FAIRCHILD SEMICONDUCTOR™

PRELIMINARY
March 1999

ACE1202
Arithmetic Controller Engine (ACEx™) for Low Power Secure Applications

General Description

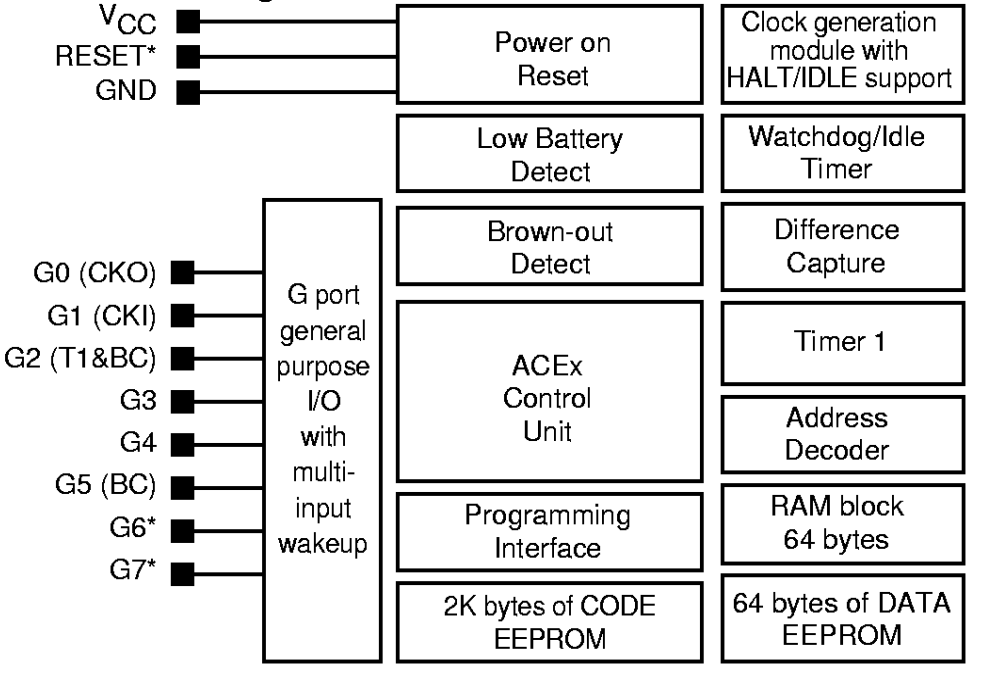
The ACE1202 (Arithmetic Controller Engine) is a dedicated programmable monolithic integrated circuit for applications requiring high performance, low power, and small size. It is a fully static part fabricated using CMOS technology.

The ACE1202 has an 8-bit microcontroller core, 64 bytes of RAM, 64 bytes of data EEPROM and 2K bytes of code EEPROM. Its on-chip peripherals include a multi-function 16-bit timer, watchdog/idle timer, and programmable undervoltage detection circuitry. On-chip clock and reset functions reduce the number of required external components. The ACE1202 is available in SO8, SO14, DIP8 and DIP14 packages.

Features

- Arithmetic Controller Engine
- 2K bytes on-board code EEPROM
- 64 bytes data EEPROM
- 64 bytes RAM
- Watchdog
- Multi-input wake-up
- Difference capture block
- 16-bit multifunction timer
- On-chip oscillator
  - — No external components
  - — 1µs instruction cycle time
- Instruction set geared for block encryption
- On-chip power on reset
- Programmable read and write disable functions
- Memory mapped I/O
- Multifunction Low Voltage Detection
- Fully static CMOS
  - — Low power HALT mode
  - — Power saving IDLE mode
  - — Single supply operation
- Software selectable I/O options
  - — Push-pull outputs with tri-state option
  - — Weak pull-up or high impedance inputs
- 40 years data retention
- 1,000,000 erase write cycles
- 8-pin and 14-pin SO and DIP packages

Block and Connection Diagram



*Available in 14-pin package only.

Absolute Maximum Ratings

Ambient Storage Temperature	-65°C to +150°C
Input Voltage not including G3	-0.3V to Vcc+0.3V
G3 Input Voltage	0.3V to 13V
Lead Temperature (10s max)	+300°C
Electrostatic Discharge on all pins	2000V min

Operating Conditions

Ambient Operating Temperature:	
ACE1202	0°C to +70°C
ACE1202E	-40°C to +85°C
ACE1202V	-40°C to +125°C
Operating Supply Voltage excluding EEPROM write:	
From 0°C to 125°C:	1.8V to 5.5V
From -40°C to 0°C:	2.0V to 5.5V
(based on preliminary data)	
Relative Humidity (non-condensing)	95%

Preliminary ACE1202 DC Electrical Characteristics

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

Parameter	Conditions	MIN	TYP	MAX	Units
Supply Current – no EEPROM write in progress	1.8V 3.3V 5.5V		0.4 1.2 3.6	0.5 1.8 4.8	mA mA mA
Supply Current – EEPROM write in progress	3.3V 5.5V		3.6 7.2	5.4 10.9	mA mA
HALT mode current	3.3V, -40°C to 25°C 5.5V, -40°C to 25°C 3.3V, 25°C to +85°C 5.5V, 25°C to +85°C 3.3V, -40°C to +125°C 5.5V, -40°C to +125°C		10 430 400 1100 2100 4000	100 1000 1000 2500 5000 8000	nA nA nA nA nA nA
EEPROM Write Voltage		2.5			V
Power supply rise time		1us		10ms	
Inputs Logic High Logic Low		 $0.8V_{CC}$		 $0.2\ V_{CC}$	 V V
Input Pull-up Current	V_{CC}=5.5V, V_{IN}=0V	30	125	350	µA
TRI-STATE Leakage	V_{CC}=5.5V		2	200	nA
Output Low Voltage - V_{OL} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 3.3V – 5.5V 5.0 mA sink 10.0 mA sink			 $0.2\ V_{CC}$ $0.2\ V_{CC}$	 V V
Output Low Voltage - V_{OL} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 2.2V – 3.3V 3.0 mA sink 5.0 mA sink			 $0.2\ V_{CC}$ $0.2\ V_{CC}$	 V V
Output Low Voltage - V_{OL} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 1.8V – 2.2V 1.0 mA sink 3.0 mA sink			 $0.2\ V_{CC}$ $0.2\ V_{CC}$	 V V
Output High Voltage - V_{OH} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 3.3V – 5.5V 0.5 mA source 1.0 mA source	 $0.8\ V_{CC}$ $0.8\ V_{CC}$			 V V
Output High Voltage - V_{OH} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 2.2V – 3.3V 0.4 mA source 0.8 mA source	 $0.8\ V_{CC}$ $0.8\ V_{CC}$			 V V
Output High Voltage - V_{OH} G0, G1, G2, G4, G6, G7 G5	V_{CC}= 1.8V – 2.2V 0.3 mA source 0.5 mA source	 $0.8\ V_{CC}$ $0.8\ V_{CC}$			 V V

Preliminary ACE1202 AC Electrical Characteristics

Parameter	Conditions	MIN	TYP	MAX	Units
Instruction cycle time from internal clock - setpoint	5.0V at 25°C	0.9	1.00	1.1	µs
Internal clock voltage variation	3.0V to 5.5V, constant temperature	-5%		+5%	
Internal clock variation	3.0V to 5.5V, full temperature range	-10%		+10%	
Frequency deviation for 0.5V drop	3.0V to 4.5V for ACE1202E, T=constant	-2%		+2%	
Crystal oscillator frequency	(Note 1)	1		4	MHz
External clock frequency	(Note 2)	1		4	MHz
EEPROM write time			5	10	ms
Internal clock start up time	(Note 2)			2	ms
Oscillator start up time	(Note 2)			2400	cycles
Reset pulse width time	(Note 2)	5			us

Note 1: The maximum permissible frequency is guaranteed by design but not 100% tested.

Note 2: The parameter is guaranteed by design but not 100% tested.

ACE1202 Electrical Characteristics for programming

All data valid over temperature range between 2.5V and 5.5V. See "EEPROM write time" in the AC characteristics for definition of the programming ready time. The following characteristics are guaranteed by design but are not production tested over the whole range.

