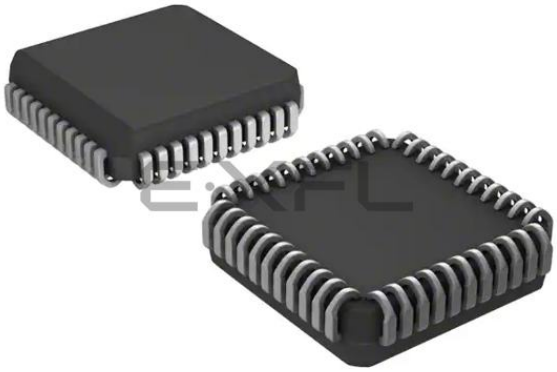


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	XA
Core Size	16-Bit
Speed	30MHz
Connectivity	UART/USART
Peripherals	PWM, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/pxag49kfa-00-512

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.

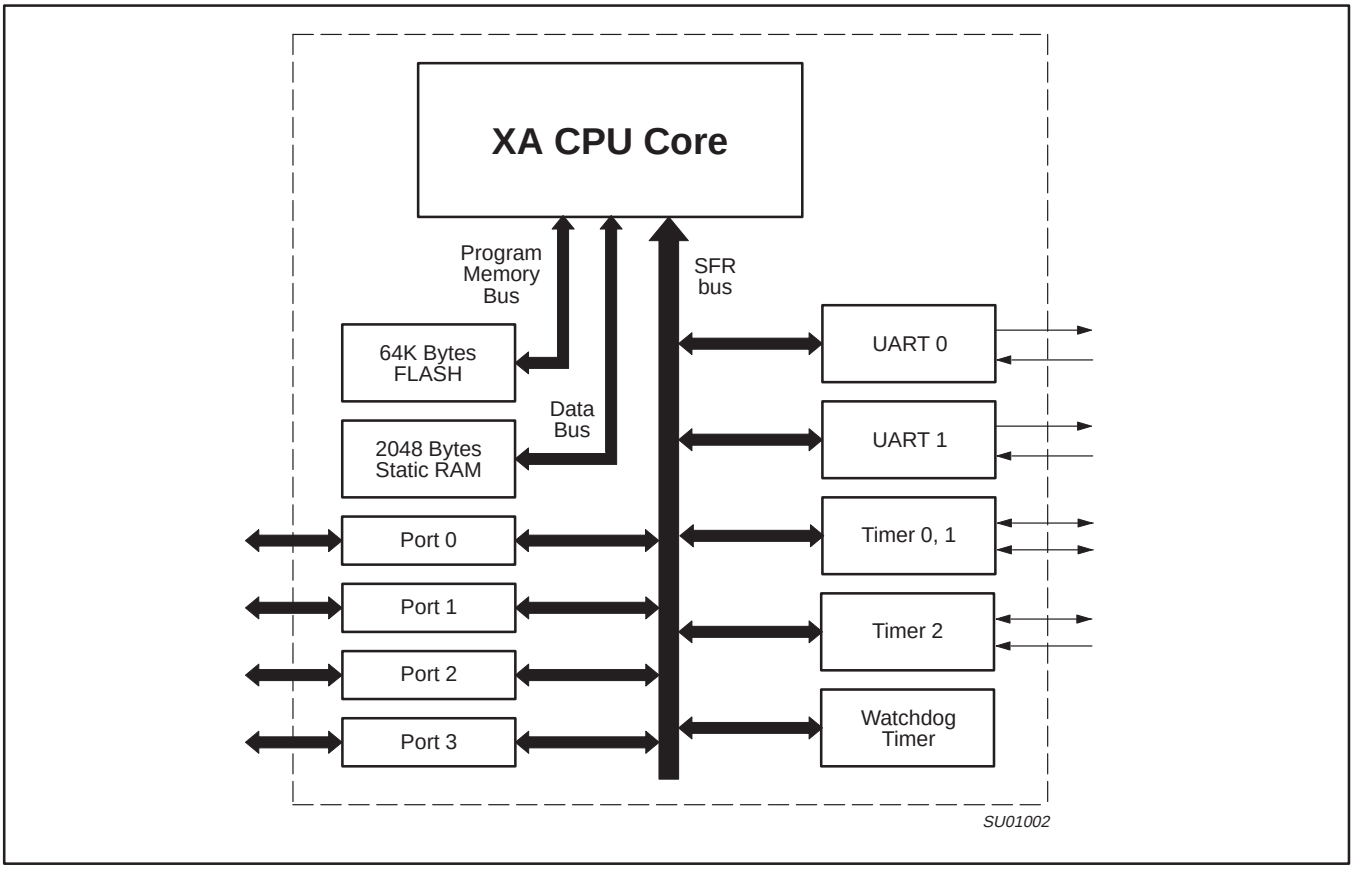
The XA-G49 contains 64 kbytes of Flash program memory, and provides three general purpose timers/counters, a watchdog timer, dual UARTs, and four general purpose I/O ports with programmable output configurations.

