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	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)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	8-TSSOP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=ace1101bemt8

Figure 2: ACE1101 Application Example (Remote Keyless Entry)

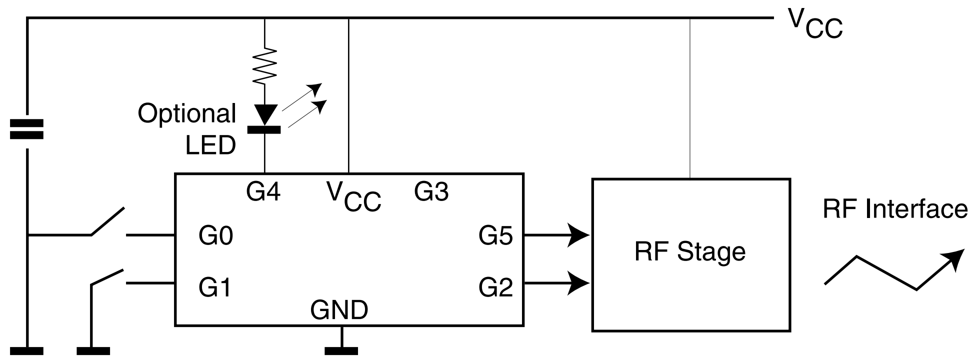


Figure 3: ACE1101 TSSOP/DIP 8-pin Device

a) Normal Mode Operation **b) Programming Mode Operation**

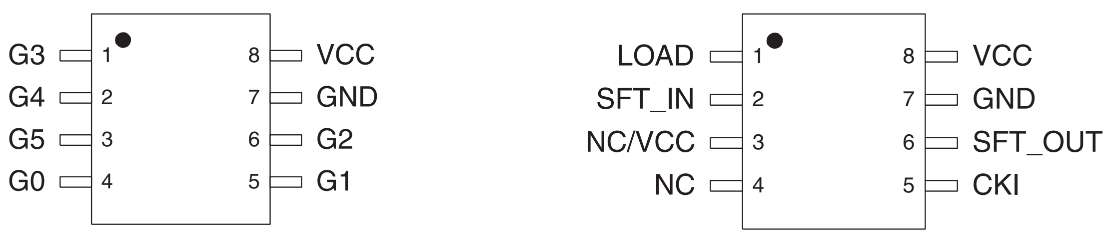
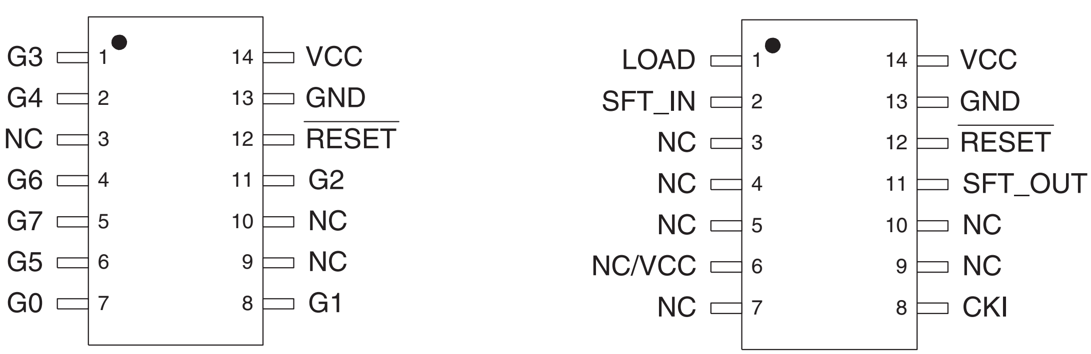


Figure 4: ACE1101 DIP 14-pin Device

a) Normal Mode Operation **b) Programming Mode Operation**



Preliminary ACE1101/1101B/1101L DC Electrical Characteristics

V_{CC} = 2.2/2.7/1.8 to 5.5V

All measurements valid for ambient operating temperature range unless otherwise stated.

Symbol	Parameter	Conditions	MIN	TYP	MAX	Units
I_{CC}[3]	Supply Current – no data EEPROM write in progress	1.8V 2.2V 2.7V 3.3V 5.5V		0.2 0.4 0.7 1.2 3.7	0.5 1.0 1.2 2.0 5.5	mA mA mA mA mA
I_{CCH}	HALT Mode current	3.3V @ -40°C to +25°C 5.5V @ -40°C to +25°C 3.3V @ +85°C 5.5V @ +85°C 3.3V @ +125°C 5.5V @+125°C		10 60 75 400 600 1550	100 1000 1000 2500 5000 8000	nA nA nA nA nA nA
I_{CCL}[4]	IDLE Mode Current	3.3V 5.5V		150 200	200 300	µA µA
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		1µs/V		10ms/V	
V_{IL}	Input Low with Schmitt Trigger Buffer	V_{CC} = 1.8 -5.5V			0.2V_{CC}	V
V_{IH}	Input High with Schmitt Trigger Buffer	V_{CC} = 1.8 - 5.5V	0.8V_{CC}			V
I_{IP}	Input Pull-up Current	V_{CC} =5.5V, V_{IN} =0V	30	65	350	µA
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, G6, G7	0.8 mA sink			0.2V_{CC}	V
	G5	1.0 mA sink			0.2V_{CC}	V
	Output Low Voltage	V_{CC} = 2.2V – 3.3V				
	G0, G1, G2, G4, G6, G7	3.0 mA sink			0.2V_{CC}	V
	G5	5.0 mA sink			0.2V_{CC}	V
	Output Low Voltage	V_{CC} = 3.3V – 5.5V				
	G0, G1, G2, G4, G6, G7	5.0 mA sink			0.2V_{CC}	V
	G5	10.0 mA sink			0.2V_{CC}	V
V_{OH}	Output High Voltage	V_{CC} = 1.8 - 2.2V				
	G0, G1, G2, G4, G6, G7	0.1 mA source	0.8V_{CC}			V
	G5	0.2 mA source	0.8V_{CC}			V
	Output High Voltage	V_{CC} = 3.3V – 5.5V				
	G0, G1, G2, G4, G6, G7	0.4 mA source	0.8V_{CC}			V
	G5	0.8 mA source	0.8V_{CC}			V
	Output High Voltage	V_{CC} = 3.3V – 5.5V				
	G0, G1, G2, G4, G6, G7	0.4 mA source	0.8V_{CC}			V
	G5	1.0 mA source	0.8V_{CC}			V

[3] I_{CC} active current is dependent on the program code.

