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®

April 2002

ACE1001 Product Family Arithmetic Controller Engine (ACEx™) for Low Power Applications

General Description

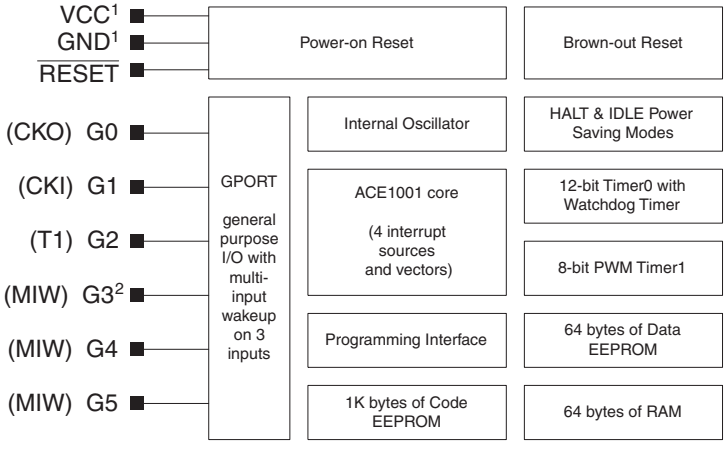
The ACE1001 is a member of the ACEx (Arithmetic Controller Engine) family of microcontrollers. It is a dedicated programmable monolithic integrated circuit for applications requiring high performance, low power, and small size. It is a fully static part fabricated using CMOS technology.

The ACE1001 product family has an 8-bit core processor, 64 bytes of RAM, 64 bytes of data EEPROM and 1K bytes of code EEPROM. Its on-chip peripherals include a programmable 8-bit timer with PWM output, watch-dog/idle timer, and programmable undervoltage detection circuitry. The on-chip clock and reset functions reduce the number of required external components. The ACE1001 product family is available in 8-pin SOIC and TSSOP packages.

Features

- Arithmetic Controller Engine
- 1K bytes on-board code EEPROM
- 64 bytes data EEPROM
- 64 bytes RAM
- Watchdog
- Multi-input wake-up 3 I/O pins
- 8-bit Timer1 with PWM output
- On-chip oscillator
  - — No external components
  - — 1µs instruction cycle time
- On-chip Power-on Reset
- Brown-out Reset
- Programmable read and write disable functions
- Memory mapped I/O
- Multilevel Low Voltage Detection
- Fully static CMOS
  - — Low power HALT mode (100nA @ 3.3V)
  - — Power saving IDLE mode
- Single supply operaton
  - — 1.8 - 5.5V (ACE1001L)
  - — 2.2 - 5.5V (ACE1001)
- 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 writes
- 8-pin SOIC and TSSOP packages.

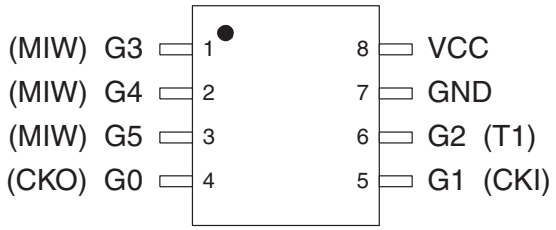
Block and Connection Diagram



1. 100nf decoupling capacitor recommended.
2. Input only

Figure 2: ACE1001 SOIC 8-Pin Device Pinout

(a) Normal Operation



(b) Programming Mode

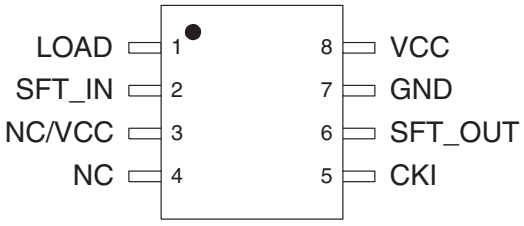
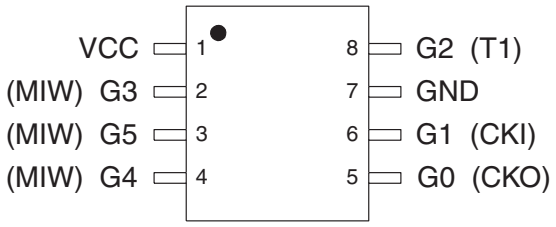
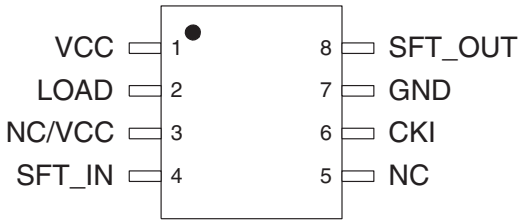


Figure 3: ACE1001 TSSOP 8-Pin Device Pinout

(a) Normal Operation



(b) Programming Mode



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

Device	Operating Voltage	Operating Temperature
ACE1001L	1.8 to 5.5V	0°C to 70°C
ACE1001	2.2 to 5.5V	0°C to 70°C
ACE1001E	2.2 to 5.5V	-40°C to +85°C

ACE1001(L) DC Electrical Characteristics

V_{CC} = 1.8/2.2 to 5.5V

All measurements valid for ambient operating temperature unless otherwise stated.

Symbol	Parameter	Conditions	MIN	TYP	MAX	Units
I_{CC}[3]	Supply Current – no data EEPROM write in progress	1.8V 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 @ +25°C 3.3V @ -40°C to +85°C		 10	100 1000	nA nA
		5.5V @ +25°C 5.5V @+125°C		 250	1000 3000	nA nA
I_{CCL}[4]	IDLE Mode Current	3.3V 5.5V		120 140	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} = 1.8V V_{CC} = 2.2 -5.5V			$0.15V_{CC}$ $0.20V_{CC}$	V V
V_{IH}	Input High with Schmitt Trigger Buffer	$V_{CC} \leq 2.2V$ $V_{CC} > 2.2V$	$0.9V_{CC}$ $0.8V_{CC}$			V 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	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	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	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	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	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	0.4 mA source	$0.8V_{CC}$			V
	G5	1.0 mA source	$0.8V_{CC}$			V

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

[4] Based on a continuous IDLE looping program.