Parameter	Description	MIN	MAX	Units
t_{HI}	CLOCK high time	0.50	DC	µs
t_{LO}	CLOCK low time	0.50	DC	
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	100		ns
t_{SV1}, t_{SV2}	LOAD supervoltage timing	50		us
t_{LOAD1}, t_{LOAD2}, t_{LOAD3}, t_{LOAD4}	LOAD timing	5		us
$V_{SUPERVOLTAGE}$	Supervoltage level	11.5	12.5	V

Figure 2: ACE1202 Application Example (Remote Keyless Entry)

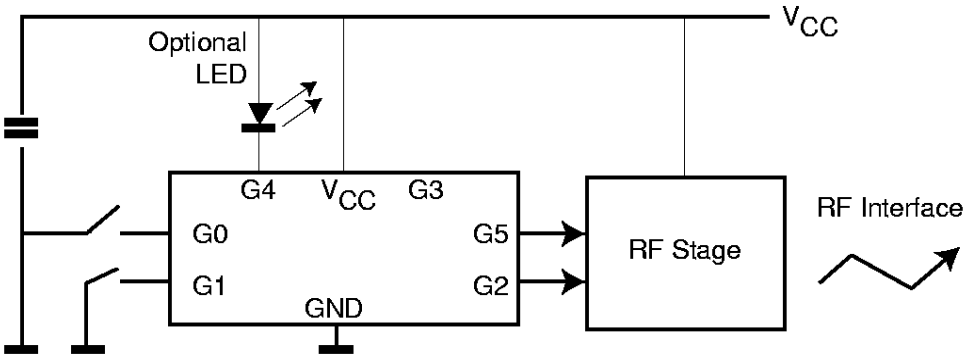
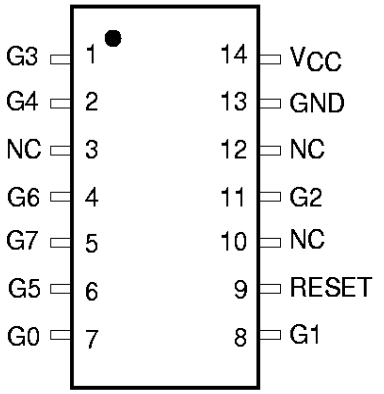


Figure 3: ACE1202 Pinout – Normal Operation



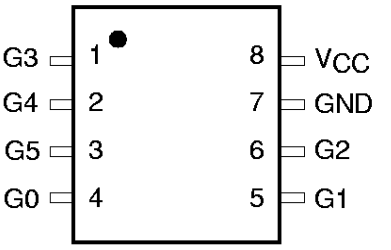
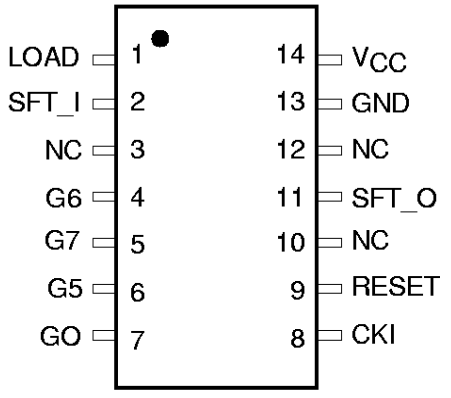
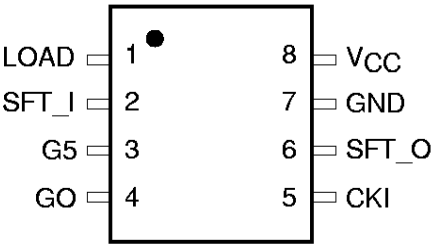
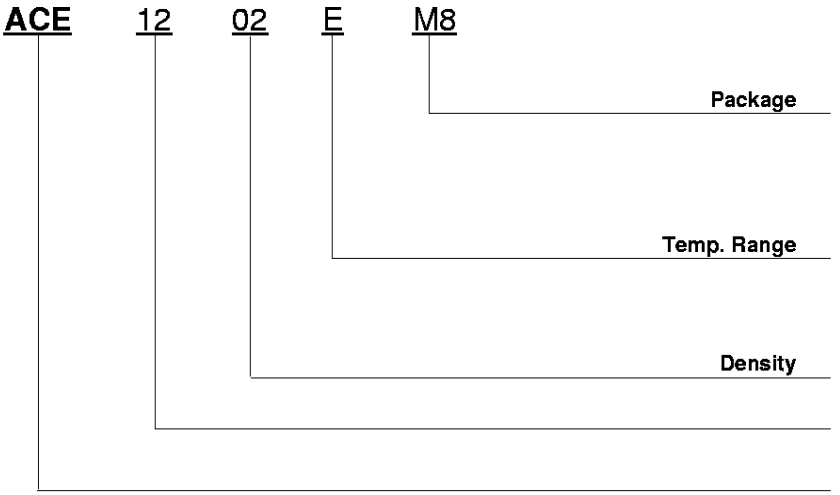


Figure 4: ACE1202 Pinout – Programming Mode





Ordering Information



	Letter	Description
Package	M8	8-pin SOIC
	M	14-pin SOIC
	N8	8-pin DIP
	N	14-pin DIP
Temp. Range	None	0 to 70°C
	E	-40 to +85°C
	V	-40 to +125°C
Density	02	2K Code EEPROM
	12	8-bit Microcontroller Core Type
	ACE	**Arithmetic Controller Engine**

3.0 Arithmetic Controller Core

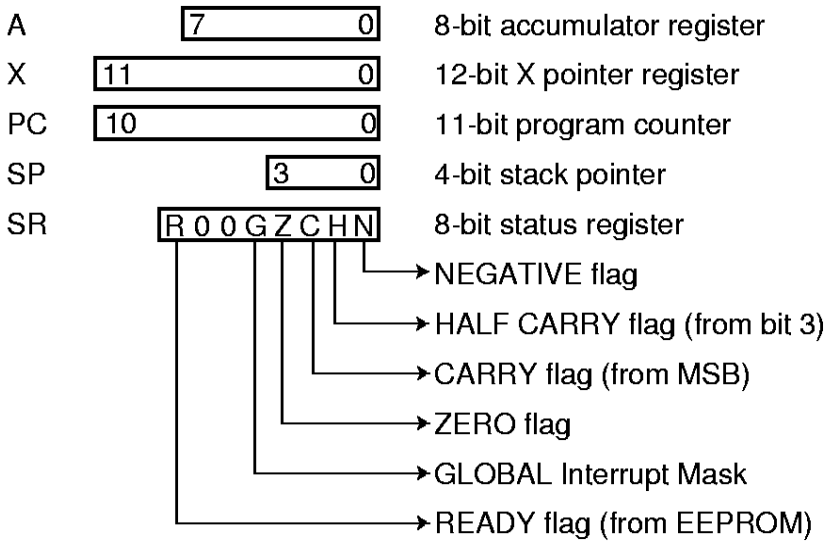
The ACE1202 core is specifically designed for low cost applications involving bit manipulation, shifting and block encryption. It is based on a modified Harvard architecture meaning peripheral, I/O, and RAM locations are addressed separately from instruction data.

The core differs from the traditional Harvard architecture by aligning the data and instruction memory sequentially. This allows the X-pointer (12-bits) to point to any memory location in either segment of the memory map. This modification improves the overall code efficiency of the ACE1202 and takes advantage of the flexibility found on Von Neumann style machines.

3.1 CPU Registers

The ACE1202 has five general purpose registers. They are the A, X, PC, SP, and SR. The X, SP and SR are memory mapped registers.

Figure 5: Programming Model



3.1.1 Accumulator

Accumulator A is general-purpose 8-bit register that holds operands and results of arithmetic calculations or data manipulations.

3.1.2 X Pointer

The X register provides a 12-bit indexing value that can be added to an 8-bit offset provided in an instruction to create an effective address. The X register can also be used as a counter or as a temporary storage register.

3.1.3 Program Counter (PC)

The program counter, an 11-bit register, contains the address of the next instruction to be executed. After reset, the program counter is initialized to 0x800 in normal mode.

3.1.4 Stack Pointer (SP)

The ACE1202 has an automatic program stack. This stack can be initialized to any location between addresses 0x30-0x3F. By default, the stack is initialized to 0x3F. Normally, the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that decrements from high memory to low memory. Each time a new address is pushed onto the stack, the SP is decremented by two. Each time an address is pulled from the stack, the SP is incremented by two. At any given time, the SP points to the next free location in the stack.

When a subroutine is called by a jump to subroutine (JSR), the address of the instruction, after the JSR 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 pulls the previously stacked return address from the stack, and loads it into the program counter. Execution then continues at this recovered return address.

3.1.5 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 ACE1202, condition codes are automatically updated by most instructions.

Carry/Borrow (C)

The C bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. 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 BCD arithmetic calculation.

Zero (Z)

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

Negative (N)

The N bit is set if the result of an arithmetic, logic, or data manipulation operation is negative (MSB = 1). Otherwise, the N bit is cleared. A result is said to be negative if its most significant bit (MSB) is a one.

Interrupt Mask (G)

The interrupt request mask (G) is a global mask that disables all maskable interrupt sources. Until the G bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted. 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. After 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 set 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.

3.2 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 continues at the address specified by the unique interrupt vector (see Table 7). This process takes five instruction cycles. At the end of the interrupt service routine, a RETI instruction is executed. The RETI instruction causes the saved address to be pulled off the stack in reverse order. Program execution resumes at the return address.

The ACE1202 is capable of supporting four interrupts. Three are maskable through the G bit of the Status register and the fourth (software interrupt) is not inhibited by the G bit. (See Table 4 for the interrupt priority sequence.) The software interrupt instruction is executed in a manner similar to other maskable interrupts in that it sets the G bit and the program counter registers are stacked.

3.3 Addressing Modes

The ACE1202 has seven addressing modes.

