Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

FMS7401L

Digital Power Controller

General Description

The FMS7401L is a Digital Power Controller designed for applications requiring ease of digital based control over analog based implementations. The FMS7401L is an ideal solution to implement ballast control, motor control and battery management functions. It integrates a wide variety of analog blocks with an 8-bit microcontroller core to offer a complementary feature set with high performance, low power and small size in a single chip.

The FMS7401L is intended for applications using a supply voltage in the 2.7V to 3.6V range. It is fabricated using CMOS technology and is fully static offering a significant power savings. The FMS7401L is available in both 8-pin and 14-pin PDIP, SOIC and TSSOP packages.

Features

- 8-bit Microcontroller Core
- 1K bytes on-board code EEPROM
- 64 bytes data EEPROM
- 64 bytes SRAM
- Watchdog Reset
- Multi-input Wakeup on all general purpose I/O pins
- Fast 12-bit PWM timer with dead time control and half-bridge output drive
  - Input Capture Mode
- 5-Ch 8-bit Analog-to-Digital Converter
  - 20 µS conversion time
  - Sample and Hold
  - Internal Voltage Reference (1.21V)
  - Gated Auto-sampling Mode
- Auto-zero Amplifier (gain 16)
- Uncommitted Amplifier
- Internal Current Source Generator (1mA)
- 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
- Programmable Comparator (63 Levels)
- Brown-out Reset
- Software selectable I/O option
- Push-pull outputs with tri-state option
- Weak pull-up or high impedance inputs
- Fully static CMOS
  - Power Saving Halt Mode
    - < 1.3µA @ 3.3V
  - Power Saving Idle Mode
    - < 180µA @ 3.3V
- Single supply operation
  - 2.7V – 3.6V
- 40 years data retention
- 100,000 data changes
- 8-/14-pin PDIP, SOIC, and TSSOP packages
- In-circuit programming
  - Fast Page-write Programming Mode

Device	Supply Voltage	Program Memory (bytes)	Data Memory (bytes)		I/O	Pin Count
			SRAM	Data EEPROM		
FMS7401L	2.7V – 3.6V	1K	64	64	6	8
FMS7401L	2.7V – 3.6V	1K	64	64	8	14

Block Diagram

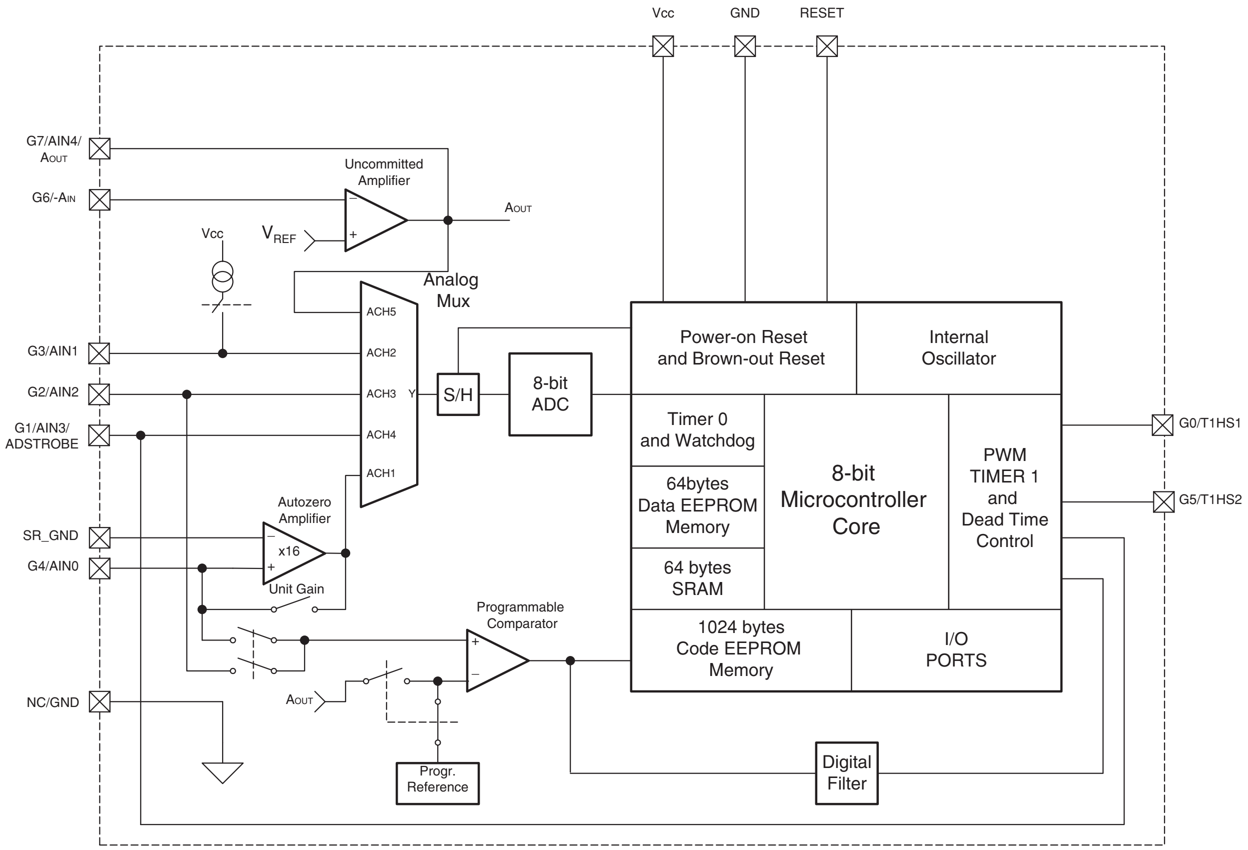
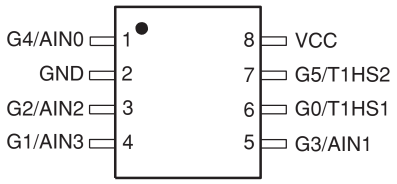
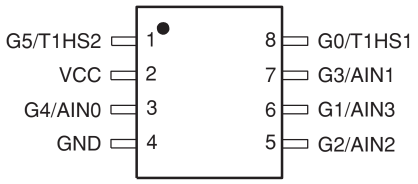