ACE1001(L) AC Electrical Characteristics

V_{CC} = 1.8/2.2 to 5.5V

All measurements valid for ambient operating temperature unless otherwise stated.

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

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

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

ACE1001(L) Electrical Characteristics for programming

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

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

ACE1001(L) Low Battery Detect (LBD) Characteristics

V_{CC} = 1.8/2.2 to 5.5V

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

ACE1001 Brown-out Reset (BOR) Characteristics

V_{CC} = 2.2 to 5.5V

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

ACE1001L Brown-out Reset (BOR) Characteristics

V_{CC} = 1.8 to 5.5V

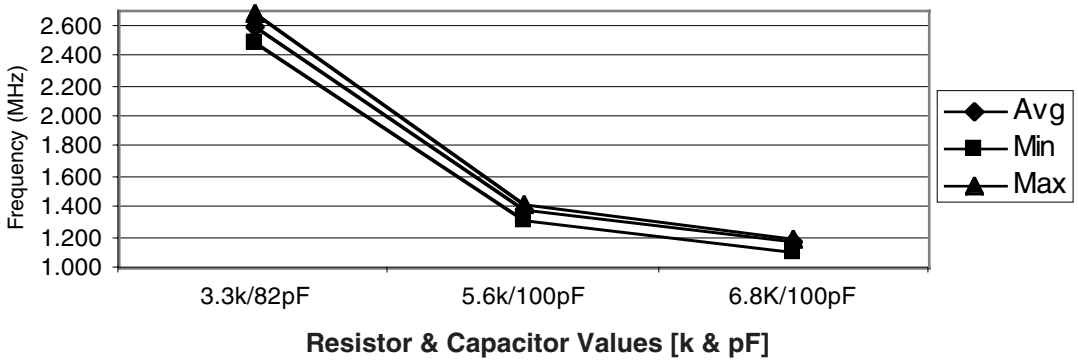
Parameter	Conditions	MIN	TYP	MAX	Units
BOR Voltage Threshold Variation (BLSEL = 0)	0°C to +70°C	1.76	1.95	2.20	V

