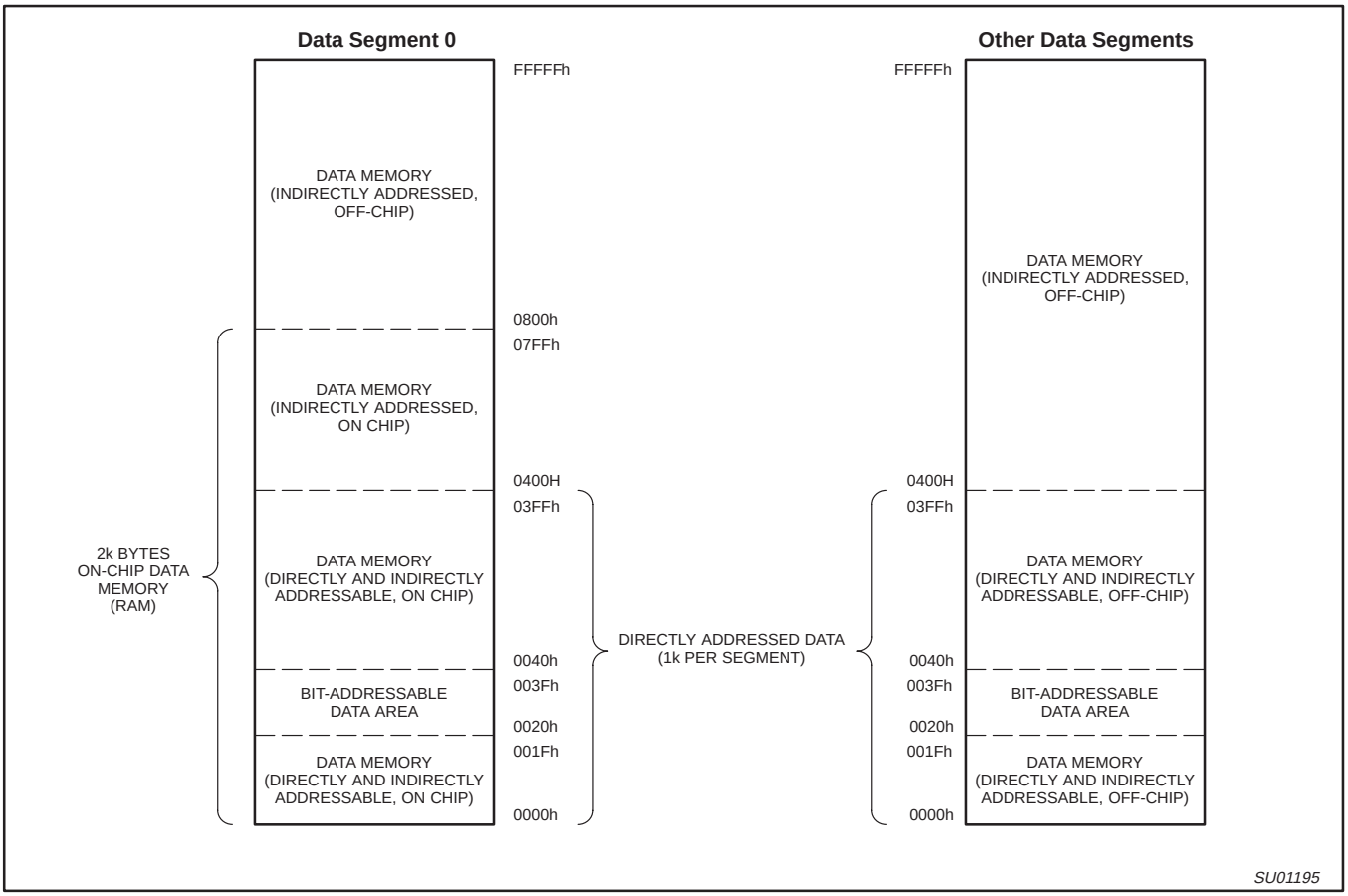


Figure 2. XA-G49 Data Memory Map

FLASH EPROM MEMORY

GENERAL DESCRIPTION

The XA-G49 Flash memory augments EPROM functionality with in-circuit electrical erasure and programming. The Flash can be read and written as bytes. The Chip Erase operation will erase the entire program memory. The Block Erase function can erase any single Flash block. In-circuit programming and standard parallel programming are both available. On-chip erase and write timing generation contribute to a user friendly programming interface.

The XA-G49 Flash reliably stores memory contents even after 10,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling. For In-System Programming, the XA-G49 can use a single +5 V power supply. Faster In-System Programming may be obtained, if required, through the use of a +12 V V_{PP} supply. Parallel programming (using separate programming hardware) uses a +12 V V_{PP} supply

FEATURES

- Flash EPROM internal program memory with Single Voltage Programming and Block Erase capability.
- Internal 2 kbyte fixed boot ROM, containing low-level programming routines and a default loader. The Boot ROM can be turned off to provide access to the full 64 kbyte Flash memory.
- Boot vector allows user provided Flash loader code to reside anywhere in the Flash memory space. This configuration provides flexibility to the user.
- Default loader in Boot ROM allows programming via the serial port without the need for a user provided loader.
- Up to 1 Mbyte external program memory if the internal program memory is disabled ($\overline{EA}$ = 0).
- Programming and erase voltage: $V_{PP} = V_{DD}$ (single 5 V ±5% chip power supply), or 12 V ±5% for In System Programming. Using 12 V V_{PP} for ISP may improve programming and erase time.
- Read/Programming/Erase:
  - Byte-wise read (60 ns access time at 4.5 V).
  - Byte Programming (1–2 minutes for 64 K flash, depending on clock frequency).
- In-circuit programming via user selected method, typically RS232 or parallel I/O port interface.
- Programmable security for the code in the Flash.
- 10,000 minimum erase/program cycles each byte over operating temperature range.
- 10 year minimum data retention.

CAPABILITIES OF THE PHILIPS XA-G49 FLASH-BASED MICROCONTROLLERS

Flash organization

The XA-G49 contains 64 kbytes of Flash program memory. This memory is organized as 5 separate blocks. The first two blocks are 8 kbytes in size, filling the program memory space from address 0 through 3FFF hex. The final three blocks are 16 kbytes in size and occupy addresses from 4000 through FFFF hex.

Figure 3 depicts the Flash memory configuration.

Flash Programming and Erasure

The XA-G49 Flash microcontroller supports a number of programming possibilities for the on-chip Flash memory. The Flash memory may be programmed in a parallel fashion on standard programming equipment in a manner similar to an EPROM microcontroller. The XA-G49 microcontroller is able to program its own Flash memory while the application code is running. Also, a default loader built into a Boot ROM allows programming blank devices serially through the UART.

Using any of these types of programming, any of the individual blocks may be erased separately, or the entire chip may be erased. Programming of the Flash memory is accomplished one byte at a time.

Boot ROM

When the microcontroller programs its own Flash memory, all of the low level details are handled by code that is permanently contained in a 2 kbyte “Boot ROM” that is separate from the Flash memory. A user program simply calls the entry point with the appropriate parameters to accomplish the desired operation. Boot ROM operations include things like: erase block, program byte, verify byte, program security lock bit, etc. The Boot ROM overlays the program memory space at the top of the address space from F800 to FFFF hex, when it is enabled by setting the ENBOOT bit at AUXR1.7.. The Boot ROM may be turned off so that the upper 2 kbytes of Flash program memory are accessible for execution.

ENBOOT and PWR_VLD

Setting the ENBOOT bit in the AUXR register enables the Boot ROM and activates the on-chip V_{PP} generator if V_{PP} is connected to V_{DD} rather than 12 V externally. The PWR_VLD flag indicates that V_{PP} is available for programming and erase operations. This flag should be checked prior to calling the Boot ROM for programming and erase services. When ENBOOT is set, it typically takes 5 microseconds for the internal programming voltage to be ready.

The ENBOOT bit will automatically be set if the status byte is non-zero during reset, or when $\overline{PSEN}$ is low, ALE is high, and $\overline{EA}$ is high at the falling edge of reset. Otherwise, ENBOOT will be cleared during reset.

When programming functions are not needed, ENBOOT may be cleared. This enables access to the 2 kbytes of Flash code memory that is overlaid by the Boot ROM, allowing a full 64 kbytes of Flash code memory.

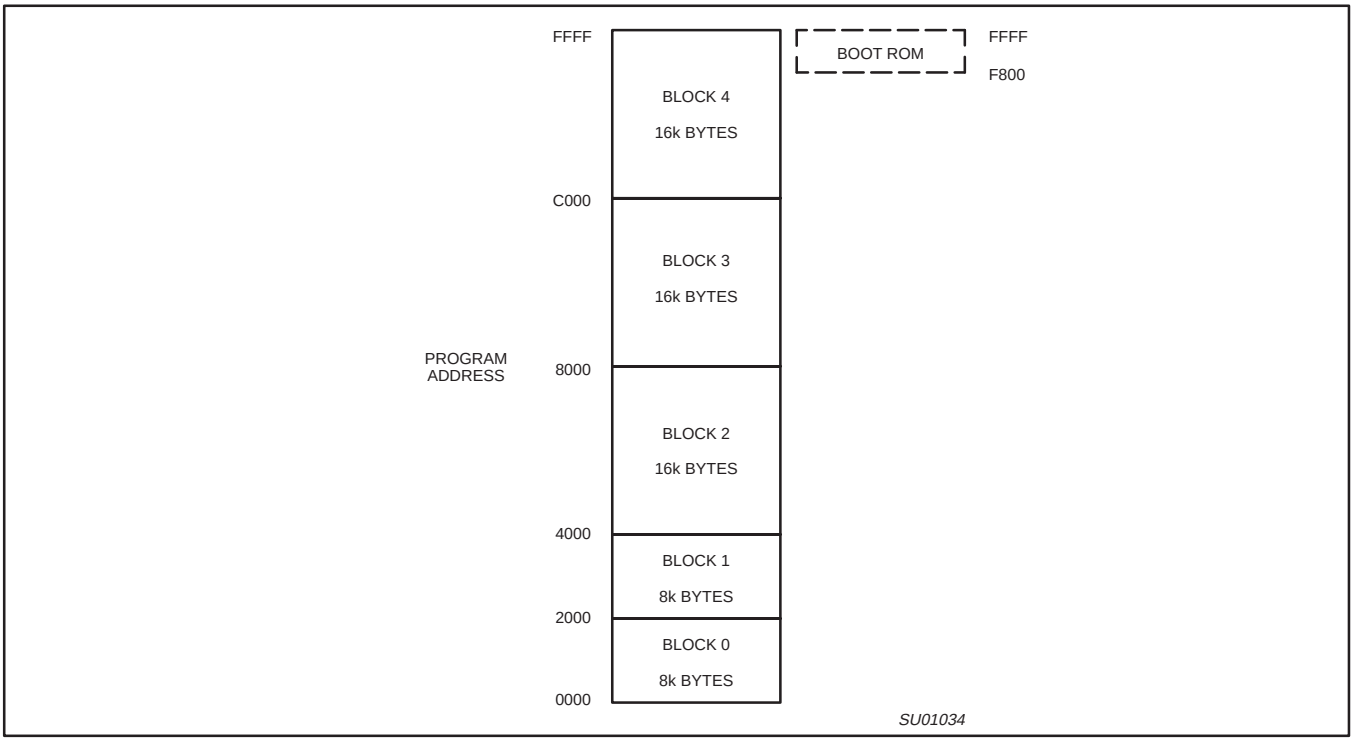


Figure 3. Flash Memory Configuration

FMIDLE

The FMIDLE bit in the AUXR register allows saving additional power by turning off the Flash memory when the CPU is in the Idle mode. This must be done just prior to initiating the Idle mode, as shown below.

```
OR      AUXR,#$40          ; Set Flash memory
                             to idle mode.
OR      PCON,#$01          ; Turn on Idle mode.
.       .                  ; Execution resumes
                             here when Idle
                             mode terminates.
```

When the Flash memory is put into the Idle mode by setting FMIDLE, restarting the CPU upon exiting Idle mode takes slightly longer, about 3 microseconds. However, the standby current consumed by the Flash memory is reduced from about 8mA to about 1mA.

Default Loader

A default loader that accepts programming commands in a predetermined format is contained permanently in the Boot ROM. A factory fresh device will enter this loader automatically if it is powered up without first being programmed by the user. Loader commands include functions such as erase block; program Flash memory; read Flash memory; and blank check.

Boot Vector

The XA-G49 contains two special FLASH registers: the BOOT VECTOR and the STATUS BYTE.

The “Boot Vector” allows forcing the execution of a user supplied Flash loader upon reset, under two specific sets of conditions. At the falling edge of reset, the XA-G49 examines the contents of the Status Byte. If the Status Byte is set to zero, power-up execution starts at location 0000H, which is the normal start address of the user’s application code.

When the Status Byte is set to a value other than zero, the Boot Vector is used as the reset vector (4 bytes), including the Boot Program Counter (BPC) and the Boot PSW (BPSW). The factory default settings are 8000h for the BPSW and F800h for the BPC, which corresponds to the address F900h for the factory masked-ROM ISP boot loader. The Status Byte is automatically set to a non-zero value when a programming error occurs. A custom boot loader can be written with the Boot Vector set to the custom boot loader.