Figure 1. FMS7401L Block and Connection Diagram

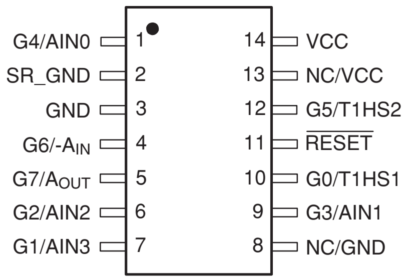
Pin Configurations



FMS7401L 8-Pin PDIP/SOIC



FMS7401L 8-Pin TSSOP



FMS7401L 14-Pin PDIP/SOIC/TSSOP

FMS7401L Pin Definitions

Pin Number				
8-Pin		14-Pin		
PDIP SOIC	TSSOP	PDIP SOIC TSSOP	Pin Name	Pin Function Description
1	3	1	G4/AIN0	General purpose I/O port (bit 4 of the I/O configuration registers).
				AIN0 analog input of the ADC (autozero amplifier's positive terminal).
				Programmable Comparator non-inverting input, if COMPSEL=0.
2	4	3	GND	Digital ground pin.
3	5	6	G2/AIN2	General purpose I/O port (bit 2 of the I/O configuration registers).
				AIN2 analog input of the ADC.
				Programmable Comparator non-inverting input, if COMPSEL=1.
4	6	7	G1/AIN3/ADSTROBE	General purpose I/O port (bit 1 of the I/O configuration registers).
				AIN3 analog input of the ADC.
				External digital clock input.
				PWM Timer 1's ADSTROBE output.
5	7	9	G3/AIN1	General purpose I/O port (bit 3 of the I/O configuration registers).
				AIN1 analog input of the ADC.
				Internal current source generator pin.
6	8	10	G0/T1HS1	General purpose I/O port (bit 0 of the I/O configuration registers).
				PWM Timer 1's T1HS1 output.
7	1	12	G5/T1HS2	General purpose I/O port (bit 5 of the I/O configuration registers).
				PWM Timer 1's T1HS2 output.
8	2	14	VCC	Supply voltage input.
–	–	2	SR_GND	AIN0 analog input of the ADC (autozero amplifier's negative terminal). SR_GND is internally connected to GND in the 8-pin FMS7401L.
–	–	4	G6/-A_{IN}	General purpose I/O port (bit 6 of the I/O configuration registers).
				Uncommitted amplifier negative analog input.
–	–	5	G7/AIN4/A_{OUT}	General purpose I/O port (bit 7 of the I/O configuration registers).
				AIN4 analog input of the ADC.
				Uncommitted amplifier analog output.
–	–	8	NC/GND	In the FMS7401L, pin 8 is internally connected to GND. Externally, pin 8 should be left unconnected or connected to GND.
–	–	11	RESET	Active low external reset input.
–	–	13	NC/VCC	In the FMS7401L, VCC is internally connected to pin 13. Externally, pin 13 should either be left unconnected or connected to pin 13.

Table of Contents

List of Figures

List of Tables

1 Reset Circuit

The reset circuit in the FMS7401L contains four input conditions that trigger a main system reset. When the main system reset is triggered, a sequence of events occur defaulting all memory mapped registers (including the initialization registers) and I/Os to their initial states (see Table 1). During the system reset sequence, the instruction core execution is halted allowing time for the internal oscillator and other analog circuits to stabilize. Once the system reset sequence completes, the device will begin with its normal operation executing the instruction program residing in the code EEPROM memory. The time required for the system reset sequence to complete (T_{RESET}) is dependent on the individual trigger condition and is defined in the Electrical Characteristics section of the datasheet. The four reset trigger conditions are as follows:

- Power-on Reset (POR)
- External Reset[1]
- Brown-out Reset (BOR)
- Watchdog Reset[2]

Table 1. Default Register States

Peripheral/Register	External Reset	POR
G1, G2, G3, G4, G6, G7	High-impedance input (tri-state input)	
G0, G5	Defined by Init Reg. 4 (see Table 28)	
SRAM Memory	No change	Unspecified
Stack Pointer	0xF	0xF
Status Register	0x80	0x80
T1CMPA, T1CMPB and T1RA Registers	0xFFF	0xFFF
DTIME Register	0x1F	0x1F
All other memory mapped register not listed above.[3]	0x00	0x00

1.1 Power-on Reset Circuit

The Power-on Reset (POR) circuit maintains the device in a reset state until Vcc reaches a voltage level high enough to guarantee proper device operation. The POR circuit is sensitive to the different Vcc ramp rates and must be within S_{Vcc} as specified in the Electrical Characteristics section of the datasheet.

The POR circuit does not generate a system reset when Vcc is falling. This feature is performed by the Brown-out Reset (BOR) circuit and must be enabled by the BOREN bit of the Initialization Register 1.[4] In the case where Vcc does not drop to 0V before the next power-up sequence, it is necessary to enable the BOR circuit and/or reset the device externally through the RESET pin.[1]

1.2 External Reset[1]

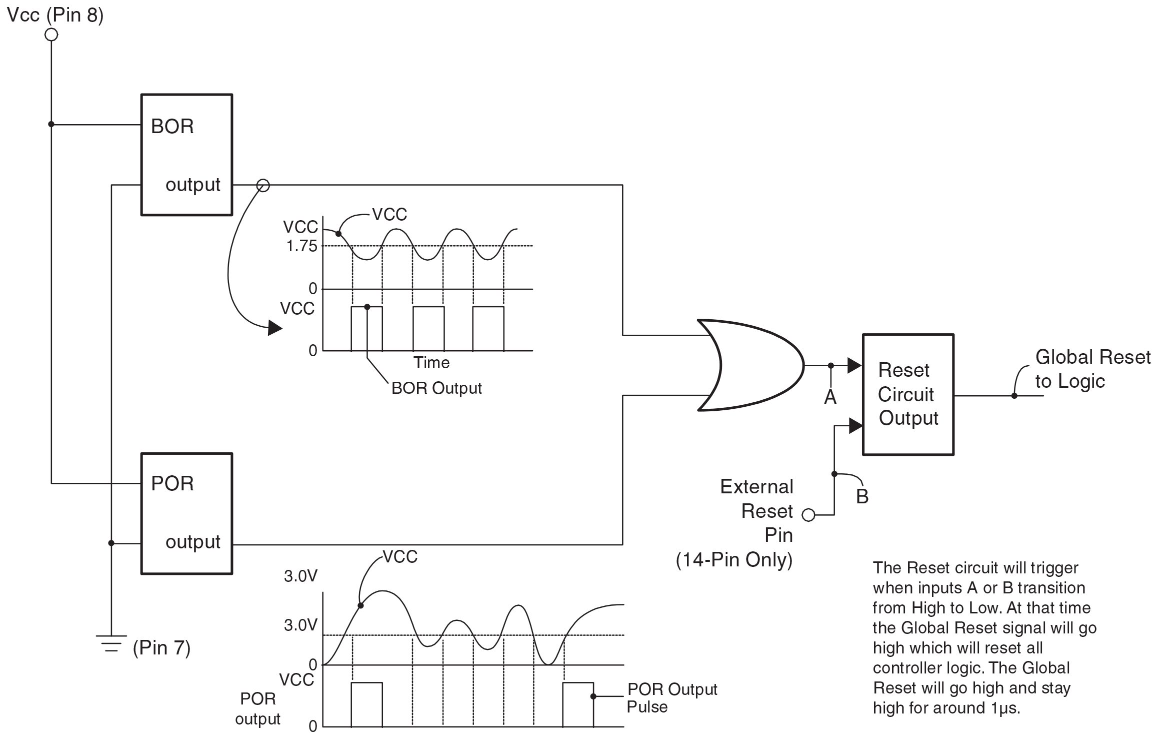
The device may be externally reset through the RESET input pin if the POR/BOR circuits cannot be used to properly reset the device in the application. The RESET input pin contains an internal pull-up resistor making it an active low signal. Therefore, to issue a device system reset the RESET input should be held low for at least 10µS before being released (i.e. returned to a high state). While the RESET input is held low, the internal oscillator and other analog circuits are kept in a low power state reducing the current consumption of the device (a state resembling Halt Mode). In addition, the I/O pins are all initialized to an input tri-state configuration unless defaulted otherwise.[5] At the rising edge of the RESET input signal, the main system reset sequence is triggered releasing the internal oscillator and other analog circuits so that they may be initialized and begin their normal operation.

