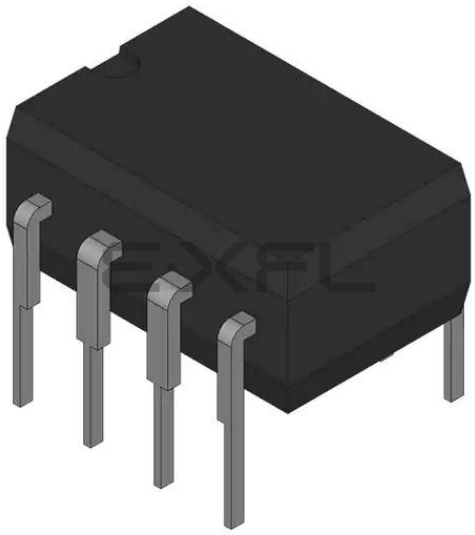


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

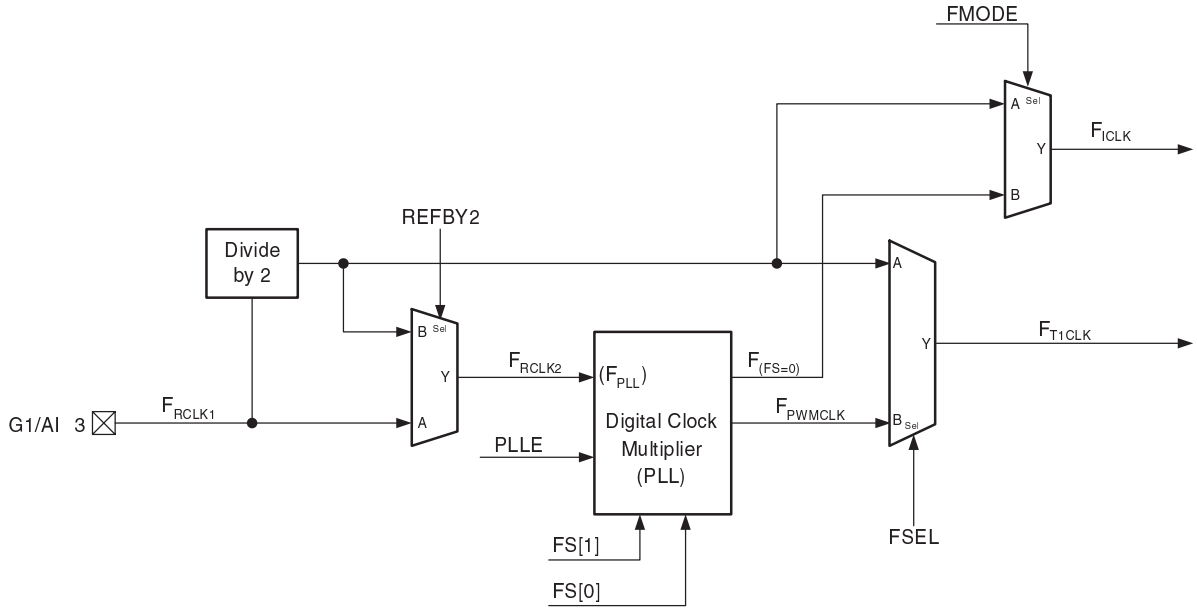
Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	Digital Power Controller
Core Processor	8-Bit
Program Memory Type	EEPROM (1kB)
Controller Series	-
RAM Size	64 x 8
Interface	-
Number of I/O	6
Voltage - Supply	2.7V ~ 3.6V
Operating Temperature	-40°C ~ 125°C
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-DIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=fms7401lvn

Table of Contents

List of Figures

Figure 4. External Clock Scheme



1. Refer to the Device Memory section of the datasheet for details regarding the Initialization Registers 1.
2. The upper F_{OSC} frequency (4MHz) is not a standard feature offered on the FMS7401L devices but is available upon request.
3. The ADCNTRL2 register is defined in the ADC Circuit section of the datasheet.
4. The PSCALE register is defined in the PWM Timer 1 Circuit section of the datasheet.
5. Software must always configure the device's entire clocking structure (see Figure 3 and Figure 4) while the PWM Timer 1 circuit is off (T1C0=0) and configured in PWM mode (T1C3=0).
6. The PLL's $F_{(FS=0)}$ output is not affected by the FS[1:0] bit value of the PSCALE register and merely shares the FS[1:0]=00 divide factor.

4 ADC Circuit

The Analog-to-Digital Converter (ADC) Circuit extends the features of the FMS7401L by offering a 5-channel 8-bit ADC. The ADC may be programmed to convert voltages on any of the eight inputs of the analog mux, where five are multifunction input channels (ACH1-ACH5) and three are used for system calibration. The integrated ADC function offers a single cost-effective solution for applications requiring voltage, current and temperature sensing. The multifunction input channels may be configured to perform standard conversions on any of the analog input pins (G4/AIN0, G3/AIN1, G2/AIN2, G3/AIN3 or G7/AIN4). Three of the multifunction input channels may be programmed to perform ADC conversions through the internal Autozero Amplifier, Uncommitted Amplifier, and Current Source Generator for special control system and battery management applications (see Figure 6).

The ADC Circuit's eight analog inputs are software selectable where their analog input voltage is converted with respect to the internal ADC reference voltage (V_{AREF}). V_{AREF} may be programmed to use the internal bandgap reference voltage (V_{REF}) or Vcc as its source. By default, the ADC circuit's V_{AREF} is configured to use the internal V_{REF} as its source.[1]

The ADC performs conversions of 8-bit resolution with accuracy as defined in the Electrical Characteristics section of the datasheet. For a standard ADC conversion, the ADC circuit converts the analog input voltage in a total of 13 conversion clock cycles, and a total of 20 conversion clock cycles when performing an autozero ADC conversion. To yield a better ADC conversion accuracy, the ADC circuit may configure the ADC clock (F_{ADCLK}) to a slower frequency, lengthening the total conversion time while improving its accuracy. As part of the total conversion time, the ADC circuit completes a sample and hold phase to measure fast changing analog signals before converting the voltage. An ADC conversion can be initiated by a software command or automatically (using the gated auto-sampling mode) by the active (on) edge transition of the ADSTROBE PWM Timer 1 output.[2] If enabled, the ADC circuit offers the use of its microcontroller hardware interrupt (ADCI) triggered after each completed ADC conversion so that the microcontroller core is freed to perform other tasks.