3.0 AC & DC Electrical Characteristic Graphs

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

Figure 4: RC Oscillator Frequency vs. Temperature

(a) V_{CC} = 5.0V



(b)V_{CC}=2.5V

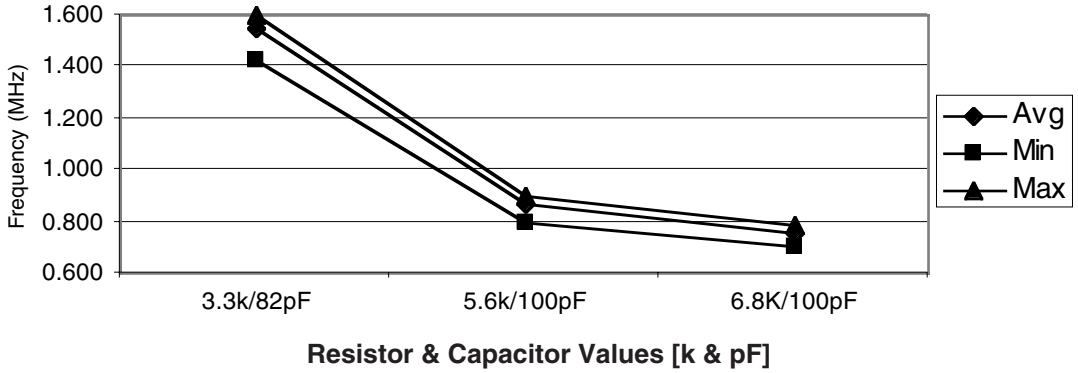


