Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

FAIRCHILD SEMICONDUCTOR®

PRELIMINARY
August 2001

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

General Description

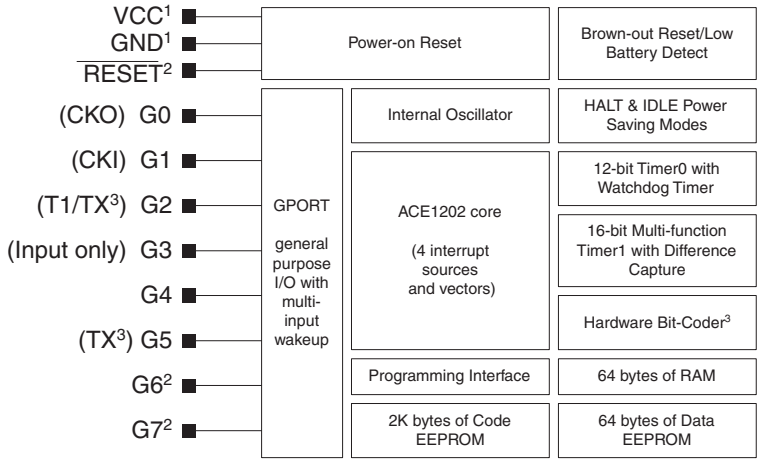
The ACE1202 (Arithmetic Controller Engine) family of microcontrollers is a dedicated programmable monolithic integrated circuit for applications requiring high performance, low power, and small size. It is a fully static part fabricated using CMOS technology.

The ACE1202 product family 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. The on-chip clock and reset functions reduce the number of required external components. The ACE1202 product family is available in 8- and 14-pin SOIC and DIP packages.

Features

- Arithmetic Controller Engine
- 2K bytes on-board code EEPROM
- 64 bytes data EEPROM
- 64 bytes RAM
- Instruction set geared for block encryption
- Watchdog
- Multi-input wake-up on all I/O pins
- 16-bit multifunction timer with difference capture
- 12-bit idle timer
- Hardware Bit - Coder (HBC) (ACE1202-2 only)
- On-chip oscillator
  - — No external components
  - — 1µs instruction cycle time
- On-chip Power-on Reset
- Programmable read and write disable functions
- Memory mapped I/O
- Multilevel Low Voltage Detection
- Brown-out Reset
- Software selectable I/O option
  - — Push-pull outputs with tri-state option
  - — Weak pull-up or high impedance
- Fully static CMOS
  - — Low power HALT mode (100nA @ 3.3V)
  - — Power saving IDLE mode
- Single supply operation
  - — 1.8-5.5V (P.N. ACE1202L)
  - — 2.2-5.5V (P.N. ACE1202, ACE12022)
  - — 2.7-5.5V (P.N. ACE1202B, ACE12022B)
- 40 years data retention
- 1,000,000 data changes
- 8 and 14-pin SOIC, 8 and 14-pin DIP packages. (CSP package available upon request)
- In-circuit programming

Block and Connection Diagram



1. 100nf Decoupling capacitor recommended
2. Available only in the 14-pin package option
3. Available only on the ACE1202-2 device

Figure 2: ACEx Application Example (Remote Keyless Entry)

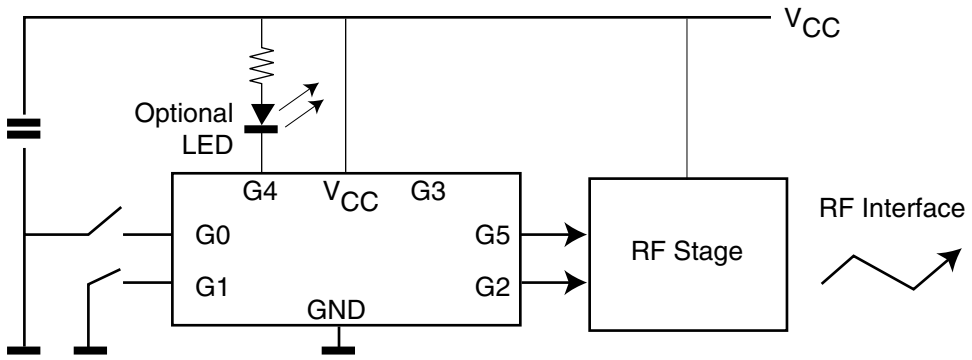
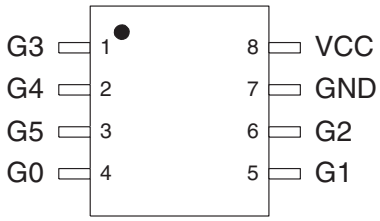


Figure 3: ACE1202/ACE1202-2 8-pin Device Pinout

a) Normal Mode Operation



b) Programming Mode Operation

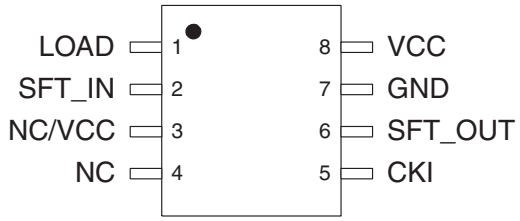
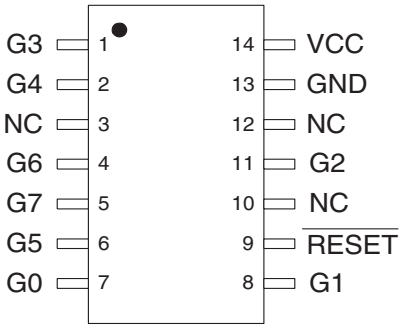
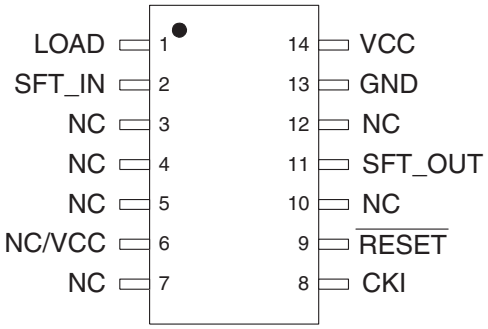


Figure 4: ACE1202/ACE1202-2 14-pin Device Pinout

a) Normal Mode Operation



b) Programming Mode Operation



2.0 Electrical Characteristics

Absolute Maximum Ratings

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

Operating Conditions

Relative Humidity (non-condensing)	95%
EEPROM write limits	See DC Electrical Characteristics

Part Number	Operating Voltage	Ambient Operating Temperature
ACE1202	2.2 to 5.5V	0°C to 70°C
ACE12022	2.2 to 5.5V	0°C to 70°C
ACE1202E	2.2 to 5.5V	-40°C to +85°C
ACE12022E	2.2 to 5.5V	-40°C to +85°C
ACE1202V	2.2 to 5.5V	-40°C to +125°C
ACE1202B	2.7 to 5.5V	0°C to 70°C
ACE12022B	2.7 to 5.5V	0°C to 70°C
ACE1202BE	2.7 to 5.5V	-40°C to +85°C
ACE12022BE	2.7 to 5.5V	-40°C to +85°C
ACE1202BV	2.7 to 5.5V	-40°C to +125°C
ACE12022BV	2.7 to 5.5V	-40°C to +125°C
ACE1202L	1.8 to 5.5V	0°C to 70°C

Preliminary ACE1202/ACE1202-2 DC Electrical Characteristics

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

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

Symbol	Parameter	Conditions	MIN	TYP	MAX	Units
I_{CC}[4]	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}[5]	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

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

[5] Based on a continuous IDLE looping program.