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

- 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

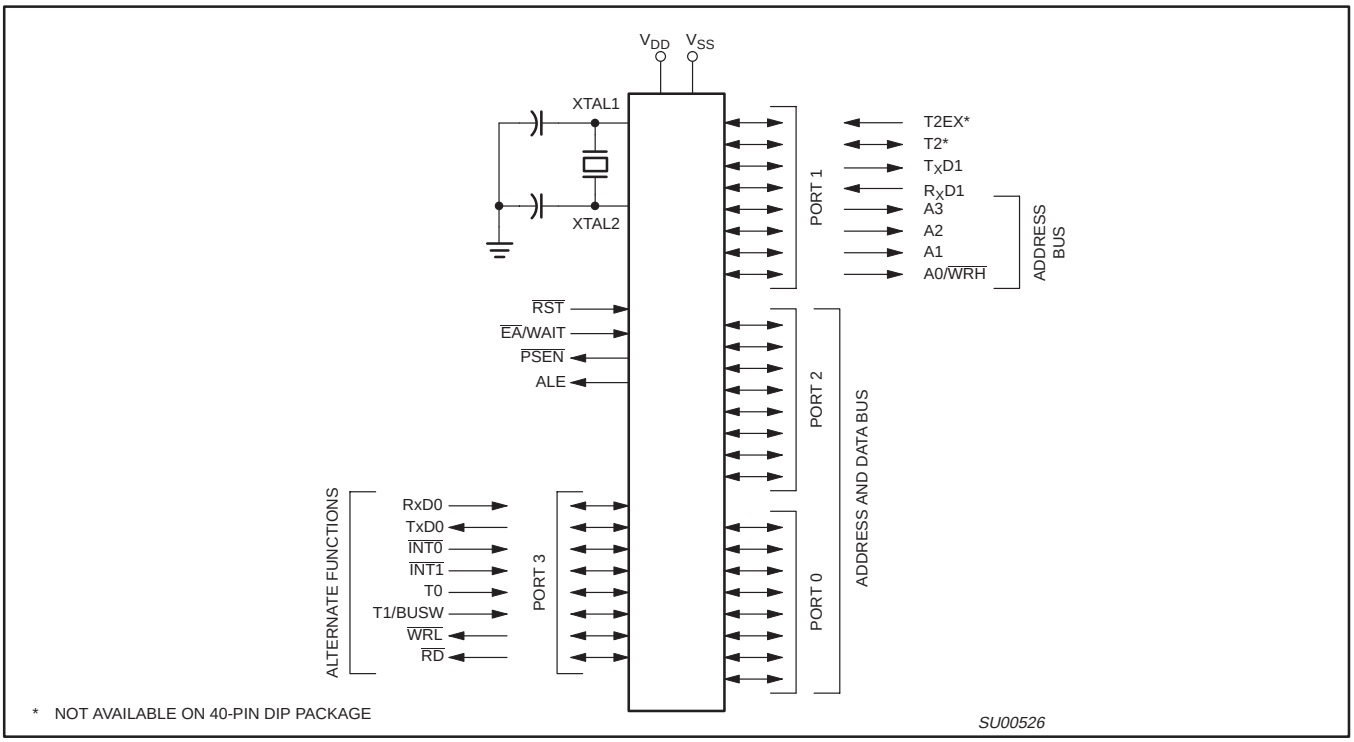
BLOCK DIAGRAM



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
PXAG49KBBD	0 to +70 44-pin Plastic Low Profile Quad Flat Package	30	SOT389-1

LOGIC SYMBOL



PIN DESCRIPTIONS

MNEMONIC	PIN. NO.		TYPE	NAME AND FUNCTION
	LCC	LQFP		
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.