Figure 5: Internal Oscillator Frequency

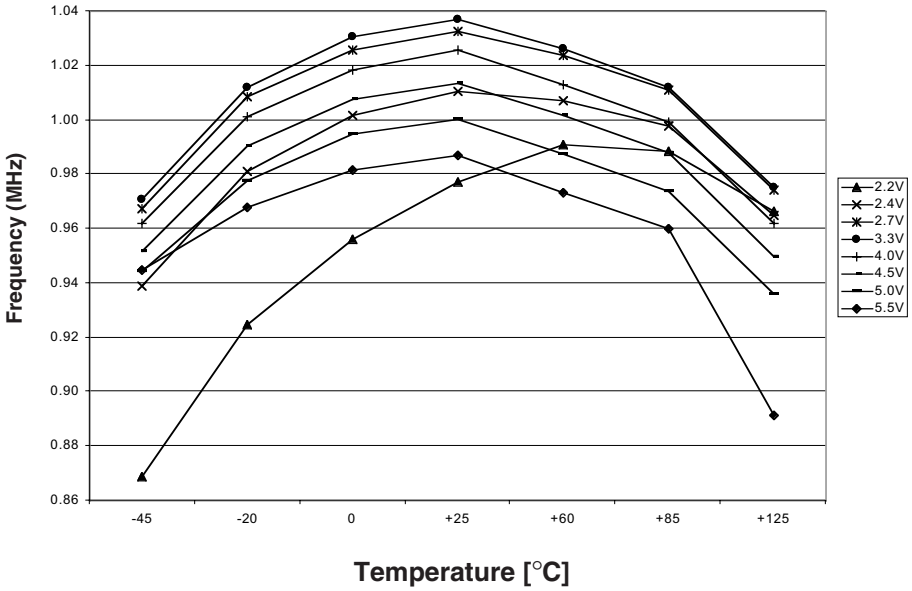
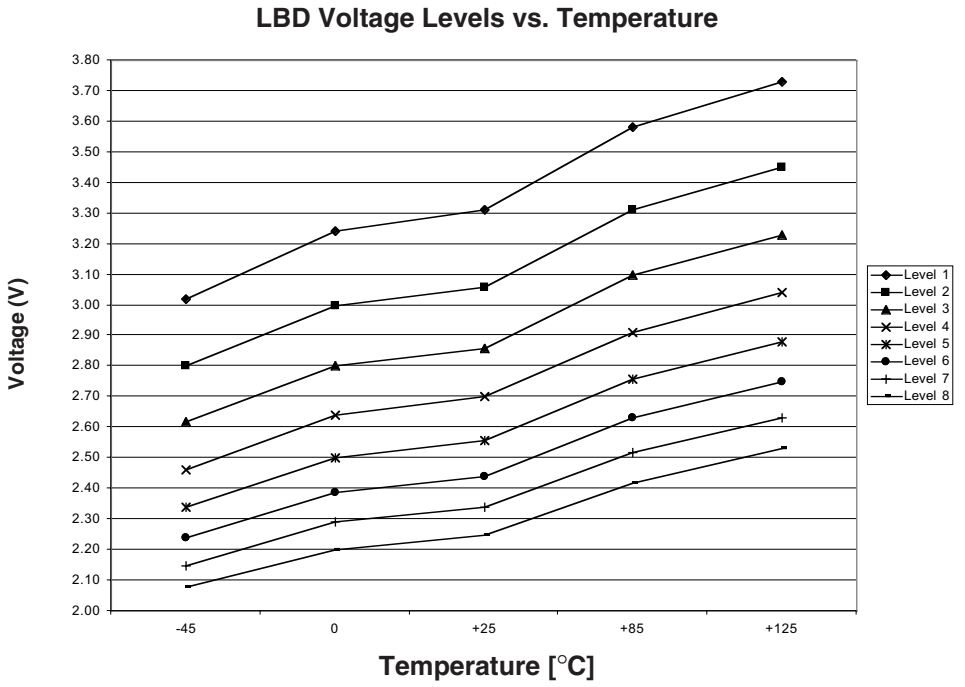


Figure 6: LBD and BOR Threshold Levels



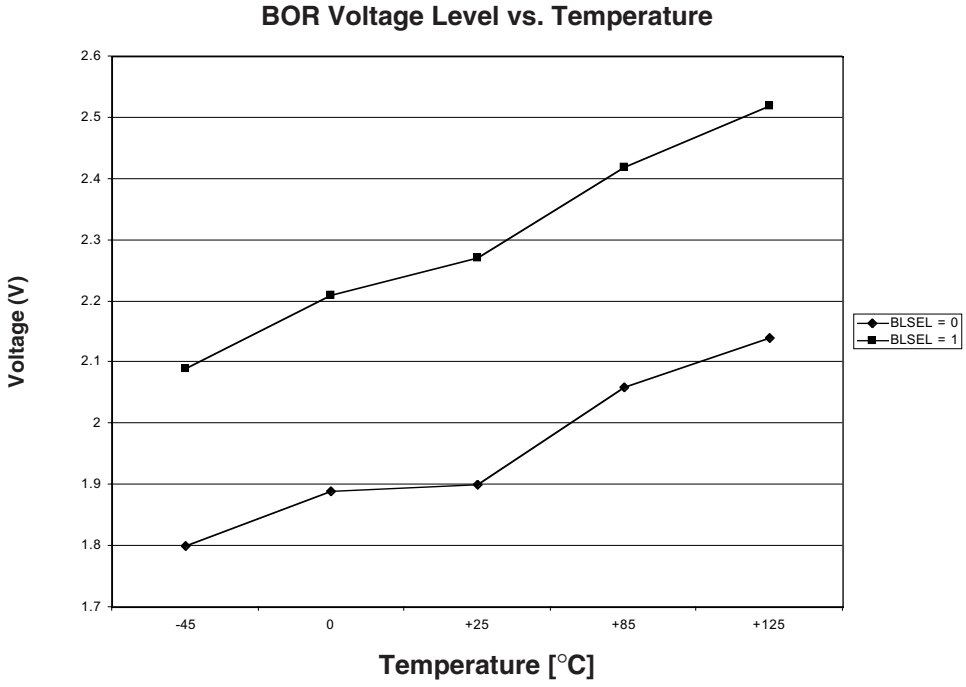
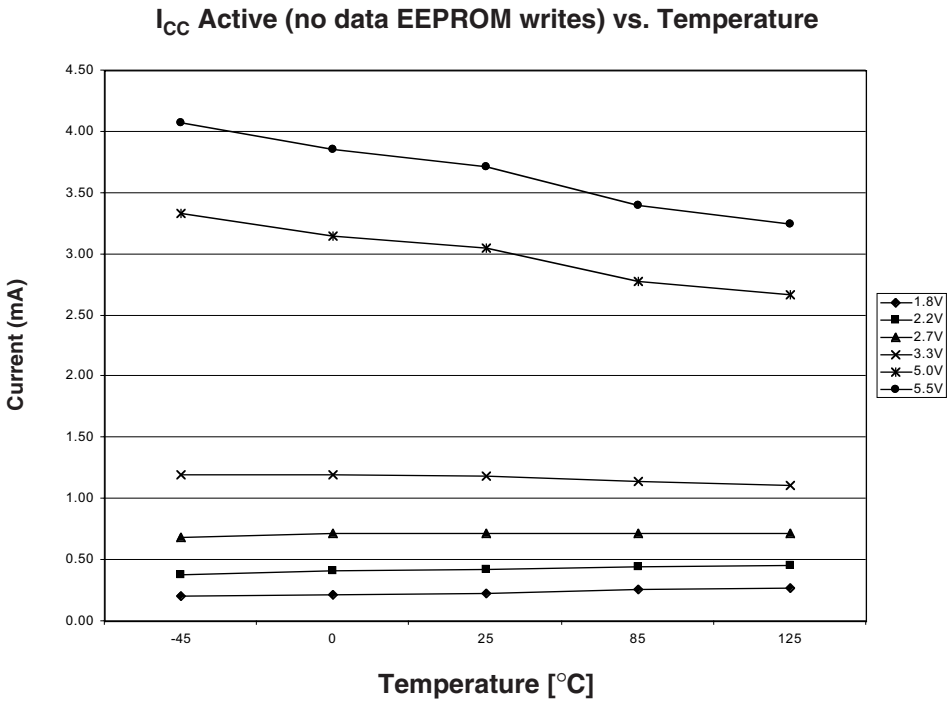