Preliminary ACE1202/ACE1202-2 AC Electrical Characteristics

V_{CC} = 1.8/2.2/2.7 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 6)			4	MHz
External clock frequency	(Note 7)			4	MHz
EEPROM write time			3	10	ms
Internal clock start up time	(Note 7)			2	ms
Oscillator start up time	(Note 7)			2400	cycles

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

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

Preliminary ACE1202/ACE1202-2 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 ACE1202/ACE1202-2 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 ACE1202/ACE1202-2 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 ACE1202L 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

3.0 AC & DC Electrical Characteristic Graphs

Figure 5: RC Oscillator Frequency vs. Temperature (V_{CC}=5.0V)

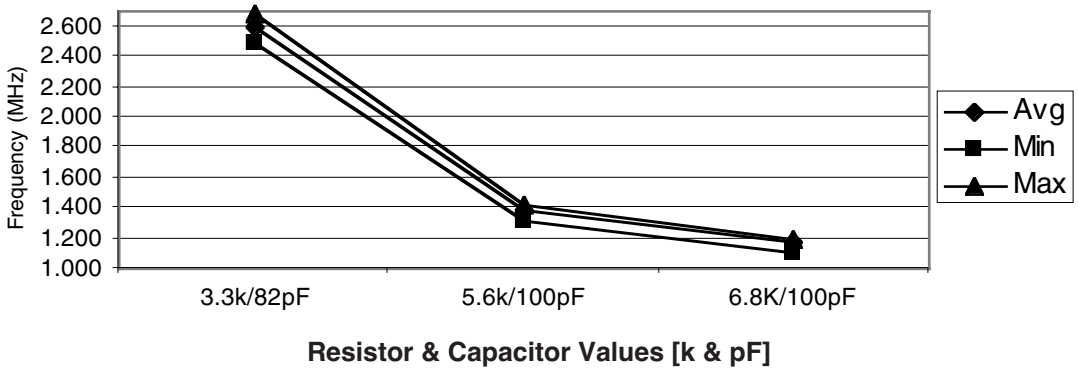


Figure 6: RC Oscillator Frequency vs. Temperature(V_{CC}=2.5V)