1.3 Brown-out Reset Circuit

The Brown-out Reset (BOR) circuit is one of the on-chip analog comparator peripherals and must be enabled through the BOREN bit of the Initialization Registers 1.[4] The BOR circuit is used to hold the device in a reset state when Vcc drops below a fixed threshold defined in the Electrical Characteristics section of the datasheet. While in reset, the device is held in its initial condition until Vcc rises above the fixed/power-on threshold. Shortly after Vcc rises above the fixed/power-on threshold, the internal system reset sequence is started. Once the system reset sequence completes, the device will begin with its normal operation executing the instruction program residing in the code EEPROM memory.

The BOR circuit should be used in situations when Vcc rises and falls slowly and in situations when Vcc does not fall to 0V before rising back to the device's normal operating range. The BOR circuit can be thought of as a supplement function to the POR circuit if Vcc does not fall below 0.7V.

Figure 2. BOR and POR Circuit Relationship Diagram



The Reset circuit will trigger when inputs A or B transition from High to Low. At that time the Global Reset signal will go high which will reset all controller logic. The Global Reset will go high and stay high for around 1µs.

1. Available only on the 14-pin package option.
2. Refer to the Timer 0 Circuit section of the datasheet for details regarding the Watchdog Reset.
3. Refer to Table 30 of the Device Memory section of the datasheet for the detailed memory map.
4. Refer to the Device Memory section of the datasheet for details regarding the Initialization Register 1.
5. Refer to Table 28 and Table 29 of the Device Memory section of the datasheet for details.

2 Clock Circuit

The FMS7401L may be clocked using its internal oscillator circuit or using an external digital clock signal. The desired clock source is selectable by the CMODE bit of the Initialization Register 1.[1] During the reset sequence, the CMODE bit is updated and the desired clock source (also called the device reference clock or F_{RCLK1}) takes control of the device. All devices are defaulted from the factory to use the internal oscillator as their main system instruction clock source. After power-up, the internal oscillator runs continuously unless entering Halt Mode or using an external clock source.

Table 2. CMODE Bit Definition

CMODE	F_{RCLK1} Clock Source
0	Internal Oscillator (@ F_{OSC})
1	External digital clock (G1/AIN3)

The internal oscillator signal is factory trimmed to yield the F_{OSC} frequency as specified in the Electrical Characteristics section of the datasheet. If the external digital clock is selected, the input signal must have a 50/50 duty cycle, can range from DC to the F_{OSC}, and must be available upon power-up. When the device is driven using an external clock source, the clock input to the device should be provided through the AIN3/G1 input.

Once the source of F_{RCLK1} is selected, the clock is then used as the reference clock for the PLL, the clock to the digital filter in the Programmable Comparator circuit, and is divided-by-2 to be used as the main system instruction clock (F_{ICLK}) of the device (see Figure 3 and Figure 4).

2.1 PLL

The FMS7401L has an internal digital clock multiplier (PLL) that steps-up the F_{RCLK1} frequency by a multiplication factor of 32. The multiplied PLL output is then divided by a factor of 1, 2, 4, and 8 in order to generate its programmable output frequencies that may be used as the main system instruction clock or by the PWM Timer 1 circuit. The PLL provides the ability to run the PWM Timer 1 circuit at a frequency as high as 64MHz while the rest of the device operates at a much slower frequency keeping the total current consumption low.

The reference clock of the PLL is defined by the F_{RCLK2} signal (as shown in Figure 3 and Figure 4) and sourced by the F_{RCLK1} signal. In order to yield the proper output frequencies offered by the PLL, F_{RCLK2} must operate at the F_{PLL} frequency as specified in the Electrical Characteristics section of the datasheet. In the case that F_{RCLK1} is operating at the upper F_{OSC} frequency,[2] the REFBY2 bit in the ADCNTRL2 register[3] must be set in order to divide the F_{RCLK1} by 2 to yield the appropriate F_{PLL} frequency of the F_{RCLK2} signal. Once the REFBY2 bit is set, the F_{RCLK2} signal that drives the PLL and digital filter in the Programmable Comparator circuit operates at a $F_{RCLK1}/2$ frequency. If an external digital clock is sourcing F_{RCLK1}, the input signal must be supplied at the specified F_{OSC} frequency in order to meet the specified F_{PLL} frequency of the F_{RCLK2} signal.

Table 3. PLL Frequency Selection (F_{PLL}/F_{OSC} = 2MHz)

FS[1:0]		F_{RCLK2}	F_{ICLK} (FMODE = 0)	F_{ICLK} (FMODE = 1)	F_{PWMCLK}
0	0	2 MHz	1 MHz	8 MHz	8 MHz
0	1	2 MHz	1 MHz	8 MHz	16 MHz
1	0	2 MHz	1 MHz	8 MHz	32 MHz
1	1	2 MHz	1 MHz	8 MHz	64 MHz

The PLL outputs may be used to clock both the PWM Timer 1 circuit and the main system clock. However, the PLL must first be enabled by setting the PLLEN bit of the PSCALE register.[4] Once set, the PLL is turned on and begins the locking phase. Before using any of the PLL outputs, software must wait the T_{PLL_LOCK} to ensure that the PLL is locked into its appropriate frequency and in phase. The PLLEN bit may not be changed while the PWM Timer 1 circuit is in run mode.[5] Any write attempts to this bit during this condition will not change its value.

The PWM Timer 1 circuit may be clocked either by the PLL's F_{PWMCLK} output or by the main system clock (F_{ICLK}). The FSEL bit of the PSCALE register[4] selects between the PLL's F_{PWMCLK} output (if FSEL=1) or F_{ICLK} (if FSEL=0). The FSEL bit may not be set if the PLL is not enabled (PLLEN=0) or changed while the PWM Timer 1 circuit is in run mode.[5] Any write attempts to this bit during these conditions will not change its value.

The FS[1:0] bits of the PSCALE register[4] select the divide factor for the F_{PWMCLK} output (see Table 3). The FS bits may be changed by software at any time; however, if the PWM Timer 1 circuit is in run mode the FS[1:0] value will not change the F_{PWMCLK} output frequency until after the PWM cycle ends (once the TMR1 counter overflows). The last FS[1:0] value at the PWM cycle end time will dictate the divide factor of the F_{PWMCLK} output for the next PWM cycle. When reading the FS[1:0], the value reported will be the last value written by software (it may not necessarily reflect the divide factor for the current PWM cycle).

The main system instruction clock (F_{ICLK}) source may be provided by the internal oscillator (F_{OSC}) or the PLL's $F_{(FS=0)}$ output with the same divide factor as the FS[1:0] = 00 selection.[6] The FMODE bit of the PSCALE register[4] selects between the $F_{(FS=0)}$ (if FMODE=1) or F_{RCLK1} divided-by-2 signal. With the FMODE bit enabled, it is possible to execute instructions at a speed eight times faster than the standard. The FMODE bit may not be set if the PLL is not enabled.[5] Any attempts to write to FMODE while PLLEN=0 will force FMODE=0 ignoring any set instruction. Once the PLL has been enabled, software may change F_{ICLK}'s source on-the-fly during normal instruction execution in order to speed-up a particular action.