4.1 ADC Circuit Configuration

Software must access the three memory mapped ADC registers to configure and control the ADC circuit.[3] The ADC Control 1 (ADCNTRL1) register is used to select the analog input channel and ADC reference voltage (V_{AREF}) for the conversion. In addition, it is used to initiate a conversion through software, monitor the ADC pending flag, and enable the ADC circuit's microcontroller hardware interrupt (ADCI). The ADC Control 2 (ADCNTRL2) register is used to enable the internal Autozero Amplifier, Uncommitted Amplifier, Current Source Generator, and/or ADC Auto-sampling Mode. The ADCNTRL2 register is also used to divide the ADC F_{ADCLK} clock to improve the conversion accuracy. Lastly, the ADC Data (ADATA) register is used by software to read the final converted 8-bit digital value. ADATA is a read only register and is updated automatically at the end of each ADC conversion.

this information because the APND bit may be triggered before the ASTART is automatically cleared. The ADC conversion completion delay may occur when the F_{ICLK} clock is slower than an ADC conversion clock cycle.

4.2.1 Analog Input Voltage and its 8-bit Digital Result

The relationship between the 8-bit digital value stored in the ADATA register and the analog input voltage is as follows:

$$V_{ADC} = \frac{V_{ACH(x)}}{V_{AREF}} \times 255$$

- V_{ADC} is the 8-bit digital result of an ADC conversion.
- $V_{ACH(x)}$ is the analog voltage applied to the selected input channel.

4.2.2 ADC Gated Auto-sampling Mode

The ADC circuit may be configured in Gated Auto-sampling Mode by setting the ENDAS bit of the ADCNTRL2 register. When in Auto-sampling Mode, all ADC conversions are automatically triggered by the active (on) edge transition of the PWM Timer 1's ADSTROBE output signal.[2] If the period of the PWM ADSTROBE signal is less than the total ADC conversion time, any triggers issued while a conversion is in progress (ASTART=1) are ignored. Once the trigger is detected, the ASTART bit of the ADCNTRL1 register is set symbolizing that a conversion is in progress. The initial conversion phase, the sample and hold or autozero (if GAIN=1), begins after a 1µS cycle delay.[7] Once all eight digital bits are determined and stored in the ADATA register, the APND flag is set to trigger a hardware interrupt (if enabled) flagging software that the ADATA register has been updated with the ADC conversion results. Once all phases of the ADC conversion cycle completes, the ASTART bit is then automatically cleared by the ADC circuit. Since software cannot change the ADC circuit configuration while an ADC conversion is in progress, the ASTART bit must be monitored to determine when the conversion cycle completes. Software cannot rely on the APND bit for this information because the APND bit may be triggered before the ASTART is automatically cleared. The ADC conversion completion delay may occur when the F_{ICLK} clock is slower than an ADC conversion clock cycle.

4.2.3 ADC Conversion Clock Configuration

The ADC conversion clock (F_{ADCLK}) is sourced either by the device's main system instruction clock (F_{ICLK}) or the PWM Timer 1's clock (F_{T1CLK}) depending on the ADC circuit's operating mode. If the standard ADC conversion mode is selected, the ADC circuit is automatically configured to source the F_{ADCLK} clock by the F_{ICLK} clock. If the ADC Conversion Auto-sampling Mode is selected, the ADC circuit is automatically configured to source the F_{ADCLK} clock by the F_{T1CLK} clock to synchronize the ADC conversions with the active (on) edge of the PWM Timer 1 ADSTROBE output signal.[2]

When in standard ADC conversion mode, the ASPEED[1:0] bits of the ADCNTRL2 register may be used to slow the total conversion time improving the ADC conversion accuracy. However, if the F_{ICLK} clock is sourced by the PLL's $F_{(FS=0)}$ output (when FMODE=1) the F_{ADCLK} will clock eight times faster than the proper conversion rate (1µS cycle time). The F_{ADCLK} clock must then be divided by setting the ASPEED[1:0]=3 divide factor to yield a $F_{ADCLK}/8$ conversion clock cycle. Otherwise, software may temporarily clear FMODE returning the conversion cycle to its proper frequency and free the ASPEED bits to be used to improve the conversion accuracy. In addition, if the internal oscillator is trimmed to its upper F_{OSC} frequency and it is sourcing the F_{ICLK} clock, the ASPEED[1:0]=1 divided factor must be selected to yield a $F_{ADCLK}/2$ conversion clock cycle.[8] A greater divide factor may still be selected by setting the ASPEED[1:0]>1.

When in ADC Conversion Auto-sampling Mode, the ADC circuit automatically configures the F_{ADCLK} clock to be sourced by the F_{T1CLK} clock so that the ADC conversions may be synchronized with the active (on) edge of the ADSTROBE signal. However, the F_{T1CLK} clock is first sent into a special divide circuit which evaluates its configuration to determine the divide factor needed to yield the proper F_{ADCLK} conversion rate (1µS cycle time). The FMODE, FSEL, and FS bits of the PSCALE register are evaluated so that the divide circuit applies the appropriate divide factor to the F_{T1CLK} clock (the PS bits do not apply). The ASPEED[1:0] bits of the ADCNTRL2 register may be used to slow the total conversion time improving the ADC conversion

5 Programmable Comparator Circuit

The Programmable Comparator circuit is an analog comparator whose outputs may be monitored by software or fed into a digital delay filter used to disable the PWM Timer 1 circuit or its PWM cycle. The comparator's non-inverting input is software selectable by the COMPSEL bit of the ADCNTRL2 register.[1] If COMPSEL=0, the non-inverting input of the Programmable Comparator is the G4/AIN0 device pin. If COMPSEL=1, the non-inverting input of the Programmable Comparator is the G2/AIN2 device pin. Before enabling the Programmable Comparator circuit, the selected analog input port pin must be configured as a tri-state input bypassing the I/O circuitry.[2] The inverting input of the comparator is controlled by the Voltage Loop (VLOOP) enable bit of the comparator control (COMP) register. If VLOOP=0, the voltage loop is disabled and the inverting input of the analog comparator is configured as one of the 63 programmable voltage levels (V_{THL}, V_{THU}). If VLOOP=1, the analog comparator is set in a voltage loop configuration with the Uncommitted (Error) Amplifier output (A_{OUT}) connected to the comparator's inverting input (see Figure 9).

The Programmable Comparator circuit may be configured and controlled by software through the two 8-bit Comparator Control (COMP) and Digital Delay (DDELAY) registers. Both the Programmable Comparator and the digital delay filter must be enabled by software by setting the Comparator Enable (COMPEN) and clearing the EPWM bits of the Digital Delay (DDELAY) register. Upon a system reset, the Programmable Comparator is disabled and the digital delay filter is enabled. The COMP circuit is automatically disabled during Halt Mode. After exiting the Halt Mode, software must wait at least 10 instruction clock cycles before reading the COUT bit to ensure that the internal circuit has stabilized.

