

Welcome to [E-XFL.COM](http://E-XFL.COM)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

Details	
Product Status	Active
Core Processor	ACE1001
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	6
Program Memory Size	1KB (1K x 8)
Program Memory Type	EEPROM
EEPROM Size	64 x 8
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-DIP
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=ace1001en">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=ace1001en</a>

**ACE1001(L) DC Electrical Characteristics** $V_{CC} = 1.8/2.2$  to  $5.5V$ 

All measurements valid for ambient operating temperature unless otherwise stated.

Symbol	Parameter	Conditions	MIN	TYP	MAX	Units
$I_{CC}^3$	Supply Current – no data EEPROM write in progress	1.8V		0.2	0.5	mA
		2.2V		0.4	1.0	mA
		2.7V		0.7	1.2	mA
		3.3V		1.2	2.0	mA
		5.5V		3.7	5.5	mA
$I_{CCH}$	HALT Mode current	3.3V @ +25°C		10	100	nA
		3.3V @ -40°C to +85°C			1000	nA
		5.5V @ +25°C		250	1000	nA
5.5V @ +125°C		3000	nA			
$I_{CCL}^4$	IDLE Mode Current	3.3V		120	200	$\mu A$
		5.5V		140	300	$\mu A$
$V_{CCW}$	EEPROM Write Voltage	Code EEPROM in Programming Mode	4.5	5.0	5.5	V
		Data EEPROM in Operating Mode	2.4		5.5	V
$S_{VCC}$	Power Supply Slope		1 $\mu s/V$		10ms/V	
$V_{IL}$	Input Low with Schmitt Trigger Buffer	$V_{CC} = 1.8V$			0.15 $V_{CC}$	V
		$V_{CC} = 2.2 - 5.5V$			0.20 $V_{CC}$	V
$V_{IH}$	Input High with Schmitt Trigger Buffer	$V_{CC} \leq 2.2V$	0.9 $V_{CC}$			V
		$V_{CC} > 2.2V$	0.8 $V_{CC}$			V
$I_{IP}$	Input Pull-up Current	$V_{CC} = 5.5V, V_{IN} = 0V$	30	65	350	$\mu A$
$I_{TL}$	TRI-STATE Leakage	$V_{CC} = 5.5V$		2	200	nA
$V_{OL}$	Output Low Voltage	$V_{CC} = 1.8 - 2.2V$				
	G0, G1, G2, G4	0.8 mA sink			0.2 $V_{CC}$	V
	G5	1.0 mA sink			0.2 $V_{CC}$	V
	Output Low Voltage	$V_{CC} = 2.2V - 3.3V$				
	G0, G1, G2, G4	3.0 mA sink			0.2 $V_{CC}$	V
	G5	5.0 mA sink			0.2 $V_{CC}$	V
	Output Low Voltage	$V_{CC} = 3.3V - 5.5V$				
	G0, G1, G2, G4	5.0 mA sink			0.2 $V_{CC}$	V
G5	10.0 mA sink			0.2 $V_{CC}$	V	
$V_{OH}$	Output High Voltage	$V_{CC} = 1.8 - 2.2V$				
	G0, G1, G2, G4	0.1 mA source	0.8 $V_{CC}$			V
	G5	0.2 mA source	0.8 $V_{CC}$			V
	Output High Voltage	$V_{CC} = 3.3V - 5.5V$				
	G0, G1, G2, G4	0.4 mA source	0.8 $V_{CC}$			V
	G5	0.8 mA source	0.8 $V_{CC}$			V
	Output High Voltage	$V_{CC} = 3.3V - 5.5V$				
	G0, G1, G2, G4	0.4 mA source	0.8 $V_{CC}$			V
G5	1.0 mA source	0.8 $V_{CC}$			V	

<sup>3</sup>  $I_{CC}$  active current is dependent on the program code.<sup>4</sup> Based on a continuous IDLE looping program.

**ACE1001(L) AC Electrical Characteristics** $V_{CC} = 1.8/2.2$  to 5.5V

All measurements valid for ambient operating temperature unless otherwise stated.

Parameter	Conditions	MIN	TYP	MAX	Units
Instruction cycle time from internal clock - setpoint	5.0V at +25°C	0.96	1.0	1.04	μs
Internal clock frequency variation	2.4V to 5.5V at constant temperature	-5		+5	%
	2.4V to 5.5V at full temperature range	-10		+10	%
Crystal oscillator frequency	(Note 5)			4	MHz
External clock frequency	(Note 5)			4	MHz
EEPROM write time			3	10	ms
Internal clock start up time	(Note 6)			2	ms
Oscillator start up time	(Note 6)			2400	cycles

<sup>5</sup> The maximum permissible frequency is guaranteed by design but not 100% tested.<sup>6</sup> The parameter is guaranteed by design but not 100% tested.**ACE1001(L) Electrical Characteristics for programming**

All data following is valid between 4.5V and 5.5V at ambient temperature. The following characteristics are guaranteed by design but are not 100% tested. See "EEPROM write time" in the AC Electrical Characteristics for definition of the programming ready time.

Parameter	Description	MIN	MAX	Units
$t_{HI}$	CLOCK high time	500	DC	ns
$t_{LO}$	CLOCK low time	500	DC	ns
$t_{DIS}$	SHIFT_IN setup time	100		ns
$t_{DIH}$	SHIFT_IN hold time	100		ns
$t_{DOS}$	SHIFT_OUT setup time	100		ns
$t_{DOH}$	SHIFT_OUT hold time	900		ns
$t_{SV1}, t_{SV2}$	LOAD supervoltage timing	50		μs
$t_{LOAD1}, t_{LOAD2}, t_{LOAD3}, t_{LOAD4}$	LOAD timing	5		μs
$V_{SUPERVOLTAGE}$	Supervoltage level	11.5	12.5	V

**ACE1001(L) Low Battery Detect (LBD) Characteristics** $V_{CC} = 1.8/2.2$  to 5.5V

Parameter	Conditions	MIN	TYP	MAX	Units
LBD Voltage Threshold Variation	+25°C	-7		+7	%
	0°C to +70°C	-12		+12	%
	-40°C to +85°C	-16		+16	%

**ACE1001 Brown-out Reset (BOR) Characteristics** $V_{CC} = 2.2$  to 5.5V

Parameter	Conditions	MIN	TYP	MAX	Units
BOR Voltage Threshold Variation (BLSEL = 1)	-40°C to +85°C	1.93	2.25	2.58	V

**ACE1001L Brown-out Reset (BOR) Characteristics** $V_{CC} = 1.8$  to 5.5V

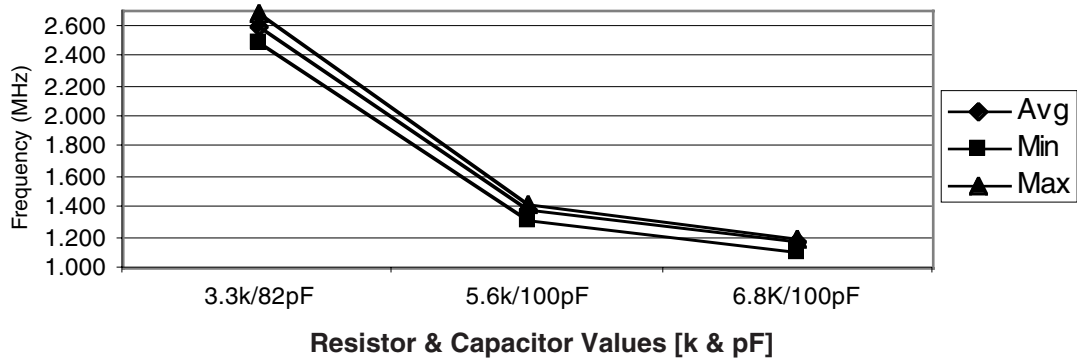
Parameter	Conditions	MIN	TYP	MAX	Units
BOR Voltage Threshold Variation (BLSEL = 0)	0°C to +70°C	1.76	1.95	2.20	V