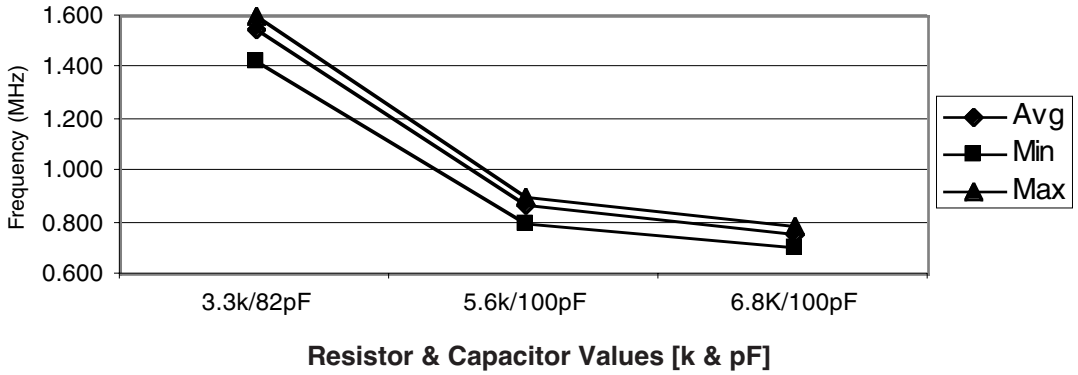


Figure 7: Internal Oscillator Frequency

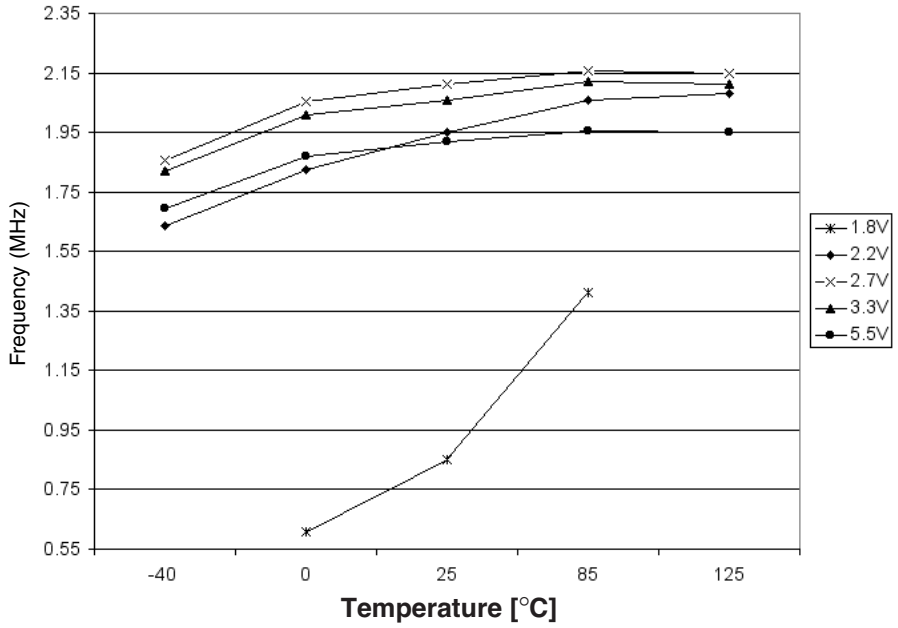
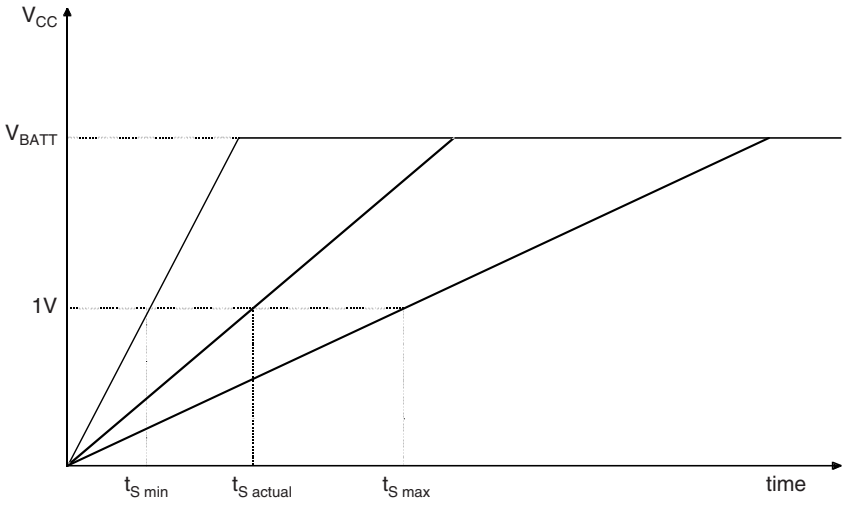
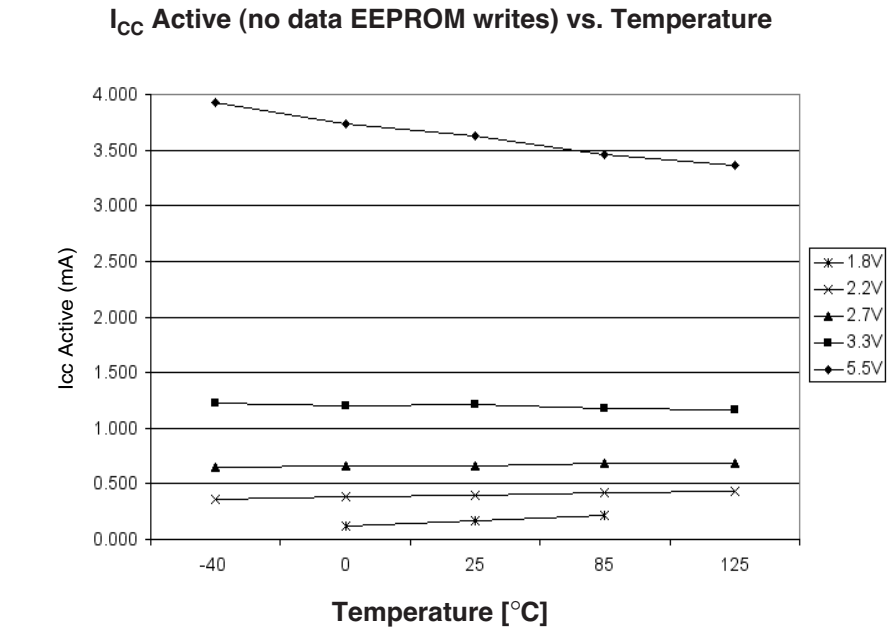


Figure 8: Power Supply Rise Time



Name	Parameter	Unit
V_{CC}	Supply Voltage	[V]
V_{BATT}	Battery Voltage (Nominal Operating Voltage)	[V]
$t_{S\ min}$	Minimum Time for V_{CC} to Rise by 1V	[ms]
$t_{S\ actual}$	Actual Time for V_{CC} to Rise by 1V	[ms]
$t_{S\ max}$	Maximum Time for V_{CC} to Rise by 1V	[ms]
S_{VCC}	Power Supply Slope	[ms/V]