In order to synchronously disable the PLL clocking structure, software must clear FSEL and FMODE before clearing the PLLEN bit in order to disable the PLL successfully e.g. using separate instructions like "RBIT PLLEN, PSCALE." There are also special conditions for Halt/Idle power saving modes that must also be considered. Please refer to the Power Saving Modes section of the datasheet for details.

Figure 3. Internal Clock Scheme

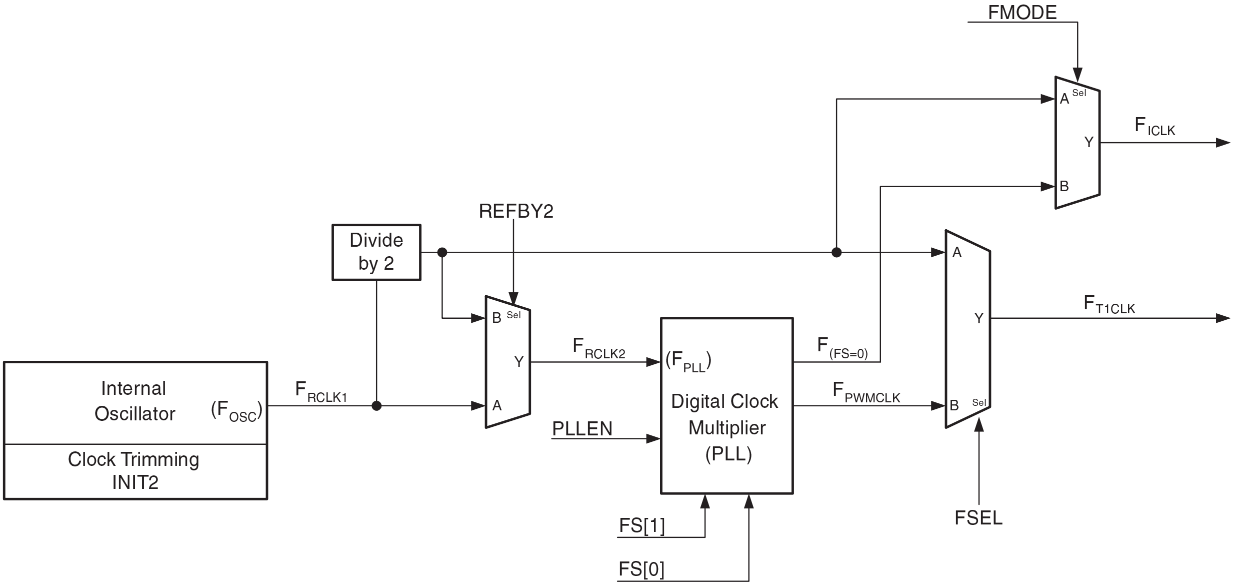
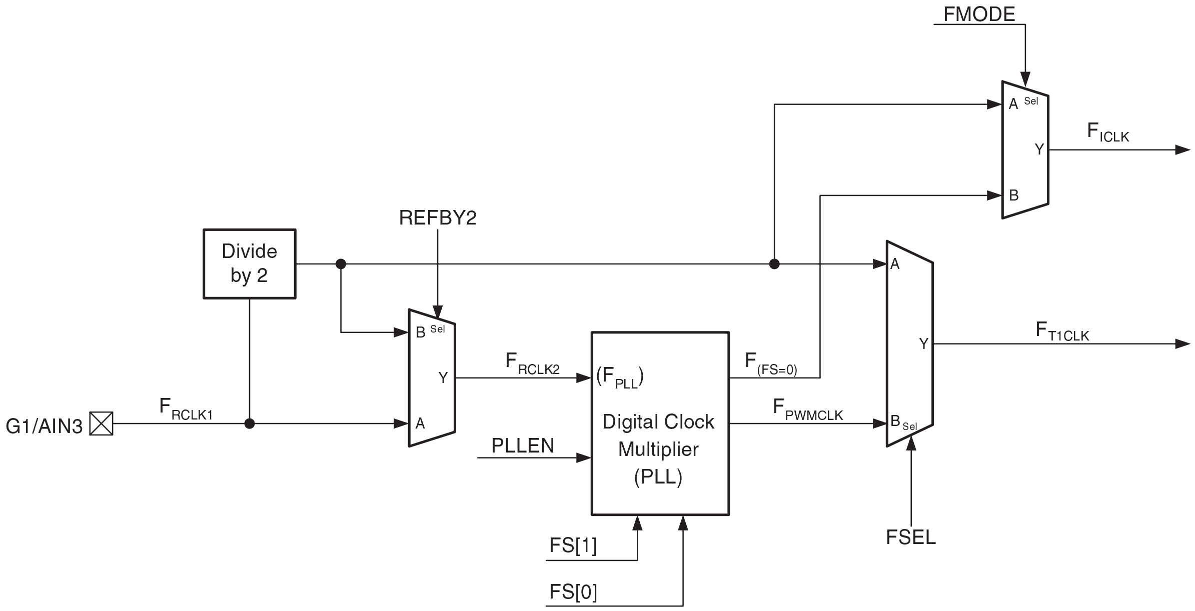


Figure 4. External Clock Scheme



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.

3 Power Saving Modes

The FMS7401L has both Halt and Idle power saving modes. Each mode is controlled by software and offers the advantage of reducing the total current consumption of the device in an application. For all current consumption details, please refer to the Electrical Characteristics section of the datasheet.

3.1 Halt Mode

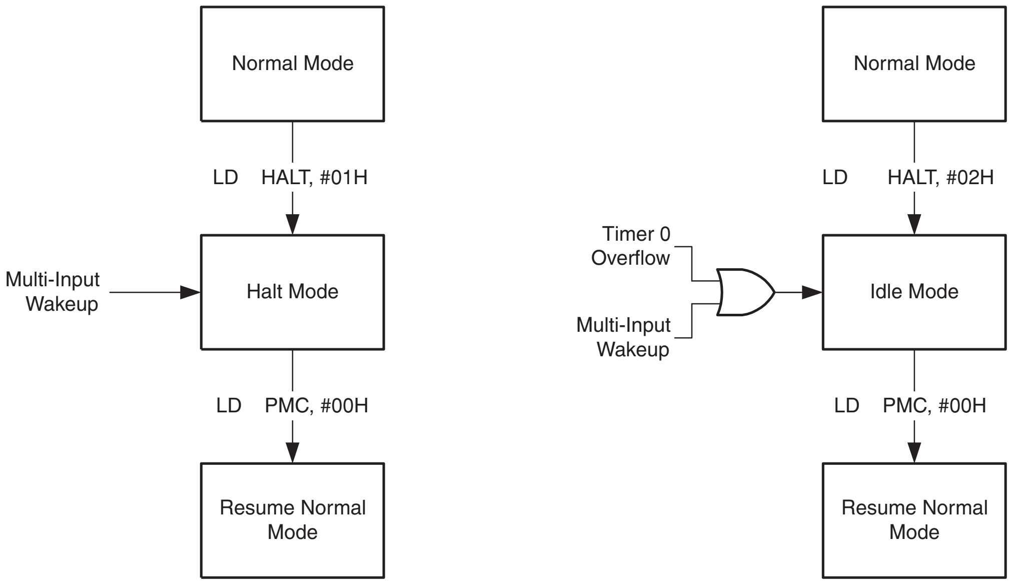
Halt Mode is a power saving feature that almost completely shuts down the device for current conservation. The device is placed into Halt Mode by setting the Halt enable bit (EHALT) of the HALT register using either the “LD M, #” or the “SBIT #, M” instructions in the software. EHALT is a write only bit and is automatically cleared upon exiting Halt Mode. When entering Halt Mode, the internal oscillator and all other on-chip systems including the Programmable Comparator (COMP) and Brown-out Reset (BOR) circuits are shut down.