[4] Based on a continuous IDLE looping program.

Preliminary ACE1101/1101B/1101L AC Electrical Characteristics

V_{CC} = 2.2/2.7/1.8 to 5.5V

All measurements valid for ambient operating temperature range unless otherwise stated.

Parameter	Conditions	MIN	TYP	MAX	Units
Instruction cycle time from internal clock - setpoint	5.0V at +25°C	0.9	1.0	1.1	µs
Internal clock voltage dependent frequency variation	3.0V to 5.5V, constant temperature			+5	%
Internal clock temperature dependent frequency variation	3.0V to 5.5V, full temperature range			+10	%
Internal clock frequency deviation for 0.5V drop	3.0V to 4.5V, constant temperature			+2	%
Crystal oscillator frequency	(Note 5)			4	MHz
External clock frequency	(Note 6)			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

[5] The maximum permissible frequency is guaranteed by design but not 100% tested.

[6] The parameter is guaranteed by design but not 100% tested.

Preliminary ACE1101/1101B/1101L 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

Preliminary ACE1101/1101L Low Battery Detect (LBD) Characteristics

V_{CC} = 2.2/1.8 to 5.5V
The following characteristics are guaranteed by design but are not 100% tested.

Parameter	Conditions	MIN	TYP	MAX	Units
LBD Voltage Threshold	Level 1 @ -40°C		2.84		V
	Level 8 @ -40°C		2.02		V
	Level 1 @ 0°C		2.98		V
	Level 8 @ 0°C		2.05		V
	Level 1 @ +25°C		3.08		V
	Level 8 @ +25°C		2.12		V
	Level 1 @ +85°C		3.31		V
	Level 8 @ +85°C		2.27		V
	Level 1 @ +125°C		3.36		V
	Level 8 @ +125°C		2.40		V

Preliminary ACE1101 Brown-out Reset (BOR) Characteristics

V_{CC} = 2.2 to 5.5V
The following characteristics are guaranteed by design but are not 100% tested.

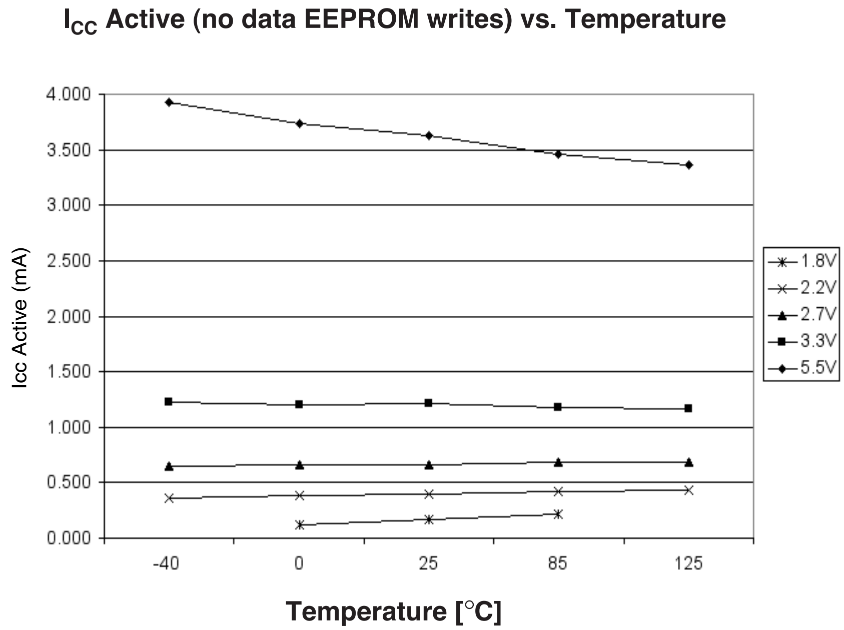
Parameter	Conditions	MIN	TYP	MAX	Units
BOR Trigger Threshold	-40°C		1.98		V
	0°C		2.06		V
	+25°C		2.12		V
	+85°C		2.27		V
	+125°C		2.37		V

Preliminary ACE1101L Brown-out Reset (BOR) Characteristics

V_{CC} = 1.8 to 5.5V
The following characteristics are guaranteed by design but are not 100% tested.

Parameter	Conditions	MIN	TYP	MAX	Units
BOR Trigger Threshold	0°C		1.78		V
	+25°C		1.82		V
	+70°C		1.96		V