Indexed

In this addressing mode, a 8-bit unsigned offset value is added to the X-pointer yielding a new effective address. This mode can be used to address any memory location (Instruction or Data).

Indirect

This is the "normal" addressing mode. The operand is the data memory addressed by the X-pointer.

Table 4: Interrupt Priority Sequence

Interrupt	Priority (4 highest, 1 lowest)
MIW	4
Timer0	3
Timer1	2
Software	1

Direct

The instruction contains an 8-bit address field that directly points to the data memory for the operand.

Immediate

The instruction contains an 8-bit immediate field as the operand.

Inherent

This instruction has no operand associated with it.

Absolute

This mode is used with the JMP and JSR instructions, with the instruction field replacing the 11-bits in the program counter. This allows jumping to any location in the memory map.

Relative

This mode is used for the JP and the bit manipulation instructions, where the instruction field being added to the program counter to get the new program location.

Table 5: Instruction Addressing Modes

Instruction	Immediate			Direct	Indexed	Indirect		Inherent	Relative	Absolute
ADC	A, #			A, M	A, [00,X]	A, [X]				
ADD	A, #			A, M	A, [00,X]	A, [X]				
AND	A, #			A, M	A, [00,X]	A, [X]				
OR	A, #			A, M	A, [00,X]	A, [X]				
SUBC	A, #			A, M	A, [00,X]	A, [X]				
XOR	A, #			A, M	A, [00,X]	A, [X]				
CLR				M		A	X			
INC				M		A	X			
DEC				M		A	X			
IFEQ	A, #	X, #	M,#	A, M	A, [00,X]	A, [X]				
IFGT	A, #	X, #		A, M	A, [00,X]	A, [X]				
IFNE	A, #			A, M	A, [00,X]	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					A, [00,X]					M,M+1
JMP					A, [00,X]					M,M+1
RET								no-op		
RETI								no-op		
INTR								no-op		

Table 6: Instruction Addressing Modes

Mnemonic	Operand	Bytes	Cycles	Flags affected
ADC	A, [00,X]	2	2	C,H,Z,N
ADC	A, [X]	1	1	C,H,Z,N
ADC	A, M	2	2	C,H,Z,N
ADC	A, #	2	2	C,H,Z,N
ADD	A, [00,X]	2	2	Z,N
ADD	A, [X]	1	1	Z,N
ADD	A, M	2	2	Z,N
ADD	A, #	2	2	Z,N
AND	A, #	2	2	Z,N
AND	A, [00,X]	2	2	Z,N
AND	A, M	2	2	Z,N
AND	A, [X]	1	1	Z,N
CLR	X	1	1	Z
CLR	A	1	1	C,Z,N
CLR	M	2	1	C,Z,N
DEC	A	1	1	Z,N
DEC	M	2	2	Z,N
DEC	X	1	1	Z
IFBIT	#, A	1	1	None
IFBIT	#, M	2	2	None
IFC		1	1	None
IFEQ	A, [00,X]	2	2	None
IFEQ	A, [X]	1	1	None
IFEQ	A, #	2	2	None
IFEQ	A, M	2	2	None
IFEQ	M, #	3	3	None
IFEQ	X, #	3	3	None
IFGT	A, #	2	2	None
IFGT	A, [00,X]	2	2	None
IFGT	A, [X]	1	1	None
IFGT	A, M	2	2	None
IFGT	X, #	3	3	None
IFNE	A, #	2	2	None
IFNE	A, [00,X]	2	2	None
IFNE	A, [X]	1	1	None
IFNE	A, M	2	2	None
IFLT	X, #	3	3	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

Mnemonic	Operand	Bytes	Cycles	Flags affected
INVC		1	1	C
JMP	[00,X]	2	3	None
JMP	M, M+1	3	4	None
JP		1	1	None
JSR	M, M+1	3	5	None
JSR	[00,X]	2	5	None
LD	A, #	2	2	None
LD	A, [00,X]	2	3	None
LD	A, [X]	1	1	None
LD	A, M	2	2	None
LD	M, #	3	3	None
LD	X, #	3	3	None
LDC	#, M	2	2	C
LD	M, M	3	3	None
NOP		1	1	None
OR	A, #	2	2	Z,N
OR	A, [00,X]	2	2	Z,N
OR	A, [X]	1	1	Z,N
OR	A, M	2	2	Z,N
RBIT	#, [X]	1	2	Z,N
RBIT	#, M	2	2	Z,N
RC		1	1	C
RET		1	5	None
RETI		1	5	None
RLC	A	1	1	C,Z,N
RLC	M	2	2	C,Z,N
RRC	A	1	1	C,Z,N
RRC	M	2	2	C,Z,N
SBIT	#, [X]	1	2	Z,N
SBIT	#, M	2	2	Z,N
SC		1	1	C
ST	A, [00,X]	2	3	None
ST	A, [X]	1	1	None
ST	A, M	2	2	None
STC	#, M	2	2	Z,N
SUBC	A, #	2	2	C,H,Z,N
SUBC	A, [00,X]	2	2	C,H,Z,N
SUBC	A, [X]	1	1	C,H,Z,N
SUBC	A, M	2	2	C,H,Z,N
XOR	A, #	2	2	Z,N
XOR	A, [00,X]	2	2	Z,N
XOR	A, [X]	1	1	Z,N
XOR	A, M	2	2	Z,N

3.4 Memory Map

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

Table 7: Memory Map

Address	Block	Contents
0x00 - 0x3F	SRAM	Data RAM
0x40 - 0x7F	Data EEPROM	Data EEPROM
0xAA	Timer1	T1RALO register
0xAB	Timer1	T1RAHI register
0xAC	Timer1	TMR1LO register
0xAD	Timer1	TMR1HI register
0xAE	Timer1	T1CNTRL register
0xAF	MIWU	WKEDG register
0xB0	MIWU	WKPND register
0xB1	MIWU	WKEN register
0xB2	I/O	PORTGD register
0xB3	I/O	PORTGC register
0xB4	I/O	PORTGP register
0xB5	Timer0	WDSVR register
0xB6	Timer0	T0CNTRL register
0xB7	Clock	HALT mode register
0xB8 - 0xBC		Reserved
0xBD	LBD	LBD register
0xBE	Core	XHI register
0xBF	Core	XLO register
0xC0	Core	Power mode clear register
0xCE	Core	SP register
0xCF	Core	Status register
0x800 - 0xFF5	Code EEPROM	Code EEPROM/ROM
0xFF6 - 0xFF7	Core	Timer0 Interrupt vector
0xFF8 - 0xFF9	Core	Timer1 Interrupt vector
0xFFA - 0xFFB	Core	MIWU Interrupt vector
0xFFC - 0xFFD	Core	Software Interrupt vector
0xFFE - 0xFFF		Reserved

3.5 Memory

The ACE1202 device has 64 bytes of RAM and 64 bytes of EEPROM available for data storage. The controller also has a 2K byte EEPROM block for instruction data only.

User software can read and write to RAM(SRAM) and to the 64 byte EEPROM block (EEPROMD) but cannot perform writes to the 2K byte EEPROM block (EEPROMC) which is protected from writes during normal operations. The instruction data in the EEPROMC block can only be written to when the device is in program mode and if the initialization register bit WDIS (Write disable) is not set.

While in normal mode, the user can write to the EEPROMD block by 1.) polling the R bit of the Status register then 2.) executing the appropriate write instruction. A "one" on the R bit indicates the EEPROMD block is ready to perform the next write. A "zero" indicates the EEPROMD block is busy. The EEPROMD block will reset the R bit on the completion of a write cycle. Attempts to read or write to the EEPROMD or enter HALT mode while the R bit is set could affect the current byte being written.

3.6 Initialization Registers

The ACE1202 has two 8-bit wide initialization registers. These registers are read from memory space on power-up and initializes certain on-chip peripherals. Figure 6 provides a details description of Initialization Register 1. The Initialization Register 2 is used to trim the internal oscillator. This register is pre-programmed in the factory to yield a 1MHz internal clock.

Both Initialization Registers 1 and 2 are read/writable in programming mode. However, retrimming the internal oscillator (writing to the Initialization Register 2) is *discouraged*.

Figure 6: Initialization Register 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMODE0	CMODE1	WDEN	BOREN	LBDEN	UBD	WDIS	RDIS

(0) RDIS	If set, disables attempts to read any memory contents in programming mode
(1) WDIS	If set, disables attempts to write any memory contents in programming mode
(2) UBD	If set, the device will not allow writes to occur in the upper block of EEPROM
(3) LBDEN	1 enables LBD. 0 disables LBD
(4) BOREN	If set, allows a brown-out reset to occur if Vcc is so low that a reliable EEPROM write cannot take place.
(5) WDEN	If set, enables the on-chip processor watchdog circuit
(6) CMODE1	Clock mode select bit one
(7) CMODE0	Clock mode select bit zero

4.0 Timer 1

Timer1 is a versatile 16-bit timer which can operate in one of four modes:

- **Pulse Width Modulation** (PWM) mode, which generates pulses of a specified width and duty cycle
- **External Event Counter** mode, which counts occurrences of an external event
- **Standard Input Capture** mode, which measures the elapsed time between occurrences of external events
- **Difference Input Capture** mode, which automatically measure the difference between edges.