### 3.0 AC & DC Electrical Characteristic Graphs

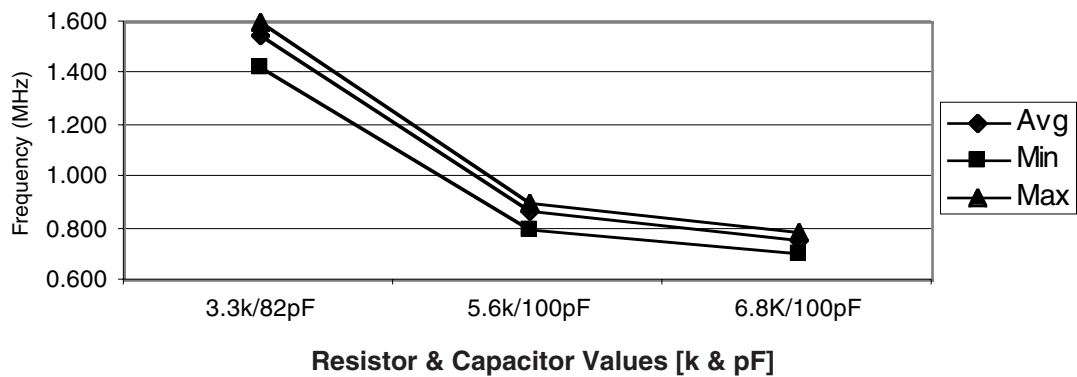
The graphs in this section are for design guidance and are based on preliminary test data.

**Figure 4: RC Oscillator Frequency vs. Temperature**

(a)  $V_{CC} = 5.0V$



(b)  $V_{CC} = 2.5V$



**Figure 5: Internal Oscillator Frequency**

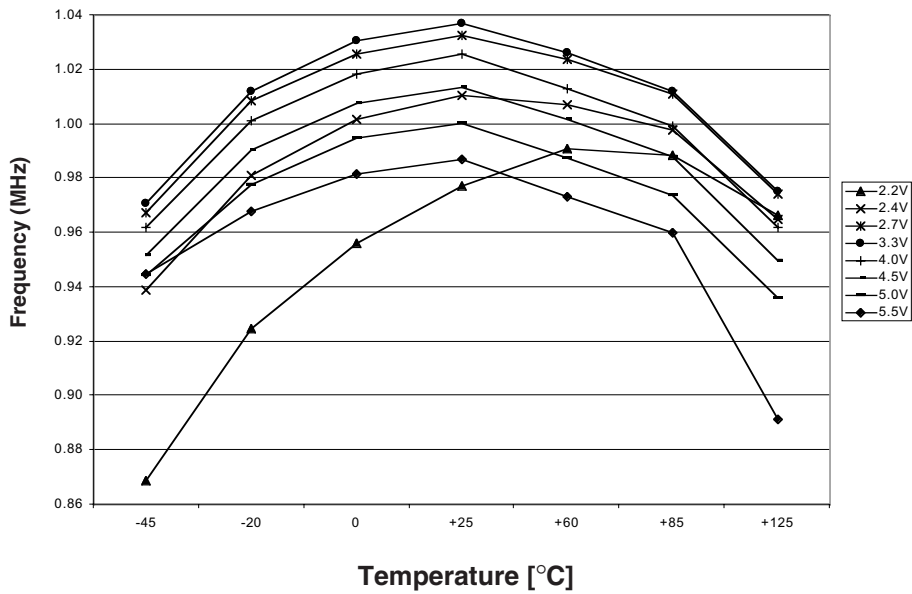


Figure 7: I<sub>CC</sub> Active Current

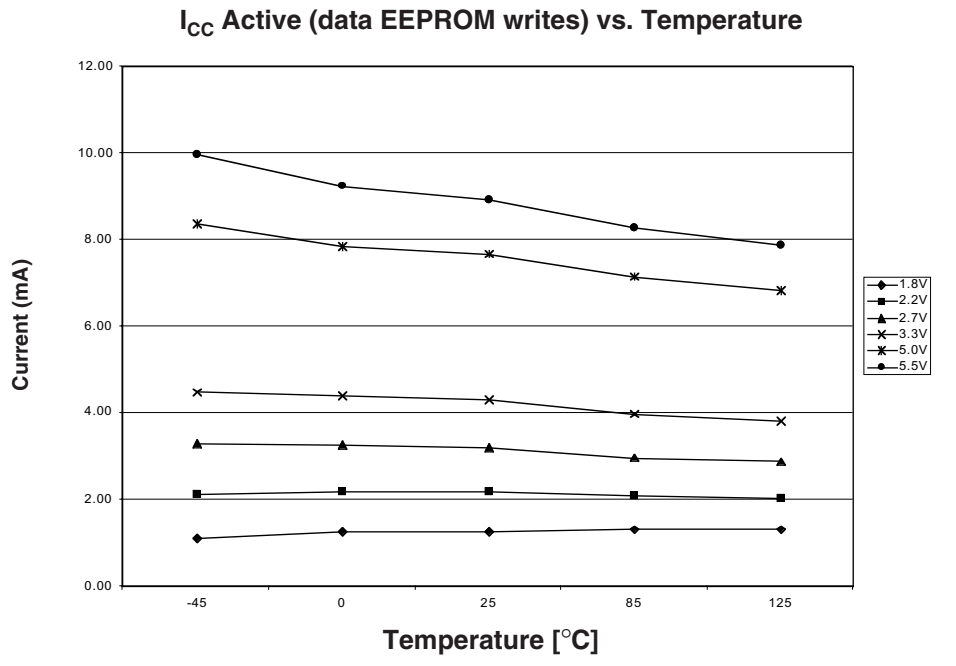
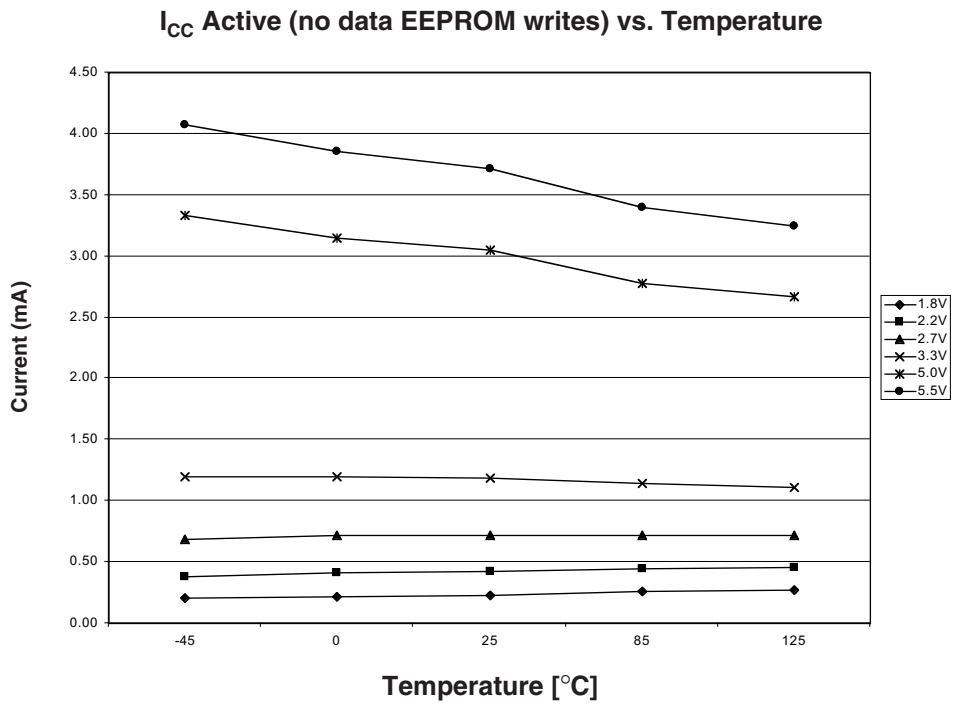