Table 8. Programmable Comparator (COMP) Control Register Bit Definitions

COMP Register (addr. 0xA0)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CL[5:0]						VLOOP	COUT

Bit	Description
CL[5:0]	Programmable Comparator Voltage Reference Level bits. Refer to Table 9 and Table 10 for details.
VLOOP	(0) Configures the inverting input of the analog comparator as one of the 63 programmable voltage levels (V_{THL}, V_{THU}). (1) Configures the analog comparator in a voltage loop configuration with the Uncommitted Amplifier output (A_{OUT}) connected to the inverting input.
COUT	(0) G2/AIN2 or G4/AIN0 non-inverting input is less than inverting input configured by VLOOP. (1) G2/AIN2 or G4/AIN0 non-inverting input is greater than inverting input configured by VLOOP.

5.1 Programmable Comparator's Voltage Threshold Levels (VLOOP=0)

The Programmable Comparator circuit is configured to compare the G4/AIN0 or G2/AIN2 non-inverting input against the programmable voltage threshold levels on its inverting input (see Table 9 and Table 10). The comparator output (C_{OUT}) is 1 when the G4/AIN0 or G2/AIN2 input pin rises above the selected voltage threshold. As long as the input stays above the selected voltage threshold, the C_{OUT} signal will hold its state. The C_{OUT} signal will equal zero if the G4/AIN0 or G2/AIN2 input voltage falls below the programmed threshold voltage or if the Programmable Comparator circuit is disabled. Software may change the programmed threshold voltage on-the-fly as needed in the application. If the digital delay filter circuit is enabled (EPWM=0), the C_{OUT} signal is monitored for its rising edge to generate the PWMOFF signal. Refer to Figure 8 and the following Digital Delay Filter with PWMOFF Output section for addition details.

Bit 6 of the ADCNTRL2 register is the Programmable Comparator non-inverting input selection (COMPSEL) bit.[1] If COMPSEL=0, the non-inverting input of the Programmable Comparator is the G4/AIN0 device pin. If COMPSEL=1, the non-inverting input of the Programmable Comparator is the G2/AIN2 device pin. Before enabling the Programmable Comparator circuit, the selected analog input port pin must be configured as a tri-state input bypassing the I/O circuitry.[2]

Bit 6 of the ADCNTRL2 register is the Programmable Comparator non-inverting input selection (COMPSEL) bit.[1] If COMPSEL=0, the non-inverting input of the Programmable Comparator is the G4/AIN0 device pin. If COMPSEL=1, the non-inverting input of the Programmable Comparator is the G2/AIN2 device pin. Before enabling the Programmable Comparator circuit, the selected analog input port pin must be configured as a tri-state input bypassing the I/O circuitry.[2]

Bit 1 of the Comparator Control (COMP) register is the Programmable Comparator circuit's voltage loop (VLOOP) configuration enable bit. If VLOOP=0, the Programmable Comparator circuit is configured to compare the analog G4/AIN0 or G2/AIN2 input (COMPSEL=0 or 1) to one of the 63 voltage threshold levels. If VLOOP=1, enables the voltage loop configuration where the analog G4/AIN0 or G2/AIN2 input (COMPSEL=0 or 1) to the Uncommitted (Error) Amplifier output (A_{OUT}).

Bit 7 of the Digital Delay (DDELAY) register is the Programmable Comparator circuit enable (COMPEN) bit. If COMPEN=0, the Programmable Comparator circuit is disabled and the C_{OUT} signal is low. If COMPEN=1, the Programmable Comparator circuit is enabled and the C_{OUT} signal generated by the comparison of the two inputs.

Bit 0 (COUT) of the Comparator Control (COMP) register is the latched comparator output (C_{OUT}) signal. If the Programmable Comparator circuit is enabled, the C_{OUT} signal is latched by the main system instruction (F_{ICLK}) clock into the COUT bit of the COMP register. Software may only read the COUT bit to monitor the comparator's activity. The COUT bit cannot cause any microcontroller hardware interrupt or any other actions.

Figure 9. Programmable Comparator Block Diagram (VLOOP = 1)