The device can exit Halt Mode only by the Multi-input Wakeup (MIW) circuit.[1] Therefore, prior to entering Halt Mode, software must first configure the MIW circuit. After a wakeup from Halt Mode, a T_{HALT_REC}[2] start-up delay is initiated to allow the internal oscillator and other analog circuits to stabilize before normal device execution resumes. Immediately after exiting Halt Mode, software must clear the Power Mode Clear (PMC) register by using only the “LD M, #” instruction (see Figure 5).

Table 4. HALT Register Definition

HALT Register (addr. 0xB7)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	EIDLE	EHALT

Figure 5. Recommended Halt/Idle Flow



3.1.1 PLL Steps for Halt Mode

When using Halt Mode and the PLL in an application, software must take the appropriate steps in order to keep the integrity of the clock structure before entering and after exiting Halt since the PLL must be disabled. While in Halt Mode, all other device circuits except for the MIW are disabled. Once the PLL is disabled, all output frequencies are turned off. If the PLL is re-enabled, it must complete the lock phase before software may enable the use of the outputs to clock any of the device circuits. Therefore, upon exiting Halt Mode software must wait the T_{PLL_LOCK}[2] to ensure that the PLL is locked into its appropriate frequency and in phase.

1. Initially, the PLLEN bit of the PSCALE register must be set in order to enable the PLL circuit.
2. If the PLL outputs are to be used to clock any of the device circuits, FMODE and/or FSEL of the PSCALE register must be set after the appropriate T_{PLL_LOCK} wait time.[3]
3. Prior to entering Halt Mode, software must clear both FMODE and FSEL (the PWM Timer 1 must be disabled in order to clear either bit) keeping the PLLEN bit 1.
4. Using a separate instruction (e.g. RBIT PLLEN, PSCALE) disable the PLL by clearing the PLLEN bit.
5. Software may then instruct the device to enter Halt Mode.
6. If all disabled circuits must be re-enabled after exiting from Halt Mode, repeat all initial steps enabling all circuits in the appropriate order as well as waiting T_{PLL_LOCK}.

3.2 Idle Mode

In addition to the Halt Mode power saving feature, the device also supports an Idle Mode operation. The device is placed into Idle Mode by setting the Idle enable bit (EIDLE) of the HALT register through software using either the "LD M, #" or the "SBIT #, M" instructions. EIDLE is a write only bit and is automatically cleared upon exiting Idle Mode. The Idle Mode operation is similar to Halt Mode except the internal oscillator, PLL, and Timer 0 circuits remain active while the other on-chip systems including the Programmable Comparator (COMP) and Brown-out Reset (BOR) circuits are shut down.

The device exits Idle Mode automatically by the Timer 0 Idle overflow every 8192 cycles and by the Multi-input Wakeup (MIW) circuit.[1] Software must first configure the MIW prior to entering Idle Mode in order to wake the device from Idle without waiting for the overflow to occur. Once a wake from Idle Mode is triggered, the normal device execution resumes by the next clock cycle. Immediately after exiting Idle Mode, software must clear the Power Mode Clear (PMC) register by using only the "LD M, #" instruction (see Figure 5).

3.2.1 PLL Steps for Idle Mode

When using Idle Mode, the PLL does not need to be disabled prior to entering Idle as it does with Halt Mode. The PLL may remain enabled the entire time the device is in Idle; however, the device will consume additional current. If current consumption is important, consider using Halt instead of Idle Mode or at least disabling the PLL while in Idle.

By keeping the PLL enabled while in Idle Mode, the PLL's outputs remain ready for use at any moment. With the PLL's outputs available, software has the option to source the main system clock (F_{ICLK}) by the PLL's $F_{(FS=0)}$ output when the FMODE bit of the PSCALE register is set.[3] In addition, if the PLL's F_{PWMCLK} output is clocking the PWM Timer 1 circuit,[4] the timer may remain operational while in Idle Mode. However, the total current consumption will increase, hence the recommendation to disable the PWM Timer 1 before entering Idle Mode. In contrast, if F_{ICLK} is clocking the PWM Timer 1, the timer circuit execution (like the main system controller) is stopped during Idle. Whether the PWM Timer 1 is operational or not during Idle Mode, the instruction execution is stopped therefore all pending flags, etc. cannot be serviced.

If the PLL is to be disabled prior to entering Idle Mode, software must take the appropriate steps in order to keep the integrity of the clock structure. Once the PLL is disabled, all output frequencies are turned off. If the PLL is re-enabled, it must complete the lock phase before software may enable the use of the outputs to clock any of the device circuits. Therefore, upon exiting Idle Mode software must wait the T_{PLL_LOCK} to ensure that the PLL is locked into its appropriate frequency and in phase.

1. Initially, the PLLEN bit of the PSCALE register must be set in order to enable the PLL circuit.

2. If the PLL outputs are to be used to clock any of the device circuits, FMODE and/or FSEL of the PSCALE register must be set after the appropriate T_{PLL_LOCK} wait time.

3. Prior to entering Idle Mode, software must clear both FMODE and FSEL (the timer must be disabled in order to clear either bit) keeping the PLLEN bit 1.

4. Using a separate instruction (e.g. RBIT PLLEN, PSCALE) disable the PLL by clearing the PLLEN bit.

5. Software may then instruct the device to enter Idle Mode.

6. If all disabled circuits must be re-enabled after exiting from Idle Mode, repeat all initial steps enabling all circuits in the appropriate order as well as waiting T_{PLL_LOCK}.

[1] The MIW and Timer 0 Circuits are described later in the datasheet.
[2] Refer to the Electrical Characteristics section of the datasheet for details.
[3] Refer to the Clock Circuit's PLL section of the datasheet for details.
[4] The FSEL bit in the PSCALE register must be set.

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).