Figure 8: HALT Mode Currents

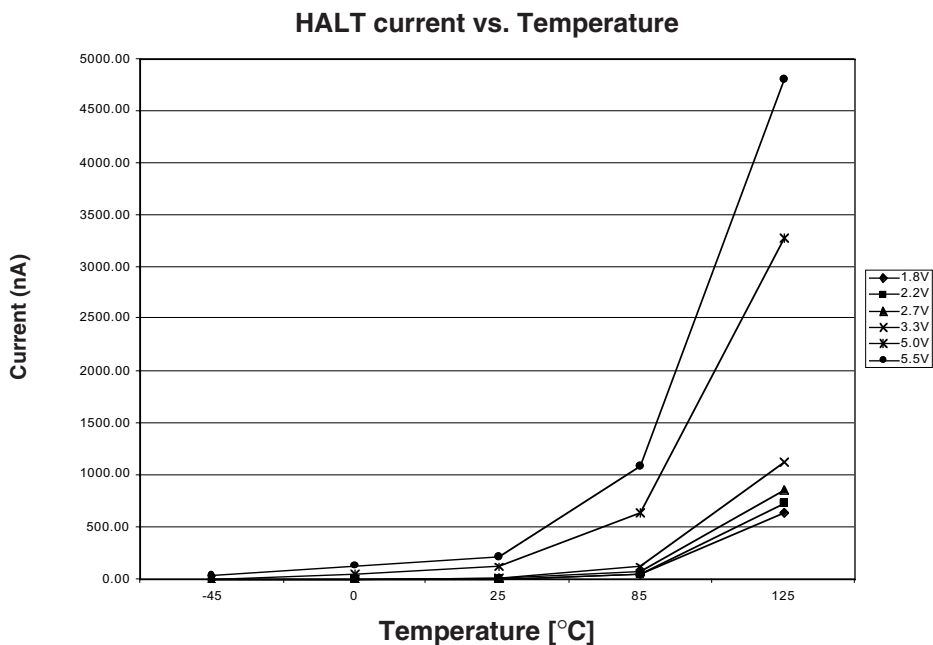


Figure 9: IDLE Mode Current

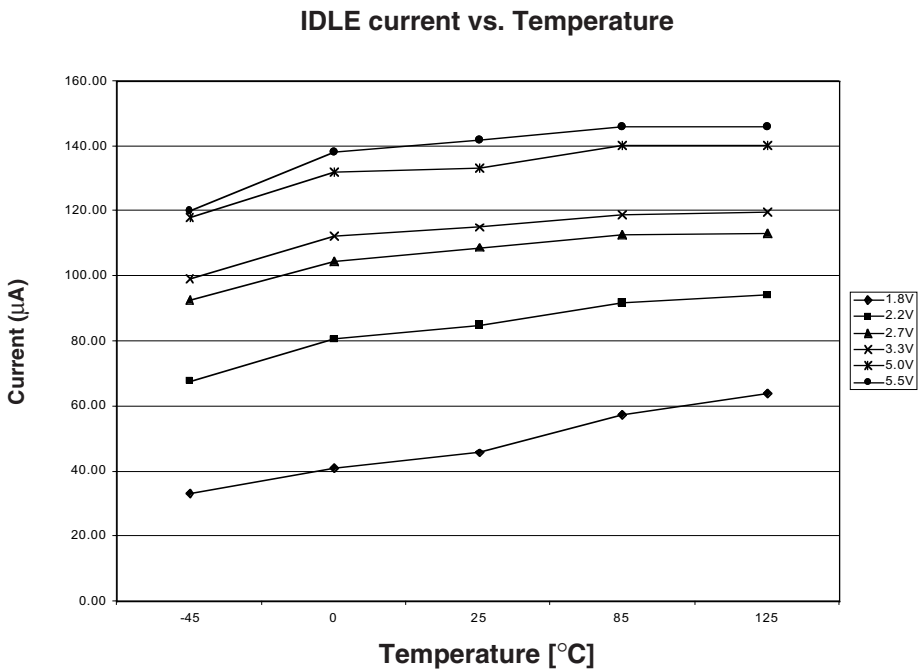
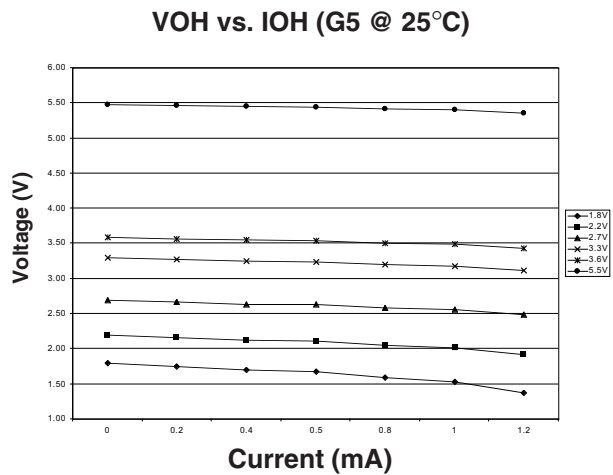
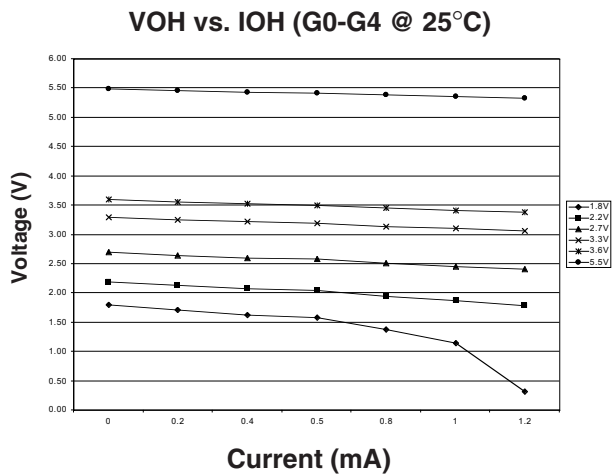
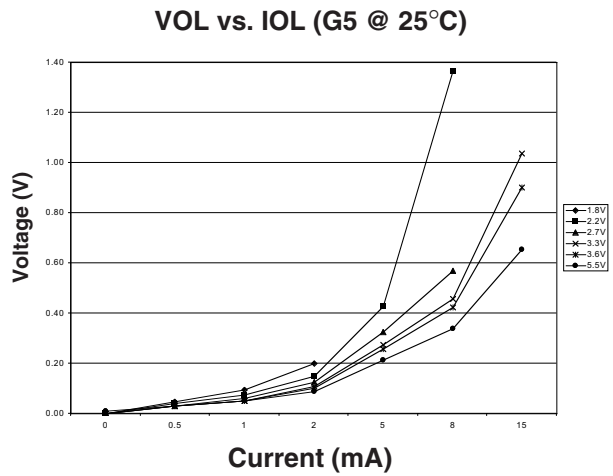
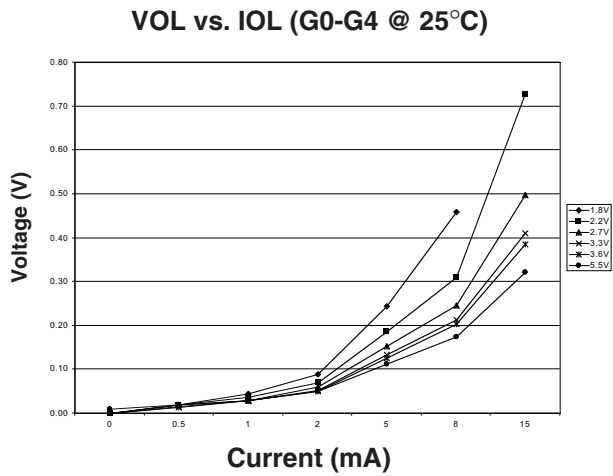


Figure 10: VOL/VOH





### 4.0 Arithmetic Controller Core

The ACEx microcontroller core is specifically designed for low cost applications involving bit manipulation, shifting and arithmetic operations. It is based on a modified Harvard architecture meaning peripheral, I/O, and RAM locations are addressed separately from instruction data.