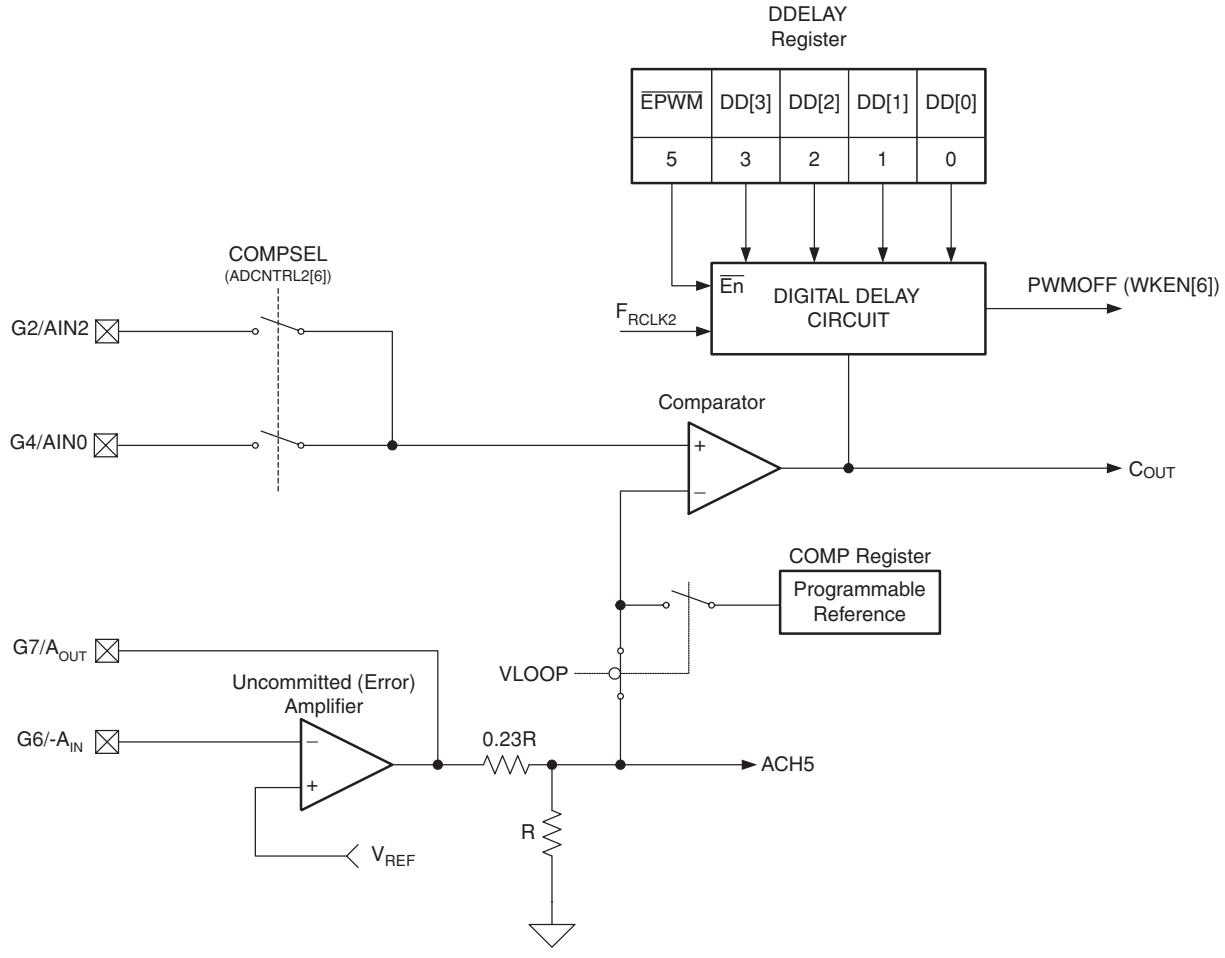


Table 16. Timer 1 Control (T1CNTRL) Register Bit Definitions

T1CNTRL Register (addr. 0xAE)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
T1C3	T1C2	T1C1	T1C0	T1PND	T1EN	X	T1BOUT

Bit	Description
T1C3	Timer 1 Mode Configuration Bit. Refer to Table 17 for details.
T1C2	Timer 1 Mode Configuration Bit. Refer to Table 17 for details.
T1C1	Timer 1 Mode Configuration Bit. Refer to Table 17 for details.
T1C0	PWM Mode (0) Stop the PWM Timer 1 circuit. (1) Start the PWM Timer 1 circuit.
	Input Capture Mode (0) Timer 1's TMR1 overflow pending flag is cleared. (1) Timer 1's TMR1 overflow pending flag is triggered.
T1PND	PWM Mode (0) Timer 1's TMR1 overflow pending flag is cleared. (1) Timer 1's TMR1 overflow pending flag is triggered.
	Input Capture Mode (0) Timer 1 capture pending flag is cleared. (1) Timer 1 capture pending flag is triggered.
T1EN	(0) Disables Timer 1 hardware interrupts. (1) Enables Timer 1 hardware interrupts.
T1BOUT	(0) Retain normal I/O function of the G1/AIN3 pin. (1) Enables Timer 1's ADSTROBE output to be sent to the G1 output port.

Table 17. Timer 1 Mode Configuration Bits

T1C3	T1C2	T1C1	Timer Mode Source	Interrupt	Timer count on
0	0	0	PWM mode no output toggle	TMR1 Overflow	Prescaler Input
0	1	1	PWM mode T1HS1 and T1HS2 toggle	TMR1 Overflow	Prescaler Input
0	0	1	PWM mode T1HS1 toggle	TMR1 Overflow	Prescaler Input
0	1	0	PWM mode T1HS2 toggle	TMR1 Overflow	Prescaler Input
1	0	0	Capture mode no T1HS1 toggle	TMR1 Overflow T1HS2 rising-edge	Prescaler Input
1	0	1	Capture mode with T1HS1 toggle	TMR1 Overflow T1HS2 rising-edge	Prescaler Input
1	1	0	Capture mode no T1HS1 toggle	TMR1 Overflow T1HS2 falling-edge	Prescaler Input
1	1	1	Capture mode with T1HS1 toggle	TMR1 Overflow T1HS2 falling-edge	Prescaler Input

6.2 Pulse Width Modulation (PWM) Mode

In PWM Mode, the Timer 1 circuit may be configured to generate pulses of a specified duty cycle and period on the T1HS1 (G0), T1HS2 (G5), and/or ADSTROBE (G1) timer outputs. The 12-bit TMR1 counter increments at the F_{T1CLK} clock rate defined by the FSEL bit of the PSCALE register. Refer to the previous PSCALE Register and Timer 1 Clock Configuration section of the datasheet for details.

A PWM cycle begins with the TMR1 counter incrementing from 0x000 until it matches the value stored in the T1RA register. At this point, the TMR1 counter completes its T1RA count and overflows (a transitions from T1RA to 0x000) setting the T1PND flag of the T1CNTRL register ending the PWM cycle. The Timer 1 circuit has two additional TMR1 compare (T1CMPA and T1CMPB) registers used to generate the T1HS1, T1HS2, and ADSTROBE output signals. All three output signals are initialized to a resting (off) state. Once the TMR1 counter is enabled (by setting the T1C0 bit of the T1CNTRL register), both compare registers are matched against the incrementing TMR1 counter. When the TMR1 completes its count equal to the value stored in the T1CMPA and T1CMPB registers, the T1HS1, T1HS2, and ADSTROBE output signals are set to an active (on) state until the TMR1 counter matches the value stored in the T1RA compare register (overflows). Once the TMR1 counter overflows, the output signals are cleared returning them to a resting (off) state. Refer to Figure 11 for a Timer 1 PWM Mode block diagram.

The PWM Timer 1 can be programmed to toggle one or both PWM output signals (T1HS1 and T1HS2) to support a variety of output configurations (half bridge, full bridge,[8] low side or high side driving). These outputs may be used to drive an external half-bridge driver and are enabled by programming the T1C1 and T1C2 bits in the T1CNTRL register (see Table 16). The T1HS1 (G0) and T1HS2 (G5) output signals may be configured with opposite phases and dead time controlled edges (see Figure 12). The phases of the output signals are configured by the bits of the PORTGD I/O configuration register.[9] Upon device power-up, the T1HS1 and T1HS2 signals may be programmed to default as active high/low outputs by the default I/O configuration register bits in the non-volatile Initialization Register 4.[10] Both G0/T1HS1 and G5/T1HS2 pins will configure to their programmed default state after T_{DIO}[2] from the system reset trigger (e.g. from a POR). The G0/T1HS1 and G5/T1HS2 pins may both be configured as outputs with common or opposite phases. If configured as outputs, the PORTGD[0] and PORTGD[5] bits configure the T1HS1 and T1HS2 signals as active high or low. If the PORTGD bit is 0, the output signal is active high, otherwise it is active low. The PORTGD[1] bit also configures the G1/ADSTROBE pin as an active high/low signal once configured as an output. The Initialization Register 4 bits only default the G0/T1HS1 and G5/T1HS2 pins not the G1/ADSTROBE pin. From factory, the G0/T1HS1, G5/T1HS2 and G1/ADSTROBE device pins are defaulted as tri-stated inputs. The pins must be configured by the Initialization Register 4 bits or by software directly through the PORTGC and PORTGD register as an output port before enabling the TMR1 counter and its outputs.