Timer1 contains a 16-bit timer (counter) register, designated TMR1, and one 16-bit autoreload (capture) register, designated T1RA. These 16-bit registers are organized as a pair of 8-bit memory mapped register bytes, TMR1HI and TMR1LO, and T1RAHI and T1RALO.

The timer (counter) block uses one I/O pin, designated T1, which is the alternate function of G2.

The timer can be started or stopped under program control. When running, the timer counts down (decrements). Depending on the operating mode, the timer counts either instruction clock cycles or transitions on the T1 pin. Occurrences of timer underflows (transitions from 0x0000 to 0xFFFF) can either generate an interrupt and/or toggle the T1 pin, also depending on the operating mode.

There is one interrupt associated with the timer, designated the Timer1 interrupt. When timer interrupt is enabled, the source of the interrupt depends on the timer operating mode: either a timer underflow, or a transfer of data to or from the T1RA register.

4.1 Timer control bits

Timer1 is controlled by reading and writing to the T1CNTRL register. By programming the control bits, the user can enable or disable the timer interrupts, set the operating mode, and start or stop the timer. The control bits operate as described in Tables 8 and 9.

Table 8: TIMER1 Control Register Bits

T1CNTRL Register	Name	Function
Bit 7	T1C3	Timer TIMER1 control bit 3 (see Table 9)
Bit 6	T1C2	Timer TIMER1 control bit 2 (see Table 9)
Bit 5	T1C1	Timer TIMER1 control bit 1 (see Table 9)
Bit 4	T1C0	Timer TIMER1 run: 1 = Start timer, 0 = Stop timer; or Timer TIMER1 underflow interrupt pending flag in input capture or difference capture modes
Bit 3	T1PNDA	Timer1 interrupt pending flag: 1 = Timer1 interrupt pending, 0 = Timer1 interrupt not pending
Bit 2	T1ENA	Timer1 interrupt enable bit: 1 = Timer1 interrupt enabled, 0 = Timer1 interrupt disabled
Bit 1	M4S1	Selects capture types 0 = Pulse capture 1 = Cycle capture
Bit 0	----------	Reserved

Table 9: Timer Operating Modes

T1 C3	T1 C2	T1 C1	M4 S1	Timer Mode Source	Interrupt A	Timer Counts On
0	0	0	x	MODE 2	Timer Underflow	T1A Pos. Edge
0	0	1	x	MODE 2	Timer Underflow	T1A Neg. Edge
1	0	1	x	MODE 1 T1A Toggle	Autoreload T1RA	Instruction Clock
1	0	0	x	MODE 1 No T1A Toggle	Autoreload T1RA	Instruction Clock
0	1	0	x	MODE 3 Captures: T1A Pos. Edge	Pos. T1A Edge	Instruction Clock
0	1	1	x	MODE 3 Captures: T1A Neg. Edge	Neg. T1A Edge	Instruction Clock
1	1	0	0	MODE 4 Difference Capture	Pos. to Neg.	Instruction Clock
1	1	0	1	MODE 4 Difference Capture	Pos. to Pos.	Instruction Clock
1	1	1	0	MODE 4 Difference Capture	Neg. to Pos.	Instruction Clock
1	1	1	1	MODE 4 Difference Capture	Neg. to Neg.	Instruction Clock

4.2 Pulse Width Modulation Mode

In the Pulse Width Modulation (PWM) mode, the timer counts down at the instruction clock rate. When an underflow occurs, the timer register is reloaded from T1RA, and decrementing proceeds from the loaded value. At every underflow interrupt, software should load the T1RA register with the alternate PWM value.

The timer can be configured to toggle the T1 output bit upon underflow. This results in the generation of a clock signal on T1 with the width and duty cycle controlled by the values stored in the T1RA.

A block diagram of the timer operating in the PWM mode is shown in Figure 7.

There is one interrupt associated with the timer, designated the Timer1 interrupt. The interrupt is maskable by the enable bit T1EN. Thus, the user can generate an interrupt on the rising edge, falling edge, or on both edges of the PWM output or disable the interrupt.

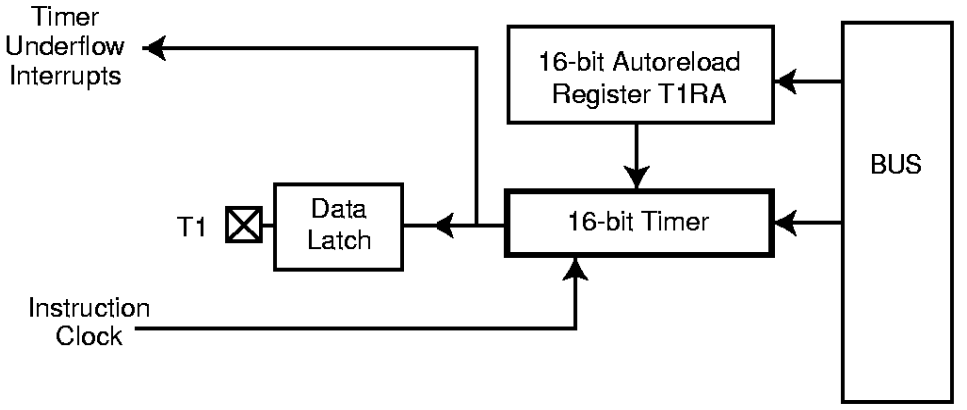
When an underflow occurs that causes a timer reload from T1RA, the interrupt pending flag bit T1PNDA is set. A CPU interrupt occurs if T1EN bit and the G (Global Interrupt enable) bit of the Status register is set. The interrupt service routine must reset the pending bit and perform whatever processing is necessary at the interrupt point.

The following steps can be used to operate the timer in the PWM mode. In this example, the T1 output pin 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 can start out either high or low; the instructions below are for starting with the T1 output high. (Follow the instructions in parentheses to start the T1 output low.)

1. Configure the T1 pin as an output by setting bit 2 of PORTGC.
2. Initialize the T1 pin value to 1 (or 0) by setting (or clearing) bit 2 of PORTGD.
3. Load the PWM "high" or "low" time into the timer register.
4. Load the PWM "low" or "high" time into the T1RA register.
5. Write the appropriate value to the timer control bits T1C3-T1C2- T1C1 to select the PWM mode, and to toggle the T1 output with every timer underflow (see Table 9).
6. Set the T1C0 bit to start the timer.
7. Upon every underflow interrupt load T1RA with alternate values, ON or OFF time.

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.

Figure 7: Pulse Width Modulation Mode



4.3 External Event Counter Mode

The external event counter mode is similar to the PWM mode, except that instead of counting instruction clock pulses, the timer counts transitions received on the T1 pin (configured as an input). The T1 pin should be connected to an external device that generates a pulse for each event to be counted. The input signal on T1 must have a pulse width equal to or greater than one instruction cycle.

The timer 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 an underflow occurs, the timer register is reloaded from the T1RA registers, and counting proceeds downward from the loaded value.

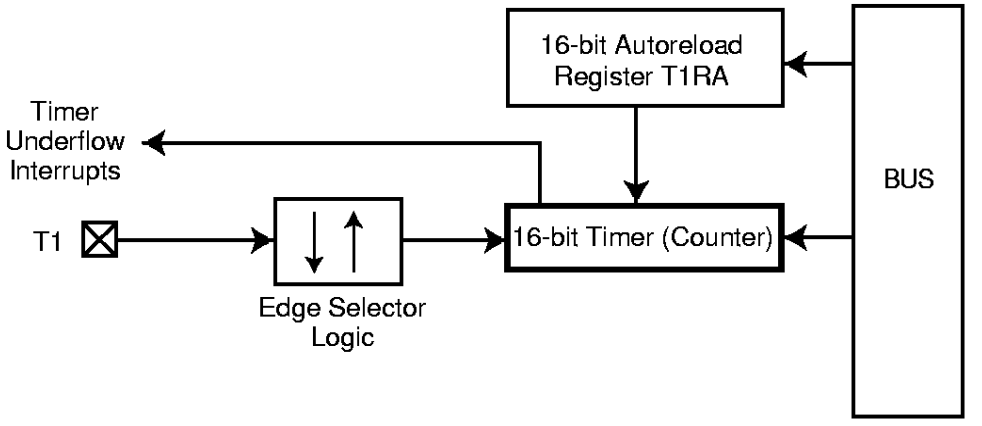
A block diagram of the timer operating in the external event counter mode is shown in Figure 8.

The following steps can be used to operate the timer in the external event counter mode.

1. Configure the T1 pin as an input by clearing bit 2 of PORTGC.
2. Load the initial count into the timer register and the T1RA register. When this 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.
3. In order to generate an interrupt each time the timer underflows, clear the T1PND pending flag and then enable the interrupt by setting the T1EN bit. The G bit must also be set.
4. Write the appropriate value to the timer control bits T1C3-T1C2- T1C1 to select the external event counter mode, and to select the type of transition to be sensed on the T1 pin (positive-going or negative-going; see Table 9).
5. Set the T1C0 bit register to start the timer.