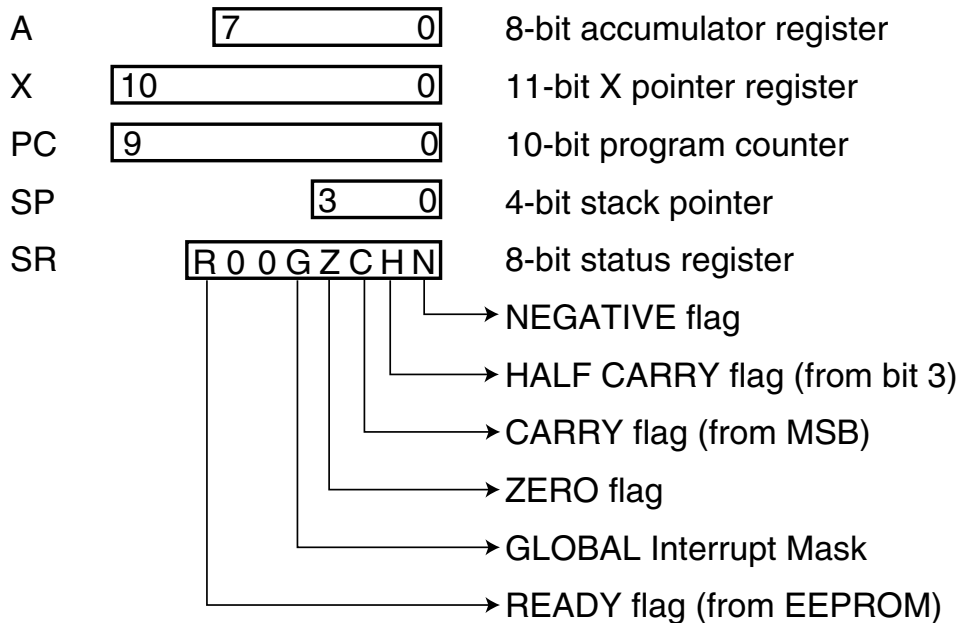
The core differs from the traditional Harvard architecture by aligning the data and instruction memory sequentially. This allows

the X-pointer (11-bits) to point to any memory location in either segment of the memory map. This modification improves the overall code efficiency of the core and takes advantage of the flexibility found on Von Neumann style machines.

### 4.1 CPU Registers

The ACEx microcontroller has five general-purpose registers. These registers are the Accumulator (A), X-Pointer (X), Program Counter (PC), Stack Pointer (SP), and Status Register (SR). The X, SP, and SR registers are all memory-mapped.

**Figure 11: Programming Model**



### 4.1.1 Accumulator (A)

The Accumulator is a general-purpose 8-bit register that is used to hold data and results of arithmetic calculations or data manipulations.

### 4.1.2 X-Pointer (X)

The X-Pointer register allows for an 11-bit indexing value to be added to an 8-bit offset creating an effective address used for reading and writing between the entire memory space. (Software can only read from code EEPROM.) This provides software with the flexibility of storing lookup tables in the code EEPROM memory space for the core's accessibility during normal operation.

The X register is divided into two sections. The 10 least significant bits (LSB) of the register is the address of the program or data memory space. The most significant bit (MSB) of the register is write only and selects between the data (0x000 to 0x0FF) or program (0xC00 to 0xFFF) memory space.

Example: If Bit 10 = 0, then the LD A, [00,X] instruction will take a value from address range 0x000 to 0x0FF and load it into A. If Bit 10 = 1, then the LD A, [00,X] instruction will take a value from address range 0xC00 to 0xFFF and load it into A.

### 4.1.3 Program Counter (PC)

The 10-bit program counter register contains the address of the next instruction to be executed. After a reset, if in normal mode the program counter is initialized to 0xC00.

### 4.1.4 Stack Pointer (SP)

The ACEx microcontroller has an automatic program stack with a 4-bit stack pointer. The stack can be initialized to any location between addresses 0x30-0x3F. After a reset, the stack pointer is defaulted to 0xF pointing to address 0x3F. Normally, the stack pointer is initialized by one of the first instructions in an application program.

The stack is configured as a data structure which decrements from high to low memory. Each time a new address is pushed onto the stack, the core decrements the stack pointer by two. Each time an address is pulled from the stack, the core increments the stack pointer by two. At any given time, the stack pointer points to the next free location in the stack.

When a subroutine is called by a jump to subroutine (JSR) instruction, the address of the instruction is automatically pushed onto the stack least significant byte first. When the subroutine is finished, a return from subroutine (RET) instruction is executed. The RET instruction pulls the previously stacked return address from the stack and loads it into the program counter. Execution then continues at the recovered return address.

### 4.1.5 Status Register (SR)

This 8-bit register contains four condition code indicators (C, H, Z, and N), an interrupt masking bit (G), and an EEPROM write flag (R). The condition code indicators are automatically updated by most instructions. (See Table 10)

### Carry/Borrow (C)

The carry flag is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation and by its dedicated instructions. The rotate instruction operates with and through the carry bit to facilitate multiple-word shift operations. The LDC and INVC instructions facilitate direct bit manipulation using the carry flag.

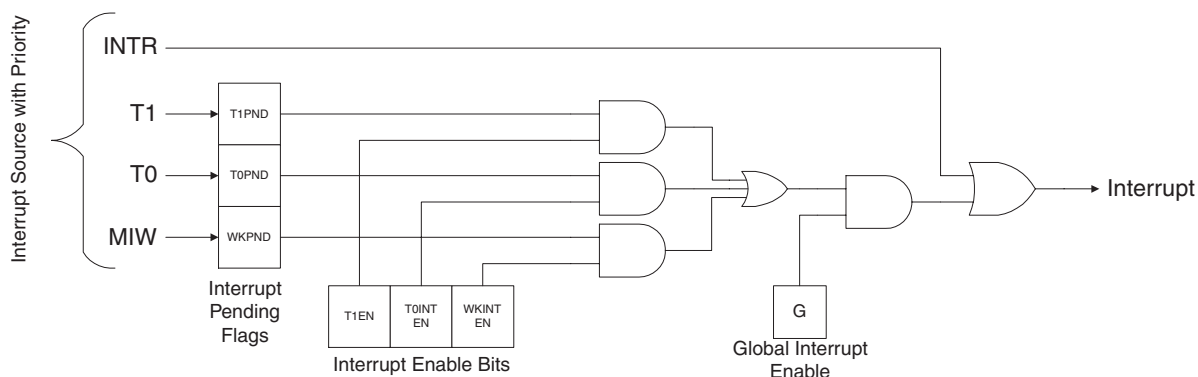
### Half Carry (H)

The half carry flag indicates whether an overflow has taken place on the boundary between the two nibbles in the accumulator. It is primarily used for Binary Coded Decimal (BCD) arithmetic calculation.

### Zero (Z)

The zero flag is set if the result of an arithmetic, logic, or data manipulation operation is zero. Otherwise, it is cleared.

**Figure 12: Basic Interrupt Structure**



**Table 9: Instruction Addressing Modes**

Instruction	Immediate			Direct	Indexed	Inherent		Relative	Absolute
ADC	A, #			A, M					
AND	A, #			A, M					
SUBC	A, #			A, M					
XOR	A, #			A, M					
CLR				M		A			
INC				M		A	X		
DEC				M		A	X		
IFEQ	A, #	M, #		A, M					
IFGT	A, #			A, M					
IFNE	A, #			A, M					
SC						no-op			
RC						no-op			
IFC						no-op			
IFNC						no-op			
INVC						no-op			
LDC				#, M					
STC				#, M					
RLC						A			
RRC						A			
LD	A, #	M, #	X, #	A, M	A, [00,X]				
ST				A, M	A, [00,X]				
LD				M, M					
NOP						no-op			
IFBIT				#, M					
SBIT				#, M					
RBIT				#, M					
JP								Rel	
JSR									M
JMP									M
RET						no-op			
RETI						no-op			
INTR						no-op			

**Table 10: Instruction Cycles and Bytes**

Mnemonic	Operand	Bytes	Cycles	Flags affected
ADC	A, #	2	2	C,H,Z,N
ADC	A, M	2	2	C,H,Z,N
AND	A, #	2	2	Z,N
AND	A, M	2	2	Z,N
CLR	A	1	1	Z,N,C,H
CLR	M	2	1	Z,N,C,H
DEC	A	1	1	Z,N
DEC	M	2	2	Z,N
DEC	X	1	1	Z
IFBIT	#, M	2	2	None
IFC		1	1	None
IFEQ	A, #	2	2	None
IFEQ	A, M	2	2	None
IFEQ	M, #	3	3	None
IFGT	A, #	2	2	None
IFGT	A, M	2	2	None
IFNE	A, #	2	2	None
IFNE	A, M	2	2	None
IFNC		1	1	None
INC	A	1	1	Z,N
INC	M	2	2	Z,N
INC	X	1	1	Z
INTR		1	5	None
INVC		1	1	C
JMP	M	3	4	None