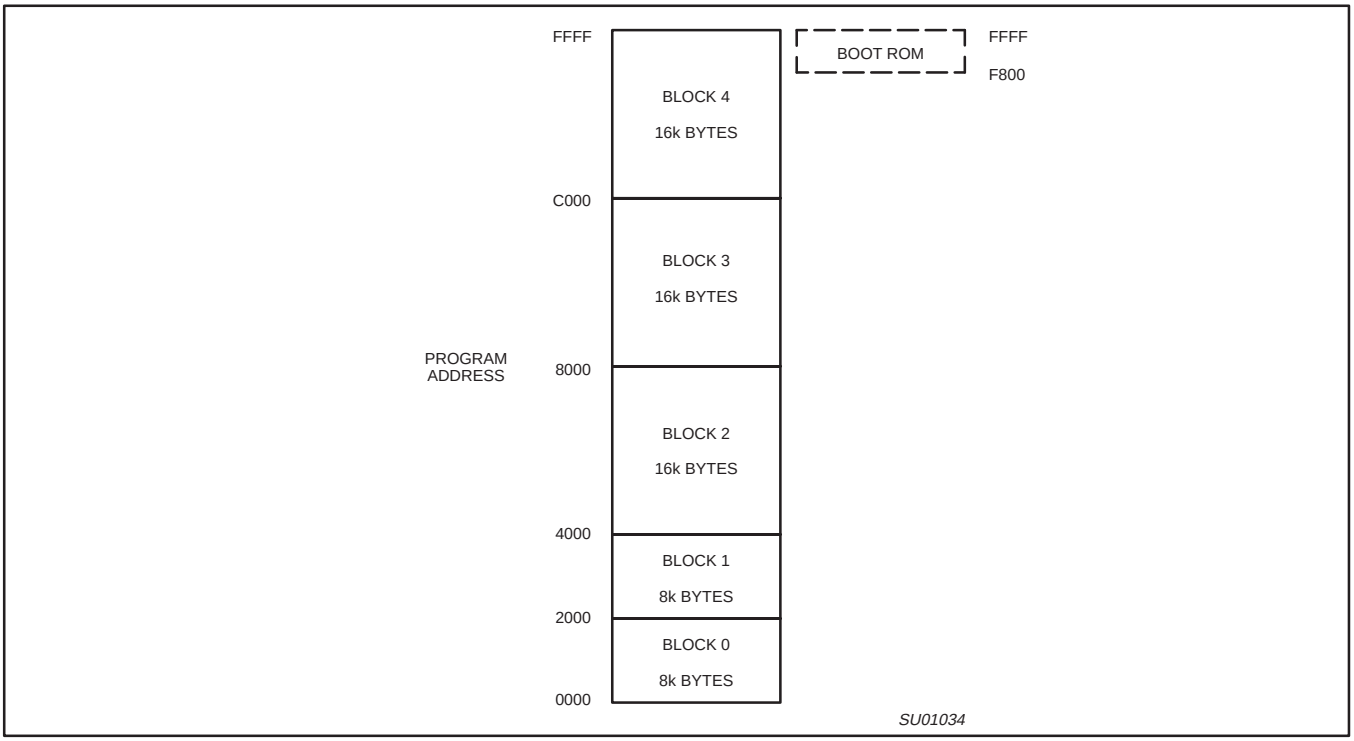


Figure 3. Flash Memory Configuration

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.

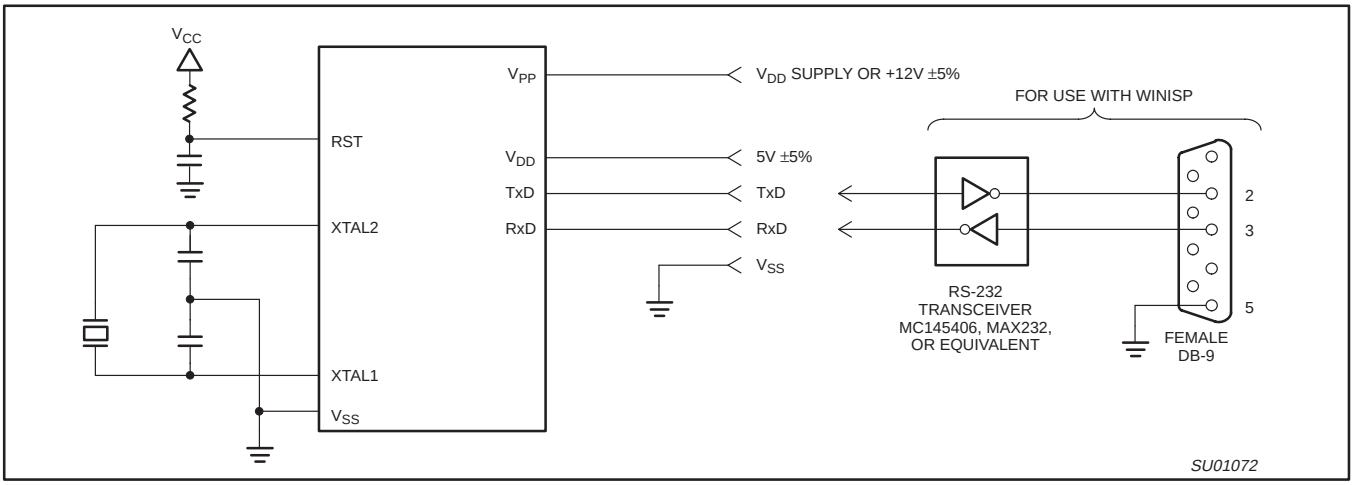


Figure 4. In-System Programming with a Minimum of Pins

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.

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`

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

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

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.

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} = 4.5 V to 5.5 V; T_{amb} = 0 to +70 °C, unless otherwise specified.
V_{DD} = 4.75 V to 5.25 V; T_{amb} = –40 to +85 °C, 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} = V_{DDmin}	2.4			V
V_{OH2}	Output high voltage, ports P0–3, ALE, $\overline{PSEN}$[2]	I_{OH} = 3.2mA, V_{DD} = V_{DDmin}	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} = 4.5 V to 5.5 V; T_{amb} = 0 to +70 °C for commercial; V_{DD} = 4.75 V to 5.25 V, –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.

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.

