Chapter 5

THE ANALOG-TO-DIGITAL CONVERTER (ADC)

5.1 ADC Overview

The Analog-to-Digital Converter (ADC) on the LF2407 allows the DSP to sample analog or "real-world" voltage signals. The output of the ADC is an integer number which represents the voltage level sampled. The integer number may be used for calculations in an algorithm. The resolution of the ADC is 10 bits, meaning that the ADC will generate a 10-bit number for every conversion it performs. However, the ADC stores the conversion results in registers that are 16 bits wide. The 10 most significant bits are the ADC result, while the least significant bits (LSBs) are filled with "0"s. We usually want to truncate the useless zeros, so the value in the result register is simply right shifted by six places.

If the ADC performs a conversion on a 3.3V signal, it will theoretically generate "1111111111000000b" (or "FFC0h") in the appropriate result register and "0h" if a 0V signal is sampled. In actuality, the least significant of the 10 bits will vary slightly; this is the result of random noise picked up by the ADC.

There are a total of 16 input channels to the single input ADC. The control logic of the ADC consists of auto-sequencers, which control the sampling of the 16 input channels to the ADC. The auto-sequencers not only control which channels (input channels) will be sampled by the ADC, but also the order of the channels that the ADC performs conversions on. The two 8-conversion auto-sequencers can operate independently or cascade together as a "virtual" 16-conversion ADC.

5.1.1 Summary of the LF2407 ADC

- 10-bit ADC with built-in Sample and Hold (S/H)
- Fast conversion time of 500 ns
- Sixteen (16) multiplexed analog inputs (ADCIN0 – ADCIN15)
- Auto-sequencing capability – up to 16 "auto-conversions" in a single session. Each conversion session can be programmed to select any one of the 16 input channels
- Two independent 8-state sequencers (SEQ1 and SEQ2) that can be operated individually in dual-sequencer mode or cascaded into one large 16-state sequencer (SEQ) in cascaded mode
- Four Sequencing Control Registers (CHSELSEQ1..4) that determine the sequence of analog channels that are taken up for conversion in a given sequencing mode
- Sixteen (individually addressable) result registers to store the converted values (RESULT0 – RESULT15)
- Multiple trigger sources for start-of-conversion (SOC)
  a. Software: Software start (using SOC SEQn bit)
  b. EVA: Event manager A (multiple event sources within EVA)

c. EVB: Event manager B (multiple event sources within EVB)
d. External: ADCSOC pin

- Interrupt control allows interrupt generation on every end-of-sequence (EOS) or every other EOS
- Sequencer can operate in start/stop mode, allowing multiple time-sequenced triggers to synchronize conversions
- EVA and EVB can independently trigger SEQ1 and SEQ2, respectively (this is applicable for dual-sequencer mode only)
- Sample-and-hold acquisition time window has separate prescale control
- Built-in calibration mode and built-in self-test mode

5.2 Operation of the ADC

Using the ADC on the LF2407 is relatively simple. The user first needs to configure the ADC for the desired operation. Like all peripherals, all registers relating to ADC operation have addresses in data memory space. The first step in configuring the ADC should be to reset the ADC. After the ADC is reset, the next step is to configure the main ADC control registers (ADCTRL1, ADCTRL2) for desired ADC operation. Then, load the MAXCONV register with the desired number of automatic conversions minus 1. For example, if seven auto-conversions are desired, MAXCONV would be loaded with "6". The desired input channels and their order of conversion need to be specified in the CHSELSEQn registers. Finally, a SOC trigger will start the sampling process. A short example of the assembly code performing the above listed steps is provided in Example 5.1.