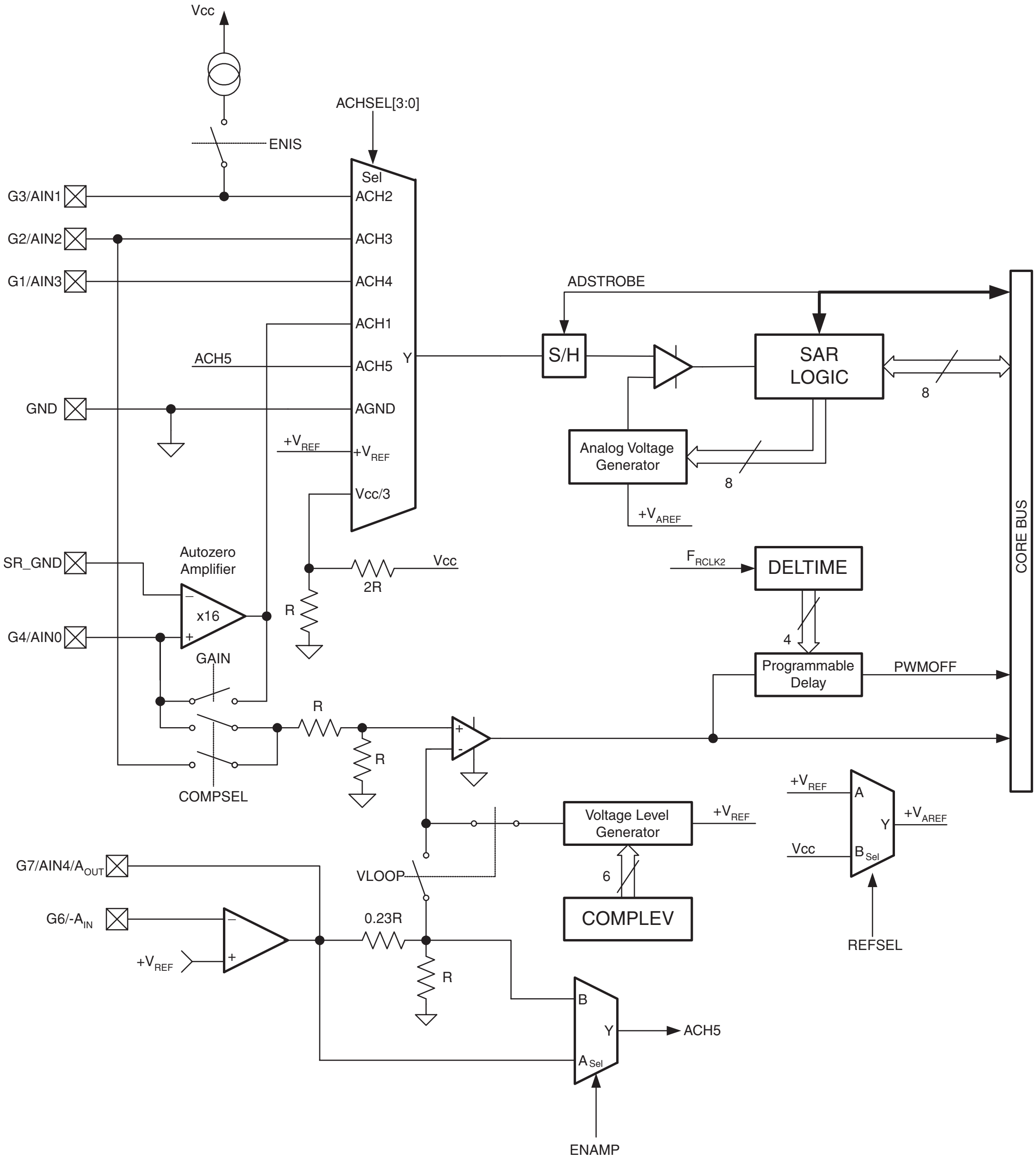
The ADC Circuit's eight analog inputs are software selectable where their analog input voltage is converted with respect to the internal ADC reference voltage (V_{AREF}). V_{AREF} may be programmed to use the internal bandgap reference voltage (V_{REF}) or Vcc as its source. By default, the ADC circuit's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

Figure 6. ADC Block Diagram[4]



4.1.1 ADCNTRL1 Register

The ADCNTRL1 is an 8-bit memory map register used to configure and control the ADC circuits. Software has both read and write access to all bits of the register.

Bit 7 of the ADCNTRL1 register is the ADC pending (APND) flag and is triggered after the 8-bit converted digital value is latched to the ADATA register towards the end of the ADC conversion cycle. The APND bit may be used by software to monitor when to access ADATA or to issue microcontroller hardware interrupts (if enabled). In order for software to monitor APND, it must be cleared before the next converted value is latched in ADATA where the APND flag is set to 1.

Bit 6 of the ADCNTRL1 register is the ADC's microcontroller hardware interrupt enabled (AINTEN) bit. If set, hardware interrupts (ADCI) are enabled and triggered by the APND pending flag.[5] As long as the ADC pending flag is set, the hardware interrupt will continue to execute software's ADC interrupt service routine until the pending flag is cleared.[6]

Bit 5 of the ADCNTRL1 register is the ADC Conversion Start/Busy (ASTART) bit. Software must set the ASTART bit to initiate an ADC conversion when the ENDAS bit of the ADCNTRL2 register is set to 0. The ASTART bit will remain high as long as an ADC conversion is in progress (whether software or the ADSTROBE signal triggered the conversion). If software attempts to clear the ASTART bit while a conversion is in progress, the write command is ignored and the ASTART bit remains high until the conversion cycle completes. Software should monitor ASTART to determine when the conversion has completed instead of the APND bit. 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.

Bit 4 of the ADCNTRL1 register is the ADC Voltage Reference Selection (REFSEL) bit. If REFSEL=0, the ADC Reference Voltage (V_{AREF}) becomes sourced by the internal bandgap voltage reference (V_{REF}). If REFSEL=1, the ADC Reference Voltage (V_{AREF}) becomes sourced by Vcc. If the ADC circuit is performing a conversion, software must avoid writing to the REFSEL bit.

Bits 3-0 of the ADCNTRL1 register are the Analog Channel Selection (ACHSEL[3:0]) bits selecting one of the eight analog input channels to convert its voltage (see Table 6). Software may write to the ACHSEL bits at any time; however, the actual ACHSEL selection signals will not change while an ADC conversion is in progress. If a read command is issued while a conversion is in progress, the current value of the ACHSEL bits may not necessarily reflect the actual state of the ACHSEL selection signals. The last value of the ACHSEL bits written by software at the time of the ADC conversion trigger, dictates the state of the ACHSEL selection signals for the triggered ADC conversion cycle.

The SBIT or RBIT instructions may be used to either set or clear one of the ADCNTRL1 register bits, like the AINTEN bit. The SBIT and RBIT instructions both take two instruction clock cycles to complete their execution. In the first cycle, all register bits are automatically read to obtain their most current value. In the second cycle, the bit to be set/cleared is given its new value and all bits are then re-written to the register. Using the SBIT/RBIT instruction to set/clear an enable bit with a pending flag in the same register may cause a potential hazard. Software may inadvertently clear a recently triggered pending flag if the trigger happened during the second phase of the SBIT/RBIT instruction execution. To avoid this condition, the LD instruction must be used to set or clear the interrupt enable bit. The ADC circuit is designed such that software may not trigger a pending flag by writing a 1 to the APND bit, it may only be cleared. The action of writing a 1 to the APND register bit holds its current bit value. The action of writing a 0 to the APND register bit clears the bit value. Therefore, the "LD T1CNTRL, #0E0H" instruction will set the ASTART and AINTEN bits without clearing APND.

Table 5. ADCNTRL1 Register Bit Definitions

ADCNTRL1 Register (addr. 0x9F)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
APND	AINTEN	ASTART	REFSEL	ACHSEL[3:0]			

Bit	Description
APND	(0) ADC's pending flag is cleared.
	(1) ADC's pending flag is triggered.
AINTEN	(0) Disables ADC hardware interrupts.
	(1) Enables ADC hardware interrupts.
ASTART	(0) ADC conversion is not in progress.
	(1) Start an ADC conversion / ADC conversion in progress.
REFSEL	(0) ADC Reference (V_{AREF}) = Internal V_{REF}
	(1) ADC Reference (V_{AREF}) = Vcc
ACHSEL[3:0]	Analog Input Channel Selection Bits. Refer to Table 6 for details.