Figure 9: I_{CC} Active



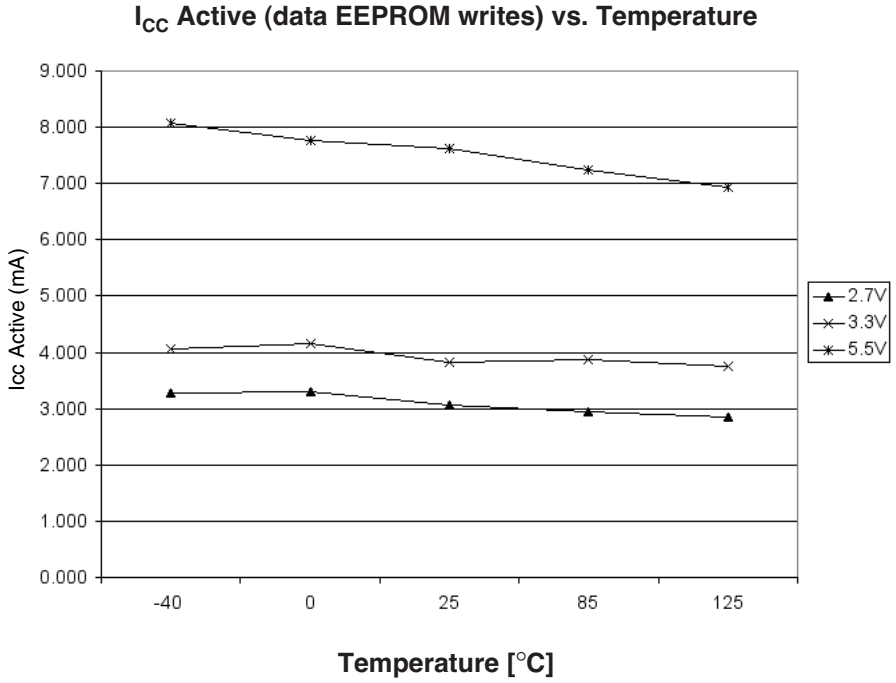


Figure 10: HALT Mode Currents

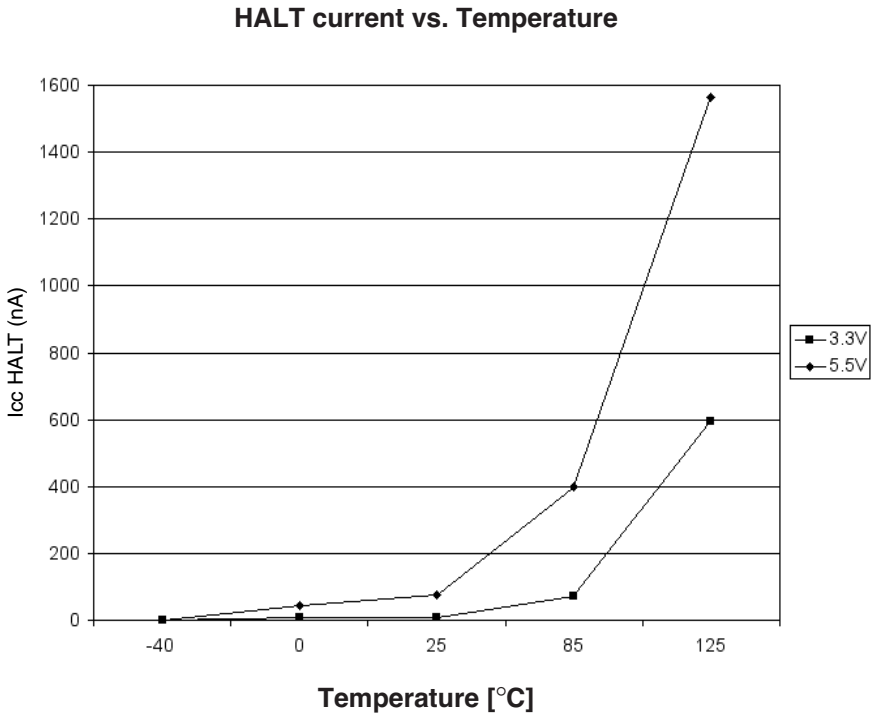
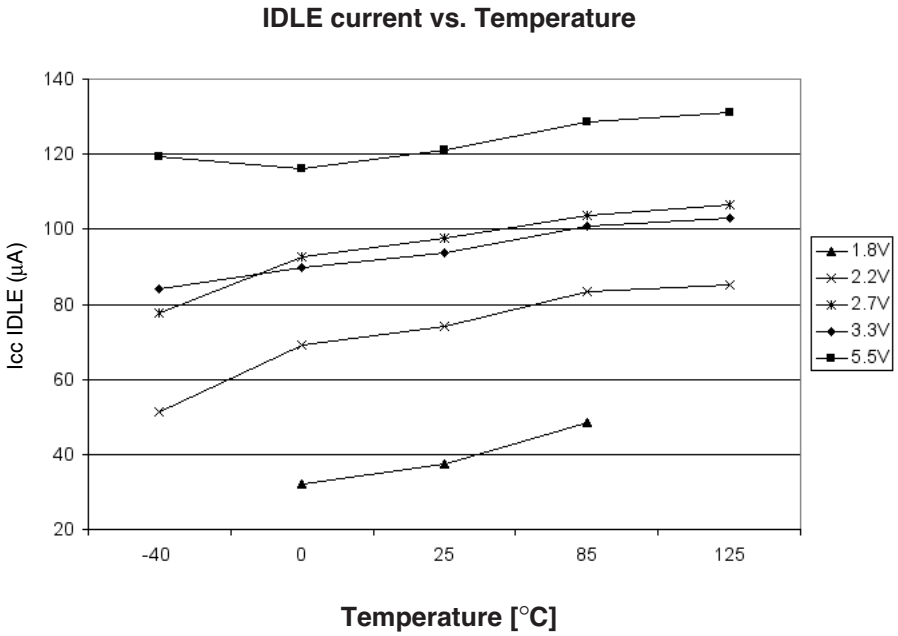


Figure 11: IDLE Mode Currents



4.0 Arithmetic Controller Core

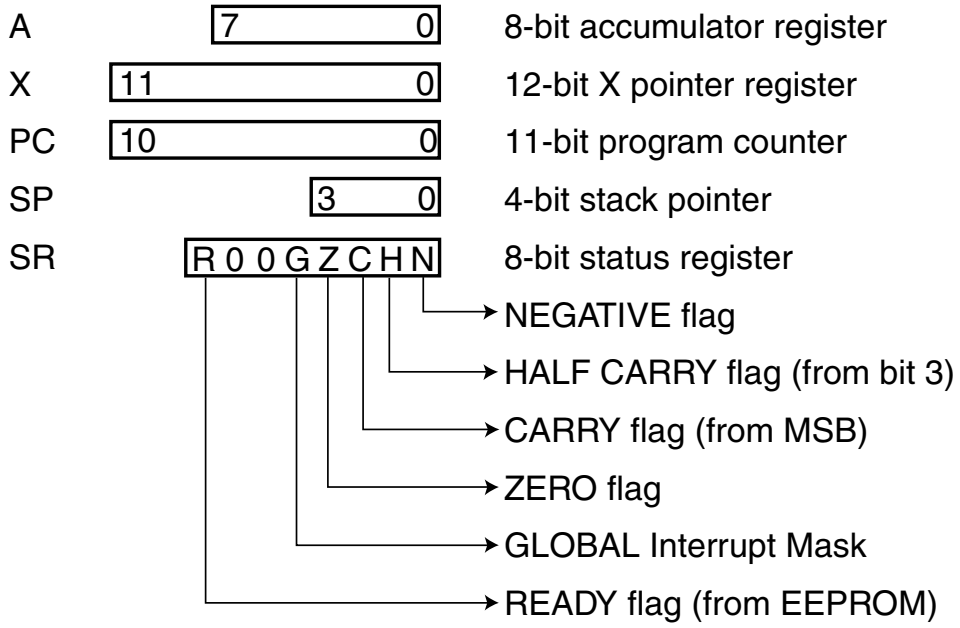
The ACEx microcontroller core is specifically designed for low cost applications involving bit manipulation, shifting and block encryption. It is based on a modified Harvard architecture meaning peripheral, I/O, and RAM locations are addressed separately from instruction data.

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 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 a 12-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 12-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[11:8]) and XLO (X[7:0]) variables located at 0xBE and 0xBF, respectively. (See Table 11)

The X register is divided into two sections. The 11 least significant bits (LSBs) 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 (0x800 to 0xFFF) memory space.

Example: If Bit 11 = 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 11 = 1, then the LD A, [00,X] instruction will take a value from address range 0x800 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 11-LSBs since the 12th 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 0x800.

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)

The 8-bit Status register (SR) contains four condition code indicators (C, H, Z, and N), one interrupt masking bit (G), and an EEPROM write flag (R). The 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 continues at the address specified by the unique interrupt vector (see Table 11). This process takes five instruction cycles. At the end of the interrupt service routine, a return from interrupt (RETI) instruction is executed. The RETI instruction causes the saved address to be pulled off the stack in reverse order. The G bit is set and instruction execution resumes at the return address.

Table 8: Interrupt Priority Sequence

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

The ACEx microcontroller is capable of supporting four interrupts. Three are maskable through the G bit of the SR and the fourth (software interrupt) is not inhibited by the G bit (see Figure 13). The software interrupt instruction is generated by the execution of the INTR instruction. once the INTR instruction is executed, the ACEx core will interrupt whether the G bit is set or not. The INTR interrupt is executed in the same manner as the other maskable interrupts where the program counter register is stacked and the G bit is cleared. This means, if the G bit was enabled prior to the software interrupt the RETI instruction must be used to return from interrupt in order to restore the G bit to its previous state. However, if the G bit was not enabled prior to the software interrupt the RET instruction must be used.