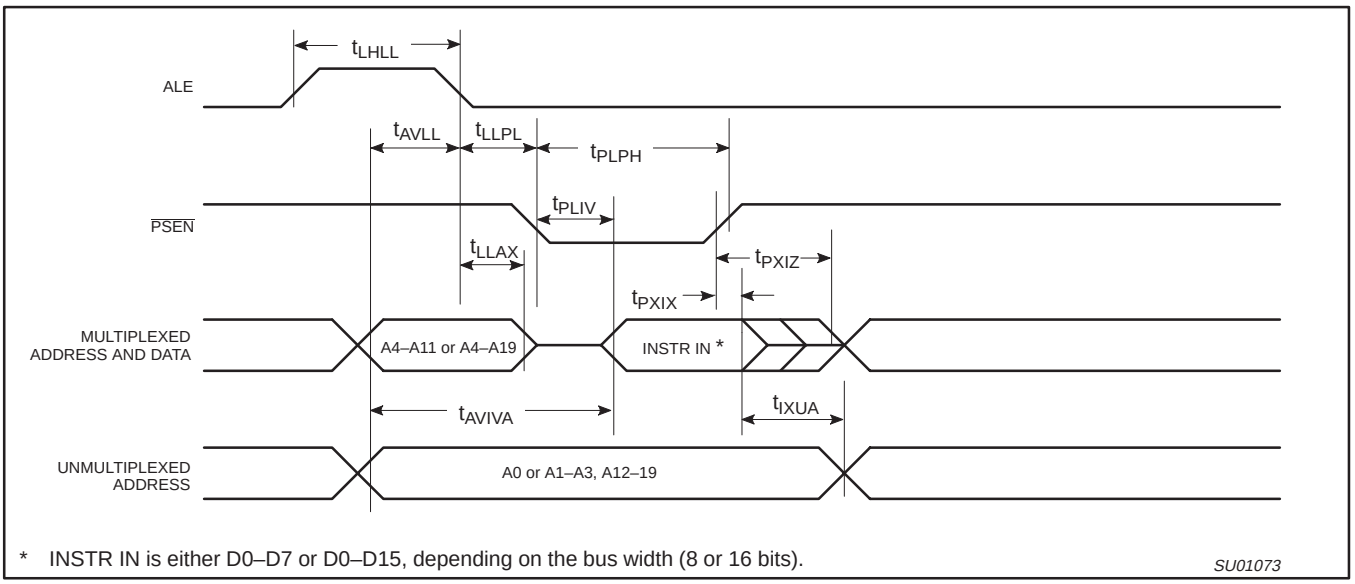


Figure 20. External Program Memory Read Cycle (ALE Cycle)

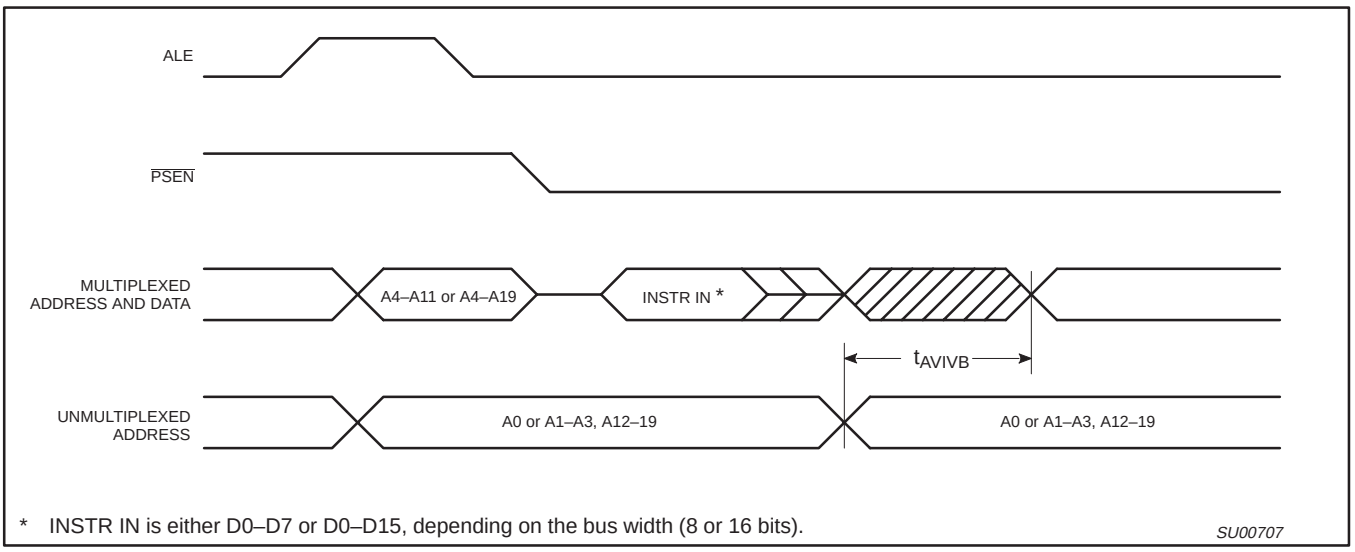


Figure 21. External Program Memory Read Cycle (Non-ALE Cycle)