Figure 7: I_{CC} Active Current



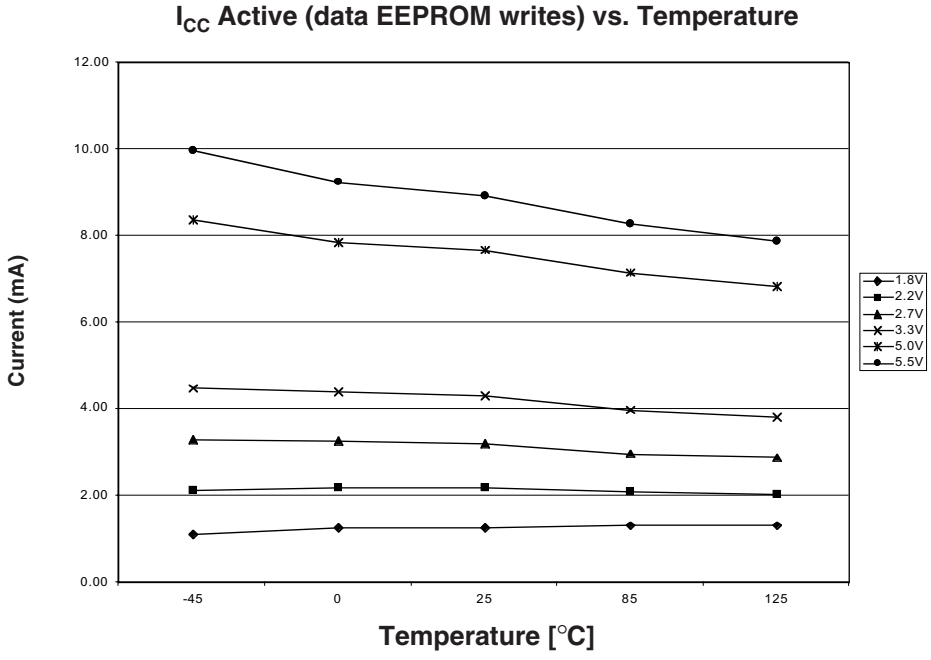


Figure 8: HALT Mode Currents

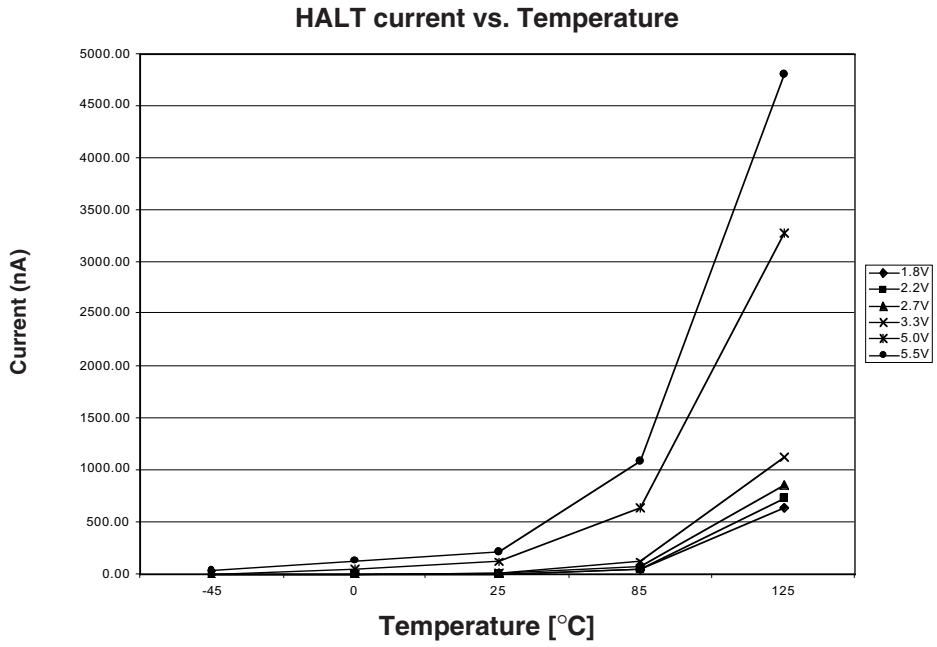


Figure 9: IDLE Mode Current

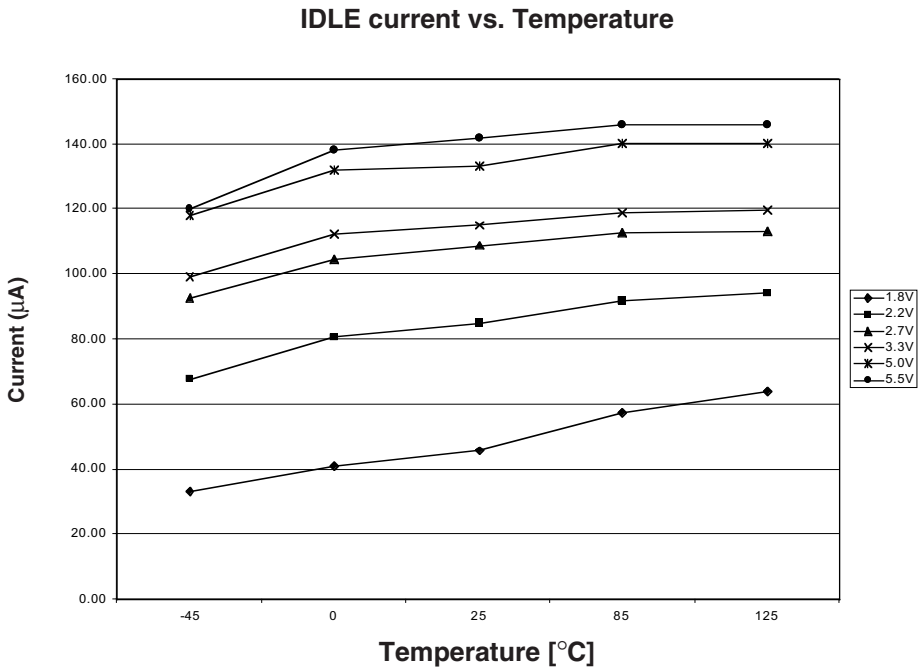
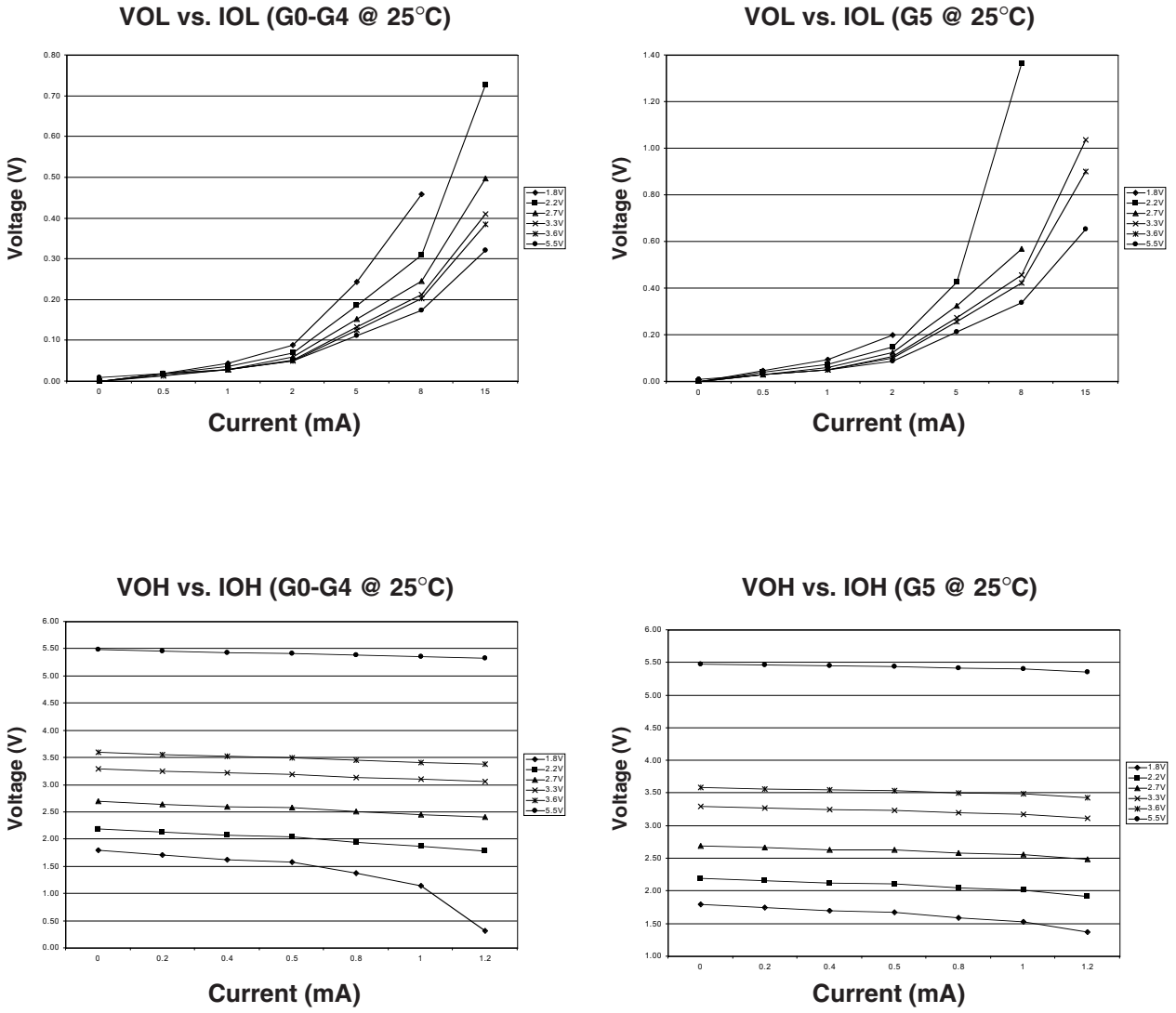