Figure 9: I_{CC} Active



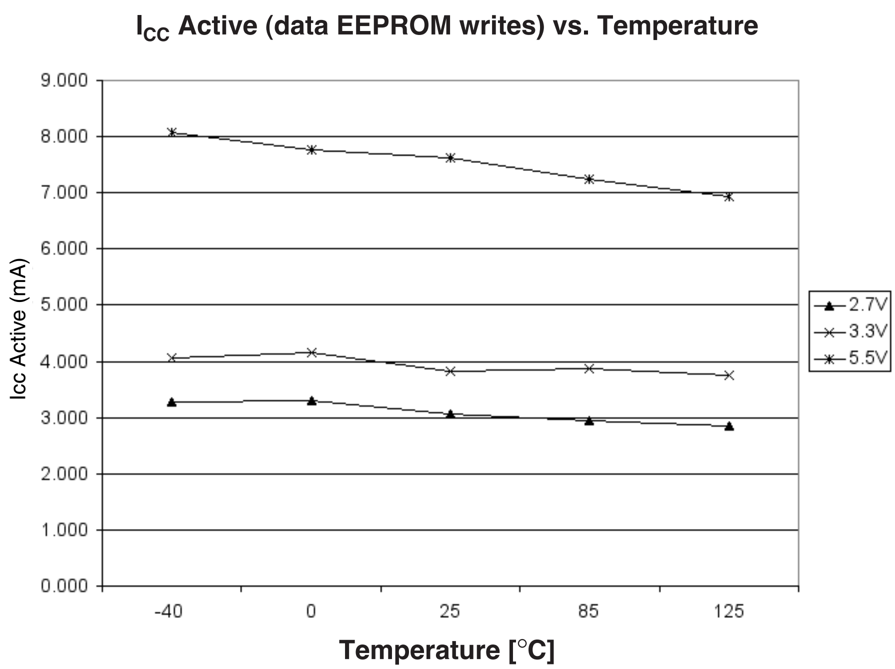
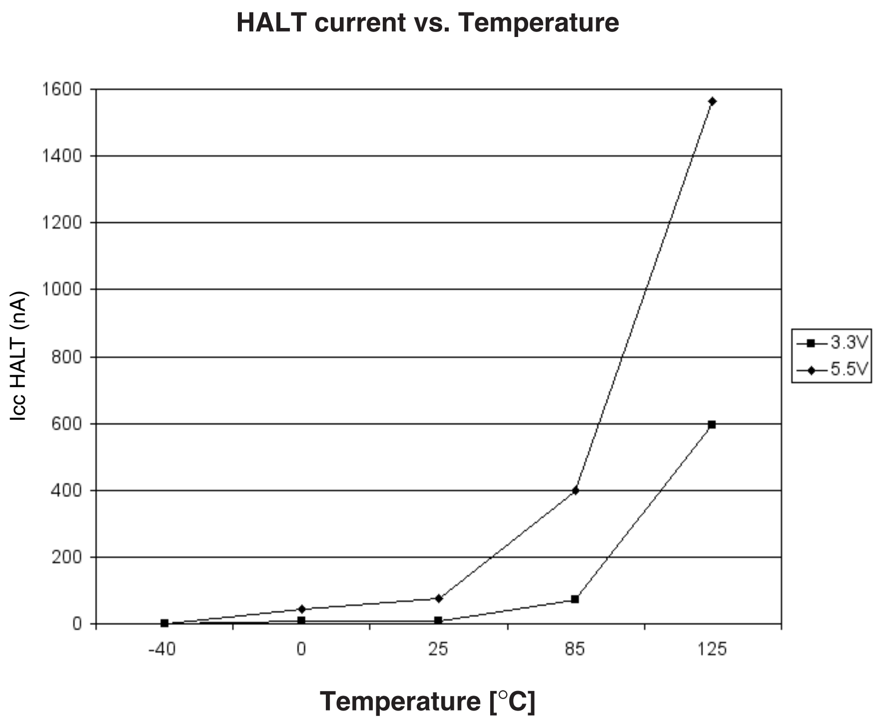


Figure 10: HALT Mode Currents



4.0 Arithmetic Controller Core

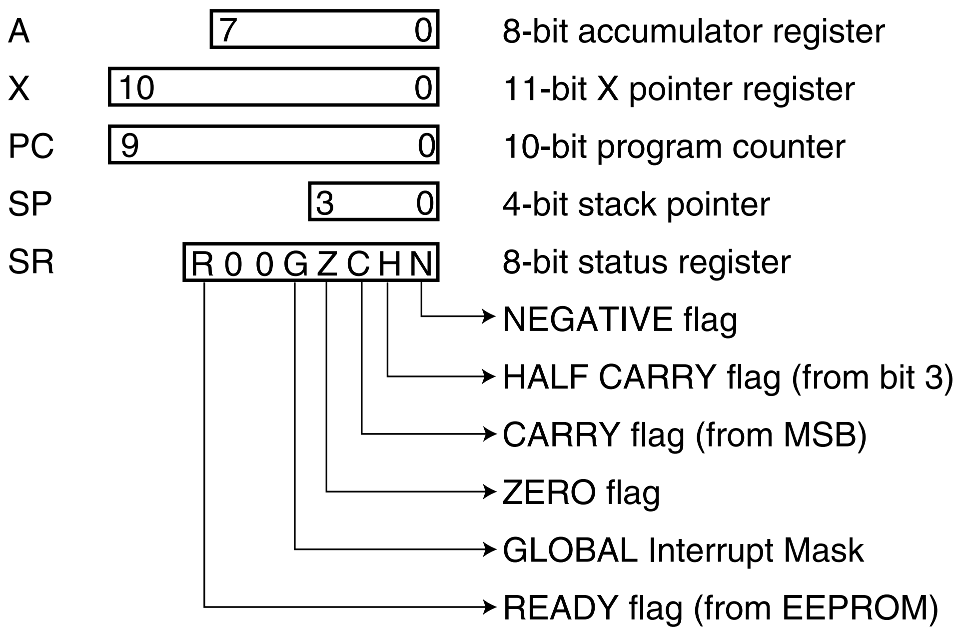
The ACEx microcontroller core is specifically designed for low cost applications involving bit manipulation, shifting block encryption. It is based on a modified Harvard architecture meaning peripheral, I/O, and RAM locations are addressed separately from instruction data.

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 ACEx microcontroller 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 12: 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 ACEx core allows software to access the entire 11-bit X-Pointer register using the special X-pointer instructions (e.g. LD X, #000H). (See Table 9) However, software may also access the register through any of the memory-mapped instructions using the XHI (X[10:8]) and XLO (X[7:0]) variables located at 0xBE and 0xBF, respectively. (See Table 11)

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.

The X register can also serve as a counter or temporary storage register. However, this is true only for the 10-LSBs since the 11th bit is dedicated for memory space selection.

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. Normally, the stack pointer is initialized by one of the first instructions in an application program. After a reset, the stack pointer is defaulted to 0xF pointing to address 0x3F.

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 is 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), one interrupt masking bit (G), and an EEPROM write flag (R). In the ACEx microcontroller, condition codes 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.

Negative (N)

The negative flag is set if the MSB of the result from an arithmetic, logic, or data manipulation operation is set to one. Otherwise, the flag is cleared. A result is said to be negative if its MSB is a one.