NOTE: When erasing the Status Byte or Boot Vector, these bytes are erased at the same time. It is necessary to reprogram the Boot Vector after erasing and updating the Status Byte.

Hardware Activation of the Boot Vector

Program execution at the Boot Vector may also be forced from outside of the microcontroller by setting the correct state on a few pins. While Reset is asserted, the $\overline{PSEN}$ pin must be pulled low, the ALE pin allowed to float high (need not be pulled up externally), and the $\overline{EA}$ pin driven to a logic high (or up to V_{PP}). Then reset may be released. This is the same effect as having a non-zero status byte. This allows building an application that will normally execute the end user’s code but can be manually forced into ISP operation. The Boot ROM is enabled when use of the Boot Vector is forced as described above, so the branch may go to the default loader. Conversely, user code in the top 2 kbytes of the Flash memory may not be executed when the Boot Vector is used.

If the factory default setting for the BPC (F800h) is changed, it will no longer point to the ISP masked-ROM boot loader code. If this happens, the only possible way to change the contents of the Boot Vector is through the parallel programming method, provided that the end user application does not contain a customized loader that provides for erasing and reprogramming of the Boot Vector and Status Byte.

After programming the FLASH, the status byte should be erased to zero in order to allow execution of the user’s application code beginning at address 0000H.

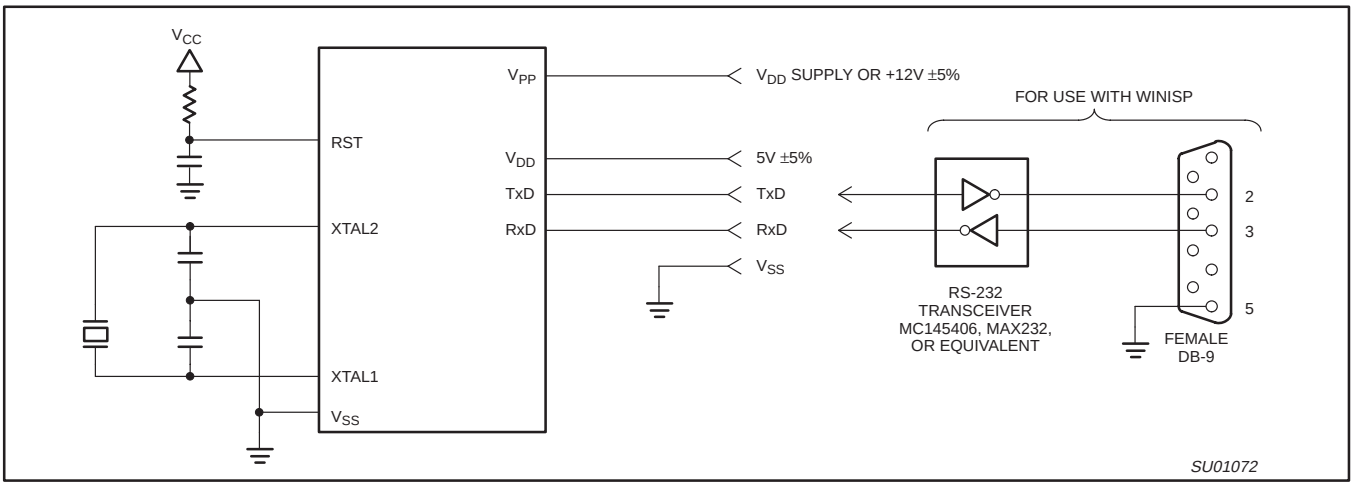


Figure 4. In-System Programming with a Minimum of Pins

In-System Programming (ISP)

In-System Programming (ISP) is performed without removing the microcontroller from the system. The In-System Programming (ISP) facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the XA-G49 through the serial port. This firmware is provided by Philips and embedded within each XA-G49 device.

The Philips In-System Programming (ISP) facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area.

The ISP function uses five pins: TxD, RxD, V_{SS}, V_{DD}, and V_{PP} (see Figure 4). Only a small connector needs to be available to interface your application to an external circuit in order to use this feature. The V_{PP} supply should be adequately decoupled and V_{PP} not allowed to exceed datasheet limits.

V_{CC}	V_{PP}	OSC FREQ	I_{DD}
5.0 V	5.0 V	22 MHz	75 ma typical
5.0 V	5.0 V	30 MHz	90 ma typical

ISP increases I_{DD} by less than 1mA.

ISP software is available on the Philips web site

1. With your browser, open this page: **www.semiconductors.com**
2. Enter **winzip.zip** into the Search box at the top of the Philips web page.
3. Click on **Microcontrollers Software** support.
4. Download disk1.zip and disk2.zip.
5. Create a directory on your hard drive named WINISP.
6. Unzip the two disk files into this new directory WINISP.

Using In-System Programming (ISP)

ISP mode is entered by holding PSEN low, asserting, un-asserting RESET, then releasing PSEN. When ISP mode is entered, the default loader first disables the watchdog timer to prevent a watchdog reset from occurring during programming.

The ISP feature allows for a wide range of baud rates to be used in the application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency. The ISP feature requires that an initial character (a lowercase f) be sent to the XA-G49 to establish the baud rate. The ISP firmware provides auto-echo of received characters.

Once baud rate initialization has been performed, the ISP firmware will only accept specific Intel Hex-type records. Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below:

:NNAAAARRDD..DDCC<crlf>

In the Intel Hex record, the “NN” represents the number of data bytes in the record. The XA-G49 will accept up to 16 (10H) data bytes. The “AAAA” string represents the address of the first byte in the record. If there are zero bytes in the record, this field is often set to 0000. The “RR” string indicates the record type. A record type of “00” is a data record. A record type of “01” indicates the end-of-file mark. In this application, additional record types will be added to indicate either commands or data for the ISP facility. The maximum number of data bytes in a record is limited to 16 (decimal). ISP commands are summarized in Table 1.

As a record is received by the XA-G49, the information in the record is stored internally and a checksum calculation is performed. The operation indicated by the record type is not performed until the entire record has been received. Should an error occur in the checksum, the XA-G49 will send an “X” out the serial port indicating a checksum error. If the checksum calculation is found to match the checksum in the record, then the command will be executed. In most cases, successful reception of the record will be indicated by transmitting a “.” character out the serial port (displaying the contents of the internal program memory is an exception).

In the case of a Data Record (record type 00), an additional check is made. A “.” character will NOT be sent unless the record checksum matched the calculated checksum and all of the bytes in the record were successfully programmed. For a data record, an “X” indicates that the checksum failed to match, and an “R” character indicates that one of the bytes did not properly program.

The ISP facility was designed so that specific crystal frequencies were not required in order to generate baud rates or time the programming pulses.

User Supplied Loader

A user program can simply decide at any time, for any reason, to begin Flash programming operations. All it has to do in advance is to instruct external circuitry to apply +5 V or +12 V to the V_{PP} pin, and make certain that the Boot ROM is enabled. User code may contain a loader designed to replace the application code contained in the Flash memory by loading new code through any communication medium available in the application. This is completely flexible and defined by the designer of the system. It could be done serially using RS-232, serially using some other method, or even parallel over a user defined I/O port. The user has the freedom to choose a method that does not interfere with the application circuit. As an added feature, the application program may also use the Flash memory as a long term data storage, saving configuration information, sensor readings, or any other desired data.

The actual loader code would typically be programmed by the user into the microcontroller in a parallel fashion or via the default loader during their manufacturing process. The entire initial Flash contents may be programmed at that time, or the rest of the application may be programmed into the Flash memory at a later time, possibly using the loader code to do the programming.

This application controlled programming capability allows for the possibility of changing the application code in the field. If the application circuit is embedded in a PC, or has a way to establish a telephone data link to a user's or manufacturer's computer, new code could be downloaded from diskette or a manufacturer's support system. There is even the possibility of conducting very specialized remote testing of a failing circuit board by the manufacturer by remotely programming a series of detailed test programs into the application board and checking the results.

Any user supplied loader should take the watchdog timer into account. Typically, the watchdog timer would be disabled upon entry to the loader if it might be running, in order to prevent a watchdog reset from occurring during programming.

XA Programming Specifications on Philips web site

Programming specifications for the XA family can be located on the Philips Semiconductors web site at:

www.semiconductors.com/mcu/

Click on **Support and Training**, then **Programming Specifications**.

Table 1. Intel-Hex Records Used by In-System Programming

RECORD TYPE	COMMAND/DATA FUNCTION

00 or 80	Data Record
	`:nnaaaa00dd....ddcc`
	Where:
	`Nn = number of bytes (hex) in record`
	`Aaaa = memory address of first byte in record`
	`dd....dd = data bytes`
	`cc = checksum`
	Example:
	`:10008000AF5F67F0602703E0322CFA92007780C3FD`

01 or 81	End of File (EOF), no operation
	`:xxxxxx01cc`
	Where:
	`xxxxxx = required field, but value is a “don’t care”`
	`cc = checksum`
	Example:
	`:00000001FF`

83	Miscellaneous Write Functions
	`:nnxxxx83ffssddcc`
	Where:
	`nn = number of bytes (hex) in record`
	`xxxx = required field, but value is a “don’t care”`
	`83 = Write Function`
	`ff = subfunction code`
	`ss = selection code`
	`dd = data input (as needed)`
	`cc = checksum`
	Subfunction Code = 01 (Erase Blocks)
	`ff = 01`
	`ss = block number in bits 7:5, Bits 4:0 = zeros`
	`block 0 : ss = 00h`
	`block 1 : ss = 20h`
	`block 2 : ss = 40h`
	`block 3 : ss = 80h`
	`block 4 : ss = C0h`
	Example:
	`:0200008301203C erase block 1`
	Subfunction Code = 04 (Erase Boot Vector and Status Byte)
	`ff = 04`
	`ss = don’t care`
	`dd = don’t care`
	Example:
	`:010000830478 erase boot vector and status byte`
	Subfunction Code = 05 (Program Security Bits)
	`ff = 05`
	`ss = 00 program security bit 1 (inhibit writing to FLASH)`
	`01 program security bit 2 (inhibit FLASH verify)`
	`02 program security bit 3 (disable external memory)`
	Example:
	`:02000083050175 program security bit 2`
	Subfunction Code = 06 (Program Status Byte or Boot Vector)
	`ff = 06`
	`ss = 00 program status byte`
	`01 program boot vector`
	NOTE: Only two bits of these Special Cells may be programmed at one time.
	Example:
	`:020000830601FC78 program boot vector to FC00h`

RECORD TYPE	COMMAND/DATA FUNCTION

84	Display Device Data or Blank Check – Record type 84 causes the contents of the entire FLASH array to be sent out the serial port in a formatted display. This display consists of an address and the contents of 16 bytes starting with that address. No display of the device contents will occur if security bit 2 has been programmed. The dumping of the device data to the serial port is terminated by the reception of any character.
	General Format of Function 84
	`:05xxxx84ssssseeeeffcc`
	Where:
	`05 = number of bytes (hex) in record`
	`xxxx = required field, but value is a “don’t care”`
	`84 = “Display Device Data or Blank Check” function code`
	`ssss = starting address`
	`eeee = ending address`
	`ff = subfunction`
	`00 = display data`
	`01 = blank check`
	`cc = checksum`
	Example:
	`:0500008440004FFF00E9 display 4000–4FFF`

85	Miscellaneous Read Functions
	General Format of Function 85
	`:02xxxx85ffsscc`
	Where:
	`02 = number of bytes (hex) in record`
	`xxxx = required field, but value is a “don’t care”`
	`85 = “Miscellaneous Read” function code`
	`ffss = subfunction and selection code`
	`0000 = read signature byte – manufacturer id (15H)`
	`0001 = read signature byte – device id # 1 (EAH)`
	`0002 = read signature byte – device id # 2 (XA–G49 = 54H))`
	`0700 = read security bits (returned value bits 3:1 = sb3,sb2,sb1)`
	`0701 = read status byte`
	`0702 = read boot vector`
	`cc = checksum`
	Example:
	`:02000085000178 read signature byte – device id # 1`

In-Application Programming Method

Several Application Program Interface (API) calls are available for use by an application program to permit selective erasing and programming of FLASH sectors. All calls are made through a common interface, PGM_MTP. The programming functions are selected by setting up the microcontroller's registers before making a call to PGM_MTP at FFF0H. Results are returned in the registers. The API calls are shown in Table 2.