Mnemonic	Operand	Bytes	Cycles	Flags affected
JP		1	1	None
JSR	M	3	5	None
LD	A, #	2	2	None
LD	A, [00,X]	2	3	None
LD	A, M	2	2	None
LD	M, #	3	3	None
LD	M, M	3	3	None
LD	X, #	3	3	None
LDC	#, M	2	2	C
NOP		1	1	None
RBIT	#, M	2	2	Z,N
RC		1	1	C,H
RET		1	5	None
RETI		1	5	None
RLC	A	1	1	C,Z,N
RRC	A	1	1	C,Z,N
SBIT	#, M	2	2	Z,N
SC		1	1	C,H
ST	A, [00,X]	2	3	None
ST	A, M	2	2	None
STC	#, M	2	2	Z,N
SUBC	A, #	2	2	C,H,Z,N
SUBC	A, M	2	2	C,H,Z,N
XOR	A, #	2	2	Z,N
XOR	A, M	2	2	Z,N

## 5.2 Pulse Width Modulation (PWM) Mode

In the PWM mode, the timer counts down at the instruction clock rate. When an underflow occurs, the timer register is reloaded from T1RA and the count down proceeds from the loaded value. At every underflow, a pending flag (T1PND) located in the T1CNTRL register is set. Software must then clear the T1PND flag and load the T1RA register with an alternate PWM value. In addition, the timer can be configured to toggle the T1 output bit upon underflow. Configuring the timer to toggle T1 results in the generation of a signal outputted from port G2 with the width and duty cycle controlled by the values stored in the T1RA. A block diagram of the timer's PWM mode of operation is shown in Figure 14.

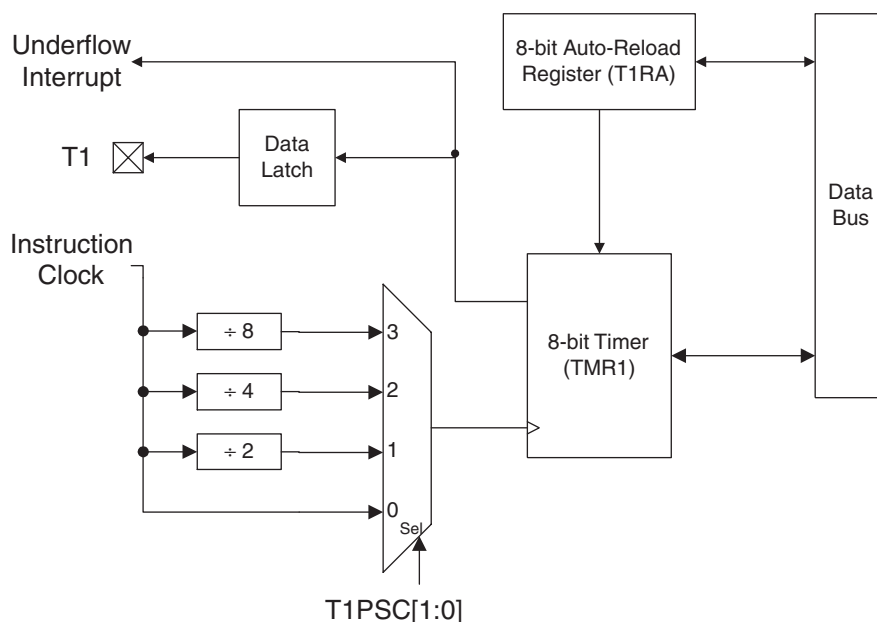
The timer has one interrupt (TMRI1) that is maskable through the T1EN bit of the T1CNTRL register. However, the core is only interrupted if the T1EN bit and the G (Global Interrupt enable) bit of the SR is set. If interrupts are enabled, the timer will generate an interrupt each time T1PND flags is set (whenever the timer underflows provided that the pending flag was cleared.) The interrupt service routine is responsible for proper handling of the T1PND flag and the T1EN bit.

The interrupt will be synchronous with every rising and falling edge of the T1 output signal. Generating interrupts only on rising or falling edges of T1 is achievable through appropriate handling of the T1EN bit or T1PND flag through software.

The following steps show how to properly configure Timer 1 to operate in the PWM mode. For this example, the T1 output signal is toggled with every timer underflow and the "high" and "low" times for the T1 output can be set to different values. The T1 output signal can start out either high or low depending on the configuration of I/O G2; the instructions below are for starting with the T1 output high. Follow the instructions in parentheses to start the T1 output low.

1. Configure T1 as an output by setting bit 2 of PORTGC.
  - SBIT 2, PORTGC ; Configure G2 as an output
2. Initialize T1 to 1 (or 0) by setting (or clearing) bit 2 of PORTGD.
  - SBIT 2, PORTGD ; Set G2 high
3. Load the initial PWM high (low) time into the timer register.
  - LD TMR1, #6FH ; High (Low) for .444ms (1MHz/4 clock)
4. Load the PWM low (high) time into the T1RA register.
  - LD T1RA, #2FH ; Low (High) for .188ms (1MHz/4 clock)
5. Write the appropriate control value to the T1CNTRL register to select PWM mode with T1 toggle, to select the divide by 4 pre-scaler, and to clear the enable and pending flags. (See Table 12)
  - LD T1CNTRL, #22H ; Setting the T1C0 bit starts the timer
6. Set te T1C0 bit to start the timer.
  - SBIT T1CP, T1CNTRL ; T1C0 equals 4
7. After every underflow, load T1RA with alternate values. If the user wishes to generate an interrupt on timer output transitions, reset the pending flags and then enable the interrupt using T1EN. The G bit must also be set. The interrupt service routine must reset the pending flag and perform whatever processing is desired.
  - RBIT T1PND, T1CNTRL ; T1PND equals 3
  - LD T1RA, #6FH ; Low for .444ms (1MHz/4 clock)

**Figure 14: Pulse Width Modulation Mode**



## 8.0 Multi-Input Wakeup/Interrupt Block

The Multi-Input Wakeup (MIW)/Interrupt contains three memory-mapped registers associated with this circuit: WKEDG (Wakeup Edge), WKEN (Wakeup Enable), and WKPND (Wakeup Pending). Each register has three bits with each bit corresponding to an input pins as shown in Figure 17. All three registers are initialized to zero upon reset.

The WKEDG register establishes the edge sensitivity for each of the wake-up input pin: either (0) rising edge or (1) falling edge.

The WKEN register enables (1) or disables (0) each of the port pins for the Wakeup/Interrupt function. The wakeup I/Os used for the Wakeup/Interrupt function must also be configured as an input pin in its associated port configuration register. However, an interrupt (EDGE1) of the core will not occur unless interrupts are enabled for the block via bit 7 of the T0CNTRL register (see Figure 15) and the G (global interrupt enable) bit of the SR is set.

The WKPND register contains the pending flags corresponding to each of the port pins (1 for wakeup/interrupt pending, 0 for wakeup/interrupt not pending).

To use the Multi-Input Wakeup/Interrupt circuit, perform the steps listed below. Performing the steps in the order shown will prevent false triggering of a Wakeup/Interrupt condition. This same procedure should be used following any type of reset because the wakeup inputs are left floating after resets resulting in unknown data on the port inputs.