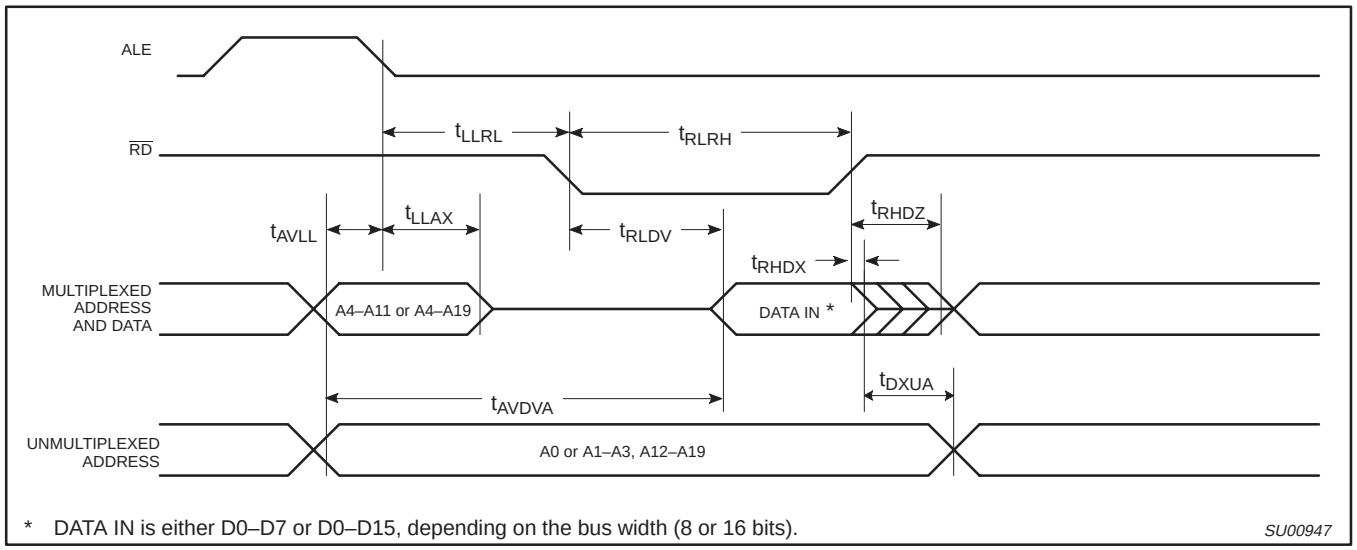


Figure 22. External Data Memory Read Cycle (ALE Cycle)

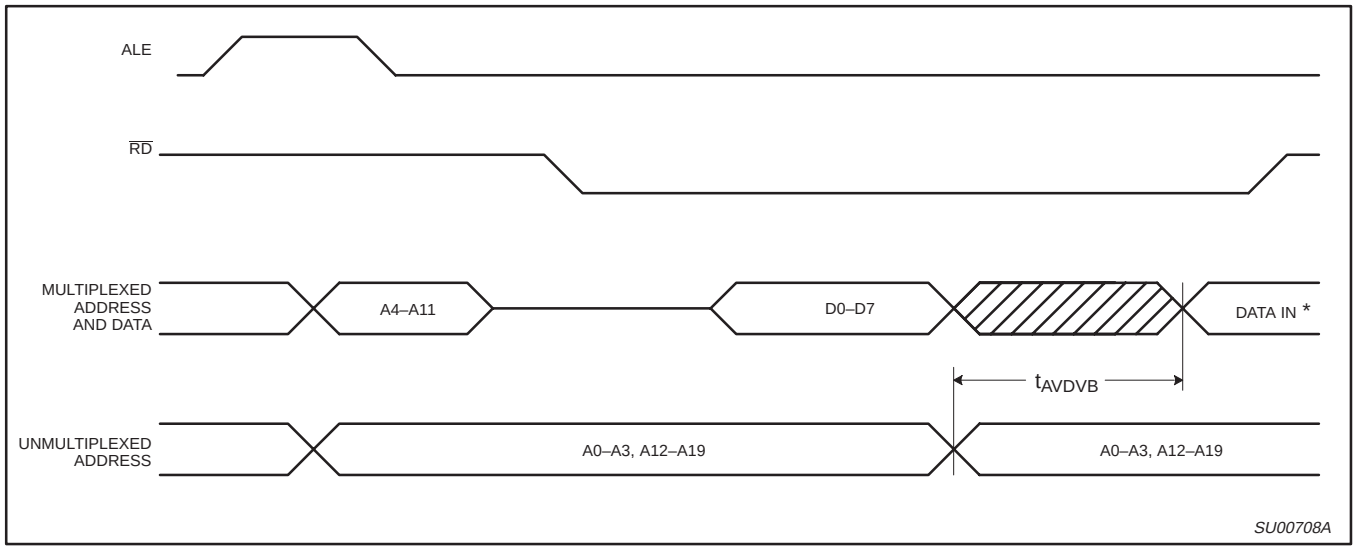


Figure 23. External Data Memory Read Cycle (Non-ALE Cycle)