Table 2. API calls

API CALL	PARAMETER
PROGRAM DATA BYTE	Input Parameters: R0H = 02h or 92h R6 = address of byte to program R4L = byte to program Return Parameter R4L = 00 if pass, non-zero if fail
ERASE BLOCK	Input Parameters: R0H = 01h or 93h R6H = block number in bits 7:5, bits 4:0 = '0' block 0 : R6H = 00h block 1 : R6H = 20h block 2 : R6H = 40h block 3 : R6H = 80h block 4 : R6H = C0h R6L = 00h Return Parameter R4L = 00 if pass, non-zero if fail
ERASE BPC and STATUS BYTE	Input Parameters: R0H = 04h Return Parameter R4L = 00 if pass, non-zero if fail
PROGRAM SECURITY BIT	Input Parameters: R0H = 05h R6H = 00h R6L = 00h – security bit # 1 (inhibit writing to FLASH) 01h – security bit # 2 (inhibit FLASH verify) 02h – security bit # 3 (disable external memory) Return Parameter none
PROGRAM STATUS BYTE	Input Parameters: R0H = 06h R6H = 00h R6L = 00h – program status byte R4L = status byte Return Parameter R4L = 00 if pass, non-zero if fail **NOTE:** Only two bits of this Special Cell may be programmed at one time.
PROGRAM BPC high byte	Input Parameters: R0H = 06h R6H = 00h R6L = 01h – program BPC R4L = BPC[15:8] (BPC[7:0] unchanged) Return Parameter R4L = 00 if pass, non-zero if fail **NOTE:** Only two bits of this Special Cell may be programmed at one time.
READ DEVICE DATA	Input Parameters: R0H = 03h R6 = address of byte to read Return Parameter R4L = value of byte read
READ MANUFACTURER ID	Input Parameters: R0H = 00h R6H = 00h R6L = 00h (manufacturer ID) Return Parameter R4L = value of byte read

API CALL	PARAMETER
READ DEVICE ID # 1	Input Parameters: R0H = 00h R6H = 00h R6L = 01h (device ID # 1) Return Parameter R4L = value of byte read
READ DEVICE ID # 2	Input Parameters: R0H = 00h R6H = 00h R6L = 02h (device ID # 2) Return Parameter R4L = value of byte read
READ SECURITY BITS	Input Parameters: R0H = 07h R6H = 00h R6L = 00h (security bits) Return Parameter R4L = value of byte read R4L[3:1] = sb3, sb2, sb1
READ STATUS BYTE	Input Parameters: R0H = 07h R6H = 00h R6L = 01h (status byte) Return Parameter R4L = value of BPC[15:8]
READ BPC	Input Parameters: R0H = 07h R6H = 00h R6L = 02h (boot vector) Return Parameter R4L = value of byte read (high byte of Boot PC)
PROGRAM ALL ZERO	Input Parameters: R0H = 90h R6H = block number in bits 7:5, bits 4:0 = ‘0’ block 0 : r6h = 00h block 1 : r6h = 20h block 2 : r6h = 40h block 3 : r6h = 80h block 4 : r6h = C0h R6L = 00h Return Parameters: R4L = 00 if pass, non–zero if fail
ERASE CHIP	Input Parameters: R0H = 91h R4L = 55h (after chip erase, return to caller) = AAh (after chip erase, reset chip) = others: error Return Parameters: R4L = 00 if pass, non–zero if fail
PROGRAM SPECIAL CELL	Input Parameters: R0H = 94h R6 = special cell address 0000h: program BPSW[7:0] 0001h: program BPSW[15:8] 0002h: program BPC[7:0] 0003h: program BPC[15:8] 0004h: program status byte 000Ah: program security bit #1 000Ch: program security bit #2 000Eh: program security bit #3 R4L = byte value to program Return Parameters: R4L = 00 if pass, non–zero if fail **NOTE:** Only two bits of these Special Cells may be programmed at one time.

API CALL	PARAMETER

ERASE SPECIAL CELL	Input Parameters:
	`R0H = 95h`
	`R6 = special cell address`
	`0000h: erase BPSW[7:0]`
	`0001h: erase BPSW[15:8]`
	`0002h: erase BPC[7:0]`
	`0003h: erase BPC[15:8]`
	`0004h: erase status byte`
	Return Parameters:
	`R4L = 00 if pass, non–zero if fail`

READ SPECIAL CELL	Input Parameters:
	`R0H = 96h`
	`R6 = special cell address`
	`0000h: read BPSW[7:0]`
	`0001h: read BPSW[15:8]`
	`0002h: read BPC[7:0]`
	`0003h: read BPC[15:8]`
	`0004h: read status byte`
	`0006h: read manufacturer ID`
	`0007h: read device ID #1`
	`0008h: read device ID #2`
	`000Ah: read security bit #1`
	`000Ch: read security bit #2`
	`000Eh: read security bit #3`
	Return Parameters:
	`R4L = value of byte read`

Security

The security feature protects against software piracy and prevents the contents of the Flash from being read. The Security Lock bits are located in Flash. The XA-G49 has 3 programmable security lock bits that will provide different levels of protection for the on-chip code and data (see Table 3).

Table 3.

SECURITY LOCK BITS[1]				PROTECTION DESCRIPTION
Level	SB1	SB2	SB3	
1	0	0	0	No program security features enabled.
2	1	0	0	Same as level 1, plus block erase is disabled. Erase or programming of the status byte or boot vector is disabled.
3	1	1	0	Same as level 2, plus program verification is disabled
4	1	1	1	Same as level 3, plus external execution is disabled.

NOTE:
1. Any other combination of the Lock bits is not defined.
2. Security bits are independent of each other. Full-chip erase may be performed regardless of the states of the security bits.
3. Setting LB doesn't prevent programming of unprogrammed bits.

XA-G49 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 of the timer/counters (Timer 0, Timer 1 and Timer 2) can be independently programmed to operate either as timers or event counters via the C/T bit in the TnCON register. All timers count up unless otherwise stated. These timers may be dynamically read during program execution.

The base clock rate of all of the timers is user programmable. This applies to timers T0, T1, and T2 when running in timer mode (as opposed to counter mode), and the watchdog timer. The clock driving the timers is called TCLK and is determined by the setting of two bits (PT1, PT0) in the System Configuration Register (SCR). The frequency of TCLK may be selected to be the oscillator input divided by 4 (Osc/4), the oscillator input divided by 16 (Osc/16), or the oscillator input divided by 64 (Osc/64). This gives a range of possibilities for the XA timer functions, including baud rate generation, Timer 2 capture. Note that this single rate setting applies to all of the timers.

When timers T0, T1, or T2 are used in the counter mode, the register will increment whenever a falling edge (high to low transition) is detected on the external input pin corresponding to the timer clock. These inputs are sampled once every 2 oscillator cycles, so it can take as many as 4 oscillator cycles to detect a transition. Thus the maximum count rate that can be supported is Osc/4. The duty cycle of the timer clock inputs is not important, but any high or low state on the timer clock input pins must be present for 2 oscillator cycles before it is guaranteed to be “seen” by the timer logic.

Timer 0 and Timer 1

The “Timer” or “Counter” function is selected by control bits C/T in the special function register TMOD. 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 will give 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.

SCR Address:440
Not Bit Addressable
Reset Value: 00H

MSB							LSB
—	—	—	—	PT1	PT0	CM	PZ

PT1	PT0	OPERATING
		Prescaler selection.
0	0	Osc/4
0	1	Osc/16
1	0	Osc/64
1	1	Reserved
CM		Compatibility Mode allows the XA to execute most translated 80C51 code on the XA. The XA register file must copy the 80C51 mapping to data memory and mimic the 80C51 indirect addressing scheme.
PZ		Page Zero mode forces all program and data addresses to 16-bits only. This saves stack space and speeds up execution but limits memory access to 64k.

SU00589

Figure 5. System Configuration Register (SCR)

TMOD Address:45C
Not Bit Addressable
Reset Value: 00H

MSB							LSB
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

GATE	Gating control when set. Timer/Counter “n” is enabled only while “$\overline{INTn}$” pin is high and “TRn” control bit is set. When cleared Timer “n” is enabled whenever “TRn” control bit is set.
C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from “Tn” input pin).

M1	M0	OPERATING
0	0	16-bit auto-reload timer/counter
0	1	16-bit non-auto-reload timer/counter
1	0	8-bit auto-reload timer/counter
1	1	Dual 8-bit timer mode (timer 0 only)

SU00605

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 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 = 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 = 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: C/T, GATE, 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.

TCON Address:410
Bit Addressable
Reset Value: 00H

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. This flag will not be set if T1OE (TSTAT.2) is set. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. This flag will not be set if T0OE (TSTAT.0) is set. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
TCON.3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.2	IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SU00604C

Figure 7. Timer/Counter Control (TCON) Register

T2CON Address:418
Bit Addressable
Reset Value: 00H

MSB							LSB
TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2

BIT	SYMBOL	FUNCTION
T2CON.7	TF2	Timer 2 overflow flag. Set by hardware on Timer/Counter overflow. Must be cleared by software. TF2 will not be set when RCLK0, RCLK1, TCLK0, TCLK1 or T2OE=1.
T2CON.6	EXF2	Timer 2 external flag is set when a capture or reload occurs due to a negative transition on T2EX (and EXEN2 is set). This flag will cause a Timer 2 interrupt when this interrupt is enabled. EXF2 is cleared by software.
T2CON.5	RCLK0	Receive Clock Flag.
T2CON.4	TCLK0	Transmit Clock Flag. RCLK0 and TCLK0 are used to select Timer 2 overflow rate as a clock source for UART0 instead of Timer T1.
T2CON.3	EXEN2	Timer 2 external enable bit allows a capture or reload to occur due to a negative transition on T2EX.
T2CON.2	TR2	Start=1/Stop=0 control for Timer 2.
T2CON.1	C/T2	Timer or counter select. 0=Internal timer 1=External event counter (falling edge triggered)
T2CON.0	CP/RL2	Capture/Reload flag. If CP/RL2 & EXEN2=1 captures will occur on negative transitions of T2EX. If CP/RL2=0, EXEN2=1 auto reloads occur with either Timer 2 overflows or negative transitions at T2EX. If RCLK or TCLK=1 the timer is set to auto reload on Timer 2 overflow, this bit has no effect.

SU01385

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) 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 (Osc/8 to Osc/8,388,608) outputs could be achieved by adjusting the prescaler along with the auto-reload register values. With a 30.0MHz oscillator, this range would be 3.58Hz to 3.75MHz.

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 C/T2 in the special function register T2CON. 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 either or both UARTs via SFRs T2MOD and T2CON). These modes are shown in Table 4.

Capture Mode

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then timer 2 is a 16-bit timer or counter, which upon overflowing sets bit TF2, 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 in T2CON 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 in T2CON is set, the timer registers will also be reloaded and the EXF2 flag 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 bit DCEN (Down Counter Enable) in the T2MOD special function register (see Table 4). 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 = 0, then Timer 2 counts up to FFFFH and sets the TF2 (Overflow Flag) bit 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 = 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 or EXF2 bit can generate the Timer 2 interrupt.

In Figure 13, the DCEN = 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 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 the 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 toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution, if needed. the EXF2 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 TCLKn and/or RCLKn in T2CON or T2MOD, the Timer 2 can be chosen as the baud rate generator for either or both UARTs. The baud rates for transmit and receive can be simultaneously different.

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. It can be programmed (1) to input the external clock for Timer/Counter 2 or (2) to output a 50% duty cycle clock ranging from 3.58Hz to 3.75MHz at a 30MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (in T2CON) must be cleared and bit T20E in T2MOD 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:

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

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 4. Timer 2 Operating Modes

TR2	CP/RL2	RCLK+TCLK	DCEN	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

TSTAT Address:411
Bit Addressable
Reset Value: 00H

MSB							LSB
—	—	—	—	—	T1OE	—	T0OE

BIT	SYMBOL	FUNCTION
TSTAT.2	T1OE	When 0, this bit allows the T1 pin to clock Timer 1 when in the counter mode. When 1, T1 acts as an output and toggles at every Timer 1 overflow.
TSTAT.0	T0OE	When 0, this bit allows the T0 pin to clock Timer 0 when in the counter mode. When 1, T0 acts as an output and toggles at every Timer 0 overflow.

SU00612B

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

T2MOD Address:419
Bit Addressable
Reset Value: 00H

MSB							LSB
—	—	RCLK1	TCLK1	—	—	T2OE	DCEN

BIT	SYMBOL	FUNCTION
T2MOD.5	RCLK1	Receive Clock Flag.
T2MOD.4	TCLK1	Transmit Clock Flag. RCLK1 and TCLK1 are used to select Timer 2 overflow rate as a clock source for UART1 instead of Timer T1.
T2MOD.1	T2OE	When 0, this bit allows the T2 pin to clock Timer 2 when in the counter mode. When 1, T2 acts as an output and toggles at every Timer 2 overflow.
T2MOD.0	DCEN	Controls count direction for Timer 2 in autoreload mode. DCEN=0 counter set to count up only DCEN=1 counter set to count up or down, depending on T2EX (see text).