Interrupt Mask (G)

The interrupt request mask (G) is a global mask that disables all maskable interrupt sources. If the G Bit is cleared, interrupts can become pending, but the operation of the core continues uninterrupted. However, if the G Bit is set an interrupt is recognized. After any reset, the G bit is cleared by default and can only be set by a software instruction. When an interrupt is recognized, the G bit is cleared after the PC is stacked and the interrupt vector is fetched. Once the interrupt is serviced, a return from interrupt instruction is normally executed to restore the PC to the value that was present before the interrupt occurred. The G bit is reset to one after a return from interrupt is executed. Although the G bit can be set within an interrupt service routine, "nesting" interrupts in this way should only be done when there is a clear understanding of latency and of the arbitration mechanism.

4.2 Interrupt handling

When an interrupt is recognized, the current instruction completes its execution. The return address (the current value in the program counter) is pushed onto the stack and execution contin-

Table 8: Interrupt Priority Sequence

Priority (4 highest, 1 lowest)	Interrupt	
4	MIW	(EDGEI)
3	Timer0	(TMRI0)
2	Timer1	(TMRI1)
1	Software	(INTR)

Table 9: Instruction Addressing Modes

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

4.4 Memory Map

All I/O ports, peripheral registers and core registers, except the accumulator and the program counter are mapped into memory space.

Table 11: Memory Map

Address	Memory Space	Block	Contents
0x00 - 0x3F	Data	SRAM	Data RAM
0x40 - 0x7F	Data	EEPROM	Data EEPROM
0xAA	Data	Timer1	T1RALO register
0xAB	Data	Timer1	T1RAHI register
0xAC	Data	Timer1	TMR1LO register
0xAD	Data	Timer1	TMR1HI register
0xAE	Data	Timer1	T1CNTRL register
0xAF	Data	MIW	WKEDG register
0xB0	Data	MIW	WKPND register
0xB1	Data	MIW	WKEN register
0xB2	Data	I/O	PORTGD register
0xB3	Data	I/O	PORTGC register
0xB4	Data	I/O	PORTGP register
0xB5	Data	Timer0	WDSVR register
0xB6	Data	Timer0	T0CNTRL register
0xB7	Data	Clock	HALT mode register
0xB8 - 0xBC			Reserved
0xBD	Data	LBD	LBD register
0xBE	Data	Core	XHI register
0xBF	Data	Core	XLO register
0xC0	Data	Core	Power mode clear (PMC) register
0xCE	Data	Core	SP register
0xCF	Data	Core	Status register (SR)
0xC00 - 0xFF5	Program	EEPROM	Code EEPROM
0xFF6 - 0xFF7	Program	Core	Timer0 Interrupt vector
0xFF8 - 0xFF9	Program	Core	Timer1 Interrupt vector
0xFFA - 0xFFB	Program	Core	MIW Interrupt vector
0xFFC - 0xFFD	Program	Core	Software Interrupt vector
0xFFE - 0xFFF			Reserved

5.2 Mode 1: 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 15.

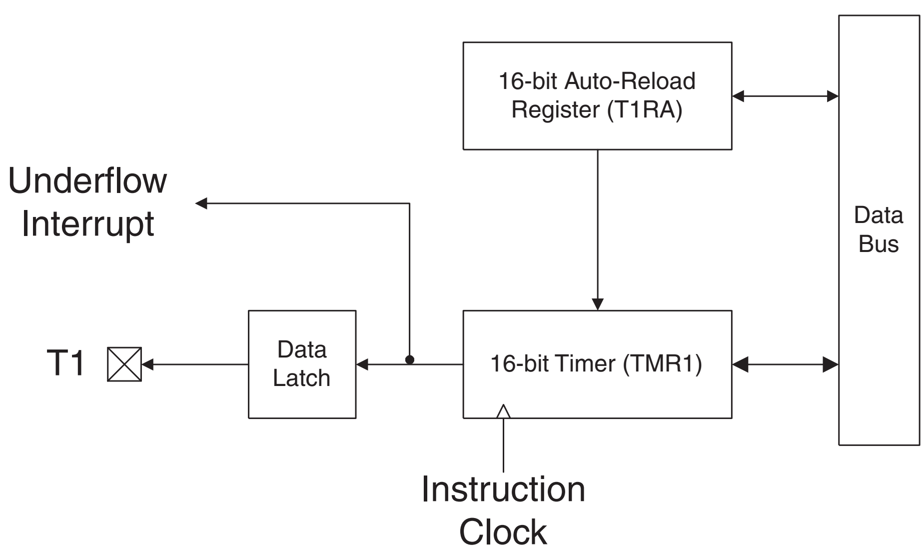
The timer has one interrupt (TMRI1) that is maskable through the T1EN bit of the T1CNTRL register. However, the core is only interrupted if the T1EN bit and the G (Global Interrupt enable) bit of the SR is set. If interrupts are enabled, the timer will generate an interrupt each time T1PND flags is set (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 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 TMR1LO, #6FH ; High (Low) for 1.391ms (1MHz clock)
   - LD TMR1HI, #05H
4. Load the PWM low (high) time into the T1RA register.
   - LD T1RALO, #2FH ; Low (High) for .303ms (1MHz clock)
   - LD T1RAHI, #01H
5. Write the appropriate control value to the T1CNTRL register to select PWM mode with T1 toggle, to clear the enable bit and pending flag, and to start the timer. (See Table 12 and Table 13)
   - LD T1CNTRL, #0B0H ; Setting the T1C0 bit starts the timer
6. After every underflow, load T1RA with alternate values. If the user wishes to generate an interrupt on a T1 output transition, 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 T1RALO, #6FH ; High (Low) for 1.391ms (1MHz clock)
   - LD T1RAHI, #05H

Figure 15: Pulse Width Modulation Mode Block Diagram



5.3 Mode 2: External Event Counter Mode

The External Event Counter mode operates similarly to the PWM mode; however, the timer is not clocked by the instruction clock but by transitions of the T1 input signal. The edge is selectable through the T1C1 bit of the T1CNTRL register. A block diagram of the timer's External Event Counter mode of operation is shown in Figure 16.