Example 5.1- The following code gives an example of initializing the ADC, setting up the CHSELSEQn registers and starting the conversion sequence:

```
LDP     #0E1h
SPLK    #0100000000000000b,ADCTRL1
NOP
SPLK    #0011000000010000b,ADCTRL1
        ; the following explains bits in ADCCTRL1:
        ; 15 - RSVD | 14 - Reset(1) | 13,12 - Soft & Free
        ; 11,10,9,8 - Acq. prescalers | 7 - Clock prescaler
        ; 6 - Cont. run (1) | 5 - Int. priority (Hi.0)
        ; 4 - Seq. casc (0-dual)
SPLK    #15, MAXCONV            ;Setup for 16 conversions
SPLK    #03210h, CHSELSEQ1      ;Conv Ch 0,1,2,3
SPLK    #07654h, CHSELSEQ2      ;Conv Ch 4,5,6,7
SPLK    #0BA98h, CHSELSEQ3      ;Conv Ch 8,9,10,11
SPLK    #0FEDCh, CHSELSEQ4      ;Conv Ch 12,13,14,15
SPLK    #2000b, ADCTRL2         ;Start the conversions by bit 13
```

After the conversion process is complete, each 10-bit result can be read from the result registers RESULTn. The conversion results are stored sequentially in result registers RESULT0 to RESULT15. The first result is stored in RESULT0, the second result in RESULT1, and so on. For example, if ADC channel 1 is selected for four consecutive conversions, the results will appear in registers RESULT0 through RESULT3. *There is no correlation between ADC Channel 1 and the RESULT register 1 or Channel 2 and RESULT 2 etc.*

We will discuss each of these steps in detail in the following sections, starting with the different operating modes of the ADC. This will aid the reader in configuring the ADC control registers by helping to determine what operating mode is needed. After the reader is familiar with the ADC operating modes, we will cover the MAXCONV and CHSELSEQn registers. Various SOC trigger methods will then be discussed. Finally, the ADC conversion result register reading will be discussed.

5.2.1 Sequencer Configurations of the ADC

The first operating parameter the user needs to select is to configure the ADC to operate as either one 16-conversion sequencer or two 8-conversion sequencers. The ADC sequencer consists of two independent 8-conversion sequencers (SEQ1 and SEQ2) that can be cascaded together to form one 16-conversion sequencer (SEQ).

When the ADC is configured to operate as one cascaded 16-convesion sequencer, it may perform up to 16 conversions on any combination of the 16 input channels. For example, it could be programmed to perform 14 conversions on channel 1, or in another instance, 10 total conversions on a combination of channels depending on what the CHSELSEQn registers are set for. The diagram Fig. 5.1 shows the configuration of the cascaded 16-conversion sequencer. When in cascaded mode there is only one sequencer (SEQ) and the MAXCONV register is programmed for the maximum number of conversions. The results are stored in RESULT0 through RESULT 15 depending on the number of conversions performed.

If the ADC is configured as two 8-conversion sequencers, then each sequencer operates independently. When the two sequencers are used independently, the current active sequencer has priority over the inactive one. The start of conversion request from the “inactive” sequencer will be taken as soon as the sequence initiated by the “active” sequencer is completed. For example, if Sequencer 1 (SEQ1) is currently performing a conversion and Sequencer 2 (SEQ2) requests a start of conversion, the ADC will finish the conversion from SEQ1, and then start the SEQ2 conversion. See Fig. 5.2 for a diagram of the dual-sequencer configuration. In dual-sequencer operation, the MAXCONV register is “split” up so that the same register contains data for the maximum number of conversions for both SEQ1 and SEQ2. The 16 result registers are also split up so SEQ1 uses RESULT0 through RESULT7 and SEQ2 uses RESULT8 through RESULT15. A summary of the cascaded and dual-sequencer configurations is listed in Table 5.1.