Table 6. Analog Input Channel Selection (ACHSEL[3:0]) Bit Definitions

ACHSEL[3]	ACHSEL[2]	ACHSEL[1]	ACHSEL[0]	Analog Channel	I/O Equiv.
0	0	0	0	ACH1	G4/AIN0
0	0	0	1	ACH2	G3/AIN1
0	0	1	0	ACH3	G2/AIN2
0	0	1	1	ACH4	G1/AIN3
1	0	0	0	ACH5	G7/AIN4/A_{OUT}
1	0	0	1	GND	-
1	0	1	0	+V_{REF}	-
1	1	0	0	Vcc/3	-

4.1.2 ADCNTRL2 Register

The ADCNTRL2 is an 8-bit memory map register used to configure the analog circuits. Six of the eight register bits are used to configure circuits directly related to the ADC circuit while the others are not related.

Bit 7 (REFBY2) of the ADCNTRL2 register is the reference clock (F_{RCLK1}) divide-by-2 enable bit. The REFBY2 bit configures the reference clock of the PLL and Programmable Comparator circuit to be sourced either by F_{RCLK1} or F_{RCLK1}/2 clock. Refer to the Clock Circuit section of the datasheet for additional details.

Bit 6 (COMPSEL) of the ADCNTRL2 register is the Programmable Comparator's non-inverting input selection bit. 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.[9] Refer to the Programmable Comparator Circuit section of the datasheet for addition details.

Bit 5 of the ADCNTRL2 register is the Uncommitted Amplifier Enable (ENAMP) bit. If ENAMP=0, the Uncommitted Amplifier circuit is disabled and its pin connections (G6/-A_{IN} and G7/A_{OUT}) may be used as normal I/O ports. The G7/AIN4 pin may still be used as a standard ADC conversion input through the analog ACH5 channel. If ENAMP=1, the Uncommitted Amplifier circuit is enabled and its pin connections must be configured as tri-state inputs where G6/-A_{IN} is the inverting input and G7/ A_{OUT} is the amplifier output.[9] If the ADC circuit is performing a conversion on the analog ACH5 input when driven by the Uncommitted Amplifier, software must avoid clearing the ENAMP bit. Refer to the following Uncommitted Amplifier section for additional details.

Bit 4 (ENDAS) of the ADCNTRL2 register enables the ADC conversion's gated auto-sampling operating mode. If ENDAS=1, the ADC circuit configures the F_{ADCLK} clock for synchronization with the PWM Timer 1's ADSTROBE output signal. The ADC circuit will then accept triggers by the active (on) edge transition of the ADSTROBE signal. All other ADC configuration options must be prepared prior to setting the ENDAS bit. Refer to the following ADC Gated Auto-sampling Mode section for additional details. The ADSTROBE signal is generated by the PWM Timer 1 circuit and is configured using its T1CMPB and T1RA registers. Refer to the PWM Timer 1 Circuit section of the datasheet for details regarding its operation. If ENDAS=0, the ADC circuit is configured to accept only ADC start commands issued by software when setting the ASTART bit of the ADCNTRL1 register to 1. Refer to the following ADC Conversion Modes section for additional details.

Bits 3 and 2 (ASPEED[1:0]) of the ADCNTRL2 register selects the divide factor (1, 2, 4, or 8) to slow the F_{ADCLK} clock extending the ADC conversion cycle time. In most cases, the F_{ADCLK} clock division is performed to improve the ADC conversion accuracy. Refer to the following ADC Conversion Clock Configuration section for addition details.

Bit 1 of the ADCNTRL2 register is the Current Source Generator Enable (ENIS) bit. If ENIS=0, the Current Source Generator circuit is disabled and its G3/AIN1 pin may be used as a normal I/O port or as a standard ADC conversion input through the analog ACH2 channel. If ENIS=1, the Current Source Generator circuit is enabled and its pin connection must be configured as a tri-state input bypassing the I/O circuitry.[9] If the ADC circuit is performing a conversion on the analog ACH2 input when driven by the Current Source Generator, software must avoid clearing the ENIS bit. Refer to the following Current Source Generator section for additional details.

Bit 0 (GAIN) of the ADCNTRL2 register is the autozero amplifier enable bit. If GAIN=0, the autozero amplifier with its gain 16 circuitry is disabled where its G4/AIN0 pin connections may be used as a normal I/O port. The G4/AIN0 pin may still be used as a standard ADC conversion input through the analog ACH1 channel. If GAIN =1, the autozero amplifier with its gain 16 circuitry is enabled and its G4/AIN0 pin connection must be configured as a tri-state input where G4/AIN0 is the non-inverting and SR_GND is the inverting input of the amplifier.[9] Software may write to the GAIN bit at any time; however, the actual GAIN enable signal will not change while an ADC conversion is in progress. If a read command is issued while a conversion is in progress, the current value of the GAIN bit may not necessarily reflect the actual state of the GAIN enable signal. The last value of the GAIN bit written by software at the time of the ADC conversion trigger, dictates the state of the GAIN enable signal for the triggered ADC conversion cycle. Refer to the following Autozero Amplifier section for additional details.

Table 7. ADCNTRL2 Register Bit Definitions

ADCNTRL2 Register (addr. 0xA0)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REFBY2	COMPSEL	ENAMP	ENDAS	ASPEED[1:0]		ENIS	GAIN

Bit	Description
REFBY2	Used to divide the reference clock for the PLL and digital filter of the Programmable Comparator circuit. Refer to the Clock Circuit section of the datasheet for details. (0) F_{RCLK2}=F_{RCLK1} (1) F_{RCLK2}=F_{RCLK1}/2
COMPSEL	(0) G4 is connected to the Programmable Comparator's non-inverting input. (1) G2 is connected to the Programmable Comparator's non-inverting input.
ENAMP	(0) Disables the Uncommitted Amplifier (G6 and G7 are normal I/Os). (1) Enables the Uncommitted Amplifier where G6/-A_{IN} is the inverting input and G7/A_{OUT} is the amplifier output. The amplifier output (A_{OUT}) is also the ACH5 input to the ADC's analog mux.
ENDAS	(0) Enables the Standard ADC Conversion Mode where software must trigger an ADC conversion by setting the ASTART bit of the ADCNTRL1 register. (1) Enables the ADC Gated Auto-sampling Mode where PWM Timer 1's ADSTROBE output to automatically triggers an ADC conversion.
ASPEED[1:0]	(0) ADC conversion clock speed = F_{ADCLK} (1) ADC conversion clock speed = F_{ADCLK}/2 (2) ADC conversion clock speed = F_{ADCLK}/4 (3) ADC conversion clock speed = F_{ADCLK}/8
ENIS	(0) Disable Current Source Generator (G3 is a normal I/O). (1) Enable Current Source Generator where G3/AIN1 sources the I_{SRC}.
GAIN	(0) Disables the Autozero Amplifier (G4 is a normal I/O). (1) Enables the Autozero Amplifier (with a gain of 16) where G4/AIN0 is its non-inverting input and SR_GND is it inverting input.