Figure 10: VOL/VOH



4.0 Arithmetic Controller Core

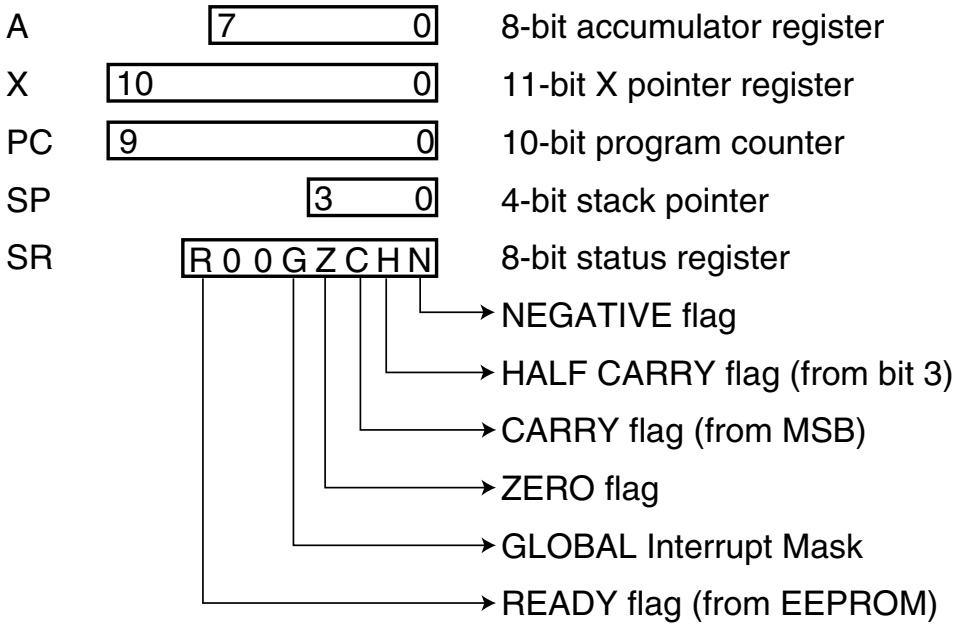
The ACEx microcontroller core is specifically designed for low cost applications involving bit manipulation, shifting and arithmetic operations. It is based on a modified Harvard architecture meaning peripheral, I/O, and RAM locations are addressed separately from instruction data.

The core differs from the traditional Harvard architecture by aligning the data and instruction memory sequentially. This allows the X-pointer (11-bits) to point to any memory location in either segment of the memory map. This modification improves the overall code efficiency of the core and takes advantage of the flexibility found on Von Neumann style machines.

4.1 CPU Registers

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

Figure 11: Programming Model



4.1.1 Accumulator (A)

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

4.1.2 X-Pointer (X)

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

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

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

4.1.3 Program Counter (PC)

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

4.1.4 Stack Pointer (SP)

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

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

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

4.1.5 Status Register (SR)

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

Carry/Borrow (C)

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

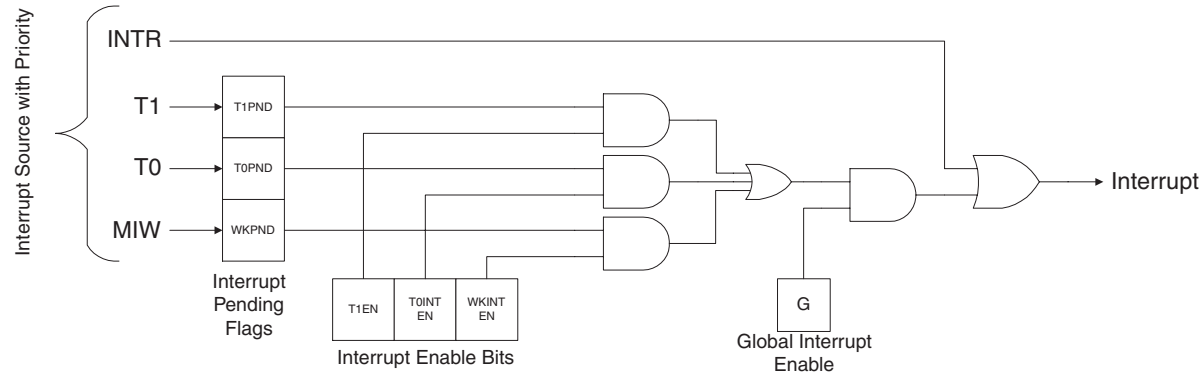
Half Carry (H)

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

Zero (Z)

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

Figure 12: Basic Interrupt Structure



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 the 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.

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 12). The software interrupt 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 six addressing modes indexed, direct, immediate, absolute jump, and relative jump.

Indexed

The instruction allows an 8-bit unsigned offset value to be added to the 10-LSBs of the X-pointer yielding a new effective address. This mode can be used to address any memory space (program or data).

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 10-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.

Table 8: Interrupt Priority Sequence

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

Table 9: Instruction Addressing Modes

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

Table 10: Instruction Cycles and Bytes

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

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

4.4 Memory Map

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

Table 11: Memory Map

Address	Memory Space	Block	Contents
0x00 - 0x3F	Data	SRAM	Data RAM
0x40 - 0x7F	Data	EEPROM	Data EEPROM
0xAA	Data	Timer1	T1RA register
0xAB, 0xAD			Reserved
0xAC	Data	Timer1	TMR1 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 - 0xBA			Reserved
0xBB	Data	Init. Reg.	Initialization register 1
0xBC	Data	Init. Reg.	Initialization register 2
0xBD	Data	LBD	LBD register
0xBE	Data	Core	XHI register
0xBF	Data	Core	XLO register
0xC0	Data	Clock	Power mode clear (PMC) register
0xCE	Data	Core	SP register
0xCF	Data	Core	Status register (SR)
0xC00 - 0xFF5	Program	EEPROM	Code EEPROM
0xFF6 - 0xFF7	Program	Core	Timer0 Interrupt vector
0xFF8 - 0xFF9	Program	Core	Timer1 Interrupt vector
0xFFA - 0xFFB	Program	Core	MIW Interrupt vector
0xFFC - 0xFFD	Program	Core	Soft 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 1K 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 13 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 13: Initialization Register 1

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

(0) RDIS [5,6] — If set, disables attempts to read the contents from the EEPROMs while in programming mode

(1) WDIS [5,6] — If set, disables attempts to write new contents to the EEPROMs while in programming mode

(2) UBD [5,6] — If set, the device will not allow any writes to occur in the upper block of data EEPROM (0x60-0x7F)

(3) BLSEL [7] — If set, the Brown-out Reset (BOR) voltage reference level is set to its higher range for the ACE1001
If not set, the BOR voltage reference level is set to its lower range for the ACE1001L

(4) BOREN — If set, allows a BOR to occur if V_{CC} 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

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

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

[6] 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.