The dead time counter of the Timer 1 circuit controls the dead time (T_{DT}) delay between the T1HS1 and T1HS2 output edge transitions through the DT[4:0] bits of the DTIME register. The dead time counter delay is first triggered after the TMR1 counter equals to the T1CMPA value and the T1HS1 signal transitions from its resting (off) to its active (on) state. Once the programmed T_{DT} completes, the T1HS2 signal then transitions from its resting (off) to its active (on) state. The dead time counter delay is triggered for a second time after the TMR1 counter equals to the T1RA value and the T1HS2 signal transitions from its active (on) to its resting (off) state. Once the programmed T_{DT} completes, the T1HS1 signal then transitions from its active (on) to its resting (off) state ending the PWM cycle. The PWM cycle is considered complete once the TMR1 counter completes the T1RA count plus T_{DT} even in the T1HS1 and T1HS2 outputs are disabled.

The T1HS1 and T1HS2 PWM output signals may be programmed to be automatically disabled by the output of the digital filter (PWMOFF) in Programmable Comparator circuit. The output may be programmed to disable the Timer 1 circuit completely or disable only the current PWM cycle. Refer to the Programmable Comparator Circuit section of the datasheet for details.

The Timer 1's ADSTROBE output signal may be configured as the G1/ADSTROBE device output if the T1BOUT bit of the T1CNTRL register is set. The ADSTROBE signal, however, is always generated by the Timer 1 circuit. Initially, the ADSTROBE begins its PWM cycle at its resting (off) state and transitions to its active (on) state once the TMR1 counter completes its count equal to the T1CMPB value. The active (on) edge transition of the ADSTROBE output may be programmed to automatically trigger an ADC conversion cycle if the ENDAS bit of the ADCNTRL2 register is set. Refer to the ADC Circuit section of the datasheet for details.

The T1PND bit of the T1CNTRL register is set once the TMR1 counter completes the count equal to the T1RA value (overflows). Software may use the T1PND bit to monitor the PWM cycles and/or trigger microcontroller hardware interrupts (TMRI1) if the T1EN bit of the T1CNTRL register is set. Software must clear the T1PND bit in order to detect a new overflow condition and/or trigger a new interrupt.[6]

Figure 11. Timer 1's PWM Mode Block Diagram

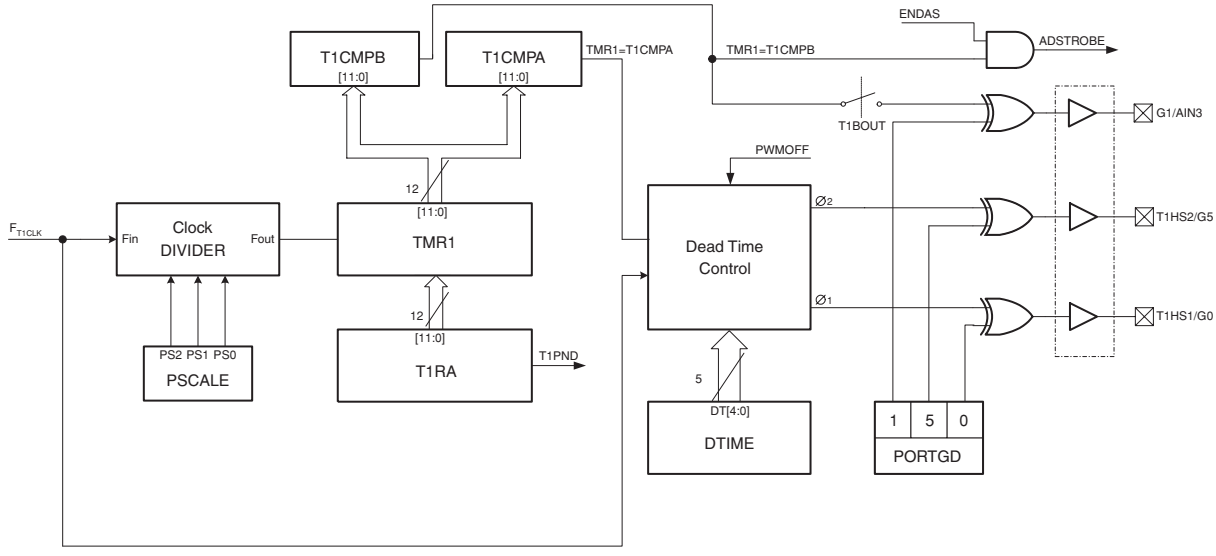
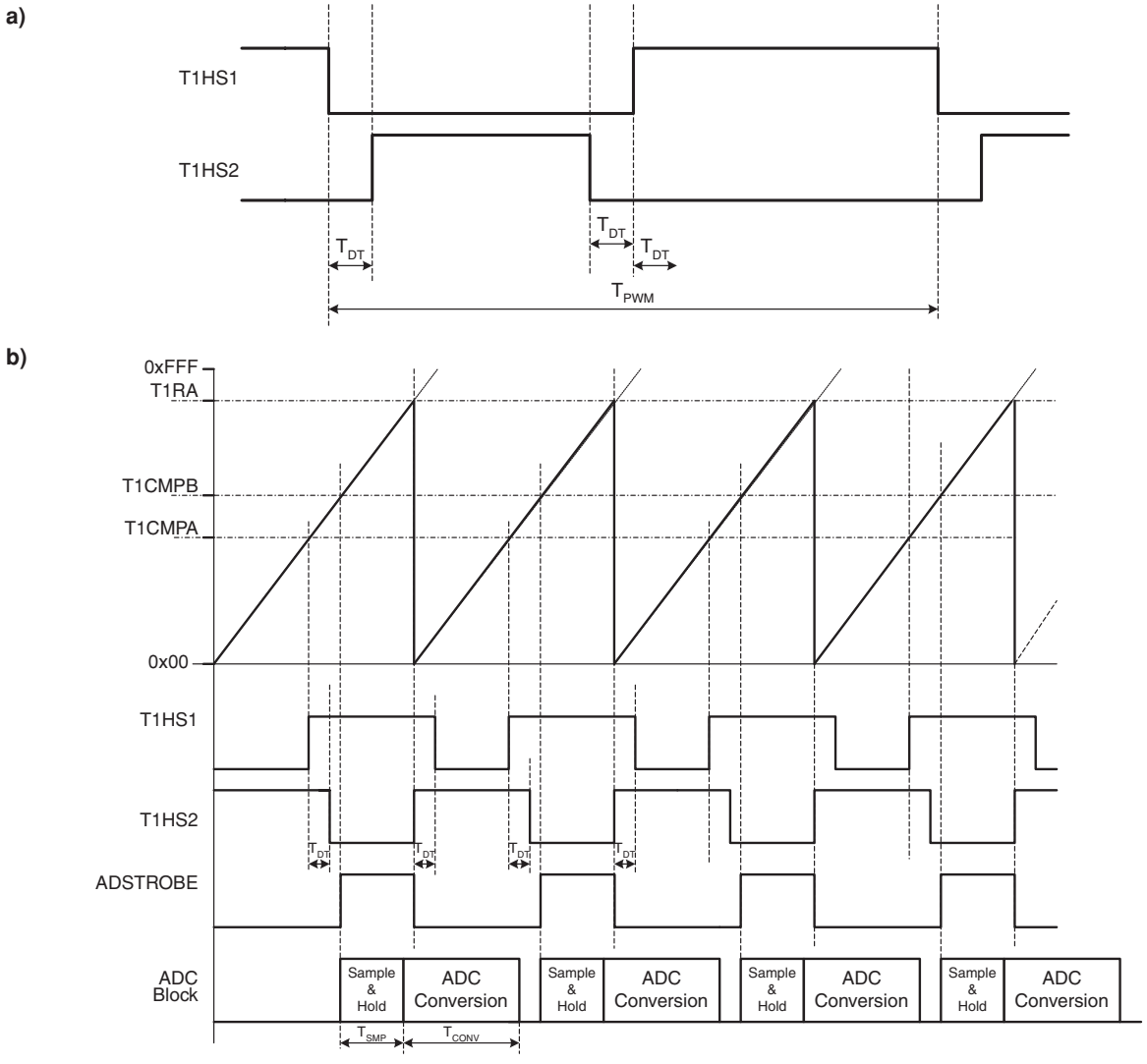