1. Clear the WKEN register.
  - CLR WKEN
2. If necessary, write to the port configuration register to select the desired port pins to be configured as inputs.
  - RBIT 4, PORTGC ; G3, G4, and/or G5
3. If necessary, write to the port data register to select the desired port pins input state.
  - SBIT 4, PORTGD ; Pull-up
4. Write the WKEDG register to select the desired type of edge sensitivity for each of the pins used.
  - LD WKEDG, #38H ; Falling edges
5. Clear the WKPND register to cancel any pending bits.
  - CLR WKPND

6. Set the WKEN bits associated with the pins to be used, thus enabling those pins for the Wakeup/Interrupt function.
  - LD WKEN, #38H ; Enabling G3, G4, G5

Once the Multi-Input Wakeup/Interrupt function has been configured, a transition sensed on any of the enabled pins will set the corresponding bit in the WKPND register. The WKPND bits can bring the device out of the HALT/IDLE mode and can also trigger an interrupt if the interrupt is enabled. The interrupt service routine can read the WKPND register to determine which pin sensed the interrupt.

The interrupt service routine or other software should clear the pending bit. The device will not enter HALT/IDLE mode as long as a WKPND pending bit is pending and enabled. The user has the responsibility of clearing the pending flags before attempting to enter the HALT/IDLE mode.

Upon reset, the WKEDG register is configured to select positive-going edge sensitivity for all wakeup inputs. If the user wishes to change the edge sensitivity of a port pin, use the following procedure to avoid false triggering of a Wakeup/Interrupt condition.

1. Clear the WKEN bit associated with the pin to disable that pin.
2. Write the WKEDG register to select the new type of edge sensitivity for the pin.
3. Clear the WKPND bit associated with the pin.
4. Set the WKEN bit associated with the pin to re-enable it.

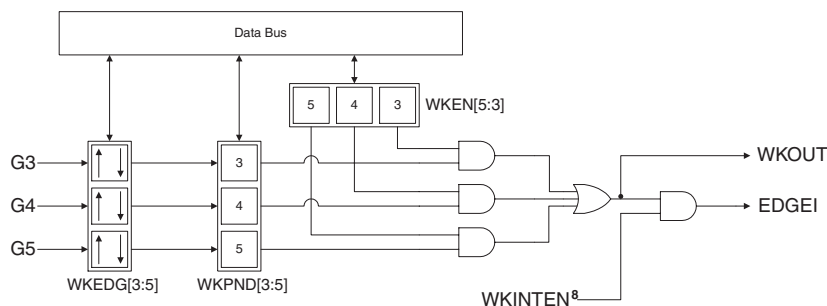
PORTG provides the user with three fully selectable, edge sensitive interrupts that are all vectored into the same service subroutine. The interrupt from PORTG shares logic with the wakeup circuitry. The WKEN register allows interrupts from PORTG to be individually enabled or disabled. The WKEDG register specifies the trigger condition to be either a positive or a negative edge. The WKPND register latches in the pending trigger conditions.

Since PORTG is also used for exiting the device from the HALT/IDLE mode, the user can elect to exit the HALT/IDLE mode either with or without the interrupt enabled. If the user elects to disable the interrupt, then the device restarts execution from the point at which it was stopped (first instruction cycle of the instruction following HALT/IDLE mode entrance instruction). In the other case, the device finishes the instruction that was being executed when the part was stopped and then branches to the interrupt service routine. The device then reverts to normal operation.

**Figure 17: MIW Register Bit Assignments**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	G5	G4	G3	x	x	x

**Figure 18: Multi-input Wakeup (MIW) Block Diagram**



<sup>8</sup> WKINTEN: Bit 7 of T0CNTRL

## 9.0 I/O Port

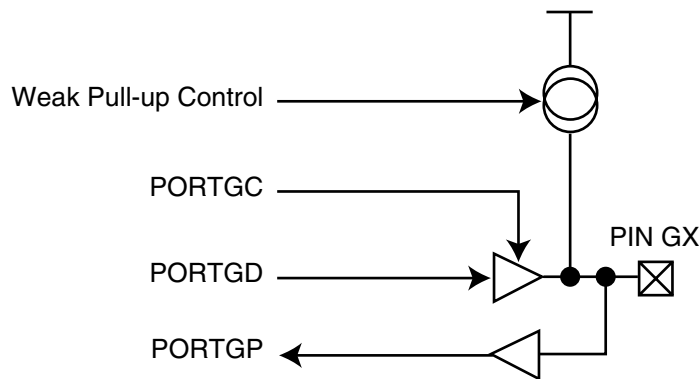
The six I/O pins are bi-directional with the exception of G3 which is always an input with weak pull-up (see Figure 19). The bi-directional I/O pins can be individually configured by software to operate as high-impedance inputs, as inputs with weak pull-up, or as push-pull outputs. The operating state is determined by the contents of the corresponding bits in the data and configuration registers. Each bi-directional I/O pin can be used for general purpose I/O, or in some cases, for a specific alternate function determined by the on-chip hardware.

### 9.1 I/O registers

The I/O pins (G0-G5) have three memory-mapped port registers associated with the I/O circuitry: a port configuration register

(PORTGC), a port data register (PORTGD), and a port input register (PORTGP). PORTGC is used to configure the pins as inputs or outputs. A pin may be configured as an input by writing a 0 or as an output by writing a 1 to its corresponding PORTGC bit. If a pin is configured as an output, its PORTGD bit represents the state of the pin (1 = logic high, 0 = logic low). If the pin is configured as an input, its PORTGD bit selects whether the pin is a weak pull-up or a high-impedance input. Table 13 provides details of the port configuration options. The port configuration and data registers are both read/writable. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports multi-input wakeup/interrupt, the PORTG inputs have Schmitt triggers.

**Figure 19: PORTGD Logic Diagram**



**Figure 20: I/O Register bit assignments**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	G5	G4	G3 <sup>9</sup>	G2	G1	G0

**Table 13: I/O configuration options**

Configuration Bit	Data Bit	Port Pin Configuration
0	0	High-impedance input (TRI-STATE input)
0	1	Input with pull-up (weak one input)
1	0	Push-pull zero output
1	1	Push-pull one output

<sup>9</sup> G3 is only an input.

## 10.0 In-circuit Programming Specification<sup>10,11</sup>

The ACEx microcontroller supports in-circuit programming of the internal data EEPROM, code EEPROM, and the initialization registers.

An externally controlled four wire interface consisting of a LOAD control pin (G3), a serial data SHIFT-IN input pin (G4), a serial data SHIFT-OUT output pin (G2), and a CLOCK pin (G1) is used to access the on-chip memory locations. Communication between the ACEx microcontroller and the external programmer is made through a 32-bit command and response word described in Table 14.

The serial data timing for the four-wire interface is shown in Figure 22 and the programming protocol is shown in Figure 21.

### 10.1 Write Sequence

The external programmer brings the ACEx microcontroller into programming mode by applying a super voltage level to the LOAD pin. The external programmer then needs to set the LOAD pin to 5V before shifting in the 32-bit serial command word using the SHIFT\_IN and CLOCK signals. By definition, bit 31 of the command word is shifted in first. At the same time, the ACEx microcontroller shifts out the 32-bit serial response to the last command on the SHIFT\_OUT pin. It is recommended that the external programmer samples this signal  $t_{ACCESS}$  (1 $\mu$ s) after the rising edge of the CLOCK signal. The serial response word, sent immediately after entering programming mode, contains indeterminate data.

After 32 bits have been shifted into the device, the external programmer must set the LOAD signal to 0V, and then apply two clock pulses as shown in Figure 21 to complete program cycle. The SHIFT\_OUT pin acts as the handshaking signal between the device and programming hardware once the LOAD signal is brought low. The device sets SHIFT\_OUT low by the time the programmer has sent the second rising edge during the LOAD =

0V phase (if the timing specifications in Figure 21 are obeyed).

The device will set the R bit of the Status register when the write operation has completed. The external programmer must wait for the SHIFT\_OUT pin to go high before bringing the LOAD signal to 5V to initiate a normal command cycle.

### 10.2 Read Sequence

When reading the device after a write, the external programmer must set the LOAD signal to 5V before it sends the new command word. Next, the 32-bit serial command word (for during a READ) should be shifted into the device using the SHIFT\_IN and the CLOCK signals while the data from the previous command is serially shifted out on the SHIFT\_OUT pin. After the Read command has been shifted into the device, the external programmer must, once again, set the LOAD signal to 0V and apply two clock pulses as shown in Figure 21 to complete READ cycle. Data from the selected memory location, will be latched into the lower 8 bits of the command word shortly after the second rising edge of the CLOCK signal.