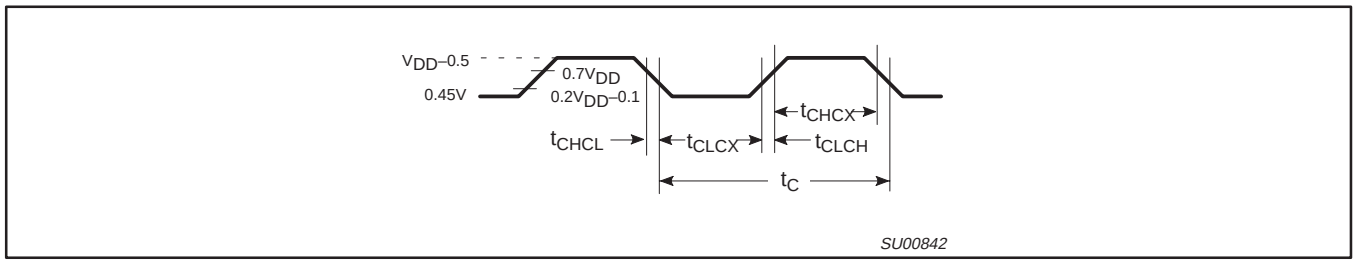


Figure 26. External Clock Drive

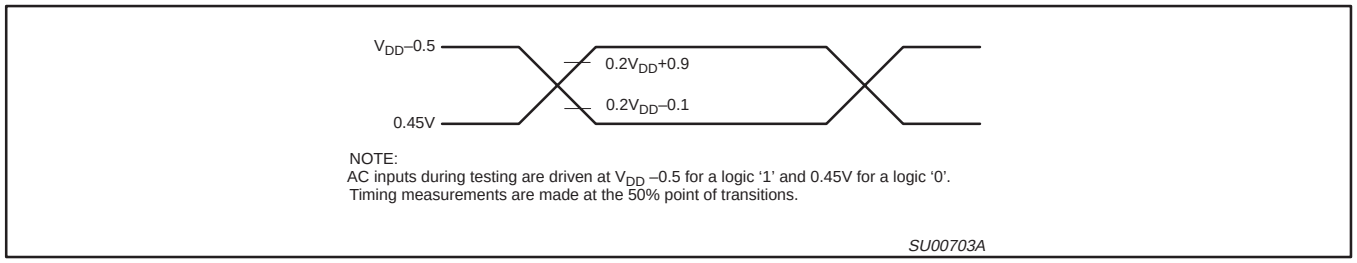


Figure 27. AC Testing Input/Output

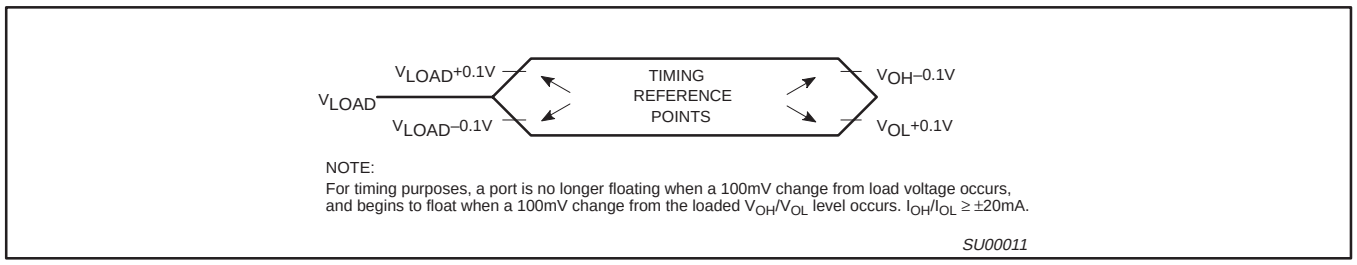


Figure 28. Float Waveform

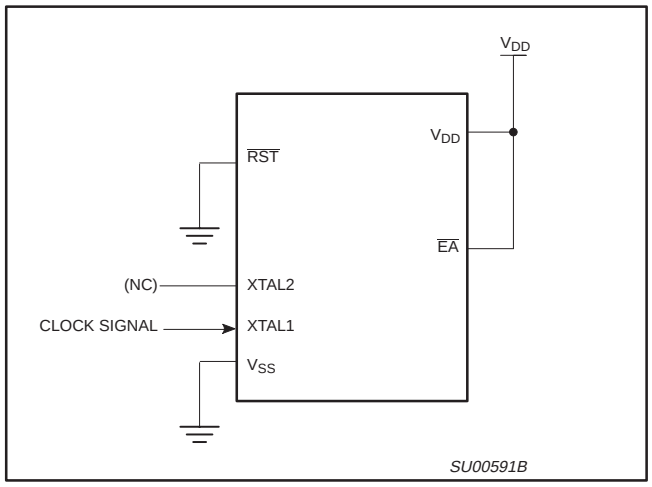


Figure 29. I_{DD} Test Condition, Active Mode
All other pins are disconnected

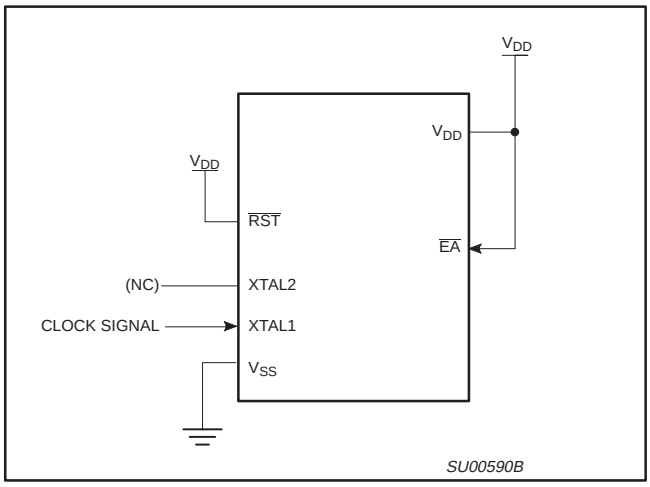