If interrupts are being used, the Timer1 interrupt service routine must clear the T1PND flag and take whatever action is required when the timer underflows. If the user wishes to merely count the number of occurrences of an event, and anticipates that the number of events may exceed 65,536, the interrupt service routine should record the number of underflows by incrementing a counter in memory. On each underflow, the timer (counter) register is reloaded with the value from the T1RA register.

Figure 8: External Event Counter Mode



4.4 Input Capture Mode

In the input capture mode, the T1 pin is configured as input. The timer counts down at the instruction clock rate. A transition received on the T1 pin causes a transfer of the timer contents to the T1RA register. The input signal on T1 must have a pulse width equal to or greater than one instruction cycle. (Refer to the AC Electrical Specifications for this device.) The values captured in the T1RA register at different times reflect the elapsed time between transitions on the T1 pin. The input pin can be configured to sense either positive-going or negative-going transitions.

A block diagram of the timer operating in the input capture mode is shown in Figure 9.

There are two interrupt events associated with the input capture mode: input capture in T1RA and timer underflow. If interrupts are enabled, a Timer1 interrupt is triggered by either an input capture in T1RA or a timer underflow.

In this operating mode, the T1C0 control bit serves as the timer underflow interrupt pending flag. The Timer1 interrupt service routine can look at this flag and the T1PND flag to determine what caused the interrupt. A set T1C0 flag means that a timer underflow occurred, whereas a set T1PND flag means that an input 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 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 timer register counts down continuously at the instruction clock rate, starting from the time that the input capture mode is selected with bits T1C3-T1C2-T1C1. To stop the timer from running, you must change from the input capture mode to the PWM or external event counter mode and reset the T1C0 bit.

The input pins can be independently configured to sense positive-going or negative-going transitions, resulting in two possible input capture mode configurations. The edge sensitivity of pin T1 is controlled by bit T1C1 as indicated in Table 9.

The edge sensitivity of a pin can be changed without leaving the input capture mode by setting or clearing the appropriate control bit (T1C1), 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 timer contents are transferred to the T1RA register, and a Timer1 interrupt is generated. The Timer1 interrupt service routine records the contents of the T1RA register and also reprograms the input capture mode, changing the T1 pin from positive to negative edge sensitivity. When the negative-going edge appears on the T1 pin, another Timer1 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 on the T1 input pin, i.e., the width of the positive pulse.

Remember that the Timer1 interrupt service routine must test the T1C0 and T1PND flags to determine what caused the interrupt.

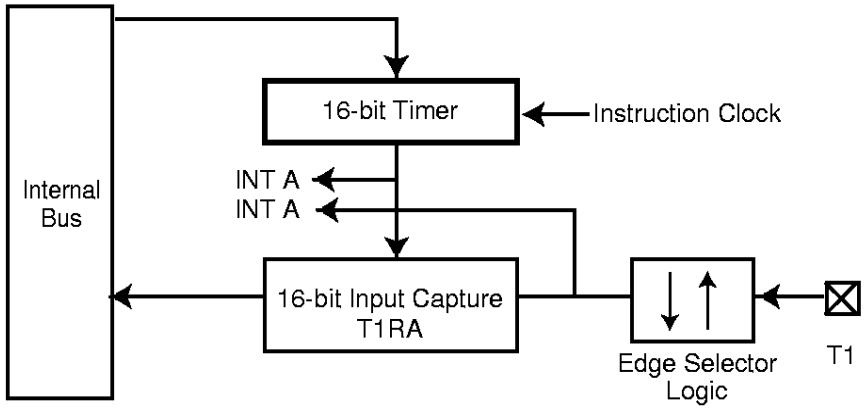
The software that measures elapsed time must take into account the possibility that an underflow occurred between the first and second readings. This can be managed by using the interrupt triggered by each underflow. The Timer1 interrupt service routine, after determining that an underflow caused the interrupt, 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 should check the status of the underflow counter and take it into account in making the calculation.

The following steps can be used to operate the timer in the input capture mode.

1. Configure the T1 pin as input by clearing bit 2 of PORTG2.
2. With the timer configured to operate in the PWM or external event counter mode (T1C2 equal to 0), reset the T1C0 bit. This stops the timer register from counting.
3. Load the initial count into the timer register, typically the value 0xFFFF to allow the maximum possible number of counts before underflow.
4. Clear the T1PND interrupt pending flag, then set the T1EN interrupt enable bit. The G bit should also be set. The interrupt is now enabled.
5. Write the appropriate value to the timer control bits T1C3-T1C2- T1C1 to select the input capture mode, and to select the types of transitions to be sensed on the T1 pin (positive-going or negative-going; see Table 9). As soon as the input capture mode is enabled, the timer starts counting.

When the programmed type of edge is sensed on the T1 pin, the T1RA register is loaded and a Timer1 interrupt is triggered. A Timer1 interrupt is also triggered when an underflow occurs in the timer register. The interrupt service routine tests both the T1PND and T1C0 flags to determine the cause of the interrupt, resets the pending bit, and performs the required task, such as recording the T1RA register contents or incrementing an underflow counter.

Figure 9: Input Capture Mode



4.5 Difference Input Capture Mode

The difference capture mode is similar to the standard capture mode. The difference is the difference capture timer will automatically capture the difference between selectable edges.

For example, a standard capture timer must be configured to capture a particular edge (rising or falling) at which time the timer value is copied into a capture register. If more information is required, software must move the captured data to RAM and reconfigure the capture timer to capture on the next edge (rising or falling). Software must then subtract the difference between the two edges to yield useful information.

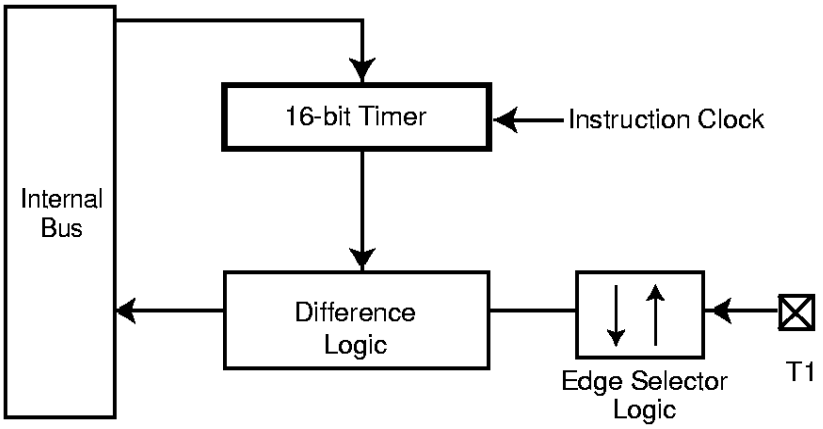
The difference capture timer eliminates the need for software intervention and allows for capturing very short pulse widths. The difference capture timer can be programmed to capture:

1. positive-edge to negative-edge
2. positive-edge to positive-edge
3. negative-edge to positive-edge
4. negative-edge to negative-edge

Once configured, the difference capture timer waits for the selected edge. When an edge transition has occurred, the 16-bit timer starts counting up based on the instruction clock. It will continue to count until the second edge transition occurs at which time the timer stops and stores the elapse time into the T1RA register.

Software can now read the difference between transitions directly without using any processor resources. This feature allows the ACE1202 to capture very small pulses where standard microcontrollers might have missed due to limited bandwidth

Figure 10: Difference Capture Mode



5.0 Timer 0

Timer 0 is a 12-bit timer. On power-up or on any reset, the timer is preset to 0xFFF and then counts down continually every instruction clock cycle. The user cannot read from or write to this timer; however, its interrupts are controlled by reading and writing to the T0CNTRL register (see Figure 11). The counter continues to count after it has underflowed. The IDLE mode uses this counter to produce a reference strobe to generate a signal every 4096 instruction clock cycles to wake the part out of IDLE mode, or to generate a regular stream of interrupts. The counter is also used as a prescaler for the watchdog timer.

The T0INTEN (IDLE timer interrupt enable) bit is a read/write bit. If set to 0, interrupt requests from the IDLE timer are ignored by the control unit. If set to 1, interrupt requests are accepted by the control unit. The T0INTEN bit is reset to zero by a reset.

The T0PND (IDLE timer pending) bit is a read/write bit. If set to 1, it indicates that an IDLE timer interrupt is pending. This bit is set by the timer underflow and reset by software writing a zero to this location.

The WKINTEN bit is used in the Multi-Input Wakeup block. See Section 7 for details.

6.0 Watchdog timer

Timer 0 is used as the clock input to the 4-bit watchdog timer. If the WDEN signal, in the initialization register is asserted, the watchdog timer must be updated at least every 65,536 cycles but no sooner than 2048 cycles since the last watchdog update. The watchdog is updated by the software writing the value 0x1B to the WDSVR register (see Figure 12). The part will be reset automatically if the watchdog is updated too frequent, or not frequent enough. So, if the WDEN bit is enabled in the initialization register 1, the Watchdog will always be powered up enabled. The Watchdog can be disabled by resetting the WDEN bit in the initialization register.