In case of multiple interrupts occurring at the same time, the ACEx microcontroller core has prioritized the interrupts. The interrupt priority sequence in shown in Table 8.

4.3 Addressing Modes

The ACEx microcontroller has seven addressing modes indexed, indirect, direct, immediate, absolute jump, and relative jump.

Indexed

The instruction allows an 8-bit unsigned offset value to be added to the 11-LSBs of the X-pointer yielding a new effective address. This mode can be used to address either data or program memory space.

Indirect

The instruction allows the X-pointer to address any location within the data memory space.

Direct

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

Immediate

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

Inherent

This instruction has no operands associated with it.

Absolute

The instruction contains a 11-bit address that directly points to a location in the program memory space. There are two operands associated with this addressing mode. Each operand contains a byte of an address. This mode is used only for the long jump (JMP) and JSR instructions.

Relative

This mode is used for the short jump (JP) instructions where the operand is a value relative to the current PC address. With this instruction, software is limited to the number of bytes it can jump, -31 or +32.

Figure 13: Basic Interrupt Structure

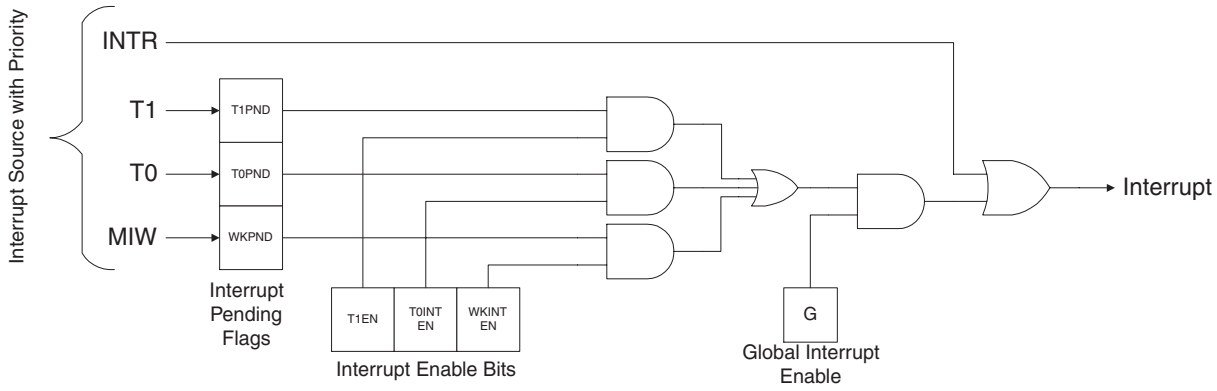


Table 9: Instruction Addressing Modes

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, [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					[00,X]					M
JMP					[00,X]					M
RET							no-op			
RETI							no-op			
INTR							no-op			

Table 10: Instruction Cycles and Bytes

Mnemonic	Operand	Bytes	Cycles	Flags affected
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, [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, M	2	2	Z,N
AND	A, [X]	1	1	Z,N
CLR	X	1	1	Z
CLR	A	1	1	C,H,Z,N
CLR	M	2	2	C,H,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	3	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	3	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	3	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
INVC		1	1	C

Mnemonic	Operand	Bytes	Cycles	Flags affected
JMP	M	3	4	None
JMP	[00,X]	2	3	None
JP		1	1	None
JSR	M	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, [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,H
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,H
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, [X]	1	1	C,H,Z,N
SUBC	A, M	2	2	C,H,Z,N
XOR	A, #	2	2	Z,N
XOR	A, [X]	1	1	Z,N
XOR	A, M	2	2	Z,N

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
0xA0	Data	HBC	HBCNTRL register (ACE1202-2 only)
0xA1	Data	HBC	PSCALE register (ACE1202-2 only)
0xA2	Data	HBC	HPATTERN register (ACE1202-2 only)
0xA3	Data	HBC	LPATTERN register (ACE1202-2 only)
0xA4	Data	HBC	BPSEL register (ACE1202-2 only)
0xA9	Data	HBC	DAT0 register (ACE1202-2 only)
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)
0x800 - 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

4.5 Memory

The ACEx microcontroller device has 64 bytes of SRAM and 64 bytes of EEPROM available for data storage. The device also has 2K bytes of EEPROM for program storage. Software can read and write to SRAM and data EEPROM but can only read from the code EEPROM. While in normal mode, the code EEPROM is protected from any writes. The code EEPROM can only be rewritten when the device is in program mode and if the write disable (WDIS) bit of the initialization register is not set to 1.

While in normal mode, the user can write to the data EEPROM array by 1) polling the ready (R) flag of the SR, then 2) executing the appropriate instruction. If the R flag is 1, the data EEPROM block is ready to perform the next write. If the R flag is 0, the data EEPROM is busy. The data EEPROM array will reset the R flag after the completion of a write cycle. Attempts to read, write, or enter HALT/IDLE mode while the data EEPROM is busy (R = 0) can affect the current data being written.

4.6 Initialization Registers

The ACEx microcontroller has two 8-bit wide initialization registers. These registers are read from the memory space on power-up to initialize certain on-chip peripherals. Figure 14 provides a detailed description of Initialization Register 1. The Initialization Register 2 is used to trim the internal oscillator to its appropriate frequency. This register is pre-programmed in the factory to yield an internal instruction clock of 1MHz.

Both Initialization Registers 1 and 2 can be read from and written to during programming mode. However, re-trimming the internal oscillator (writing to the Initialization Register 2) once it has left the factory is *discouraged*.

Figure 14: Initialization Register 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMODE[0]	CMODE[1]	WDEN	BOREN	BLSEL [10]	UBD [8,9]	WDIS [8,9]	RDIS [8,9]

(0) RDIS [8,9] — If set, disables attempts to read the contents from the memory while in programming mode

(1) WDIS [8,9] — If set, disables attempts to write new contents to the memory while in programming mode

(2) UBD [8,9] — If set, the device will not allow any writes to occur in the upper block of data EEPROM (0x60-0x7F)

(3) BLSEL [10] — If set, the Brown-out Reset (BOR) voltage reference level is set to its higher range for P.N. ACE1202/ACE12022 If not set, the BOR voltage reference level is set to its lower range for P.N. ACE1202L

(4) BOREN — If set, allows a BOR to occur if VCC falls below the voltage reference level

(5) WDEN — If set, enables the on-chip processor watchdog circuit

(6) CMODE[1] — Clock mode select bit 1 (See Table 17)

(7) CMODE[0] — Clock mode select bit 0 (See Table 17)

[8] If both the WDIS and RDIS bits are set, the device will no longer be able to be placed into program mode.