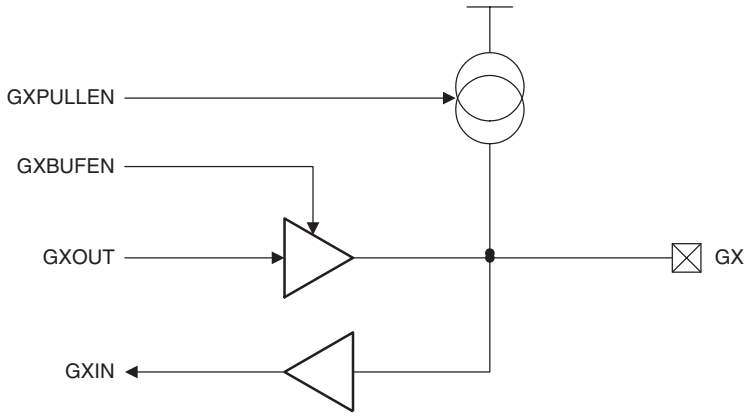
Figure 12. Example PWM Output Signals a) and b)



8 I/O Ports

The eight I/O pins (six on the 8-pin package option) are bi-directional (see Figure 14). 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.

Figure 14. PORTGD Logic Diagram



8.1 I/O Registers

The I/O pins (G0–G7) have three memory mapped port registers associated with the I/O circuitry: a Port Configuration (PORTGC), Port Data (PORTGD) and Port Input (PORTGP) register.[1] PORTGC is used to configure the pins as inputs or outputs. A pin may be configured as an input by writing a 0 or as an output by writing a 1 to its corresponding PORTGC bit. If a pin is configured as an output, its PORTGD bit represents the state of the pin (1 = logic high, 0 = logic low). If the pin is configured as an input, its PORTGD bit selects whether the pin is a weak pull-up or a high-impedance input. Table 20 provides details of the port configuration options. The port configuration and data registers can both be read from or written to. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports MIW, all input ports have Schmitt triggers.

Upon power-up, the PORTGC and PORTGD registers are initialized to 0x00. However, the G0/T1HS1 and G5/T1HS2 pins may be defaulted to the different I/O configurations defined by the default I/O configuration bits of the Initialization Register 4. Refer to Table 29 in the Device Memory section of the datasheet for details.

Table 20. I/O Register Bit Assignments

PORTGC, PORTGD, PORTGD Registers (addr. 0xB3, 0xB2, 0xB4)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
G7[2]	G6[2]	G5	G4	G3	G2	G1	G0

Table 21. I/O Configuration Options

PORTGC Bit	PORTGD Bit	Port Pin Configuration
0	0	High-impedance input (tri-state input)
0	1	Input with pull-up (weak one input)
1	0	Push-pull zero output
1	1	Push-pull one output

1. Refer to Table 30 of the Device Memory section of the datasheet for the detailed memory map.
2. Available only on the 14-pin package option.

9 Multi-input Wakeup Circuit

The Multi-input Wakeup (MIW) circuit may be used to wake the device from either Halt or Idle Mode[1] with an external event, generate flags for software monitoring and microcontroller hardware interrupts by any one or all I/O ports (G0–G7). The MIW circuit is configured using the Wakeup Enable (WKEN), Wakeup Edge (WKEDG), Wakeup Pending (WKPND) and T0CNTRL memory mapped registers.[2] The WKEN, WKEDG and WKPND are 8-bit registers where each bit corresponds to an I/O port pin (see Table 21). All four registers are initialized to 0x00 upon a system reset.

The PWMOFF output signal may also be programmed as an input of the G6 port MIW circuit. Interrupts may be triggered if the PWMOFF/G6 input MIW circuit is enabled and configured to trigger its microcontroller hardware interrupt (EDGEI). Bit 6 (PWMINT) of the DDELAY register, if set to 1, selects the PWMOFF signal in place of its G6 input to the MIW circuit. Software must then enable the MIW PWMOFF/G6 circuit by setting the WKEN[6] bit. The WKEDG[6] bit must also be cleared to select the rising edge transitions on the PWMOFF signal as its WKPND[6] bit trigger. Software may monitor the WKPND[6] flag or enable the MIW hardware interrupt (EDGEI) to help detect when the PWMOFF signal is triggered. Refer to the Programmable Comparator Circuit sections of the datasheet for addition details.

9.1 MIW Configuration Registers

The Wakeup Enable (WKEN) register individually enables an I/O port's edge transition to trigger a wakeup/interrupt pending flag. If the WKEN register bit is 1, the corresponding I/O port's MIW circuitry (defined by its bit number) is enabled; otherwise, the port circuitry remains disabled and the pending flag may not be triggered.

The Wakeup Edge (WKEDG) register bits are used to program an enabled I/O port's pending flag to be triggered from either a rising-/falling-edge transition. If the WKEDG register bit is 1, a falling-edge transition of the enabled I/O port will trigger the pending flag. If zero, a rising-edge transition of the enabled I/O port will trigger the pending flag.