WARNING

Ensure that the Watchdog Timer has been updated before entering idle mode.

Figure 11: Timer 0 Control Register (T0CNTRL)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WKINTEN	x	x	x	x	x	T0PND	T0INTEN

Figure 12: Watchdog Server Register (WDSVR)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	1	0	1	1

7.0 Multi-Input Wakeup Block

There are three memory-mapped registers associated with this circuit: WKEDG (Wakeup Edge), WKEN (Wakeup Enable), and WKPND (Wakeup Pending). Each register has eight bits, with each bit corresponding to one of the input pins shown in Figure 13. All three registers are initialized to zero with a Reset.

The WKEDG register establishes the edge sensitivity for each of the port input pins: either positive-going edges (0) or negative-going edges (1).

The WKEN register enables (1) or disables (0) each of the port pins for the Wakeup/Interrupt function. Any pin to be used for the Wakeup/Interrupt function must also be configured as an input pin in the PORTGC configuration register.

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).

The T0CNTRL register is the Timer0 control register; however, bit 7 (WKINTEN) is used as the wakeup interrupt enable bit (see Figure 17). By setting this bit the device can interrupt in the event of a multi-input wakeup (if the global interrupt bit is set).

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 a Reset because the Wakeup inputs will be set to high-impedence, resulting in unknown data on the port inputs.

1. Clear the WKEN register.
2. If necessary, write to the port configuration register to change the desired port pins from outputs to inputs.
3. Write the WKEDG register to select the desired type of edge sensitivity for each of the pins used.
4. Clear the WKPND register to cancel any pending bits.
5. Set the WKEN bits associated with the pins to be used, thus enabling those pins for the Wakeup/Interrupt function.

Once the Multi-Input Wakeup/Interrupt function has been set up, a transition sensed on any of the enabled pins will set the corresponding bit in the WKPND register. This brings the device out of the HALT or IDLE mode (if in that mode), and also triggers a maskable interrupt if that interrupt is enabled. The interrupt service routine can read the WKPND register to determine which pin triggered the interrupt.

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

After 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.

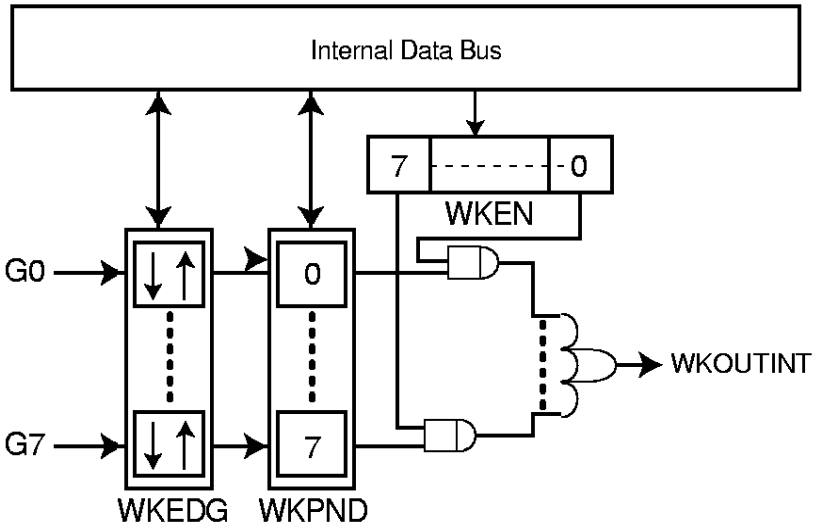
1. Disable the pin by clearing the associated bit in the WKEN register.
2. Write to the associated bit of the WKEDG register selecting the new edge sensitivity of the pin.
3. Clear the WKPND bit associated with the pin.
4. Re-enable the pin by setting the associated WKEN bit.

PORTG provides the user with eight fully selectable, edge sensitive interrupts which are all vectored into the same service subroutine.

The interrupt from PORTG shares logic with the wake up 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 the pending trigger conditions.

Since PORTG is also used for exiting the device from the HALT or IDLE mode, the user can elect to exit the HALT or 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 the enter HALT or IDLE mode instruction). In the other case, the device finishes the instruction which was being executed when the part was stopped (the NOP instruction following the enter HALT or IDLE mode instruction), and then branches to the interrupt service routine. The device then reverts to normal operation.

Figure 13: Multi-input Wakeup (MIWU) Block Diagram



8.0 I/O Port

The eight PORTG pins are bi-directional (see Figure 14) with the exception of PORT G3 which is always an input. The bi-directional I/O pins can individually be configured by software to operate as a high-impedance input, an input with weak pull-up, or as a push-pull output. 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.

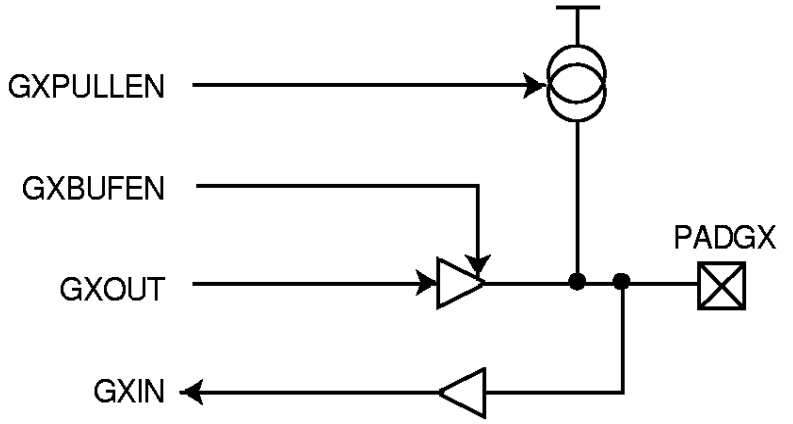
8.1 I/O registers

PORTG has three associated memory mapped port registers; 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 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 10 provides details of the port configuration options. The port configuration and data registers are all read/writeable. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports multi-input wakeup/interrupt, PORTG inputs have Schmitt triggers.

Table 10: I/O configuration options

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

Figure 14: PORTG Logic Diagram



9.0 In-circuit Programming Specification for ACE1202

The ACE1202 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 ACE1202 and the external programmer is made through a 32-bit command and response word described in Table 11.

The serial data timing for the four wire interface is shown in Figure 15. The programming protocol is shown in Figure 16.

The external programmer brings the ACE1202 into programming mode by applying a supervoltage level ($V_{SUPERVOLTAGE}$) to the LOAD pin. The external programmer then needs to set the LOAD pin to V_{CC} before shifting in the 32-bit serial command word using the SHIFT_IN and the CLOCK signals. By definition, bit 31 of the command word is shifted in first. At the same time, the ACE1202 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 on the falling 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 ACE1202, the external programmer must set the LOAD signal to 0V, and then apply two clock pulses as shown in Figure 16. When reading the device, the external programmer must set the LOAD signal to Vcc before it sends a new command word. When writing to the ACE1202, the SHIFT_OUT signal acts as the READY signal. The ACE1202 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 22 are obeyed). The ACE1202 will set the R bit of the Status register when the write operation has been completed. The external programmer must wait for the R bit to go high before bringing the LOAD signal to 5V to initiate a new command cycle.

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

Writing a series of bytes to the ACE1202 is achieved by sending a series of command words with bit 24 set to 0. Reading a series of bytes from the ACE1202 is achieved by sending a series of command words with the desired addresses in sequence and reading the following response words to verify the correct address and reading the data contents.

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

Table 11: 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, 0 otherwise	X
bit 28	Set to 1 to read/write code EEPROM, 0 otherwise	X
bits 27 – 25	Must be set to 0	X
bit 24	Set to 1 to read, 0 to write	X
bits 23 – 20	Must be set to 0	X
bits 19 – 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