SU00610B

Figure 10. Timer 2 Mode Control (T2MOD)

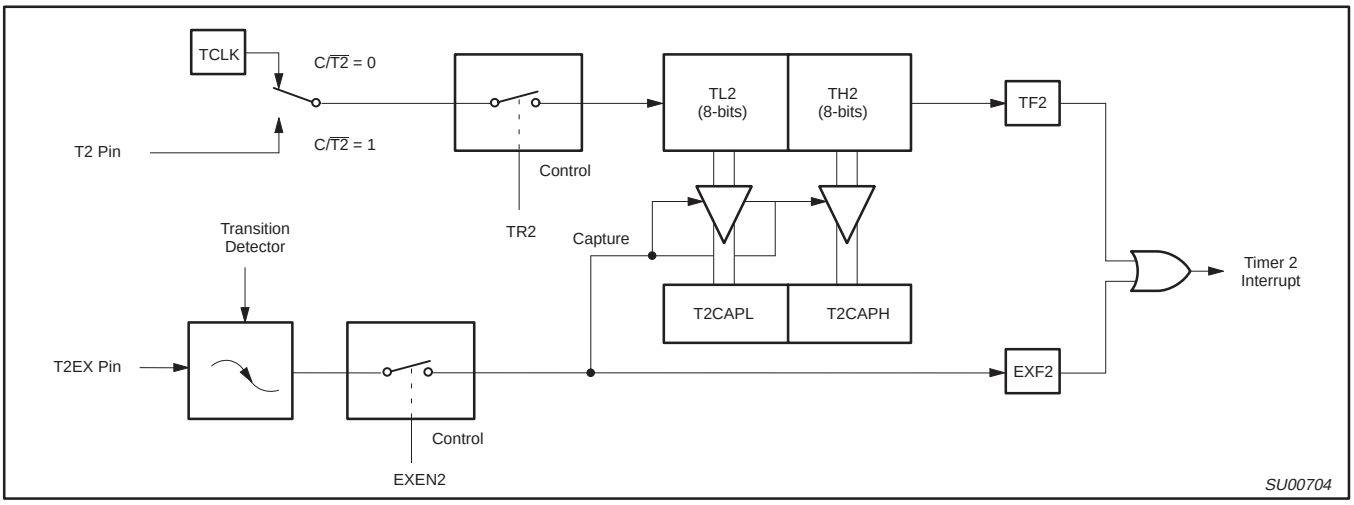


Figure 11. Timer 2 in Capture Mode

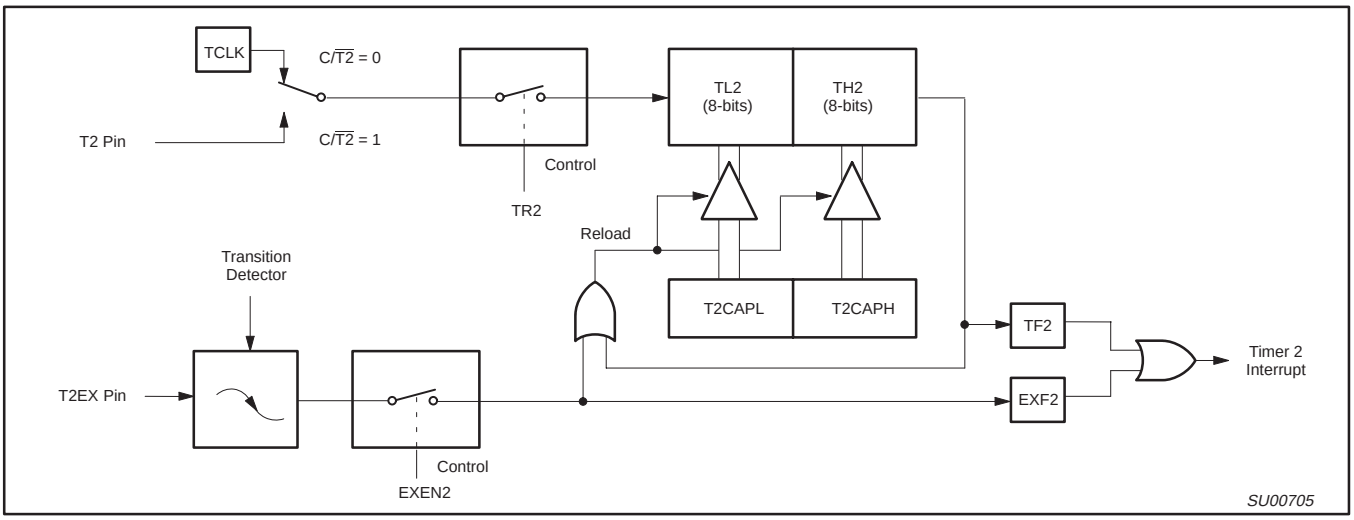


Figure 12. Timer 2 in Auto-Reload Mode (DCEN = 0)

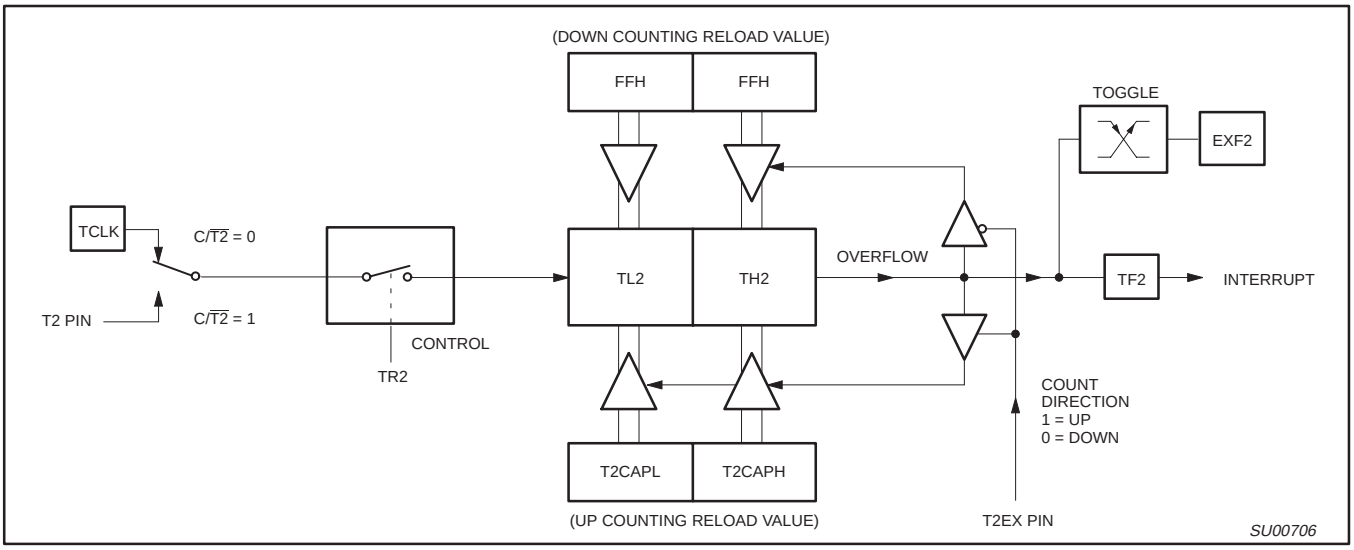


Figure 13. Timer 2 Auto Reload Mode (DCEN = 1)

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 (see Table 5). 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. Definitions: 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 5, 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.

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

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

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

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 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 5. 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 $\overline{\text{RESET}}$ 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 (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.

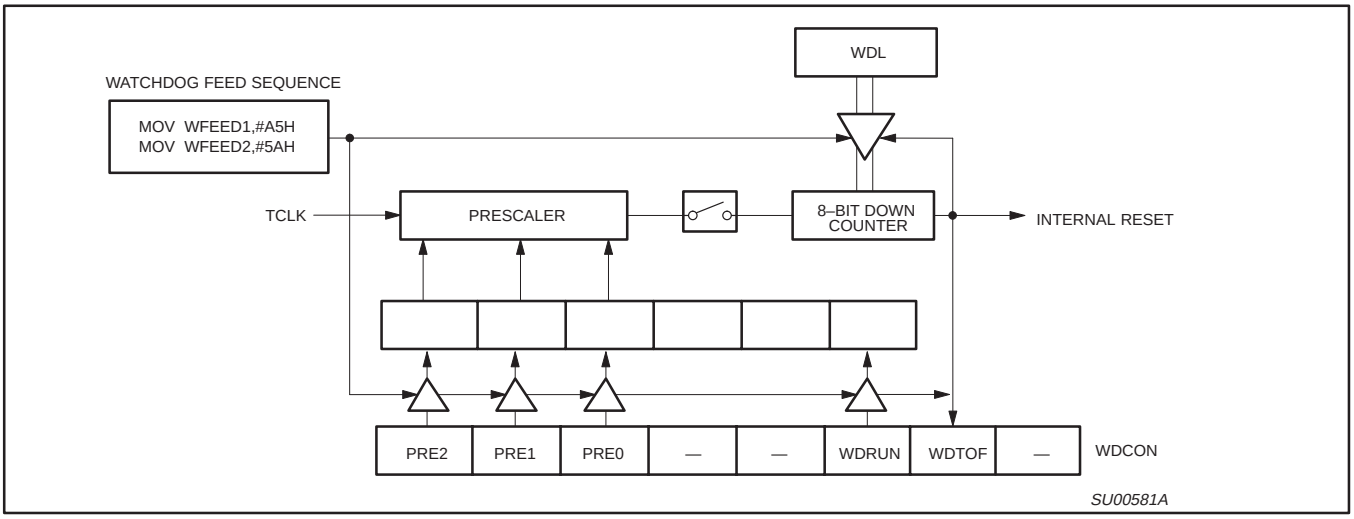


Figure 14. Watchdog Timer in XA-G49

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.4	—	
WDCON.3	—	
WDCON.2	WDRUN	Watchdog Run Control bit, reset to 1
WDCON.1	WDTOF	Timeout flag
WDCON.0	—	

UARTs

The XA-G49 includes 2 UART ports that are 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.

Some other enhancements have been made to UART operation. The first is that there are separate interrupt vectors for each UART's transmit and receive functions. The UART transmitter has been double buffered, allowing packed transmission of data with no gaps between bytes and less critical interrupt service routine timing. A break detect function has been added to the UART. This operates independently of the UART itself and provides a start-of-break status bit that the program may test. Finally, an Overrun Error flag has been added to detect missed characters in the received data stream. The double buffered UART transmitter may require some software changes in code written for the original XA-G49 single buffered UART.

Each UART baud rate is determined by either a fixed division of the oscillator (in UART modes 0 and 2) or by the timer 1 or timer 2 overflow rate (in UART modes 1 and 3).

Timer 1 defaults to clock both UART0 and UART1. Timer 2 can be programmed to clock either UART0 through T2CON (via bits R0CLK and T0CLK) or UART1 through T2MOD (via bits R1CLK and T1CLK). In this case, the UART not clocked by T2 could use T1 as the clock source.

The serial port receive and transmit registers are both accessed at Special Function Register SnBUF. Writing to SnBUF loads the transmit register, and reading SnBUF accesses a 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 RxDn. TxDn 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 TxDn) or received (through RxDn): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SnCON. The baud rate is variable.

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_n in SnCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8_n. On receive, the 9th data bit goes into RB8_n in Special Function Register SnCON, 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 TxDn) or received (through RxDn): 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.

In all four modes, transmission is initiated by any instruction that uses SnBUF as a destination register. Reception is initiated in Mode 0 by the condition RI_n = 0 and REN_n = 1. Reception is initiated in the other modes by the incoming start bit if REN_n = 1.

Serial Port Control Register
The serial port control and status register is the Special Function Register SnCON, shown in Figure 16. 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 15. Serial Port Extended Status (SnSTAT) Register (See also Figure 17 regarding Framing Error flag)

INTERRUPT SCHEME

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

Table 6. Vector Locations for UARTs in XA

Vector Address	Interrupt Source	Arbitration
A0H – A3H	UART 0 Receiver	7
A4H – A7H	UART 0 Transmitter	8
A8H – ABH	UART 1 Receiver	9
ACH – AFH	UART 1 Transmitter	10

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
The UARTs in XA has the following error flags; see Figure 15.

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 18.

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 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

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

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

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

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

Upon reset SADDR and SADEN 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.

SnCON Address: S0CON 420
S1CON 424

Bit Addressable
Reset Value: 00H

MSB							LSB
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	$f_{OSC}/16$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{OSC}/32$
1	1	3	9-bit UART	variable

BIT	SYMBOL	FUNCTION
SnCON.5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2=1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.
SnCON.4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
SnCON.3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired. The TB8 bit is not double buffered. See text for details.
SnCON.2	RB8	In Modes 2 and 3, is the 9th data bit that was received. In Mode 1, if SM2=0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
SnCON.1	TI	Transmit interrupt flag. Set when another byte may be written to the UART transmitter. See text for details. Must be cleared by software.
SnCON.0	RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the end of the stop bit time in the other modes (except see SM2). Must be cleared by software.