[9] If the RDIS or UBD bits are not set while the WDIS bit is not set, then the RDIS and UBD bits can be reset.

[10] The BLSEL bit is set to its appropriate level in the factory. If writing to the initialization register is necessary, be sure to maintain BLSEL set value.

5.0 Timer 1

Timer 1 is a versatile 16-bit timer that 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 measures the difference between edges

Timer 1 contains a 16-bit timer/counter register (TMR1), a 16-bit auto-reload/capture register (T1RA), and an 8-bit control register (T1CNTRL). All register are memory-mapped for simple access through the core with both the 16-bit registers organized as a pair of 8-bit register bytes {TMR1HI, TMR1LO} and {T1RAHI, T1RALO}. Depending on the operating mode, the timer contains an external input or output (T1) that is multiplexed with the I/O pin G2. By 12 and 13.

default, the TMR1 is reset to 0xFFFF, T1RA is reset to 0x0000, and T1CNTRL is reset to 0x00.

The timer can be started or stopped through the T1CNTRL register bit T1C0. When running, the timer counts down (decrements) every clock cycle. Depending on the operating mode, the timer's clock is either the instruction clock or a transition on the T1 input. In addition, occurrences of timer underflow (transitions from 0x0000 to 0xFFFF/T1RA value) can either generate an interrupt and/or toggle the T1 output pin.

Timer 1's interrupt (TMRI1) can be enabled by interrupt enable (T1EN) bit in the T1CNTRL register. When the timer interrupt is enabled, depending on the operating mode, the source of the interrupt is a timer underflow and/or a timer capture.

5.1 Timer control bits

Reading and writing to the T1CNTRL register controls the timer's operation. By writing to the control bits, the user can enable or disable the timer interrupts, set the mode of operation, and start or stop the timer. The T1CNTRL register bits are described in Tables

Table 12: TIMER1 Control Register (T1CNTRL)

T1CNTRL Register	Name	Function
Bit 7	T1C3	Timer TIMER1 control bit 3 (see Table 13)
Bit 6	T1C2	Timer TIMER1 control bit 2 (see Table 13)
Bit 5	T1C1	Timer TIMER1 control bit 1 (see Table 13)
Bit 4	T1C0	Timer TIMER1 run: 1 = Start timer, 0 = Stop timer; or Timer TIMER1 underflow interrupt pending flag in input capture mode
Bit 3	T1PND	Timer1 interrupt pending flag: 1 = Timer1 interrupt pending, 0 = Timer1 interrupt not pending
Bit 2	T1EN	Timer1 interrupt enable bit: 1 = Timer1 interrupt enabled, 0 = Timer1 interrupt disabled
Bit 1	M4S1	Capture type: 0 = Pulse capture, 1 = Cycle capture (see Table 13)
Bit 0	----------	Reserved

Table 13: TIMER1 Operating Modes

T1 C3	T1 C2	T1 C1	M4 S1	Timer Mode Source	Interrupt A	Timer Counts On
0	0	0	x	MODE 2	TIMER1 Underflow	T1 Pos. Edge
0	0	1	x	MODE 2	TIMER1 Underflow	T1 Neg. Edge
1	0	1	x	MODE 1 T1 Toggle	Autoreload T1RA	Instruction Clock
1	0	0	x	MODE 1 No T1 Toggle	Autoreload T1RA	Instruction Clock
0	1	0	x	MODE 3 Captures: T1 Pos. edge	Pos. T1 Edge	Instruction Clock
0	1	1	x	MODE 3 Captures: T1 Neg. Edge	Neg. T1 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

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.

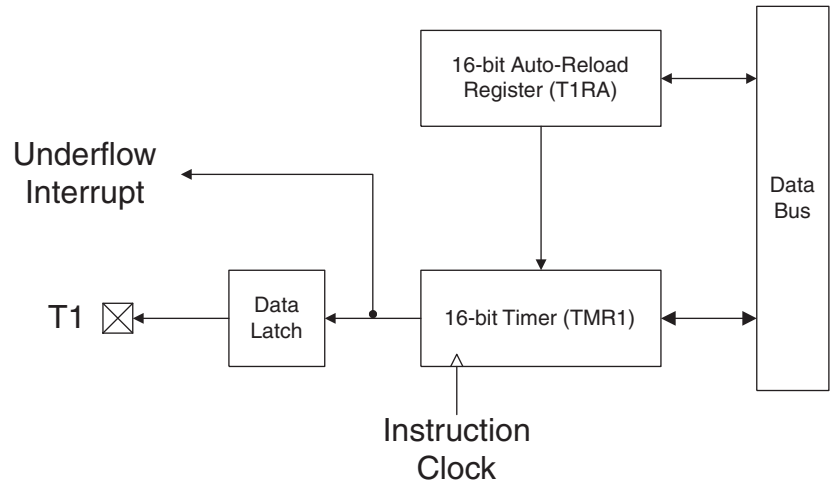
The timer has one interrupt (TMRI1) that is maskable through the T1EN bit of the T1CNTRL register. However, the core is only interrupted if the T1EN bit and the G (Global Interrupt enable) bit of the SR is set. If interrupts are enabled, the timer will generate an interrupt each time T1PND flags is set (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 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



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.

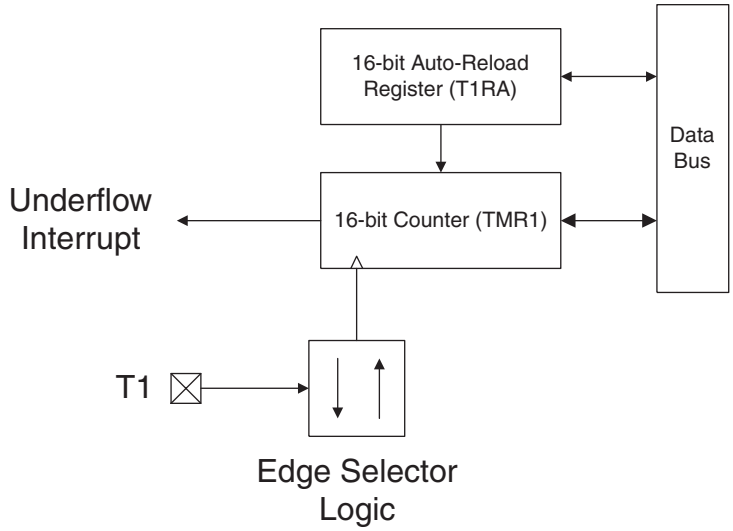
As with the PWM mode, when the counter underflows the counter is reloaded from the T1RA register and the count down proceeds from the loaded value. At every underflow, a pending flag (T1PND) located in the T1CNTRL register is set. Software must then clear the T1PND flag and can then load the T1RA register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



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 12 and 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.