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

5.0 Timer 1

Timer 1 is a versatile 8-bit timer. Its main function is to operate as a Pulse Width Modulation (PWM) generator that generates pulses of a specified width and duty cycles.

Timer 1 contains an 8-bit timer register (TMR1), an 8-bit auto-reload register (T1RA), and an 8-bit control register (T1CNTRL). All registers are memory-mapped for simple access through the core. For the PWM signal generation the timer contains an output (T1) that is multiplexed with the I/O pin G2.

The timer can be started or stopped through the T1CNTRL register bit T1C0. When running, the timer counts down (decrements) every clock cycle. The timer's clock has a pre-scalar and is selectable through two T1CNTRL register bits T1PSC[1:0]. Depending on the selected operating mode, occurrences of timer underflow (transitions from 0x00 to 0xFF or reload) can either generate an interrupt and/or toggle the T1 output pin.

Timer 1's interrupt (TMRI1) can be enabled by the interrupt enable (T1EN) bit in the T1CNTRL register. When the timer interrupt is enabled, the source of the interrupt is a timer underflow. By default, the timer register is reset to 0xFF and the auto-reload register is reset to 0x00.

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, start or stop the timer, and select the clock. The T1CNTRL register bits are described in Table 12.

Table 12: TIMER1 Control Register Bits

T1CNTRL Register	Name	Function
Bit 7	----------	Reserved
Bit 6	----------	Reserved
Bit 5	T1C1	T1 toggle enable bit: 1 = T1 toggle enabled, 0 = T1 toggle disabled
Bit 4	T1C0	TMR1 run: 1 = Start timer, 0 = Stop timer
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 1,0	T1PSC	Pre-scalar selection bits: Selects the 1MHz clock divider to be by 1 (00b), 2 (01b), 4 (10b), or 8 (11b)

5.2 Pulse Width Modulation (PWM) Mode

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

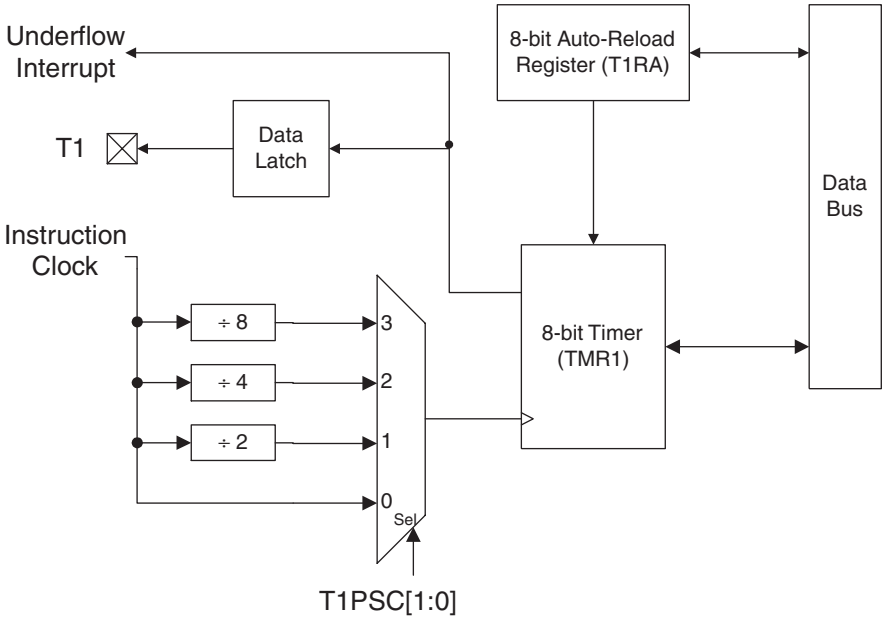
The timer has one interrupt (TMRI1) that is maskable through the T1EN bit of the T1CNTRL register. However, the core is only interrupted if the T1EN bit and the G (Global Interrupt enable) bit of the SR is set. If interrupts are enabled, the timer will generate an interrupt each time T1PND flags is set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

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

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

1. Configure T1 as an output by setting bit 2 of PORTGC.
   - SBIT 2, PORTGC ; Configure G2 as an output
2. Initialize T1 to 1 (or 0) by setting (or clearing) bit 2 of PORTGD.
   - SBIT 2, PORTGD ; Set G2 high
3. Load the initial PWM high (low) time into the timer register.
   - LD TMR1, #6FH ; High (Low) for .444ms (1MHz/4 clock)
4. Load the PWM low (high) time into the T1RA register.
   - LD T1RA, #2FH ; Low (High) for .188ms (1MHz/4 clock)
5. Write the appropriate control value to the T1CNTRL register to select PWM mode with T1 toggle, to select the divide by 4 pre-scalar, and to clear the enable and pending flags. (See Table 12)
   - LD T1CNTRL, #22H ; Setting the T1C0 bit starts the timer
6. Set te T1CO bit to start the timer.
   - SBIT T1CP, T1CNTRL ; T1CO equals 4
7. After every underflow, load T1RA with alternate values. If the user wishes to generate an interrupt on timer output transitions, reset the pending flags and then enable the interrupt using T1EN. The G bit must also be set. The interrupt service routine must reset the pending flag and perform whatever processing is desired.
   - RBIT T1PND, T1CNTRL ; T1PND equals 3
   - LD T1RA, #6FH ; Low for .444ms (1MHz/4 clock)

Figure 14: Pulse Width Modulation Mode



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 or after the watchdog has been- serviced). The T0PND flag is set every other time the timer overflows (transitions from 0xFFF to 0x000). 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 16.0 for details.)
2. Start up delay from HALT mode
3. Watchdog pre-scalar (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 8.0 for details.