SU00597C

Figure 16. Serial Port Control (SnCON) Register

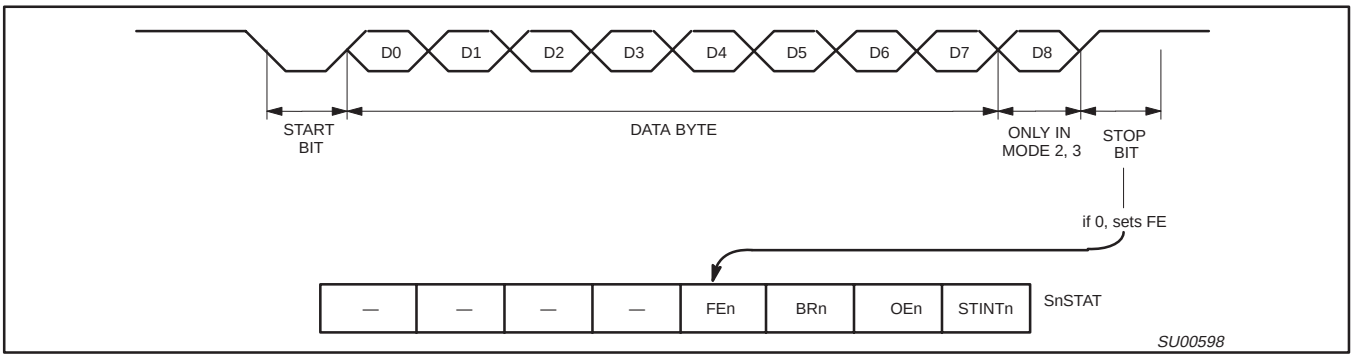


Figure 17. UART Framing Error Detection

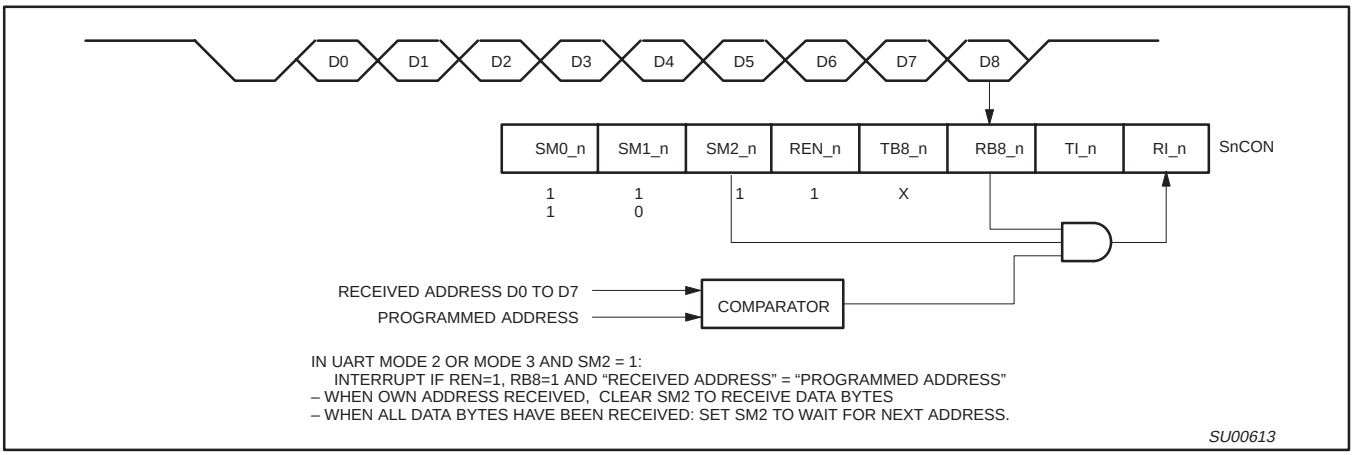


Figure 18. UART Multiprocessor Communication, Automatic Address Recognition

I/O PORT OUTPUT CONFIGURATION

Each I/O port pin can be user configured to one of 4 output types. The types are Quasi-bidirectional (essentially the same as standard 80C51 family I/O ports), Open-Drain, Push-Pull, and Off (high impedance). The default configuration after reset is Quasi-bidirectional. However, in the ROMless mode (the $\overline{EA}$ pin is low at reset), the port pins that comprise the external data bus will default to push-pull outputs.

I/O port output configurations are determined by the settings in port configuration SFRs. There are 2 SFRs for each port, called PnCFGA and PnCFGB, where “n” is the port number. One bit in each of the 2 SFRs relates to the output setting for the corresponding port pin, allowing any combination of the 2 output types to be mixed on those port pins. For instance, the output type of port 1 pin 3 is controlled by the setting of bit 3 in the SFRs P1CFGA and P1CFGB.

Table 7 shows the configuration register settings for the 4 port output types. The electrical characteristics of each output type may be found in the DC Characteristic table.

Table 7. Port Configuration Register Settings

PnCFGB	PnCFGA	Port Output 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 an input at reset (see Reset Options below), while the address size is set by the program in a configuration register. If all off-chip code is selected (through the use of the $\overline{EA}$ pin), the initial code fetches will be done with the maximum address size (20 bits).

RESET

The device is reset whenever a logic “0“ is applied to $\overline{RST}$ for at least 10 microseconds, placing a low level on the pin re-initializes the on-chip logic. Reset must be asserted when power is initially applied to the XA and held until the oscillator is running.

The duration of reset must be extended when power is initially applied or when using reset to exit power down mode. This is due to the need to allow the oscillator time to start up and stabilize. For most power supply ramp up conditions, this time is 10 milliseconds.

As $\overline{RST}$ is brought high again, an exception is generated which causes the processor to jump to the reset address. Typically, this is the address contained in the memory location 0000. The destination of the reset jump must be located in the first 64 K of code address on power-up, all vectors are 16-bit values and so point to page zero addresses only. After a reset the RAM contents are indeterminate.

Alternatively, the Boot Vector may supply the reset address. This happens when use of the Boot Vector is forced or when the Flash status byte is non-zero. These cases are described in the section “Hardware Activation of the Boot Vector” on page 11.

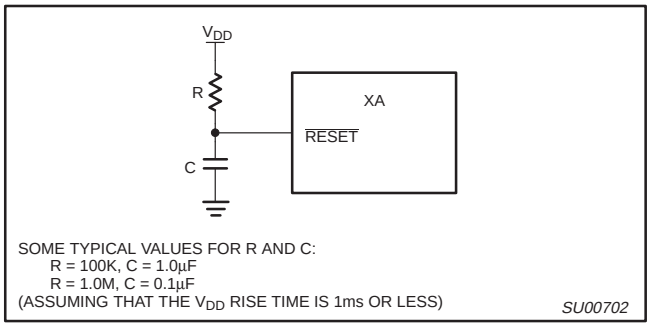


Figure 19. Recommended Reset Circuit

RESET OPTIONS

The $\overline{EA}$ pin is sampled on the rising edge of the $\overline{RST}$ pulse, and determines whether the device is to begin execution from internal or external code memory. $\overline{EA}$ pulled high configures the XA in single-chip mode. If $\overline{EA}$ is driven low, the device enters ROMless mode. After Reset is released, the $\overline{EA}$/WAIT pin becomes a bus wait signal for external bus transactions.

The BUSW/P3.5 pin is weakly pulled high while reset 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 pullup will cause a 1 to be loaded for the bus width, giving a 16-bit external bus. BUSW may be pulled low with a 2.7 K or smaller value resistor, giving an 8-bit external bus. The bus width setting from the BUSW pin may be overridden by software once the user program is running.

Both $\overline{EA}$ and BUSW must be held for three oscillator clock times after reset is deasserted to guarantee that their values are latched correctly.

POWER REDUCTION MODES

The XA-G49 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 (2 V), retaining the RAM, register, and SFR values at the point where the power down mode was entered.

INTERRUPTS

The XA-G49 supports 38 vectored interrupt sources. These include 9 maskable event interrupts, 7 exception interrupts, 16 trap interrupts, and 7 software interrupts. The maskable interrupts each have 8 priority levels and may be globally and/or individually enabled or disabled.

The XA defines four types of interrupts:

- **Exception Interrupts** – These are system level errors and other very important occurrences which include stack overflow, divide-by-0, and reset.
- **Event interrupts** – These are peripheral interrupts from devices such as UARTs, timers, and external interrupt inputs.
- **Software Interrupts** – These are equivalent of hardware interrupt, but are requested only under software control.
- **Trap Interrupts** – These are TRAP instructions, generally used to call system services in a multi-tasking system.

Exception interrupts, software interrupts, and trap interrupts are generally standard for XA derivatives and are detailed in the *XA User Guide*. Event interrupts tend to be different on different XA derivatives.

The XA-G49 supports a total of 9 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-G49. 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-G49, 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.

Table 8. Interrupt Vectors

EXCEPTION/TRAPS PRECEDENCE

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

EVENT INTERRUPTS

DESCRIPTION	FLAG BIT	VECTOR ADDRESS	ENABLE BIT	INTERRUPT PRIORITY	ARBITRATION RANKING
External interrupt 0	IE0	0080–0083	EX0	IPA0.2–0 (PX0)	2
Timer 0 interrupt	TF0	0084–0087	ET0	IPA0.6–4 (PT0)	3
External interrupt 1	IE1	0088–008B	EX1	IPA1.2–0 (PX1)	4
Timer 1 interrupt	TF1	008C–008F	ET1	IPA1.6–4 (PT1)	5
Timer 2 interrupt	TF2(EXF2)	0090–0093	ET2	IPA2.2–0 (PT2)	6
Serial port 0 Rx	RI.0	00A0–00A3	ERI0	IPA4.2–0 (PRIO)	7
Serial port 0 Tx	TI.0	00A4–00A7	ETI0	IPA4.6–4 (PTIO)	8
Serial port 1 Rx	RI.1	00A8–00AB	ERI1	IPA5.2–0 (PRT1)	9
Serial port 1 Tx	TI.1	00AC–00AF	ETI1	IPA5.6–4 (PTI1)	10

SOFTWARE INTERRUPTS

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

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Operating temperature under bias	–55 to +125	°C
Storage temperature range	–65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ pin to V_{SS}	0 to +13.0	V
Voltage on any other pin to V_{SS}	–0.5 to V_{DD}+0.5 V	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

V_{DD} = 4.75 V to 5.25 V unless otherwise specified;
T_{amb} = 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	
Supplies						
I_{DD}	Supply current operating	5.25 V, 30 MHz			110	mA
I_{ID}	Idle mode supply current	5.25 V, 30 MHz			40	mA
I_{PD}	Power-down current				30	µA
I_{PDI}	Power-down current (–40°C to +85°C)				150	µA
V_{RAM}	RAM-keep-alive voltage	RAM-keep-alive voltage	1.5			V
V_{IL}	Input low voltage		–0.5		0.22 V_{DD}	V
V_{IH}	Input high voltage, except XTAL1, RST	At 5.0 V	2.2			V
V_{IH1}	Input high voltage to XTAL1, RST	At 5.0 V	0.8V_{DD}			V
V_{IL1}	Input low voltage to XATL1, RST	At 5.0 V			0.12 V_{DD}	
V_{OL}	Output low voltage all ports, ALE, $\overline{PSEN}$[3]	I_{OL} = 3.2mA, V_{DD} = 5.0 V			0.5	V
V_{OH1}	Output high voltage all ports, ALE, $\overline{PSEN}$[1]	I_{OH} = –100µA, V_{DD} = 4.75 V	2.4			V
V_{OH2}	Output high voltage, ports P0–3, ALE, $\overline{PSEN}$[2]	I_{OH} = 3.2mA, V_{DD} = 4.75 V	2.4			V
C_{IO}	Input/Output pin capacitance				15	pF
I_{IL}	Logical 0 input current, P0–3[6]	V_{IN} = 0.45 V		–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 5.25 V			–650	µA

NOTES:
1. Ports in Quasi bi-directional mode with weak pull-up (applies to ALE, $\overline{PSEN}$ only during RESET).
2. Ports in Push-Pull mode, both pull-up and pull-down assumed to be 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 2 V.
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 V_{DD} = 5 V.)
   Maximum I_{OL} per 8-bit port: 26mA
   Maximum total I_{OL} for all output: 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.

AC ELECTRICAL CHARACTERISTICS (5 V)