Writing a series of bytes to the device is achieved by sending a series of Write command words while observing the devices handshaking requirements.

Reading a series of bytes from the device is achieved by sending a series of Read command words with the desired addresses in sequence and reading the following response words to verify the correct address and data contents.

The addresses of the data EEPROM and code EEPROM locations are the same as those used in normal operation.

Powering down the device will cause the part to exit programming mode.

**Table 14: 32-Bit Command and Response Word**

Bit number	Input command word	Output response word
bits 31 – 30	Must be set to 0	X
bit 29	Set to 1 to read/write data EEPROM, or the initialization registers, otherwise 0	X
bit 28	Set to 1 to read/write code EEPROM, otherwise 0	X
bits 27 – 25	Must be set to 0	X
bit 24	Set to 1 to read, 0 to write	X
bits 23 – 18	Must be set to 0	X
bits 17 – 8	Address of the byte to be read or written	Same as Input command word
bits 7 – 0	Data to be programmed or zero if data is to be read	Programmed data or data read at specified address

<sup>10</sup> Application Note reference: "How to In-Circuit Program the ACEx Family of Microcontrollers."

<sup>11</sup> During in-circuit programming, G5 must be either not connected or driven high.



## 11.0 Brown-out/Low Battery Detect Circuit

The Brown-out Reset (BOR) and Low Battery Detect (LBD) circuits on the ACE<sup>™</sup> microcontroller have been designed to offer two types of voltage reference comparators. The sections below will describe the functionality of both circuits.

### 11.1 Brown Out Reset

The Brown-out Reset (BOR) function is used to hold the device in reset when  $V_{CC}$  drops below a fixed threshold. While in reset, the device is held in its initial condition until  $V_{CC}$  rises above the threshold value. Shortly after  $V_{CC}$  rises above the threshold value, an internal reset sequence is started. After the reset sequence, the core fetches the first instruction and starts normal operation.

On the devices, the BOR should be used in situations when  $V_{CC}$  rises and falls slowly and in situations when  $V_{CC}$  does not fall to zero before rising back to operating range. The BOR can be thought of as a supplement function to the Power-on Reset when  $V_{CC}$  does not fall below ~1.5V. The Power-on Reset circuit works best when  $V_{CC}$  starts from 0V and rises sharply. So in applications where  $V_{CC}$  is not constant, the BOR will give added device stability.

The BOR circuit must be enabled through the BOR enable bit (BOREN) in the initialization register. The BOREN bit can only be set while the device is in programming mode. Once set, the BOR will always be

powered-up enabled. Software cannot disable the BOR. The BOR can only be disabled in programming mode by resetting the BOREN bit as long as the global write protect (WDIS) feature is not enabled.

### 11.2 Low Battery Detect

The Low Battery Detect (LBD) circuit allows software to monitor the  $V_{CC}$  level at the lower voltage ranges. LBD has eight software programmable voltage reference threshold levels ranging from 2.2V to 3.3V that can be changed on the fly. Once  $V_{CC}$  falls below the selected threshold, the LBD flag in the LBD control register is set. The LBD flag will hold its value until  $V_{CC}$  rises above the threshold. (See Figure 23)

The LBD bit is read only. If LBD is 0, it indicates that the  $V_{CC}$  level is higher than the selected threshold. If LBD is 1, it indicates that the  $V_{CC}$  level is below the selected threshold. The threshold level can be adjusted up to eight levels using the three trim bits (Bat\_trim[2:0]) of the LBD control register. The LBD flag does not cause any hardware actions or an interruption of the processor. It is for software monitoring only.

The LBD function is disabled during HALT/IDLE mode. After exiting HALT/IDLE, software must wait at least 10 $\mu$ s before reading the LBD bit to ensure that the internal circuit has stabilized.

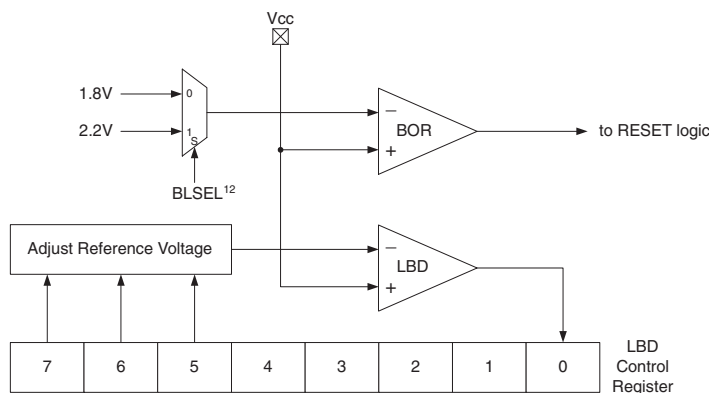
**Figure 23: LBD Control Register Definition**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bat_trim[2:0]			0	X	X	X	LBD

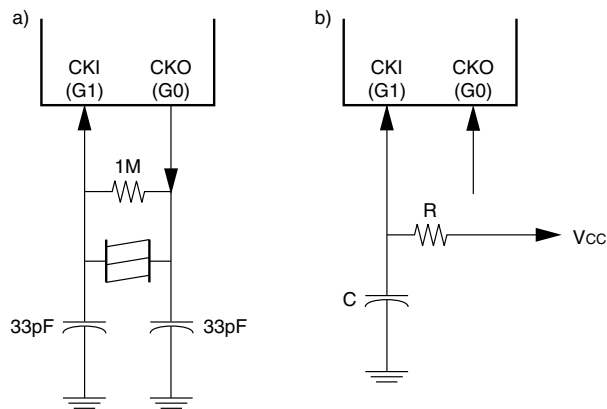
Bat_trim[2]	Bat_trim[1]	Bat_trim[0]	Voltage Threshold
0	0	0	3.3
0	0	1	3.1
0	1	0	2.9
0	1	1	2.7
1	0	0	2.5
1	0	1	2.4
1	1	0	2.3
1	1	1	2.2

**Figure 24: BOR/LBD Block Diagram**



<sup>12</sup> See Figure 13 for information on BLSEL.

**Figure 26: Crystal 9 (a) and RC (b) Oscillator Diagrams**



**15.0 HALT Mode**

The HALT mode is a power saving feature that almost completely shuts down the device for current conservation. The device is placed into HALT mode by setting the HALT enable bit (EHALT) of the HALT register through software using only the “LD M, #” instruction. EHALT is a write only bit and is automatically cleared upon exiting HALT. When entering HALT, the internal oscillator and all the on-chip systems including the LBD and the BOR circuits are shut down.

The device can exit HALT mode only by the MIW circuit. Therefore, prior to entering HALT mode, software must configure the MIW circuit accordingly. (See Section 8.0) After a wakeup from HALT, a 64 clock cycle start-up delay is initiated to allow the internal oscillator to stabilize before normal execution resumes. Immediately after exiting HALT, software must clear the Power Mode Clear (PMC) register by only using the “LD M, #” instruction. (See Figure 28)

**16.0 IDLE Mode**

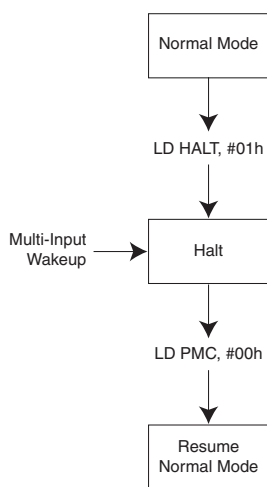
In addition to the HALT mode power saving feature, the device also supports an IDLE mode operation. The device is placed into IDLE mode by setting the IDLE enable bit (EIDLE) of the HALT register through software using only the “LD M, #” instruction. EIDLE is a write only bit and is automatically cleared upon exiting IDLE. The IDLE mode operation is similar to HALT except the internal oscillator, the Watchdog, and the Timer 0 remain active while the other on-chip systems including the LBD and the BOR circuits are shut down.