Figure 15 - Serial Data Timing

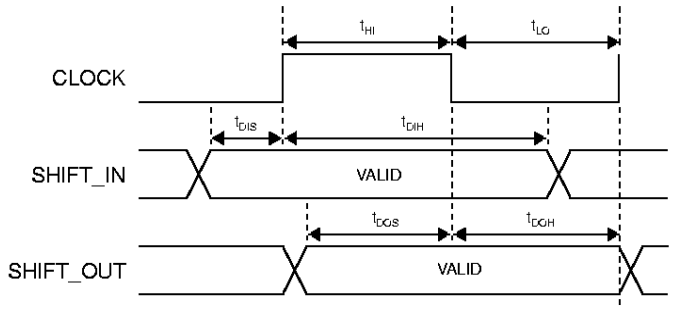
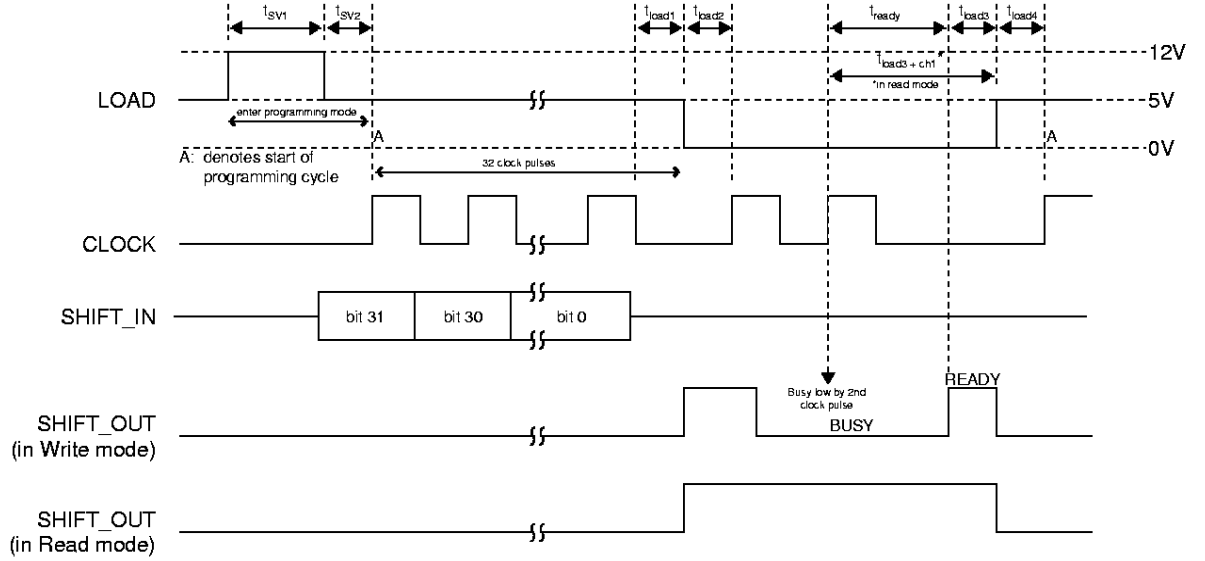


Figure 16 - Programming Protocol



10.0 Low battery detect circuit

The Low Battery Detect (LBD) circuit sets the LBD bit in the LBAT register (see Figure 17) when V_{CC} drops below the selected threshold voltage. The threshold voltage can be adjusted from 2.0V to 2.7V using the three most significant bits of the LBAT register. The LBDEN (Low Battery Detect enable) bit in the initialization register is used to enable or disable the low battery detection.

The LBD bit is read only. If set to 0, it indicates that the V_{CC} level is higher than the desired threshold. If set to 1, it indicates that the V_{CC} level is below the desired threshold.

The LBD circuit is disabled during HALT mode. On exiting HALT mode, the software must wait 10µs before reading the LBD bit to ensure that the circuit has stabilized.

11.0 Brown-out detection circuit

The Brown-out detect circuit is used to reset the device when Vcc falls below a 2.0V threshold. Once V_{CC} rises above the 2.0V threshold, a reset sequence will be generated. The BOREN (Brown-out Reset enable) bit in the initialization register is used to enable or disable the brown-out detection. This bit must be set after the device has been programmed.

12.0 RESET block

When a RESET sequence is initiated, all I/O registers will be reset, setting all I/O's to high impedence inputs. The system clock is restarted after the required clock start-up delay. A reset is generated by any one of the following three conditions:

- Power-on RESET
- Brown-out RESET
- Watchdog RESET (as described in Section 6)
- External RESET

The Power-on RESET circuit is guaranteed to work if the rate of rise of V_{DD} is no slower than 0.06V per ms. It is also necessary that V_{DD} starts from 0V. If using an external crystal, the on-chip Power-on RESET is also not adequate for low frequency crystals which require longer than 20 ms to start-up and stabilize. For such situations, we recommend the external RC circuits are used for a longer Power-on RESET.

13.0 CLOCK AND HALT/IDLE BLOCK

The ACE1202 has an on-board oscillator with a frequency of 2MHz and a tolerance over temperature, voltage, and device of ±10%. On power-up, the on-chip oscillator runs continuously unless entering HALT mode.

If required, an external oscillator circuit may be used depending on the states of the CMODE bits (see Table 12). When the device is driven by an external clock, its input is on the CKI input pin (G1) which can be between DC and 4MHz. The CKO output clock is on pin G0 (crystal configuration). If the internal oscillator of 2MHz, an external crystal of 4MHz, or an external RC of 4MHz is used, it will be divided down to 1MHz by the clock circuit, yielding an instruction clock cycle time of 1µs. However, if an external square clock is used it will not be divided.

The ACE1202 is forced into HALT mode by setting the HALT bit of the HALTCON register (see Figure 18). The clock is stopped, the wake-up circuit is initialized and all on-chip subsystems are held static or shut down completely to minimize power consumption. The device exits HALT mode either through an external reset, or from the multi-input wakeup block. The clock circuit is reactivated and the system clock is restarted after the required clock start-up delay

The EHALT (enable HALT) bit is a write only bit. If set to 0, nothing happens. If set to 1, the part enters HALT mode. It is automatically reset to 0 on leaving HALT mode.

The EIDLE (enable IDLE) bit is a write only bit. If set to 0, nothing happens. If set to 1, the part enters IDLE mode. It is automatically reset to 0 on leaving IDLE mode. Both the EIDLE and the EHALT bits are reset to zero by a reset.

Table 12: CMODEx Bit Definition

CMODE1	CMODE0	Clock Type
0	0	Internal 1 MHz clock
0	1	External square clock
1	0	External crystal/resonator
1	1	External RC clock

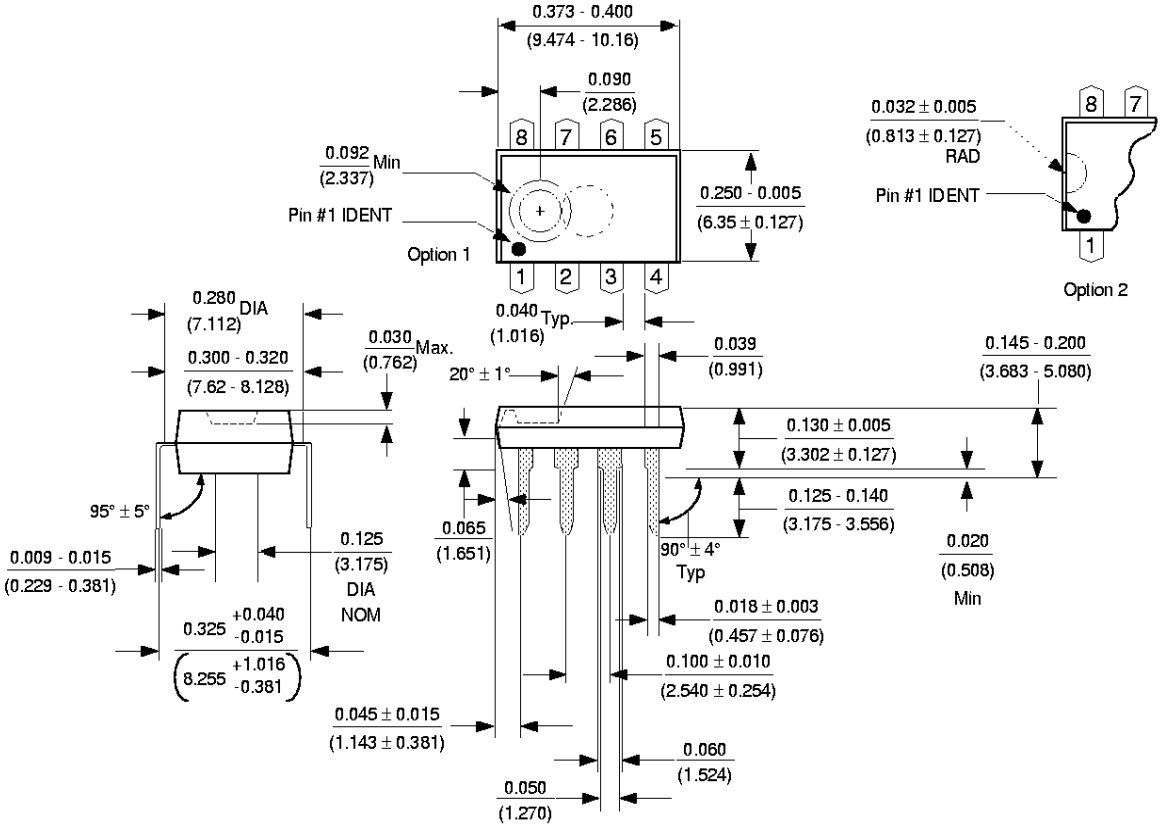
Figure 17: LBAT Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bat_trim2	Bat_trim1	Bat_trim0	undefined	undefined	Undefined	undefined	LBD