V_{DD} = 4.75 V to 5.25 V; 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	30	MHz
t_C	26	Clock period and CPU timing cycle	$1/f_C$		ns
t_{CHCX}	26	Clock high time	t_C * 0.5 [7]		ns
t_{CLCX}	26	Clock low time	t_C * 0.4 [7]		ns
t_{CLCH}	26	Clock rise time		5	ns
t_{CHCL}	26	Clock fall time		5	ns
Address Cycle					
t_{CRAR}	25	Delay from clock rising edge to ALE rising edge	5	46	ns
t_{LHLL}	20	ALE pulse width (programmable)	(V1 * t_C) – 6		ns
t_{AVLL}	20	Address valid to ALE de-asserted (set-up)	(V1 * t_C) – 14		ns
t_{LLAX}	20	Address hold after ALE de-asserted	(t_C/2) – 10		ns
Code Read Cycle					
t_{PLPH}	20	$\overline{PSEN}$ pulse width	(V2 * t_C) – 10		ns
t_{LLPL}	20	ALE de-asserted to $\overline{PSEN}$ asserted	(t_C/2) – 7		ns
t_{AVIVA}	20	Address valid to instruction valid, ALE cycle (access time)		(V3 * t_C) – 36	ns
t_{AVIVB}	21	Address valid to instruction valid, non-ALE cycle (access time)		(V4 * t_C) – 29	ns
t_{PLIV}	20	$\overline{PSEN}$ asserted to instruction valid (enable time)		(V2 * t_C) – 29	ns
t_{PXIX}	20	Instruction hold after $\overline{PSEN}$ de-asserted	0		ns
t_{PXIZ}	20	Bus 3-State after $\overline{PSEN}$ de-asserted (disable time)		t_C – 8	ns
t_{IXUA}	20	Hold time of unlatched part of address after instruction latched	0		ns
Data Read Cycle					
t_{RLRH}	22	$\overline{RD}$ pulse width	(V7 * t_C) – 10		ns
t_{LLRL}	22	ALE de-asserted to $\overline{RD}$ asserted	(t_C/2) – 7		ns
t_{AVDVA}	22	Address valid to data input valid, ALE cycle (access time)		(V6 * t_C) – 36	ns
t_{AVDVB}	23	Address valid to data input valid, non-ALE cycle (access time)		(V5 * t_C) – 29	ns
t_{RLDV}	22	$\overline{RD}$ low to valid data in, enable time		(V7 * t_C) – 29	ns
t_{RHDX}	22	Data hold time after $\overline{RD}$ de-asserted	0		ns
t_{RHDZ}	22	Bus 3-State after $\overline{RD}$ de-asserted (disable time)		t_C – 8	ns
t_{DXUA}	22	Hold time of unlatched part of address after data latched	0		ns
Data Write Cycle					
t_{WLWH}	24	$\overline{WR}$ pulse width	(V8 * t_C) – 10		ns
t_{LLWL}	24	ALE falling edge to $\overline{WR}$ asserted	(V12 * t_C) – 10		ns
t_{QVWX}	24	Data valid before $\overline{WR}$ asserted (data setup time)	(V13 * t_C) – 22		ns
t_{WHQX}	24	Data hold time after $\overline{WR}$ de-asserted (Note 6)	(V11 * t_C) – 7		ns
t_{AVWL}	24	Address valid to $\overline{WR}$ asserted (address setup time) (Note 5)	(V9 * t_C) – 22		ns
t_{UAWH}	24	Hold time of unlatched part of address after $\overline{WR}$ is de-asserted	(V11 * t_C) – 7		ns
Wait Input					
t_{WTH}	25	WAIT stable after bus strobe ($\overline{RD}$, $\overline{WR}$, or $\overline{PSEN}$) asserted		(V10 * t_C) – 30	ns
t_{WTL}	25	WAIT hold after bus strobe ($\overline{RD}$, $\overline{WR}$, or $\overline{PSEN}$) 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 $\overline{PSEN}$ 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, $\overline{PSEN}$ 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 $\overline{RD}$ 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, $\overline{RD}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 $\overline{RD}$ (for data read cycles), $\overline{WRL}$ and/or $\overline{WRH}$ (for data write cycles), or $\overline{PSEN}$ (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 $\overline{PSEN}$. V10 = V8 for a data write cycle using $\overline{WRL}$ and/or $\overline{WRH}$. V10 = V7–1 for a data read cycle using $\overline{RD}$. 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 $\overline{RD}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 $\overline{WRL}$ and/or $\overline{WRH}$ 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 does not have $\overline{PSEN}$ 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 $\overline{RD}$ 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 $\overline{WR}$ strobe. This is not usually the case, and in most applications this parameter is not used.
6. Please note that the XA-G49 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 or ceramic resonator is connected to the XTAL1 and XTAL2 pins.

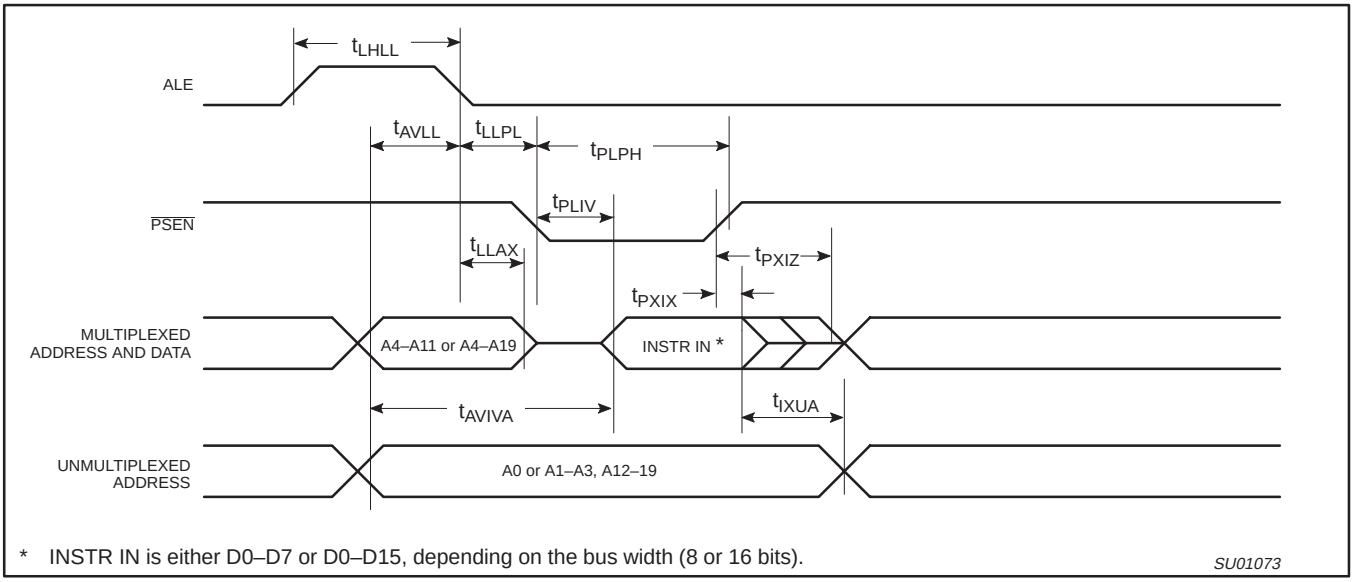


Figure 20. External Program Memory Read Cycle (ALE Cycle)

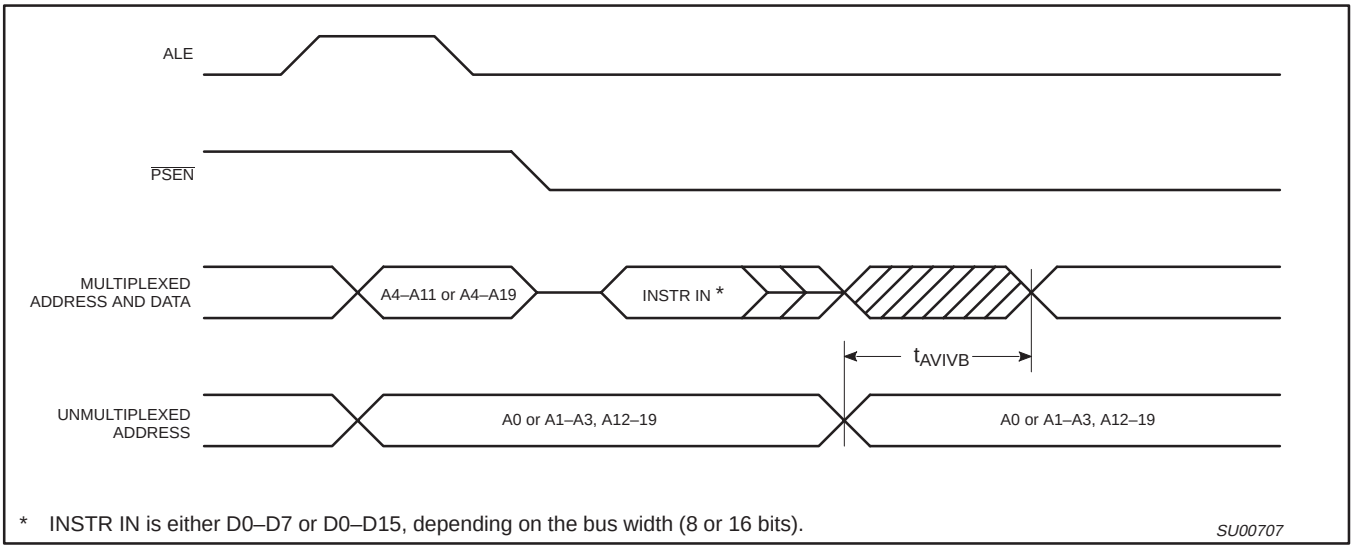


Figure 21. External Program Memory Read Cycle (Non-ALE Cycle)

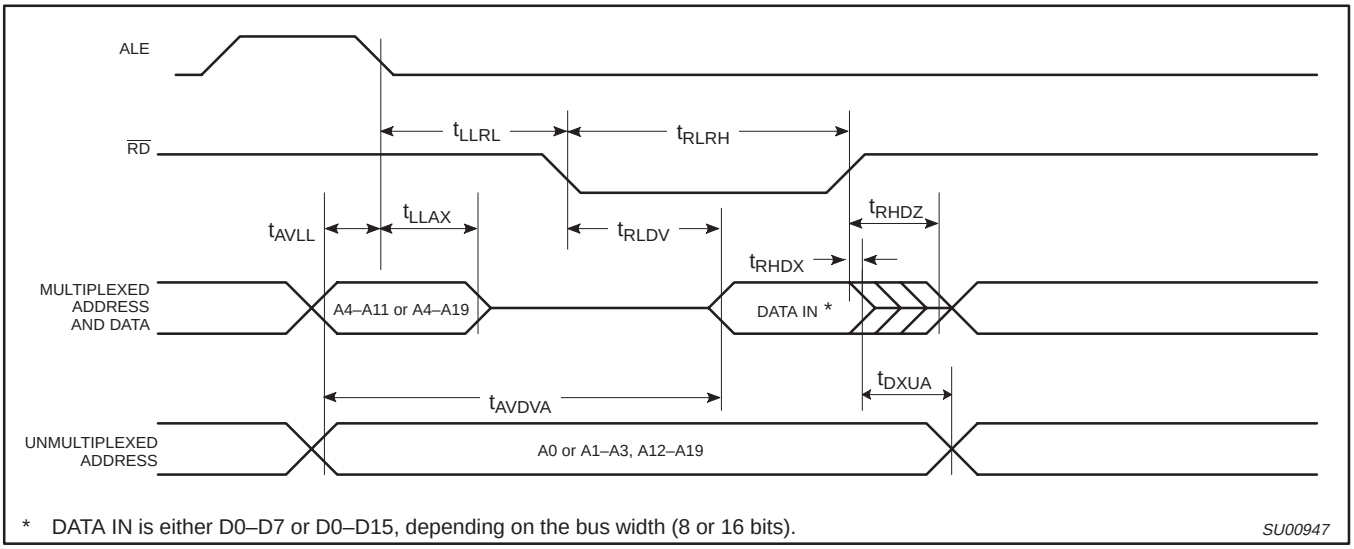


Figure 22. External Data Memory Read Cycle (ALE Cycle)

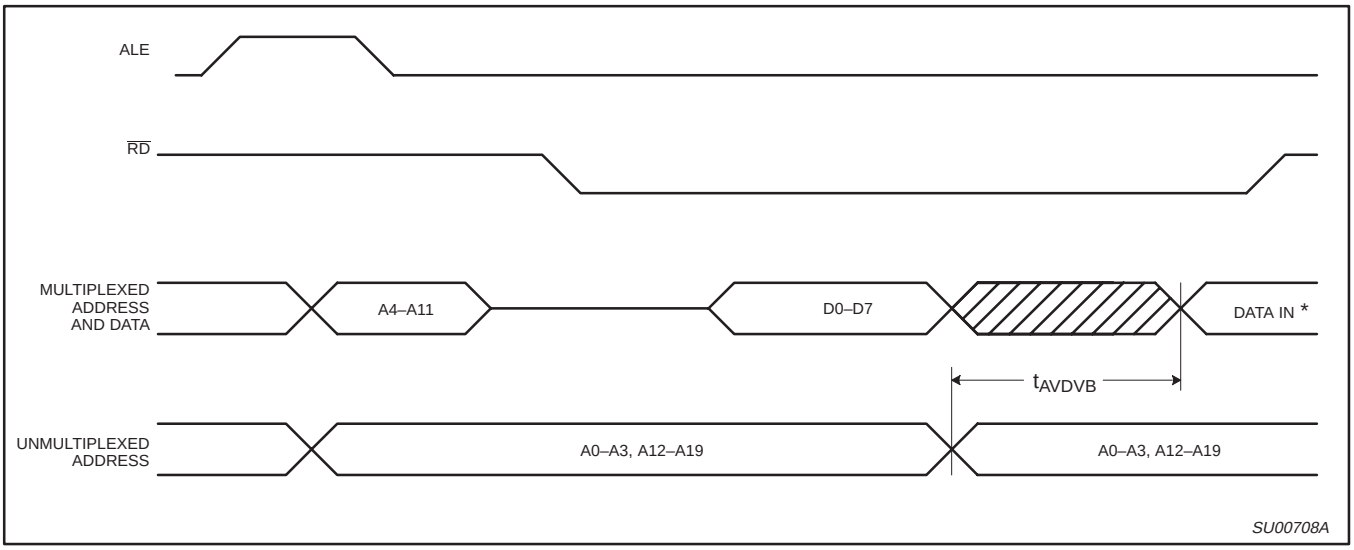


Figure 23. External Data Memory Read Cycle (Non-ALE Cycle)