The MIW circuit shares a single hardware interrupt (EDGEI) among all pending flags and is enabled by the Wakeup Interrupt enable (WKINTEN) bit of the T0CNTRL register.[2] The WKINTEN bit enables hardware interrupts for the MIW circuit if set to 1.[3]

The Wakeup Pending (WKPND) register contains the pending flags corresponding to each of the I/O port pins. If a WKPND register bit is 1, the programmed I/O port edge transition has triggered its pending flag. If zero, the flag is not pending and no transition has occurred from the last pending reset. A pending flag may only be triggered by enabled I/O ports (if its WKEN register bit is 1). Once a pending flag is triggered, all flags are logically-ORed together to trigger a WAKEOUT if in Halt/Idle Mode and/or hardware interrupts (if enabled). If software is to re-enter Halt/Idle Mode, all pending flags must be cleared, otherwise the command is ignored. Since all MIW pending flags share a single hardware interrupt, software must take care with the handling of the pending flags when more than one pending flag is enabled. As long as a MIW pending flag is set, the hardware interrupt will continue to execute software's MIW interrupt service routine with highest priority until all pending flags are cleared.[4]

Upon exiting Halt/Idle Mode or before leaving software's MIW interrupt service routine, the RBIT instruction may be used to clear a particular pending flag. The RBIT instruction takes two instruction clock cycles to complete its execution. In the first cycle, all eight register bits are automatically read to obtain their most current value. In the second cycle, the bit to be cleared is given its new value and all bits are then re-written to the register. Using the RBIT instruction to clear an individual pending flag causes no potential hazards if only one wakeup I/O port is enabled. However, if more than one I/O port is enabled software may inadvertently clear a recently triggered pending flag if the trigger happened during the second phase of the RBIT instruction execution. To avoid this condition, the LD instruction must be used to clear a set pending flag. The MIW circuit is designed such that software may not trigger a pending flag by writing a 1 to a WKPND register bit, it may only be cleared. The action of writing a 1 to a WKPND register bit holds the current bit value. The action of writing a 0 to a WKPND register bit clears the bit value. Therefore, the "LD WKPND, #0F7H" instruction will clear the WKPND[3] while all others bits remain the same.

10 8-Bit Microcontroller Core

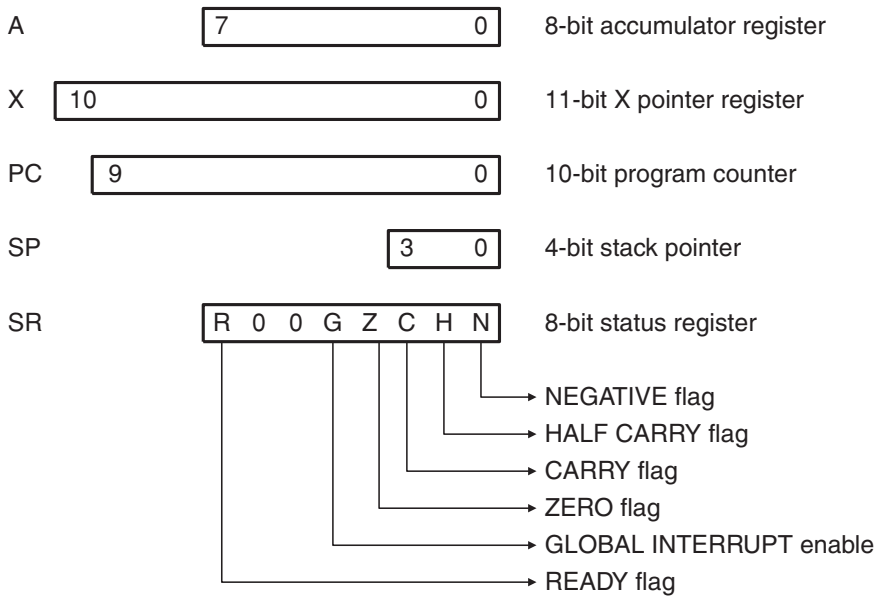
The FMS7401L's 8-bit microcontroller core is specifically designed for low cost applications involving bit manipulation, shifting and block encryption. It is based on a modified Harvard architecture meaning peripheral, I/O and RAM locations are addressed separately from instruction data.

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

10.1 Core Registers

The microcontroller core 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 memory mapped while A and PC are not.

Figure 16. Core Program Model



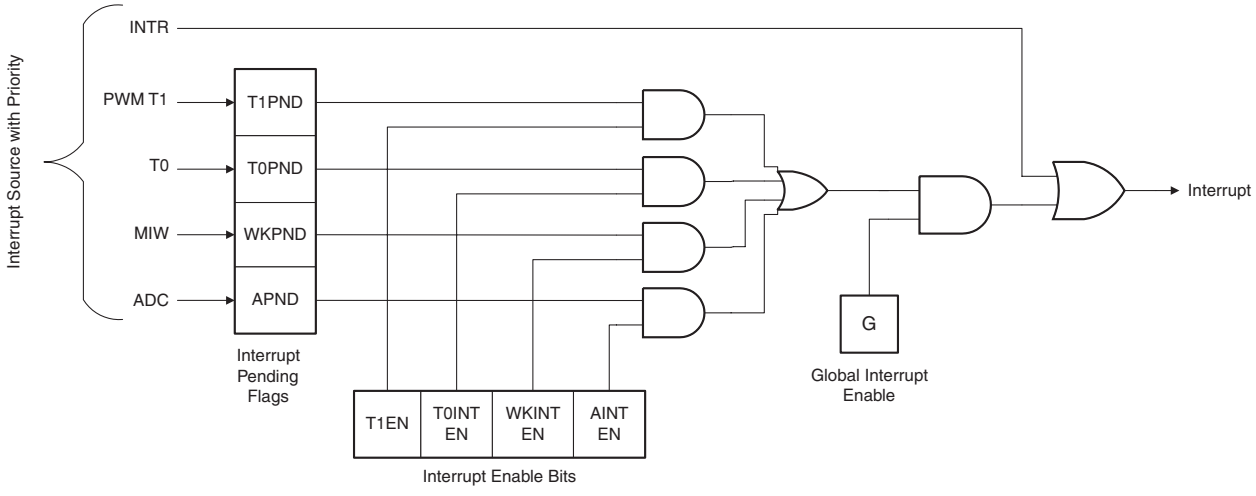
10.1.6 Interrupt Handling

When an interrupt is recognized, the current instruction completes its execution. The return address (the current value in the program counter, PC) is pushed onto the stack, the global interrupt (G) mask of the status register (SR) is cleared, and execution continues at the address specified by the respective interrupt vector (see Table 30). This process takes five instruction cycles to complete. The interrupt vector contains the address of the software's interrupt service routine (ISR). Initially, the ISR may save (if necessary) the status register's contents. Software, however, cannot restore SR using the traditional microcontroller methods. Just before ending the ISR, software may restore the contents of SR by using only the special inherent instructions (e.g. SC, RC and LDC) or specially defined software routines since all SR bits except for G are read only when using direct, indirect, or indexed instructions (e.g. LD , ST, RBIT or SBIT). Upon exiting the ISR, software must clear the appropriate triggering pending flag and execute a return-from-interrupt (RETI) instruction. The RETI instruction pulls the saved return address off the stack in reverse order restoring PC and setting G of SR to one. Instruction execution resumes at the restored the program counter address.