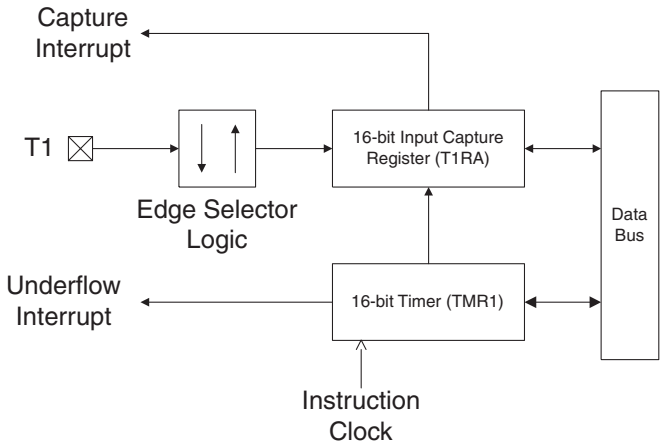
For example, the T1 pin can be programmed to be sensitive to a positive-going edge. When the positive edge is sensed, the TMR1 register contents is transferred to the T1RA register and a Timer 1 interrupt is generated. The Timer 1 interrupt service routine records the contents of the T1RA register, changes 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 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



5.5 Mode 4: Difference Input Capture Mode

The Difference Input Capture mode works similarly to the standard Input Capture mode. However, for the Difference Input Capture the timer automatically captures the elapsed time between the selected edges without the core needing to perform the calculation.

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

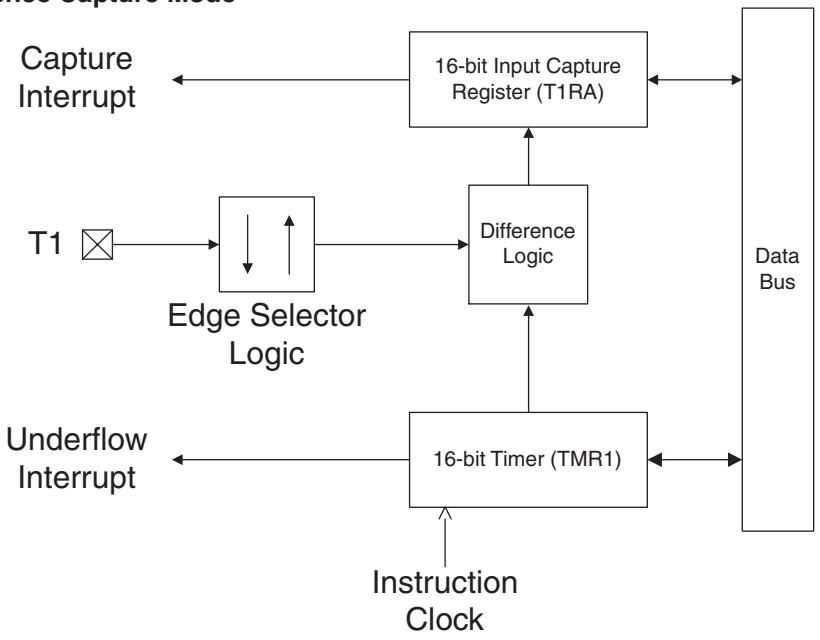
The Difference Capture mode eliminates the need for software intervention and allows for capturing very short pulse or cycle widths. it can be configured to capture the elapsed time between:

1. positive to negative-going edges
2. positive to positive-going edges
3. negative to positive-going edges
4. negative to negative-going edges

Once configured, the Difference Capture timer waits for the first selected edge. when the edge transition has occurred, the 16-bit timer starts counting up based every instruction clock cycle. It will continue to count until the second selected edge transition occurs at which time the timer stops and stores the elapse time into the T1RA register.

Software can now read the differnce between transitions directly without using any processor resources. However, like the standard Input Capture mode both the capture (T1PND) and the underflow (T1C0) flags must be monitored and handled appropriately. This feature allows the ACEx microcontroller to capture very small pulses where standard microcontrollers might have missed cycles due to the limited bandwidth.

Figure 18: Difference Capture Mode



6.0 Timer 0

Timer 0 is a 12-bit free running idle timer. Upon power-up or any reset, the timer is reset to 0x000 and then counts up continuously based on the instruction clock of 1MHz (1 μs). Software cannot read from or write to this timer. However, software can monitor the timer's pending (T0PND) bit that is set every 8192 cycles (initially 4096 cycles after a reset). The T0PND flag is set every other time the timer overflows (transitions from 0xFFF to 0x000) through a divide-by-2 circuit. After an overflow, the timer will reset and restart its counting sequence.

Software can either poll the T0PND bit or vector to an interrupt subroutine. In order to interrupt on a T0PND, software must be sure to enable the Timer 0 interrupt enable (T0INTEN) bit in the Timer 0 control (T0CNTRL) register and also make sure the G bit is set in SR. Once the timer interrupt is serviced, software should reset the T0PND bit before exiting the routine. Timer 0 supports the following functions:

1. Exiting from IDLE mode (See Section 17.0 for details.)
2. Start up delay from HALT mode
3. Watchdog pre-scaler (See Section 7.0 for details.)

The T0INTEN bit is a read/write bit. If set to 0, interrupt requests from the Timer 0 are ignored. If set to 1, interrupt requests are accepted. Upon reset, the T0INTEN bit is reset to 0.

The T0PND bit is a read/write bit. If set to 1, it indicates that a Timer 0 interrupt is pending. This bit is set by a Timer 0 overflow and is reset by software or system reset.

The WKINTEN bit is used in the Multi-input Wakeup/Interrupt block. See Section 8 for details.

7.0 Watchdog timer

The Watchdog timer is used to reset the device and safely recover in the rare event of a processor "runaway condition." The 12-bit Timer 0 is used as a pre-scaler for Watchdog timer. The Watchdog timer must be serviced before every 61,440 cycles but no sooner than 4096 cycles since the last Watchdog reset. The Watchdog is serviced through software by writing the value 0x1B to the Watchdog Service (WDSVR) register (see Figure 20). The part resets automatically if the Watchdog is serviced too frequent, or not frequent enough.

The Watchdog timer must be enabled through the Watchdog enable bit (WDEN) in the initialization register. The WDEN bit can only be set while the device is in programming mode. Once set, the Watchdog will always be powered-up enabled. Software cannot disable the Watchdog. The Watchdog timer can only be disabled in programming mode by resetting the WDEN bit as long as the memory write protect (WDIS) feature is not enabled.