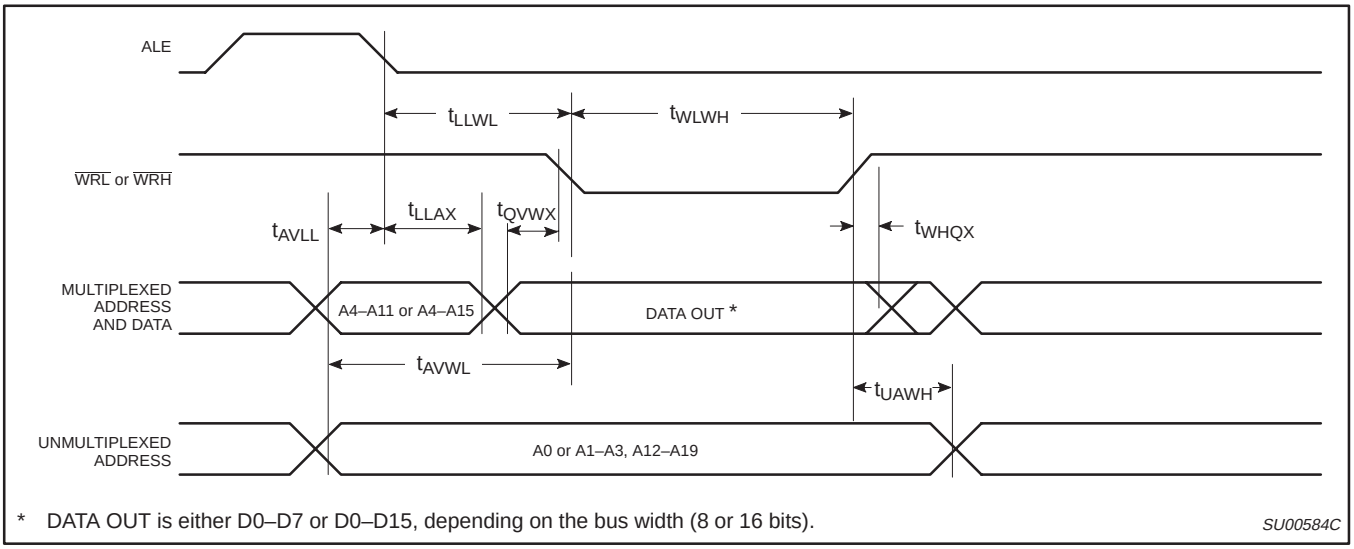


Figure 24. External Data Memory Write Cycle

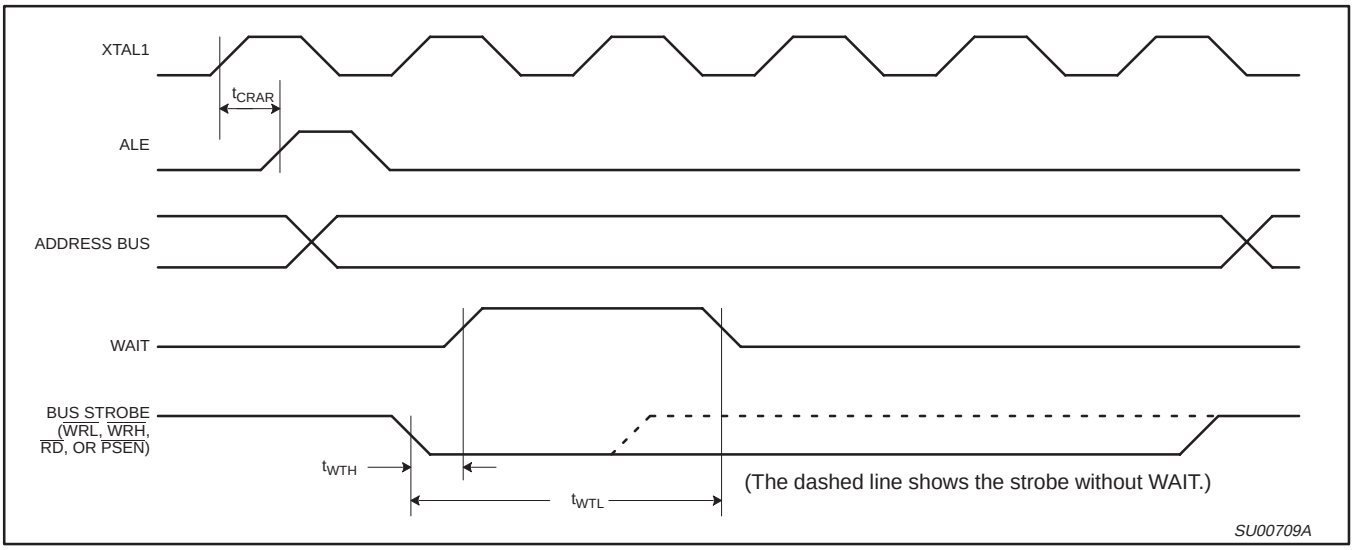


Figure 25. WAIT Signal Timing

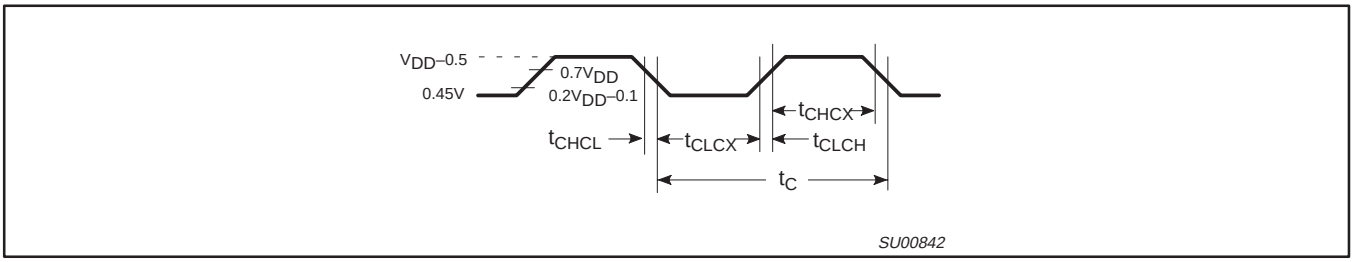


Figure 26. External Clock Drive

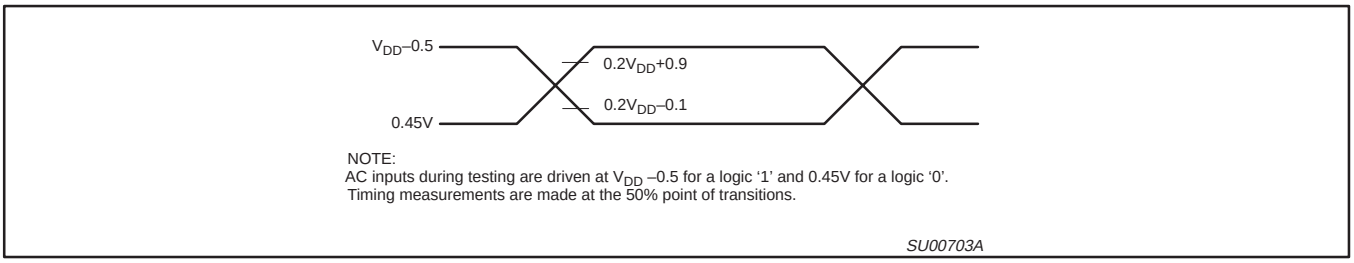


Figure 27. AC Testing Input/Output

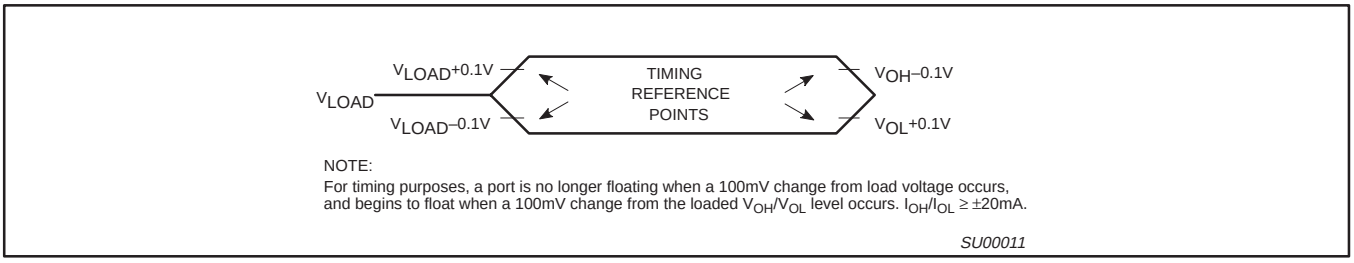


Figure 28. Float Waveform

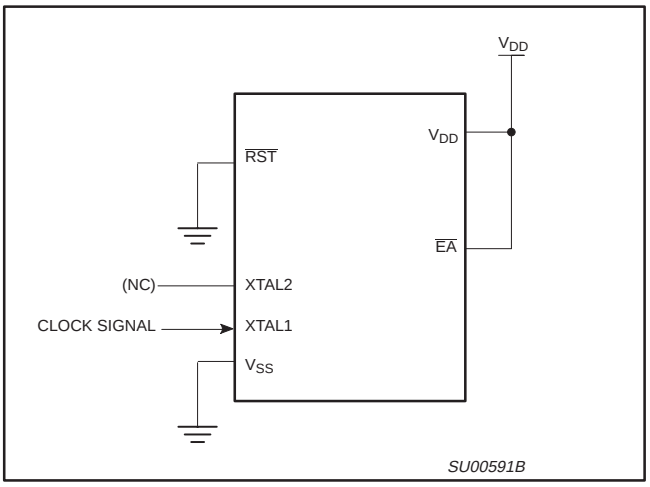


Figure 29. I_{DD} Test Condition, Active Mode
All other pins are disconnected

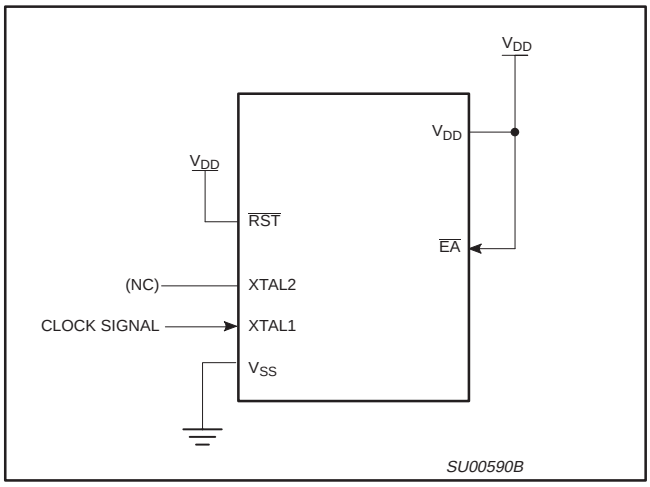


Figure 30. I_{DD} Test Condition, Idle Mode
All other pins are disconnected

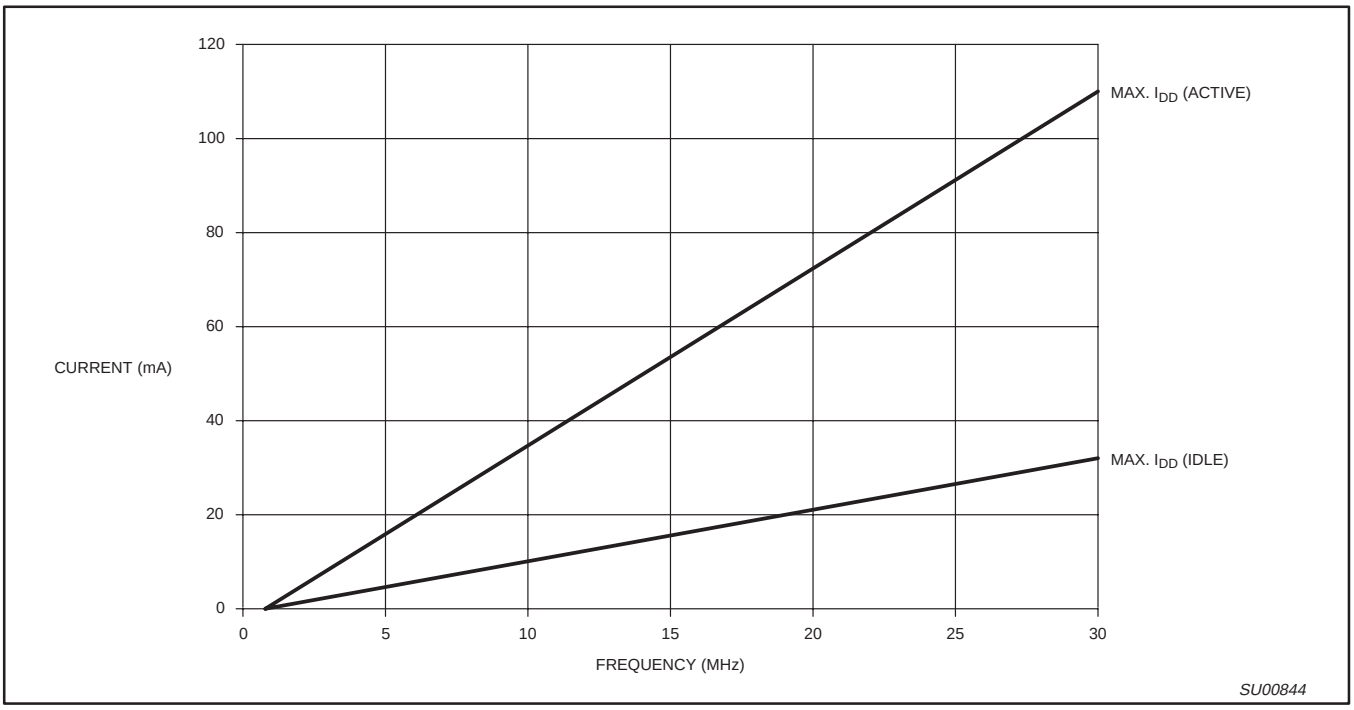


Figure 31. I_{DD} vs. Frequency
Valid only within frequency specification of the device under test.