The device can exit IDLE by a Timer 0 overflow every 8192 cycles or/and by the MIW circuit. If exiting IDLE mode with the MIW, prior to entering, software must configure the MIW circuit accordingly. (See Section 8.0) Once a wake from IDLE mode is triggered, the core will begin normal operation by the next clock cycle. Immediately after exiting IDLE mode, software must clear the Power Mode Clear (PMC) register by using only the “LD M, #” instruction. (See Figure 29)

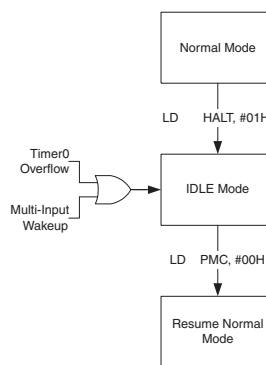
**Figure 27: HALT Register Definition**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	x	x	x	x	EIDLE	EHALT

**Figure 28: Recommended HALT Flow**



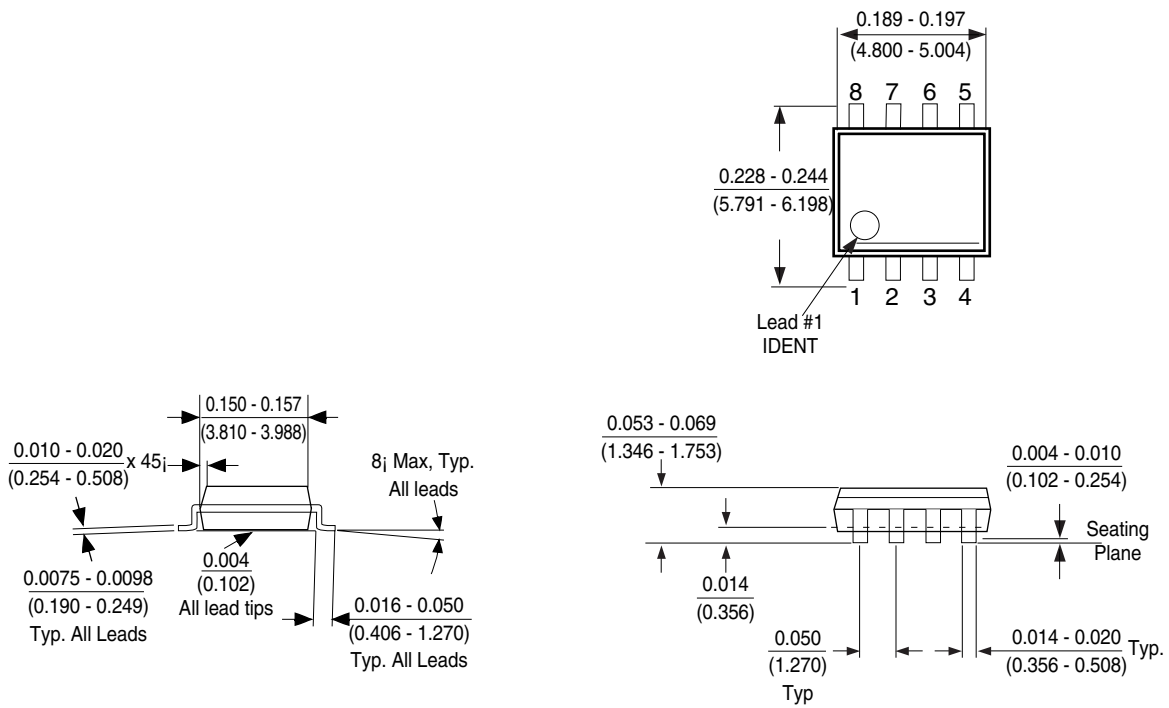
**Figure 29: Recommended IDLE Flow**



## Ordering Information

Part Number	Core Type			Max. # I/Os	Program Memory Size		Operating Voltage Range		Temperature Range			Package		Tape and Reel
	0	1	2		8	1K	2K	1.8 – 5.5V	2.2 – 5.5V	0 to 70°C	-40 to +85C	-40 to +125°C	8-pin SOIC	
ACE1001M8		X		X	X			X	X			X		
ACE1001M8X		X		X	X			X	X			X		X
ACE1001MT8		X		X	X			X	X				X	
ACE1001MT8X		X		X	X			X	X				X	X
ACE1001EM8		X		X	X			X		X		X		
ACE1001EM8X		X		X	X			X		X		X		X
ACE1001EMT8		X		X	X			X		X			X	
ACE1001EMT8X		X		X	X			X		X			X	X
ACE1001LM8		X		X	X		X		X			X		
ACE1001LM8X		X		X	X		X		X			X		X
ACE1001LMT8		X		X	X		X		X				X	
ACE1001LMT8X		X		X	X		X		X				X	X

**Physical Dimensions** inches (millimeters) unless otherwise noted



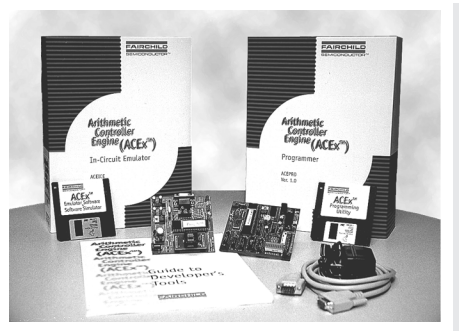
**Molded Small Out-Line Package (M8)**  
**Order Number ACE1001M8/ACE1001EM8/ACE1001LM8**  
**Package Number M08A**

## ACEx Development Tools

### General Information

Fairchild Semiconductor offers different possibilities to evaluate and emulate software written for ACEx.

**Simulator:** Is a Windows program able to load, assemble, and debug ACEx programs. It is possible to place as many breakpoints as needed, trace the program execution in symbolic format, and program a device with the proper options. The ACEx Simulator is available free-of-charge and can be downloaded from Fairchild's web site at [www.fairchildsemi.com/products/micro](http://www.fairchildsemi.com/products/micro)



**ACEx Emulator Kit:** Fairchild also offers a low cost real-time in-circuit emulator kit that includes:

- Emulator board
- Emulator software
- Assembler and Manuals
- Power supply
- DIP14 target cable
- PC cable

The ACEx emulator allows for debugging the program code in a symbolic format. It is possible to place one breakpoint and watch various data locations. It also has built-in programming capability.

**Prototype Board Kits:** Fairchild offer two solutions for the simplification of the breadboard operation so that ACEx Applications can be quickly tested.

- 1) ACEDEMO is can be used for general purpose applications
- 2) ACETXRX for transmitting / receiving (RF, IR, RS232, RS485) applications.

ACEDEMO has 8 switches, 8 LEDs, RS232 voltage translator, buzzer, and a lamp with a small breadboard area.

### Ordering P/Ns

#### Programming Adapters:

- DIP8 - ACEADAPT8
- DIP14 - ACEADAPT14
- TSSOP8 - ACEADAPTMT8
- SO8 - ACEADAPTM8
- SO14 - ACEADAPTM

#### Emulator Kit:

- ACEICE (110Vac)
- ACEICEEU (220Vac)

#### Prototype Boards:

- ACEDEMO
- ACETXRX (315MHz)
- ACETXRXEU (433MHz)

## Life Support Policy

Fairchild's products are not authorized for use as critical components in life support devices or systems without the express written approval of the President of Fairchild Semiconductor Corporation. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Fairchild Semiconductor  
Americas  
Customer Response Center  
Tel: 1-888-522-5372

Fairchild Semiconductor  
Europe  
Deutsch Fax: +44 (0) 1793-856858  
Tel: +49 (0) 8141-6102-0  
English Tel: +44 (0) 1793-856856  
Français Tel: +33 (0) 1-6930-3696  
Italiano Tel: +39 (0) 2-249111-1

Fairchild Semiconductor  
Hong Kong  
8/F, Room 808, Empire Centre  
68 Mody Road, Tsimshatsui East  
Kowloon, Hong Kong  
Tel: +852-2722-8338  
Fax: +852-2722-8383

Fairchild Semiconductor  
Japan Ltd.  
4F, Natsume Bldg.  
2-18-6, Yushima, Bunkyo-ku  
Tokyo, 113-0034 Japan  
Tel: 81-3-3818-8840  
Fax: 81-3-3818-8841

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.