Figure 30. I_{DD} Test Condition, Idle Mode
All other pins are disconnected

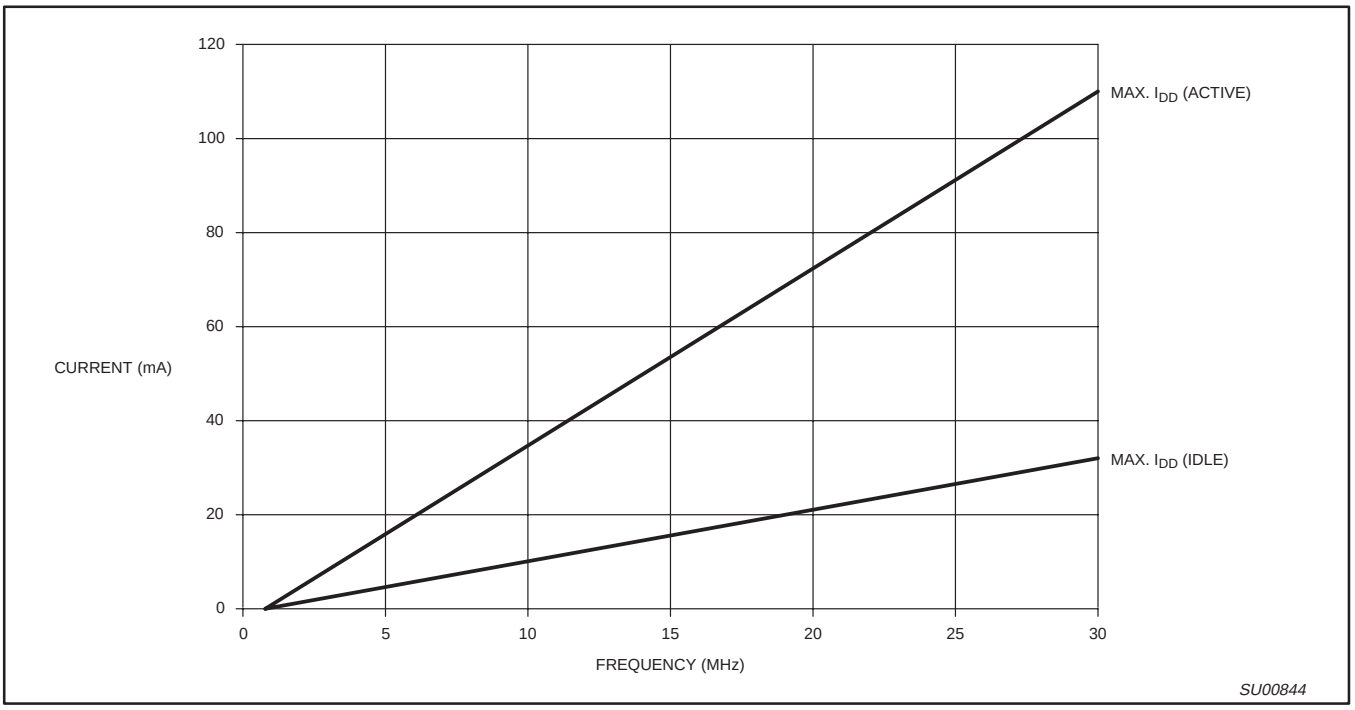


Figure 31. I_{DD} vs. Frequency
Valid only within frequency specification of the device under test.

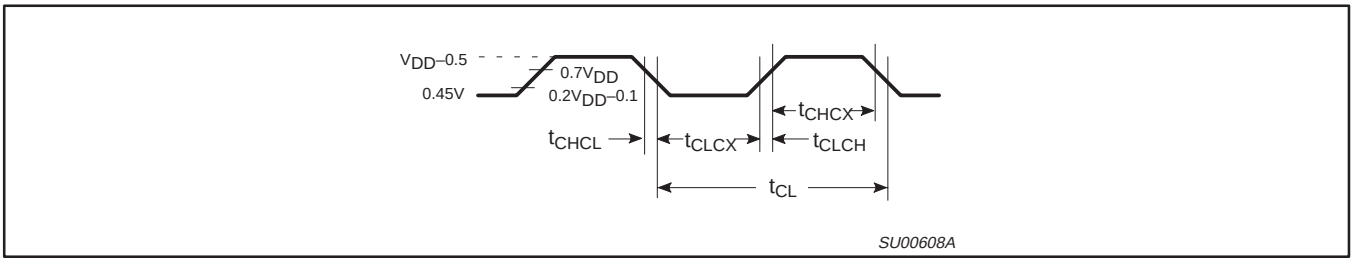


Figure 32. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes
t_{CLCH} = t_{CHCL} = 5ns