Figure 18: HALTCON Register Definition

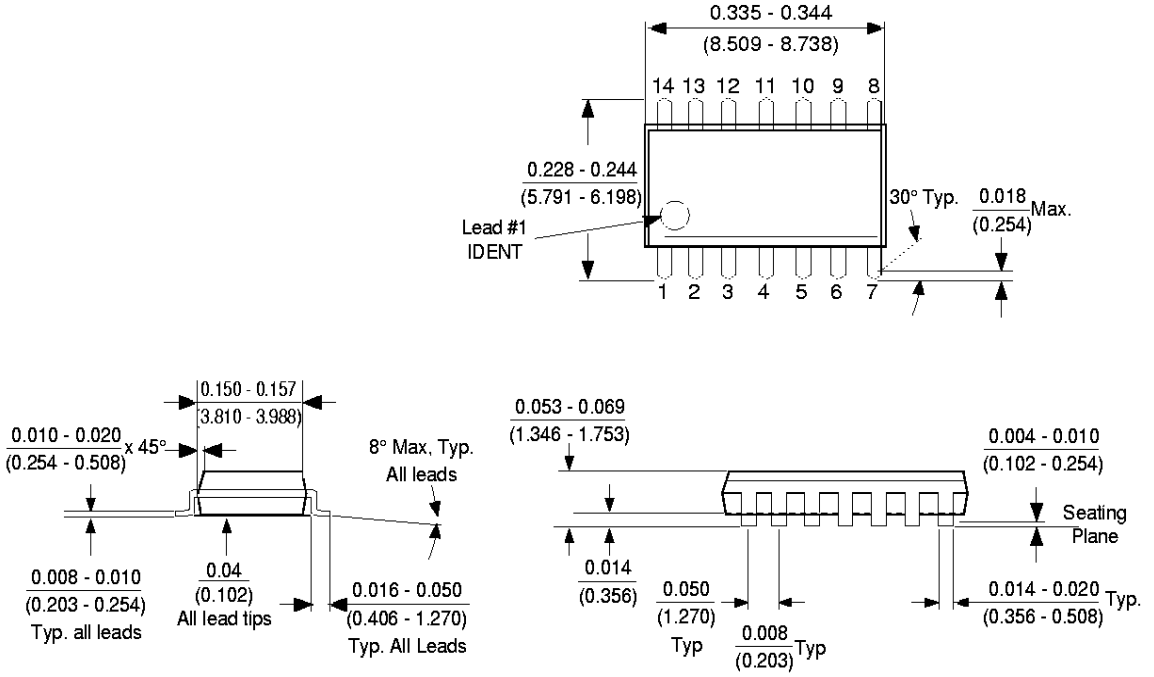
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
undefined	undefined	undefined	undefined	undefined	undefined	EIDLE	EHALT

Physical Dimensions inches (millimeters) unless otherwise noted



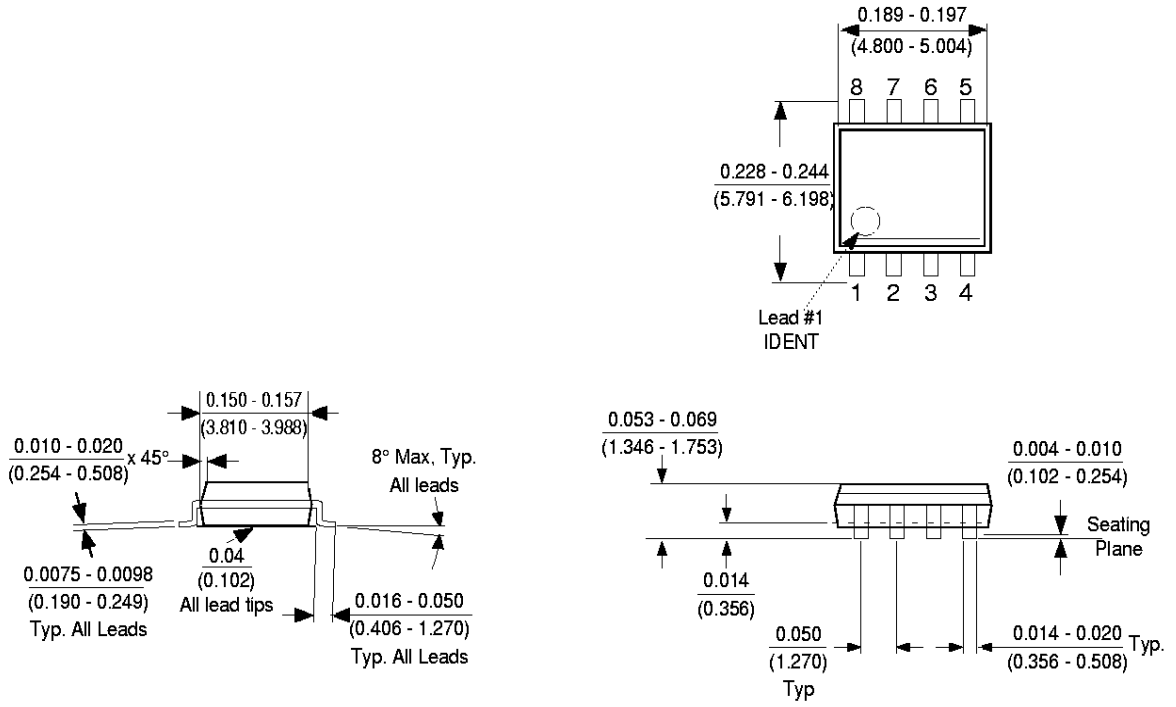
Molded Dual-In-Line Package (N)
Order Number ACE1202N8/ACE1202EN8/ACE1202VN8
Package Number N08E

Physical Dimensions inches (millimeters) unless otherwise noted



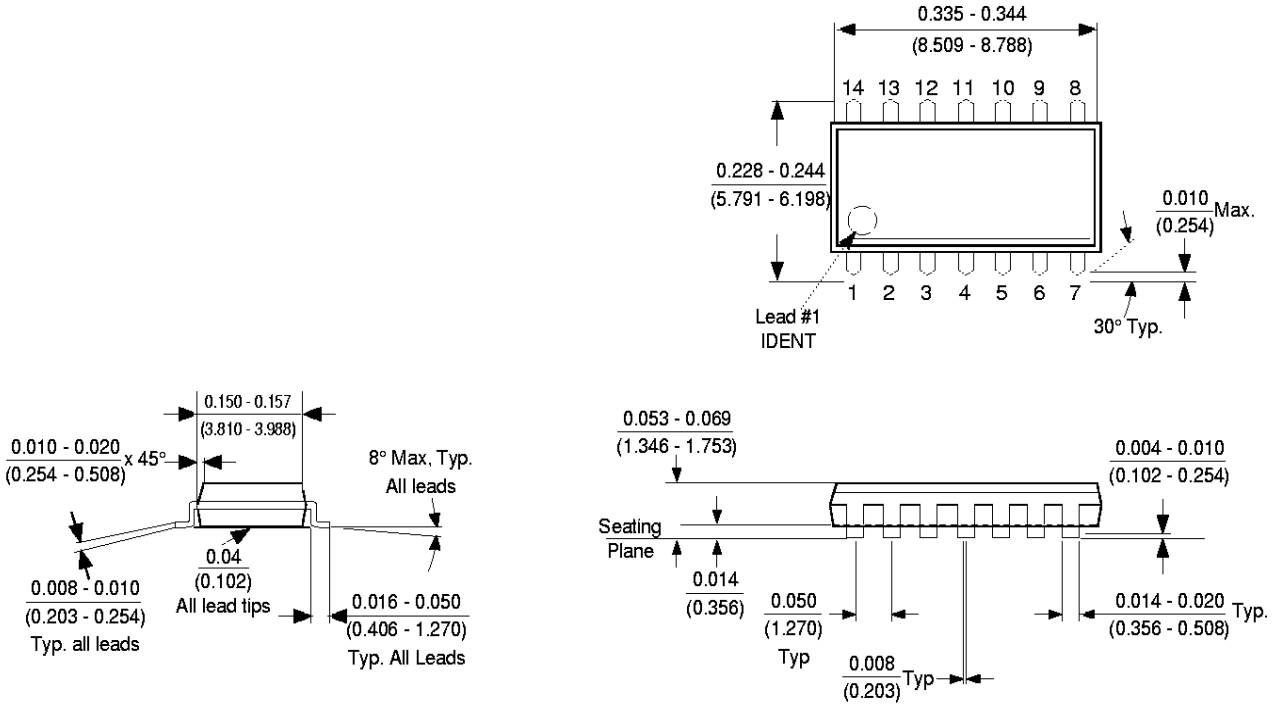
Molded Dual-In-Line Package (N)
Order Number ACE1202N/ACE1202EN/ACE1202VN
Package Number N14A

Physical Dimensions inches (millimeters) unless otherwise noted



Molded Small Out-Line Package (M8)
Order Number ACE1202M8/ACE1202EM8/ACE1202VM8
Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted



Molded Small Out-Line Package (M)
Order Number ACE1202M/ACE1202EM/ACE1202VM
Package Number M14A

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
Fax +44 (0) 1793-856858
Deutsch 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
2-16-6, Yushima, Bunkyo-ku
Tokyo, 113-0034 Japan
Tel 81-3-3818-8840
Fax 81-3-3818-8841