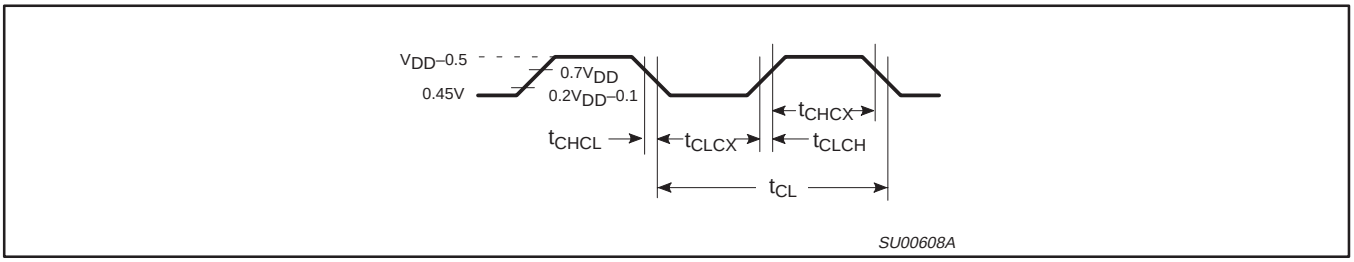


Figure 32. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes
t_{CLCH} = t_{CHCL} = 5ns

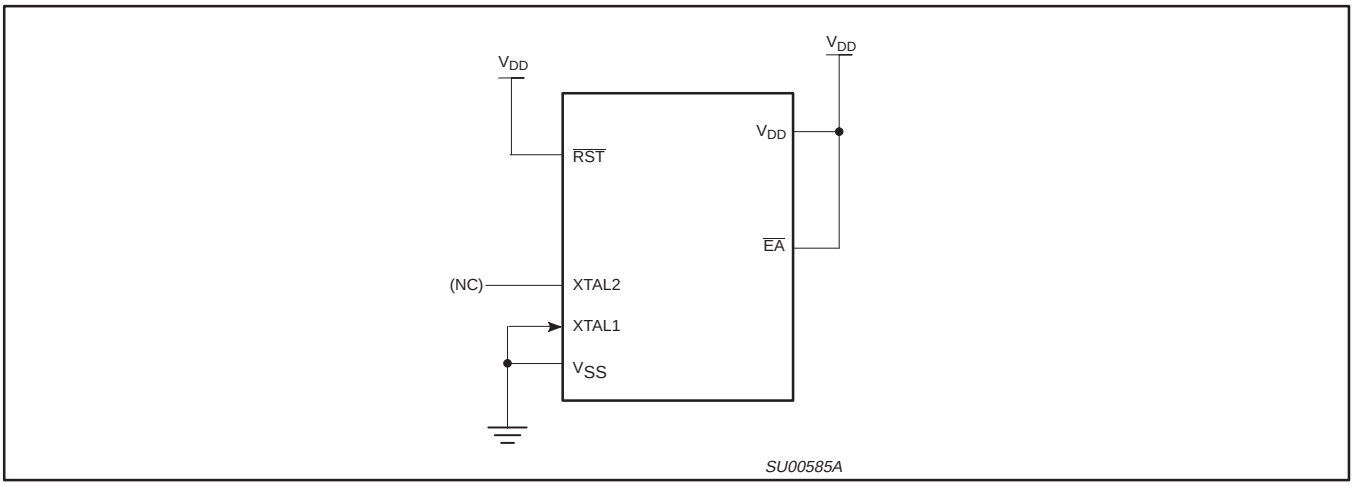
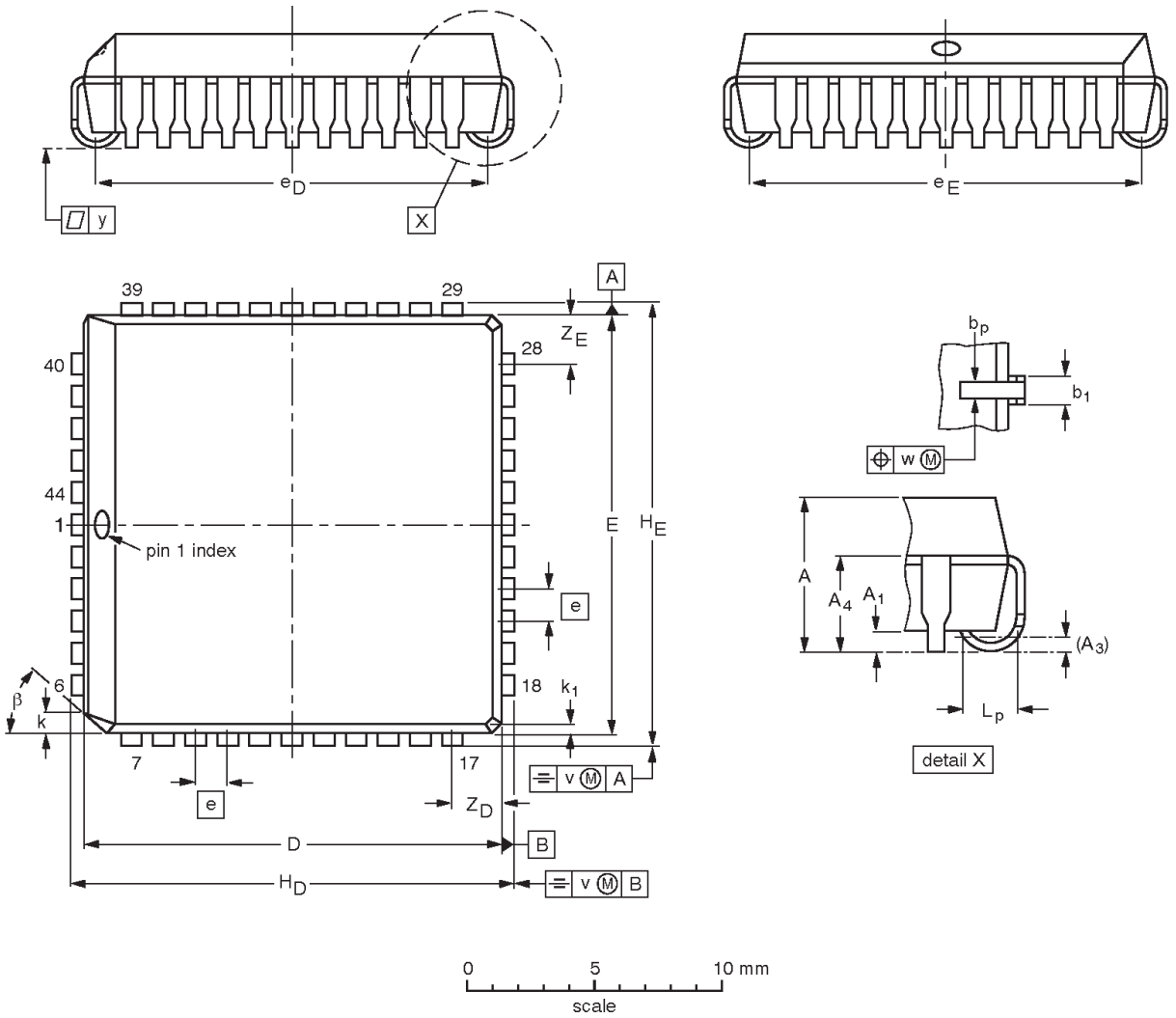


Figure 33. I_{DD} Test Condition, Power Down Mode
All other pins are disconnected. V_{DD}=2 V to 5.5 V

PLCC44: plastic leaded chip carrier; 44 leads **SOT187-2**



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

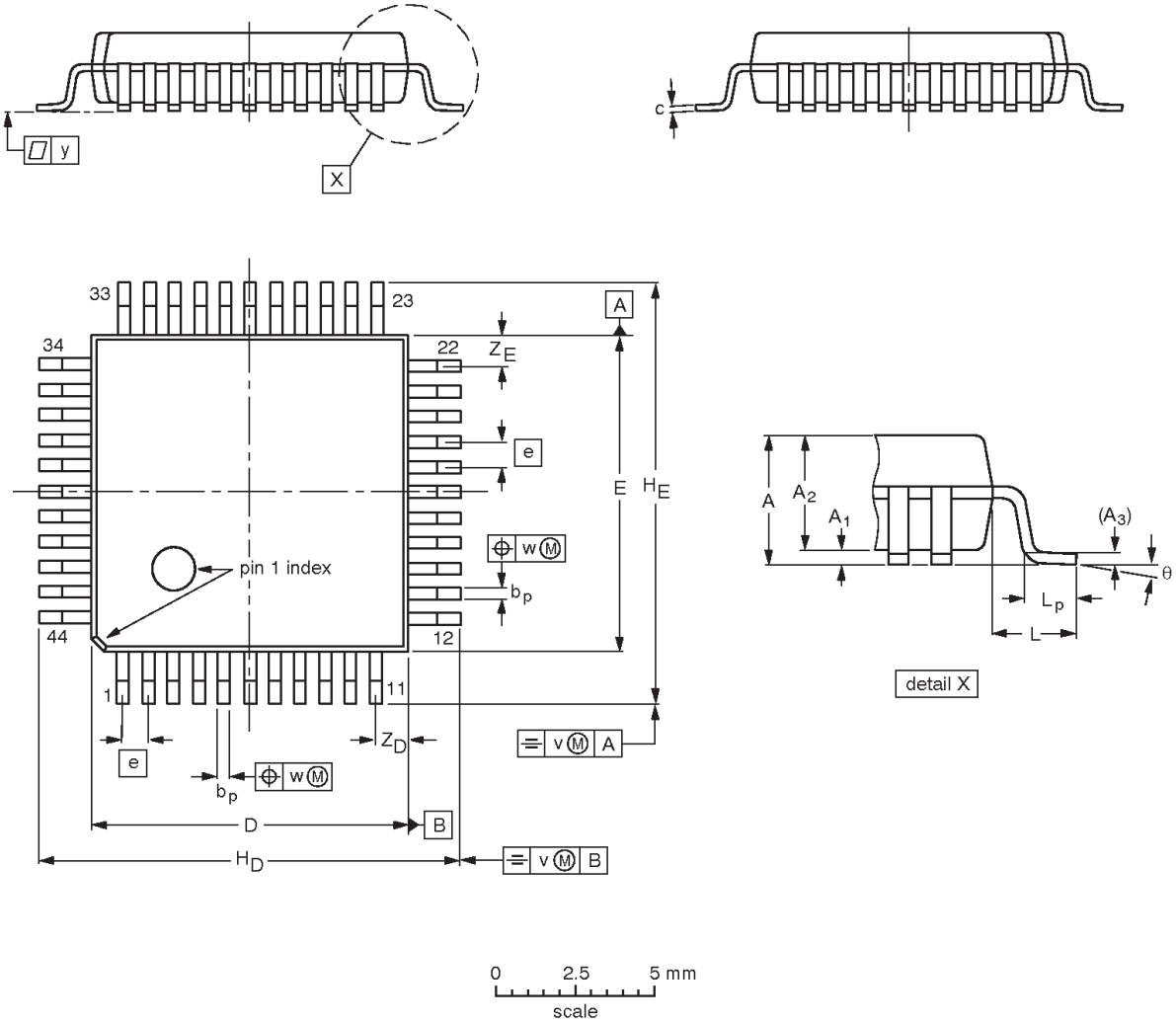
UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	D(1)	E(1)	e	e_D	e_E	H_D	H_E	k	k_1 max.	L_p	v	w	y	Z_D(1) max.	Z_E(1) max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.630 0.590	0.630 0.590	0.695 0.685	0.695 0.685	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT187-2	112E10	MO-047				~~97-12-16~~ 99-12-27

LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm SOT389-1



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.60	0.15 0.05	1.45 1.35	0.25	0.45 0.30	0.20 0.12	10.10 9.90	10.10 9.90	0.80	12.15 11.85	12.15 11.85	1.0	0.75 0.45	0.20	0.20	0.10	1.14 0.85	1.14 0.85	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT389-1	136E08	MS-026				~~99-12-17~~ 00-01-19

NOTES

XA 16-bit microcontroller family 64K Flash/2K RAM, watchdog, 2 UARTs

XA-G49

Data sheet status

Data sheet status	Product status	Definition [1]
Objective specification	Development	This data sheet contains the design target or goal specifications for product development. Specification may change in any manner without notice.
Preliminary specification	Qualification	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product specification	Production	This data sheet contains final specifications. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

[1] Please consult the most recently issued datasheet before initiating or completing a design.

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

© Copyright Philips Electronics North America Corporation 2000
All rights reserved. Printed in U.S.A.

Date of release: 06-00

Document order number: 9397 750 07265

Let's make things better.

Philips Semiconductors

PHILIPS