The T1 input should be connected to an external device that generates a positive/negative-going pulse for each event. By clocking the timer through T1, the number of positive/negative transitions can be counted therefore allowing software to capture the number of events that occur. The input signal on T1 must have a pulse width equal to or greater than one instruction clock cycle.

The counter can be configured to sense either positive-going or negative-going transitions on the T1 pin. The maximum frequency at which transitions can be sensed is one-half the frequency of the instruction clock.

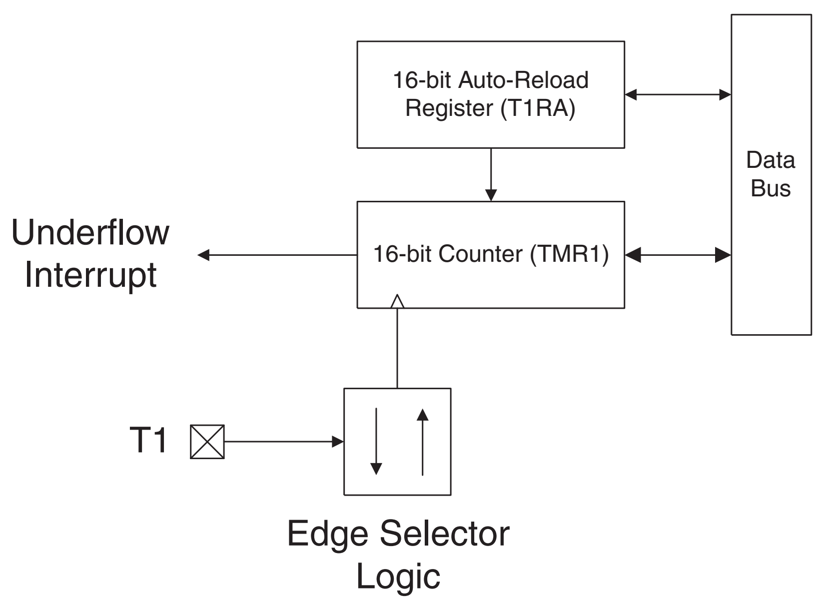
As with the PWM mode, when the counter underflows the counter is reloaded from the T1RA register and the count down proceedsfrom the loaded value. At every underflow, a pending flag (T1PND) located in the T1CNTRL register is set. Software must then clear the T1PND flag and can then load the T1RA register with an alternate value.

The counter 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 counter will generate an interrupt each time the T1PND flag is set (whenever 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 following steps show how to properly configure Timer 1 to operate in the External Event Counter mode. For this example, the counter is clocked every falling edge of the T1 input signal. Follow the instructions in parentheses to clock the counter every rising edge.

1. Configure T1 as an input by clearing bit 2 of PORTGC.
   - RBIT 2, PORTGC ; Configure G2 as an input
2. Initialize T1 to input with pull-up by setting bit 2 of PORTGD.
   - SBIT 2, PORTGD ; Set G2 high
3. Enable the global interrupt enable bit.
   - SBIT 4, STATUS
4. Load the initial count into the TMR1 and T1RA registers. When the number of external events is detected, the counter will reach zero; however, it will not underflow until the next event is detected. To count N pulses, load the value N-1 into the registers. If it is only necessary to count the number of occurrences and no action needs to be taken at a particular count, load the value 0xFFFF into the registers.
   - LD TMR1LO, #0FFH
   - LD TMR1HI, #00H
   - LD T1RALO, #0FFH
   - LD T1RAHI, #00H
5. Write the appropriate control value to the T1CNTRL register to select External Event Counter mode, to clock every falling edge, to set the enable bit, to clear the pending flag, and to start the counter. (See Table 12 and Table 13)
   - LD T1CNTRL, #34H (#00h) ; Setting the T1C0 bit starts the timer
6. When the counter underflows, the interrupt service routine must clear the T1PND flag and take whatever action is required once the number of events occurs. If the software wishes to merely count the number of events and the anticipated number may exceed 65,536, the interrupt service routine should record the number of underflows by incrementing a counter in memory. Software can then calculate the correct event count.
   - RBIT T1PND, T1CNTRL ; T1PND equals 3

Figure 16: External Event Counter Mode Block Diagram



5.4 Mode 3: Input Capture Mode

In the Input Capture mode, the timer is used to measure elapsed time between edges of an input signal. Once the timer is configured for this mode, the timer starts counting down immediately at the instruction clock rate. The Timer 1 will then transfer the current value of the TMR1 register into the T1RA register as soon as the selected edge of T1 is sensed. The input signal on T1 must have a pulse width equal to or greater than one instruction clock cycle. At every T1RA capture, software can then store the values into RAM to calculate the elapsed time between edges on T1. At any given time (with proper consideration of the state of T1) the timer can be configured to capture on positive-going or negative-going edges. A block diagram of the timer's Input Capture mode of operation is shown in Figure 17.

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. The Input Capture mode contains two interrupt pending flags 1) the TMR1 register capture in T1RA (T1PND) and 2) timer underflow (T1C0). If interrupts are enabled, the timer will generate an interrupt each time a pending flag is set (provided that the pending flag was previously cleared.) The interrupt service routine is responsible for proper handling of the T1PND flag, T1C0 flag, and the T1EN bit.

For this operating mode, the T1C0 control bit serves as the timer underflow interrupt pending flag. The Timer 1 interrupt service routine must read both the T1PND and T1C0 flags to determine the cause of the interrupt. A set T1C0 flag means that a timer underflow occurred whereas a set T1PND flag means that a capture occurred in T1RA. It is possible that both flags will be found set, meaning that both events occurred at the same time. The interrupt service routine should take this possibility into consideration.

Because the T1C0 bit is used as the underflow interrupt pending flag, it is not available for use as a start/stop bit as in the other modes.

The TMR1 register counts down continuously at the instruction clock rate starting from the time that the input capture mode is selected. (See Table 12and Table 13) To stop the timer from running, you must change the mode to an alternate mode (PWM or External Event Counter) while resetting the T1C0 bit.