The microcontroller core is capable of supporting five interrupts. Four are maskable through G of the SR and the fifth (software interrupt) is not inhibited by G (see Figure 17). The execution of the INTR instruction generates a software interrupt. Once the INTR instruction is executed, the microcontroller core will interrupt whether G is set or not. The INTR interrupt is executed in the same manner as the other maskable interrupts where PC is stacked and G is cleared. This means, if G was enabled prior to the software interrupt the RETI instruction must be used to return from interrupt in order to restore G to one. However, if G was not enabled prior to the software interrupt the RET instruction must be used.

In the case of simultaneous multiple interrupts, the microcontroller core prioritizes the interrupts. See Table 23 for the interrupt service priority sequence.

Figure 17. Basic Interrupt Structure



1. The FMS7401L's normal mode operation begins after a system reset and is when the 8-bit microcontroller core begins executing the instruction program residing in the code EEPROM memory. During this time, the code EEPROM memory may only be read by software not written. Refer to the Device Memory section of the datasheet for additional memory addressing information.
2. A 3-bit value in cases like the IFBIT and IFNBIT instructions.
3. A 12-bit value in the case of instructions writing to X.
4. The content of XHI (X[10:8]) is ignored.
5. The program memory space for the FMS7401L is 0xC00 to 0xFFF; however, the program counter will use only the 10 least significant bits of the address provided.
6. Although the JP instruction can jump forward 31 bytes, it can only jump backwards 30 bytes because the program counter is automatically incremented while the JP instruction is being executed.

prior to leaving the factory with the appropriate calibration value and with the ports configured as tri-state inputs. In programming mode, the default port configuration may be changed; however, be sure to maintain the factory current source calibration value since writes to a single register must affect all bits.

The Initialization Registers 1, 2, 3 and 4 can be read from and written to while in programming mode. However, re-trimming the internal oscillator and re-calibrating the analog circuits once the device has left the factory is discouraged and will void all device guarantees.

Table 26. Initialization Register 1 Bit Definitions

Initialization Register 1 (volatile/non-volatile addr. 0xB9, 0xBB)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CLK_ADJ	CMODE	unused	WDEN	BOREN	UBD	WDIS[6]	RDIS[6]

(7)	CLKADJ	When set, the internal clock trimming register (volatile Initialization Register 2, Addr. 0xBA) can be accessed by the core in order to modify the internal clock frequency.
(6)	CMODE	Clock mode select: 0 = Internal Oscillator, 1 = External Oscillator
(4)	WDEN	If set, the on-chip processor Watchdog Timer resets are enabled.
(3)	BOREN	If set, the on-chip Brown-out Reset comparator circuit is enabled.
(2)	UBD	If set, write access of the upper 32 bytes of the data EEPROM (0x60-0x7F) is disabled in both programming and normal mode.[1,2]
(1)	WDIS[6]	If set, write access of the device memory is permanently disabled while in programming mode.[2]
(0)	RDIS[6]	If set, read access of the device memory is permanently disabled while in programming mode.[2]

Table 27. Initialization Register 3 Bit Definitions

Initialization Register 3 (volatile/non-volatile addr. 0xD0, 0xD1)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
unused	unused	BOR_TRIM			COMP_TRIM		

(5:3)	BOR_TRIM	These three bits allow for the calibration of the Brown-out Reset comparator circuit.
(2:0)	COMP_TRIM	These three bits allow for the calibration of the Programmable Comparator's upper range circuit.

Table 28. Initialization Register 4 Bit Definitions

Initialization Register 4 (volatile/non-volatile addr. 0xD3, 0xD4)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
T1HS_DIR	T1HS2_LEV	T1HS1_LEV	ISOURCE_TRIM[4:0]				

(7)	T1HS_DIR	Initializes during reset, the T1HS1 (G0) and T1HS2 (G5) I/O ports both either inputs or outputs. This bit shadows directly to bits 0 and 5 of PORTGC.
(6:5)	T1HSx_LEV	Initializes during reset, the individual T1HS1 (G0) and T1HS2 (G5) I/O port level. These bits shadow directly to bits 0 and 5 of PORTGD.
(4:0)	ISOURCE_TRIM	These five bits allow for the calibration of internal current source generator.

AC Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OSC}[5]	Internal oscillator frequency (factory trim set-point)	Vcc=3.3V	1.96	2.00	2.04	MHz
Δckv	Internal oscillator frequency voltage variation		-0.5		+0.5	%
Δckt	Internal oscillator frequency temperature variation	Vcc=3.3V -40°C to 85°C	-3		+3	%
		Vcc=3.3V -40°C to 125°C	-4		+4	%
F_{PLL}	PLL input reference frequency			2.00		MHz
T_{PLL_LOCK}[3]	PLL Lock time	-40°C to 125°C			60	µS
T_{EEW}	EEPROM Writing time			3.7	5	mS
T_{RESET}[3]	System reset time	-40°C to 125°C	2.5	3.7	4.7	mS
T_{DIO}[3]	T1HS1 and T1HS2 default I/O configuration settling time			40		µS
T_{HALT_REC}[3]	Internal device start time after exiting from Halt where $F_{ICLK} = F_{OSC}$[2]	-40°C to 125°C		5	7	µS

Brown-out Reset (BOR) Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Parameter	Conditions	Min.	Typ.	Max.	Units
BOR Trigger Vcc Threshold Level		2.64	2.71	2.78	V
	-40°C to +85°C	2.60		2.83	V
	-40°C to 125°C	2.59		2.83	V

Programmable Comparator Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Parameter	Conditions	Min.	Typ.	Max.	Units
All 32 thresholds (V_{THU})		-6%	V_{THU}	+6%	V
Upper Range (0.45V to 2.0V)	-40°C to 125°C	-8%	V_{THU}	+8%	V
All 31 thresholds (V_{THL})		V_{THU} – 30mV	V_{THL}	V_{THU} + 30mV	V
Lower Range (0.03V to 0.43V)	-40°C to 125°C	V_{THU} – 35mV	V_{THL}	V_{THU} + 35mV	V
Comparator Response Time[3]	2mV overdrive		359		nS
	5mV overdrive		173		nS
	10mV overdrive		95		nS

Ordering Information

FSID	Package	Supply Voltage	Temperature Range	Packaging Option	
				Method	Qty
FMS7401LEN	PDIP8	2.7V to 3.6V	-40°C to 85°C	Rail	40
FMS7401LVN	PDIP8	2.7V to 3.6V	-40°C to 125°C	Rail	40
FMS7401LEN14	PDIP14	2.7V to 3.6V	-40°C to 85°C	Rail	25
FMS7401LVN14	PDIP14	2.7V to 3.6V	-40°C to 125°C	Rail	25
FMS7401LEM8X	SOIC8	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMX	SOIC14	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMT8X	TSSOP8	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMTX	TSSOP14	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

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

1/24/05 0.0m 005
Stock#DS30007401