WARNING

Ensure that the Watchdog timer has been serviced before entering IDLE mode because it remains operational during this time.

Figure 19: 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 20: 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

8.0 Hardware Bit-Coder (ACE1202-2 only)

The ACE1202-2 contains a dedicated hardware bit-encoding peripheral block, Hardware Bit-Coder (HBC), for IR/RF data transmission (see Figure 21.) The HBC is completely software programmable and can be configured to emulate various bit-encoding formats. The software developer has the freedom to encode each bit of data into a desired pattern and output the encoded data at the desired frequency through either the G2 or G5 output (TX) ports.

The HBC contains six 8-bit memory-mapped configuration registers PSCALE, HPATTERN, LPATTERN, BPSEL, HBCNTRL, and DAT0. The registers are used to select the transmission frequency, store the data bit-encoding patterns, configure the data bit-pattern/frame lengths, and control the data transmission flow.

To select the IR/RF transmission frequency, an 8-bit divide constant must be written into the IR/RF Pre-scalar (PSCALE) register. The IR/RF transmission frequency generator divides the 1MHz instruction clock down by 4 and the PSCALE register is used to select the desired IR/RF frequency shift. Together, the transmission frequency range can be configured between 976Hz (PSCALE = 0xFF) and 125kHz (PSCALE = 0x01). Upon a reset, the PSCALE register is initialized to zero disabling the IR/RF transmission frequency generator. However, once the PSCALE register is programmed, the desired IR/RF frequency is maintained as long as the device is powered.

Once the transmission frequency is selected, the data bit-encoding patterns must be stored in the appropriate registers. The HBC contains two 8-bit bit-encoding pattern registers, High-pattern (HPATTERN) and Low-pattern (LPATTERN). The encoding pattern stored in the HPATTERN register is transmitted when the data bit value to be encoded is a 1. Similarly, the pattern stored in the LPATTERN register is transmitted when the data bit value to be encoded is a 0. The HBC transmits each encoded pattern MSB first.

The number of bits transmitted from the HPATTERN and LPATTERN registers is software programmable through the Bit Period Configuration (BPSEL) register (see Figure 22). During the transmission of HPATTERN, the number of bits transmitted is configured by BPH[2:0] (BPSEL[2:0]) while BPL[2:0] (BPSEL[5:3]) configures the number of transmitted bits for the LPATTERN. The HBC allows from 2 (0x1) to 8 (0x7) encoding pattern bits to be transmitted from each register. Upon a reset, BPSEL is initially 0 disabling the HBC from transmitting pattern bits from either register.

The Data (DAT0) register is used to store up to 8 bits of data to be encoded and transmitted by the HBC. This data is shifted, bit by bit, MSB to LSB into a 1-bit decision register. If the active bit shifted into the decision register is 1, the pattern in the HPATTERN register is shifted out of the output port. Similarly, if the active bit is 0 the pattern in the LPATTERN register is shifted out.

The HBC control (HBCNTRL) register is used to configure and control the data transmission. HBCNTRL is divided in 5 different controlling signal FRAME[2:0], IOSEL, TXBUSY, START/$\overline{\text{STOP}}$, and OCFLAG (see Figure 23.)

FRAME[2:0] selects the number of bits of DAT0 to encode and transmit. The HBC allows from 2 (0x1) to 8 (0x7) DAT0 bits to be encoded and transmitted. Upon a reset, FRAME is initialized to zero disabling the DAT0's decision register transmitting no data.

The IOSEL signal selects the transmission to output (TX) through either port G2 or G5. If IOSEL is 1, G5 is selected as the output port otherwise G2 is selected.

The TXBUSY signal is read only and is used to inform software that a transmission is in progress. TXBUSY goes high when the encoded data begins to shift out of the output port and will remains high during each consecutive DAT0 frame bit transmission (see Figure 25). The HBC will clear the TXBUSY signal when the last DAT0 encoded bit of the frame is transmitted and the STOP signal is 0.

The START / $\overline{\text{STOP}}$ signal controls the encoding and transmission process for each data frame. When software sets the START / $\overline{\text{STOP}}$ bit the DAT0 frame transmission process begins. The START signal will remain high until the beginning of the last encoded DAT0 frame bit transmission. The HBC then clears the START / $\overline{\text{STOP}}$ bit allowing software to either continue with a new DAT0 frame transmission or stop the transmission all together (see Figure 25). If TXBUSY is 0 when the START signal is enabled, a synchronization period occurs before any data is transmitted lasting the amount of time to transmit a 0 encoded bit (see Figure 24).

The OCFLAG signal is read only and goes high when the last encoded bit of the DAT0 frame is transmitting. The OCFLAG signal is used to inform software that the DAT0 frame transmission operation is completing (see Figure 25). If multiple DAT0 frames are to be transmitted consecutively, software should poll the OCFLAG signal for a 1. Once OCFLAG is 1, DAT0 must be reload and the START / $\overline{\text{STOP}}$ bit must be restored to 1 in order to begin the new frame transmission without interruptions (the synchronization period). Since OCFLAG remains high during the entire last encoded DAT0 frame bit transmission, software should wait for the HBC to clear the OCFLAG signal before polling for the new OCFLAG high pulse. If new data is not reloaded into DAT0 and the START signal ($\overline{\text{STOP}}$ is active) is not set before the OCFLAG is 0, the transmission process will end (TXBUSY is cleared) and a new process will begin starting with the synchronization period.

Figure 24 and 25 shows how the HBC performs its data encoding. In the example, two frames are encoded and transmitted consecutively with the following bit encoding format specification:

1. Transmission frequency = 62.5KHz
2. Data to be encoded = 0x52, 0x92 (all 8-bits)
3. Each bit should be encoded as a 3-bit binary value, '1' = 110b and '0' = 100b
4. Transmission output port : G2

To perform the data transmission, software must first initialize the PSCALE, BPSEL, HPATTERN, LPATTERN, and DAT0 registers with the appropriate values.

```
LD  PSCALE, #03H        ; (1MHz ÷ 4) ÷ 4 = 62.5KHz
LD  BPSEL, #012H        ; BPH = 2, BPL = 2 (3 bits each)
LD  HPATTERN, #0C0H     ; HPATTERN = 0xC0
LD  LPATTERN, #090H     ; LPATTERN = 0x90
LD  DAT0, #052H         ; DAT0 = 0x52
```

Once the basic registers are initialized, the HBC can be started. (At the same time, software must set the number of data bits per data frame and select the desired output port.)

```
LD  HBCNTRL, #27H       ; START / STOP = 1,
                          FRAME = 7, IOSEL = 0
```