The input pins can be independently configured to sense positive-going or negative-going transitions. The edge sensitivity of pin T1 is controlled by bit T1C1 as indicated in Table 13.

The edge sensitivity of a pin can be changed without leaving the input capture mode even while the timer is running. This feature allows you to measure the width of a pulse received on an input pin.

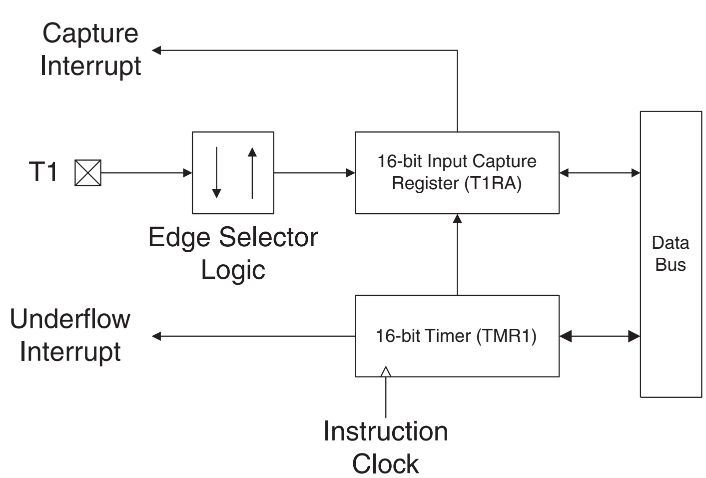
For example, the T1 pin can be programmed to be sensitive to a positive-going edge. When the positive edge is sensed, the TMR1 register contents is transferred to the T1RA register and a Timer 1 interrupt is generated. The Timer 1 interrupt service routine records the contents of the T1RA register, changes the edge sensitivity from positive to negative-going edge, and clears the T1PND flag. When the negative-going edge is sensed another Timer 1 interrupt is generated. The interrupt service routine reads the T1RA register again. The difference between the previous reading and the current reading reflects the elapsed time between the positive edge and negative edge of the T1 input signal i.e. the width of the positive-going pulse.

Remember that the Timer1 interrupt service routine must test the T1C0 and T1PND flags to determine the cause of the interrupt. If the T1C0 flag caused the interrupt, the interrupt service routine should record the occurrence of an underflow by incrementing a counter in memory or by some other means. The software that calculates the elapsed time between captures should take into account the number of underflow that occurred when making its calculation.

The following steps show how to properly configure Timer 1 to operate in the Input Capture mode.

1. Configure T1 as an input by clearing bit 2 of PORTGC.
   - RBIT 2, PORTGC ; Configure G2 as an input
2. Initialize T1 to input with pull-up by setting bit 2 of PORTGD.
   - SBIT 2, PORTGD ; Set G2 high
3. Enable the global interrupt enable bit.
   - SBIT 4, STATUS
4. With the timer stopped, load the initial time into the TMR1 register (typically the value is 0xFFFF.)
   - LD TMR1LO, #0FFH
   - LD TMR1HI, #00H
5. Write the appropriate control value to the T1CNTRL register to select Input Capture mode, to sense the appropriate edge, to set the enable bit, and to clear the pending flags. (See Table 12 and Table 13)
   - LD T1CNTRL, #64H ; T1C1 is the edge select bit
6. As soon as the input capture mode is enabled, the timer starts counting. When the selected edge is sensed on T1, the T1RA register is loaded and a Timer 1 interrupt is triggered.

Figure 17: Input Capture Mode Block Diagram



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 8-bits with each bit corresponding to an input pins as shown in Figure 20. 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 positive going-edge (0) or negative-going edge (1).

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 of the core will not occur unless interrupts are enabled for the block via bit 7 of the T0CNTRL register (see Figure 18) 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 ; G4
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, #0FFH ; All negative-going 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, #10H ; Enabling G4

Once the Multi-Input Wakeup/Interrupt function has been configured, a transition sensed on any of the I/O pins will set the corresponding bit in the WKPND register. The WKPND bits , where the corresponding enable (WKEN) bits are set, will bring the device out of the HALT/IDLE mode and can also trigger an interrupt if interrupts are 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.

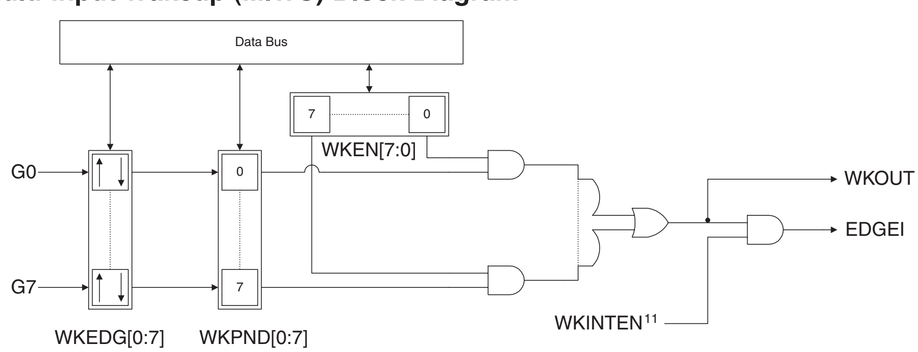
Since PORTG is also used for exiting the device from the HALT/IDLE mode, the user can elect to exit the HALT/IDLE mode either with or without the interrupt enabled. If the user elects to disable the interrupt, then the device restarts execution from the point at which it was stopped (first instruction 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 20: Multi-input Wakeup (MIW) Register Definition

WKEDG, WKEN, WKPND							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
[10]G7	[10]G6	G5	G4	G3	G2	G1	G0

[10] Available only on the 14-pin package option

Figure 21: Multi-input Wakeup (MIWU) Block Diagram



[11] WKINTEN: Bit 7 of T0CNTRL

9.0 I/O Port

The six I/O pins (eight on 14-pin package option) are bi-directional (see Figure 22) with the exception of G3 which is always an input with weak pull-up. 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-G7) 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-impedence input. Table 14 provides details of the port configuration options. The port configuration and data registers can both be read from or written to. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports MIW, PORTG inputs have Schmitt triggers.