7.0 Watchdog

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-scalar 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 16). 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 15: Timer 0 Control Register Definition (T0CNTRL)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WKINTEN	x	x	x	x	x	T0PND	T0EN

Figure 16: 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 Multi-Input Wakeup/Interrupt Block

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

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

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

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

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

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

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

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

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

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

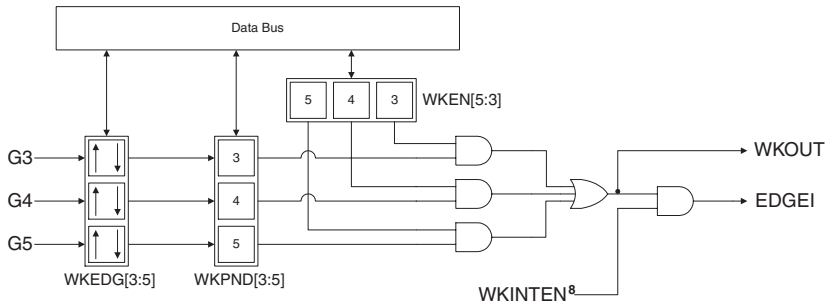
PORTG provides the user with three fully selectable, edge sensitive interrupts that are all vectored into the same service subroutine. The interrupt from PORTG shares logic with the wakeup circuitry. The WKEN register allows interrupts from PORTG to be individually enabled or disabled. The WKEDG register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

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

Figure 17: MIW Register Bit Assignments

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

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



[8] WKINTEN: Bit 7 of T0CNTRL

9.0 I/O Port

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

9.1 I/O registers

The I/O pins (G0-G5) have three memory-mapped port registers associated with the I/O circuitry: a port configuration register (PORTGC), a port data register (PORTGD), and a port input register (PORTGP). PORTGC is used to configure the pins as inputs or outputs. A pin may be configured as an input by writing a 0 or as an output by writing a 1 to its corresponding PORTGC bit. If a pin is configured as an output, its PORTGD bit represents the state of the pin (1 = logic high, 0 = logic low). If the pin is configured as an input, its PORTGD bit selects whether the pin is a weak pull-up or a high-impedence input. Table 13 provides details of the port configuration options. The port configuration and data registers are both read/writable. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports multi-input wakeup/interrupt, the PORTG inputs have Schmitt triggers.

Figure 19: PORTGD Logic Diagram

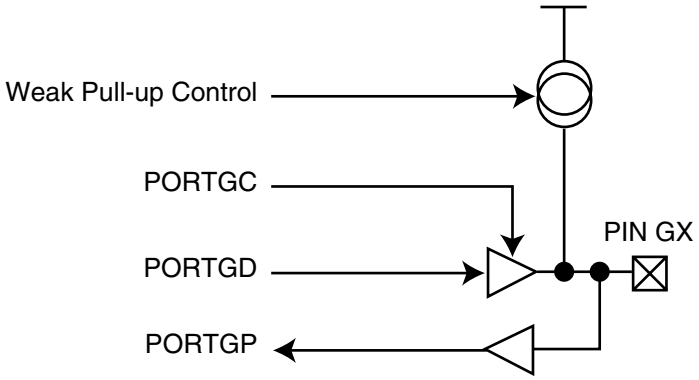


Figure 20: I/O Register bit assignments

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	G5	G4	G3[9]	G2	G1	G0

Table 13: I/O configuration options

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

[9] G3 is only an input.

10.0 In-circuit Programming Specification[10,11]

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

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

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

10.1 Write Sequence

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

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

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

10.2 Read Sequence

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

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

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

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

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

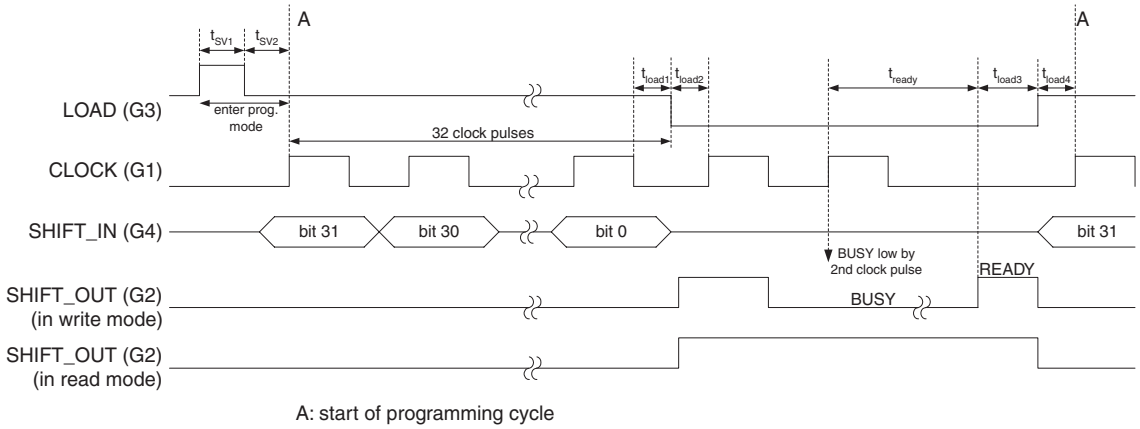
Table 14: 32-Bit Command and Response Word

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

[10] Application Note reference: "How to In-Circuit Program the ACEx Family of Microcontrollers."

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

Figure 21: Programming Protocol[10]



A: start of programming cycle

Figure 22: Serial Data Timing