4.2 ADC Conversion Modes

The ADC circuit may be configured to convert analog voltages with a conversion cycle time determined by the ADC clock (F_{ADCLK}) and the ASPEED bits of the ADCNTRL2 register. Refer to the following ADC Conversion Clock Configuration section for details. By default, the ADC circuit performs a conversion with every trigger initiated by software setting the ASTART bit of the ADCNTRL1 register to 1. The ADC circuit may also be configured to perform a conversion automatically (using the gated auto-sampling mode) with every active (on) edge of the PWM Timer 1 ADSTROBE output signal. Refer to the following ADC Gated Auto-sampling Mode section for details.

Before any ADC conversion triggers are issued (by software or automatically) software must configure the voltage reference (V_{AREF}) and analog input channel appropriately. This is done by programming the REFSEL and ACHSEL bits of the ADCNTRL1 register. If using the internal Autozero Amplifier, Uncommitted Amplifier, and Current Source Generator circuits in the application, software must also configure and enable the desired circuits. Lastly, the F_{ADCLK} must be configured to improve the ADC conversion accuracy.

When performing an ADC conversion where software triggers the conversion, the ASTART bit of the ADCNTRL1 register remains high (1) symbolizing that a conversion is in progress. The ADC conversion is divided in two phases lasting a total of 13 conversion clock cycles. However, an autozero ADC conversion lasts a total of 20 conversion clock cycles. Refer to the following Autozero Amplifier section for details. In the first phase of a standard conversion, occupying the first four conversion cycles, the ADC circuit performs a sample and hold operation to measure fast changing analog signals before converting the input voltage. The second phase, occupying the last nine cycles, converts the analog input voltage to an 8-bit digital value and stores it in the ADATA register for easy access by software. Once the converted value is stored in ADATA, the APND bit is triggered and ASTART bit is cleared (symbolizing completion of the conversion cycle). 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.

4.2.1 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 accuracy. However, if the internal oscillator is trimmed to its upper F_{OSC} frequency while FMODE and FSEL are zero, 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.

4.3 Autozero Amplifier

The GAIN bit of the ADCNTRL2 register enables the Autozero Amplifier circuit. To perform a proper ADC conversion using the autozero amplifier, software must configure its non-inverting input (G4/AIN0) as a tri-state input bypassing the I/O circuitry. The autozero amplifier has a gain of 16, but is not defined as a true differential amplifier because the inverting SR_GND input must be connected as close to ground as possible (e.g. to act as a Kelvin connection) to reduce noise and improve the precision of the measurement.

To perform an ADC conversion through the autozero amplifier, the ACH1 input channel of the analog mux must be selected. The autozero ADC conversion is divided in three phases lasting a total of 20 conversion clock cycles. In the first phase, occupying the first six conversion cycles, it calculates the offset voltage of the amplifier. The second phase, occupying the next five cycles, adds or subtracts the offset voltage to the amplified input voltage which now has a gain of 16. The final phase, occupying the last nine cycles, converts the autozero input voltage to an 8-bit digital value and stores it in the ADATA register for easy access by software.

4.4 Uncommitted Amplifier

The Uncommitted Amplifier Enable (ENAMP) bit of the ADCNTRL2 register enables the Uncommitted Amplifier (AMP) circuit whose inverting input is connected to the G6/-A_{IN} and output to the G7/A_{OUT} port pins. Before enabling the AMP circuit, software must configure both the G6/-A_{IN} and G7/A_{OUT} pins as tri-state input ports bypassing all I/O circuitry. The AMP circuit may be used in any control or battery management applications.

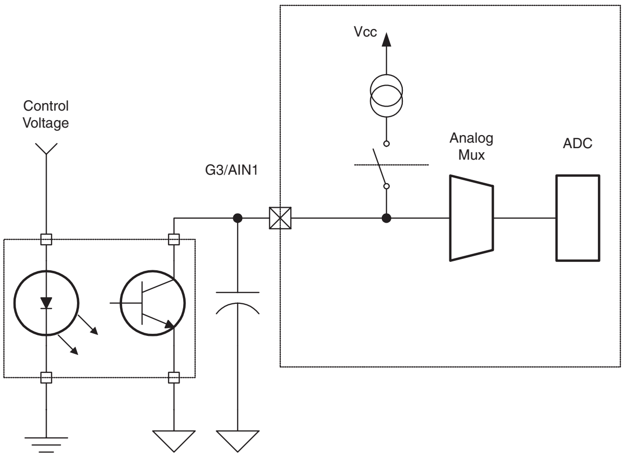
In control applications, the AMP circuit is used as the error amplifier in a hardware closed loop whose input connections are part of the external compensation loop circuit. The output of the amplifier (A_{OUT}) is internally fed to the Programmable Comparator circuit to control the PWM T1HS1 and T1HS2 inputs of the control plant block. An ADC conversion may be triggered to monitor A_{OUT} by selecting the ACH5 input channel of the analog mux.

The AMP circuit may be configured as a general uncommitted amplifier whose non-inverting input is connected to V_{REF}. Therefore, the AMP circuit may only amplify differences with respect to V_{REF}. An ADC conversion may be triggered to convert the voltage at A_{OUT} by selecting the ACH5 input channel of the analog mux. In battery management applications, the AMP circuit may be used to improve the resolution of the battery voltage measurement by adding a gain through the feedback loop. Voltage variation at a typical point will be amplified with a gain for better resolution, for example, to sense the Negative Delta V (NDV) to determine the end of change for a NiCD or NiMH battery.

4.5 Current Source Generator

The Current Source Enable (ENIS) bit of the ADCNTRL2 register enables the Current Source Generator (ISOURCE) circuit connected to the G3/AIN1 pin. Before enabling the ISOURCE circuit, software must configure the G3/AIN1 port as a tri-state input bypassing all I/O circuitry.[9] Once the ENIS bit is set, the ISOURCE circuit begins to generate I_{SRC} of current typically used to interface to an opto-coupler output.[1] Figure 7 provides an example of a typical ISOURCE application where the voltage developed at the G3/AIN1 input can be converted by the ADC circuit if the ACH2 analog input channel is selected. The ISOURCE and ADC circuit combination may also be used to measure Capacitive Sensors or the resistance of a thermistor (NTC/PTC) to indirectly measure the temperature.

Figure 7. Current Generator Interface



1. Refer to the Electrical Characteristics section of the datasheet for details.
2. Refer to the PWM Timer 1 Circuit section of the datasheet for details regarding the ADSTROBE signal configuration.
3. Refer to Table 30 of the Device Memory section of the datasheet for the detailed memory map.
4. On the FMS7401L 8-pin device, the SR_GND is internally bonded to the GND pin.
5. Hardware interrupts are not executed by the microcontroller core unless the Global Interrupt enable (G) flag of the Status register is set. Refer to the 8-Bit Microcontroller Core section of the datasheet for details.
6. The ADC hardware interrupt will be executed in the defined priority order. Refer to the 8-Bit Microcontroller Core section of the datasheet for details.
7. Assuming the internal oscillator frequency is F_{OSC}=2MHz as specified in the Electrical Characteristics section of the datasheet.
8. The upper F_{OSC} frequency (4MHz) is not a standard feature offered on the FMS7401L devices but is available upon request.
9. Refer to the I/O Ports section of the datasheet for details.

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]