Figure 22: PORTG Logic Diagram

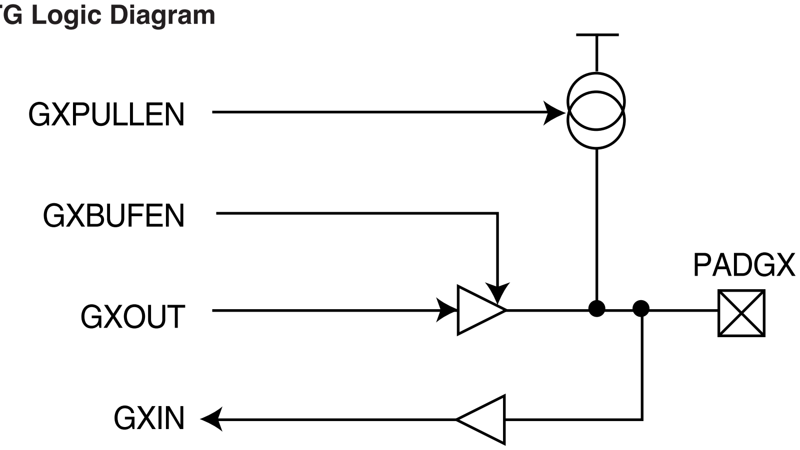


Figure 23: I/O Register bit assignments (PORTGC, PORTGD, PORTGD)

PORTGC, PORGD, PORTGD							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
G7[12]	G6[12]	G5	G4	G3[13]	G2	G1	G0

[12] Available only on the 14-pin package option

[13] G3 is always an input with weak pull-up.

Table 14: I/O configuration options

Configuration Bit	Data Bit	Port Pin Configuration
0	0	High-impedence 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

10.0 In-circuit Programming Specification[14, 15]

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 15.

The serial data timing for the four-wire interface is shown in Figure 25 and the programming protocol is shown in Figure 24.

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} (1µs) 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 24 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 24 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 24 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 15: 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 programm ed or zero if data is to be read	Programmed data or data read at specified address

[14] Ffor further information see Application Note AN-8005.

[15] During in-circuit programming, G5 must be either not connected or driven high.

Figure 24- Programming Protocol[15]

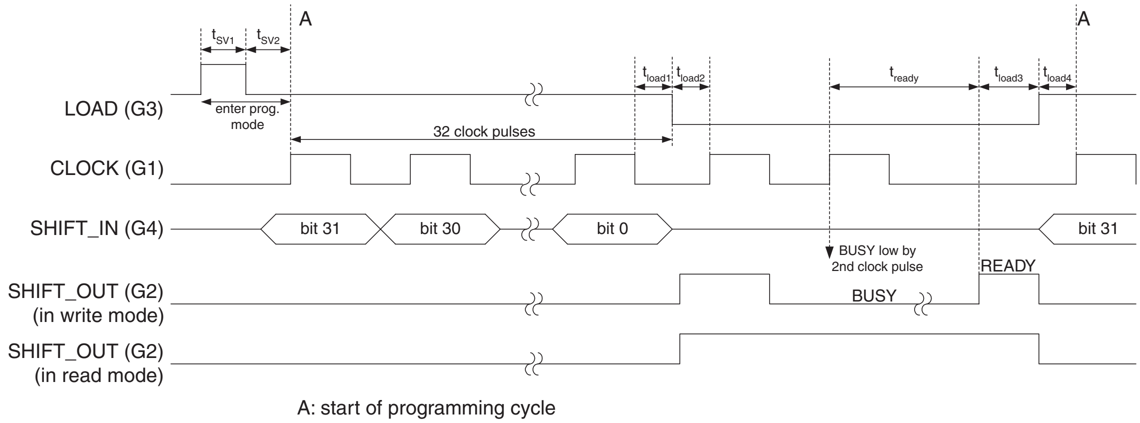


Figure 25- Serial Data Timing

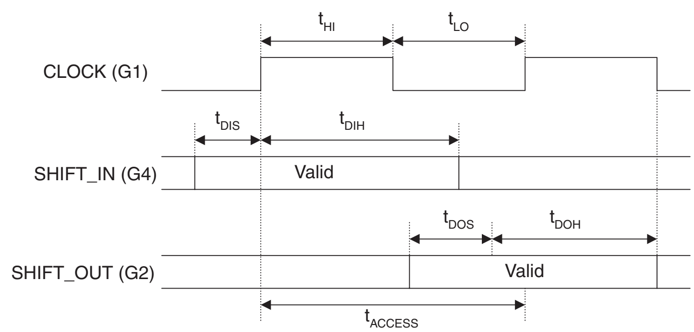
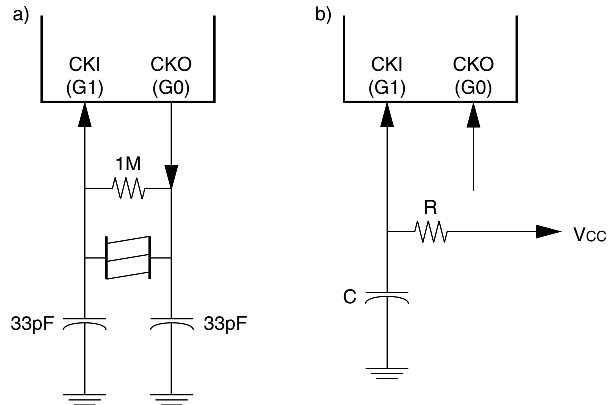


Figure 28: Crystal (a) and RC (b) Oscillator Diagrams



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) After a wakeup from HALT, a 1ms 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 30)

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) 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 31)