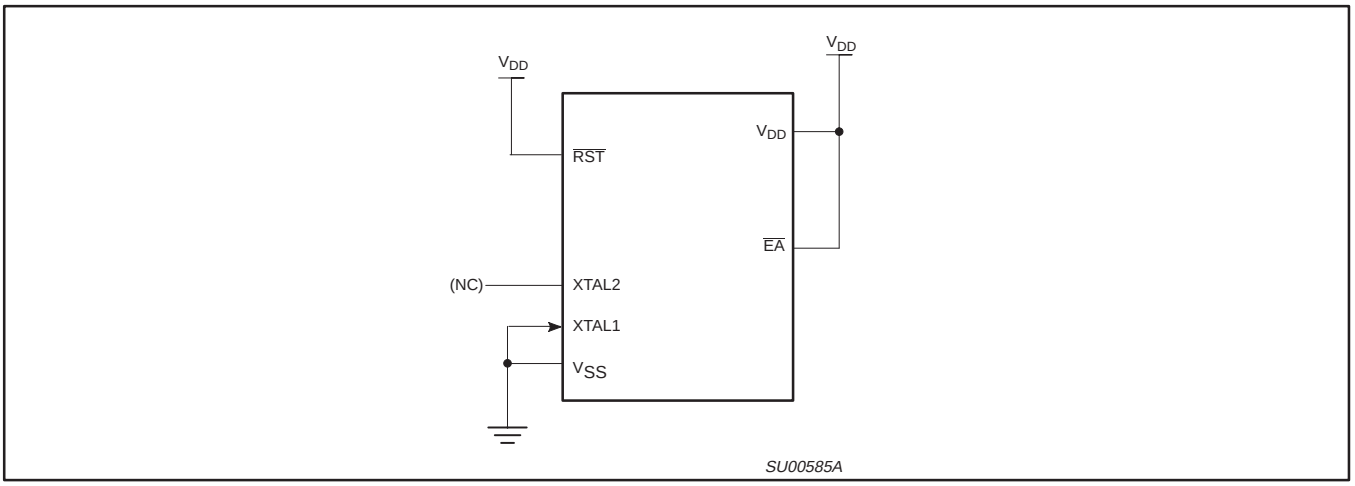
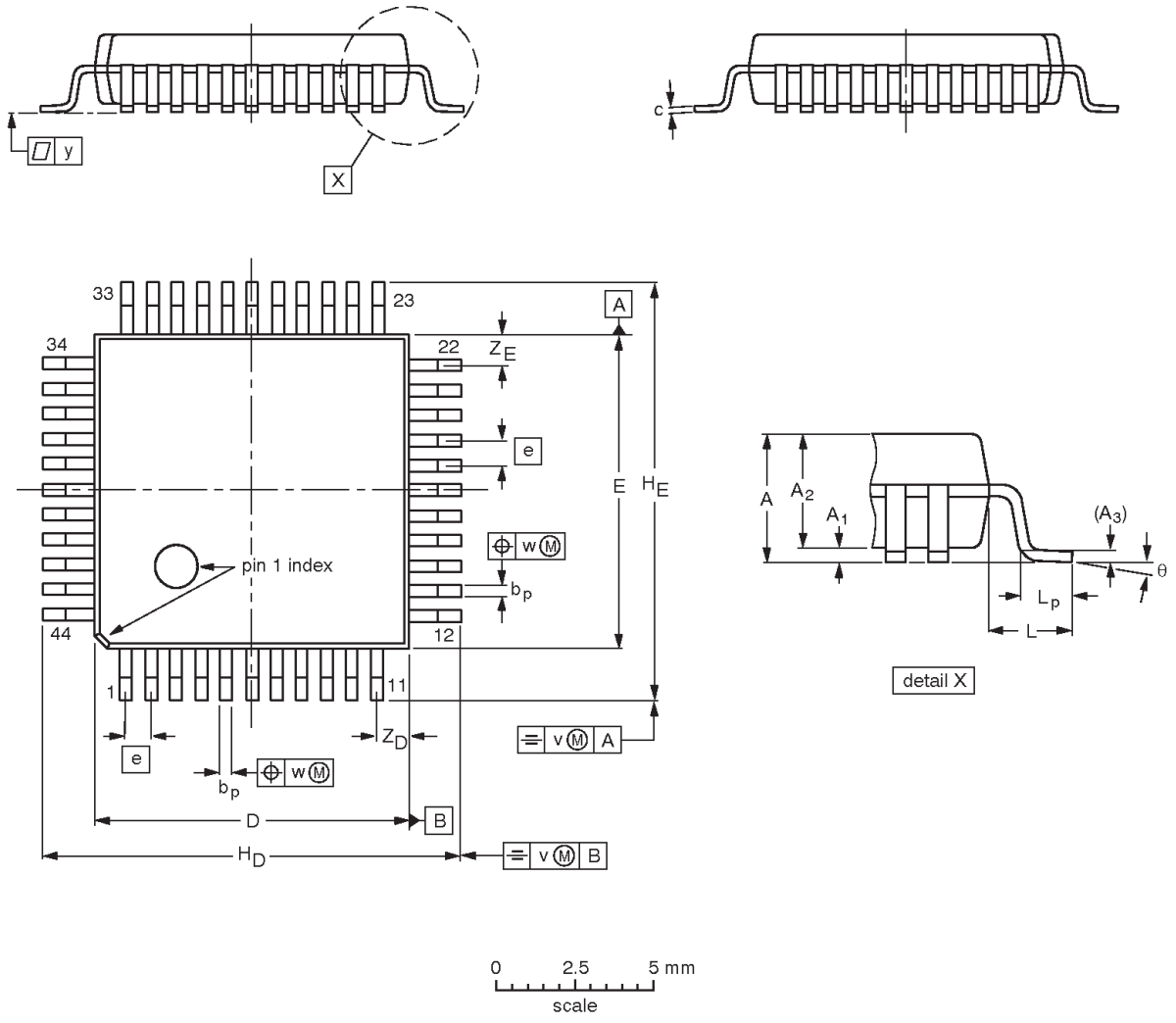


Figure 33. I_{DD} Test Condition, Power Down Mode
All other pins are disconnected. V_{DD}=2 V to 5.5 V

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^o 0^o

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

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

XA-G49

Data sheet status

Data sheet status [1]	Product status [2]	Definitions
Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Changes will be communicated according to the Customer Product/Process Change Notification (CPCN) procedure SNW-SQ-650A.

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

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

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 60134). 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

Date of release: 06-01

Document order number: 9397 750 08524

Let's make things better.

Philips Semiconductors

PHILIPS