Figure 29: HALT Register Definition

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

Figure 30: Recommended HALT Flow

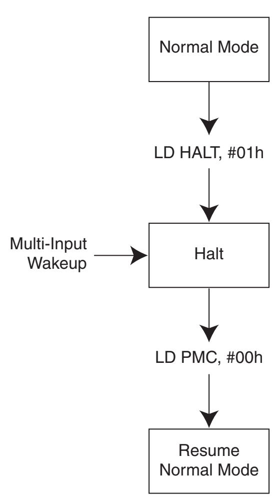
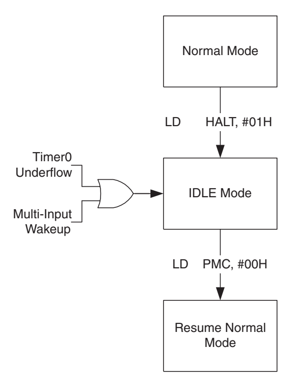


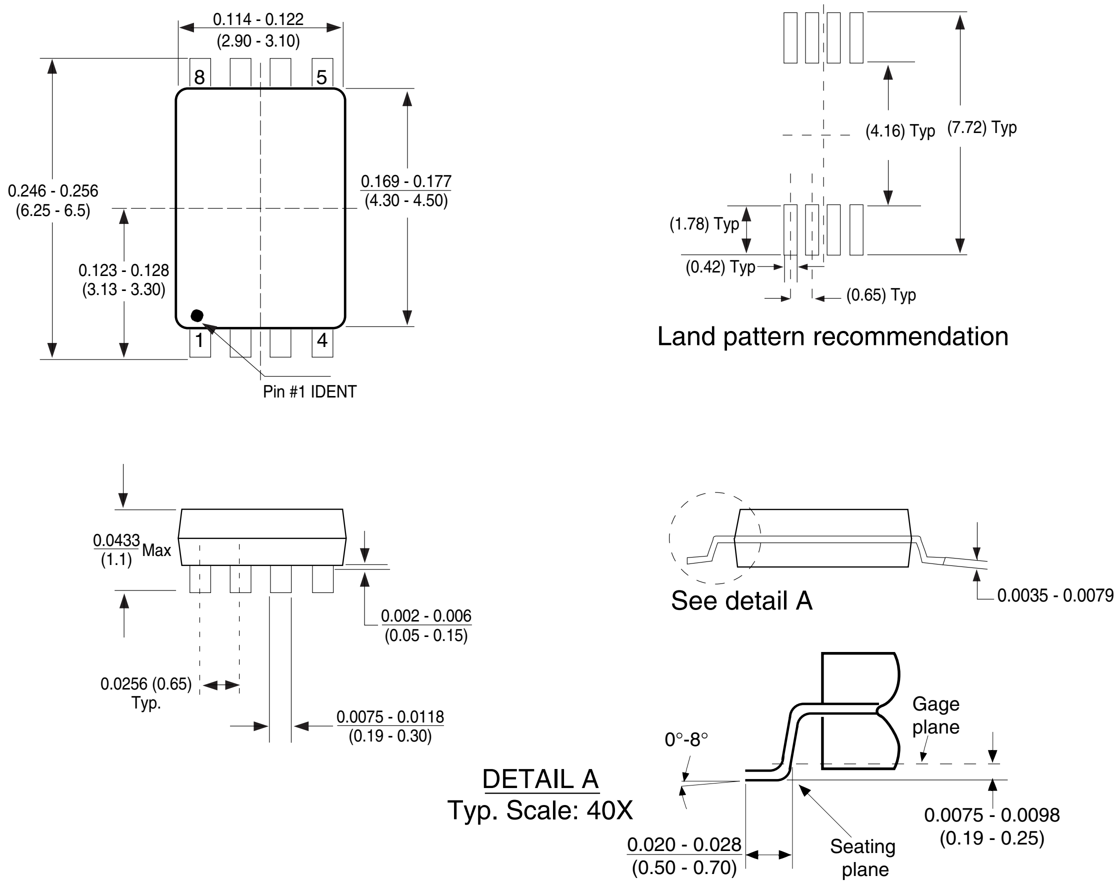
Figure 31: Recommended IDLE Flow



Ordering Information

Part Numbe	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	2.7 – 5.5V	0 to 70°C	-40 to +85C	-40 to +125°C	8-pin TSSOP	8-pin DIP	14-pin DIP	
ACE1101MT8		X		X	X			X		X			X			
ACE1101MT8X		X		X	X			X		X			X			X
ACE1101N		X		X	X			X		X				X		
ACE1101N14		X		X	X			X		X					X	
ACE1101EMT8		X		X	X			X			X		X			
ACE1101EMT8X		X		X	X			X			X		X			X
ACE1101EN		X		X	X			X			X			X		
ACE1101EN14		X		X	X			X			X				X	
ACE1101VMT8		X		X	X			X				X	X			
ACE1101VMT8X		X		X	X			X				X	X			X
ACE1101VN		X		X	X			X				X		X		
ACE1101VN14		X		X	X			X				X			X	
ACE1101BMT8		X		X	X				X	X			X			
ACE1101BMT8X		X		X	X				X	X			X			X
ACE1101BN		X		X	X				X	X				X		
ACE1101BN14		X		X	X				X	X					X	
ACE1101BEMT8		X		X	X				X		X		X			
ACE1101BEMT8X		X		X	X				X		X		X			X
ACE1101BEN		X		X	X				X		X			X		
ACE1101BEN14		X		X	X				X		X				X	
ACE1101BVMT8		X		X	X				X			X	X			
ACE1101BVMT8X		X		X	X				X			X	X			X
ACE1101BVN		X		X	X				X			X		X		
ACE1101BVN14		X		X	X				X			X			X	
ACE1101LMT8		X		X	X		X			X			X			
ACE1101LMT8X		X		X	X		X			X			X			X
ACE1101LN		X		X	X		X			X				X		
ACE1101LN14		X		X	X		X			X					X	

Physical Dimensions inches (millimeters) unless otherwise noted



Notes: Unless otherwise specified
1. Reference JEDEC registration MO153. Variation AA. Dated 7/93

8-Pin Molded TSSOP (MT8)
Order Number ACE1101(1101L)MT8/ACE1101EMT8/ACE1101VMT8
ACE1101BMT8/ACE1101BEMT8/ACE1101BVMT8
Package Number MTC08