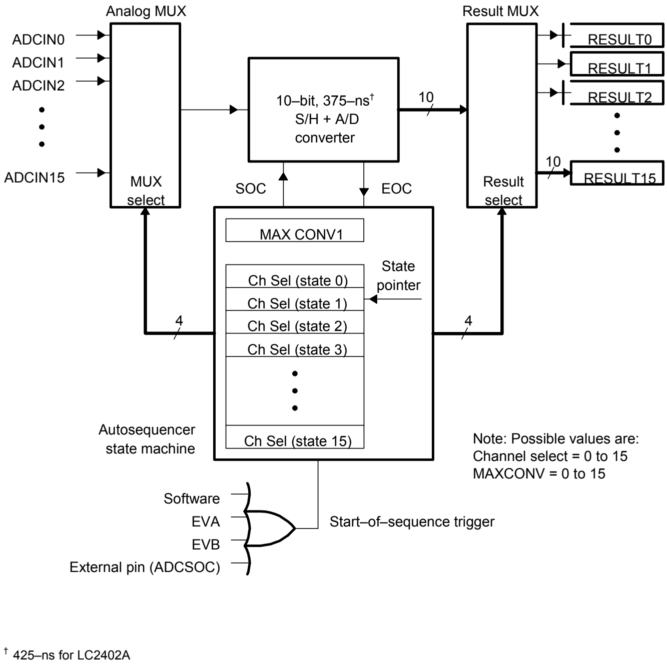


Figure 5.1 Block diagram of ADC in cascaded sequencer mode. (Courtesy of Texas Instruments)

Table 5.1 Comparasion table of dual (SEQ1 and SEQ2) versus cascaded sequencer configuration

Feature	Single 8–state sequencer #1 (SEQ1)	Single 8–state sequencer #2 (SEQ2)	Cascaded 16–state sequencer (SEQ)
Start–of–conversion triggers	EVA, software, external pin	EVB, software	EVA, EVB, software, external pin
Maximum number of autoconversions (i.e., sequence length)	8	8	16
Autostop at end–of–sequence (EOS)	Yes	Yes	Yes
Arbitration priority	High	Low	Not applicable
ADC conversion result register locations	0 to 7	8 to 15	0 to 15
CHSELSEQn bit field assignment	CONV00 to CONV07	CONV08 to CONV15	CONV00 to CONV15

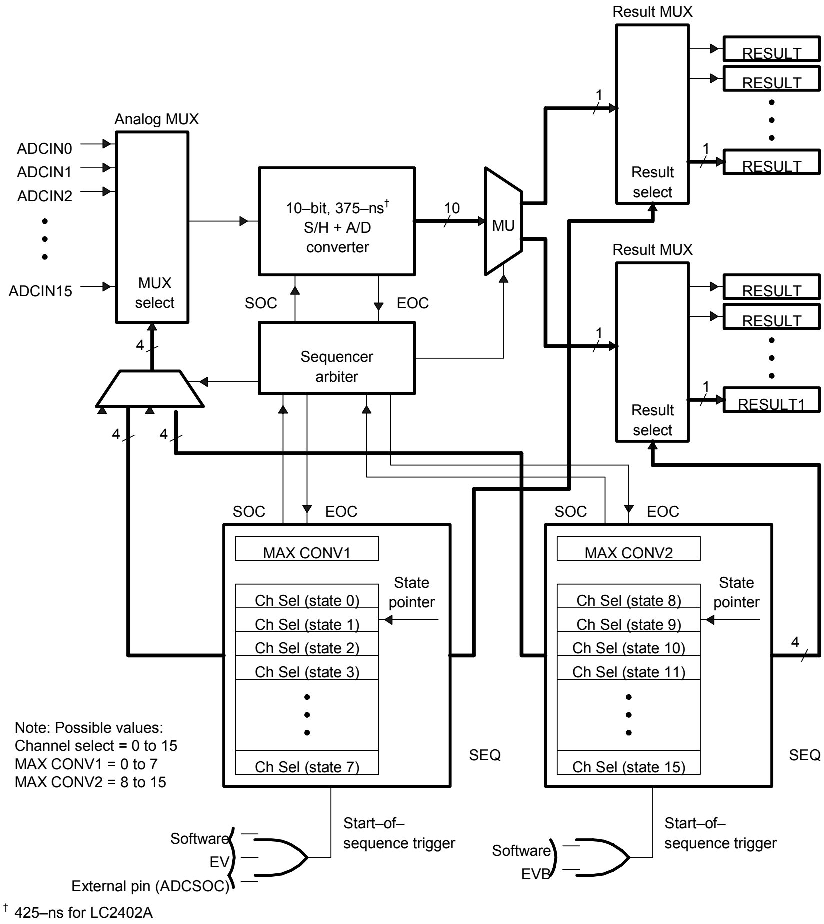


Figure 5.2 Block diagram of ADC in dual sequencer mode. (Courtesy of Texas Instruments)

5.2.2 Sequencer Operating Modes

Once the sequencer configuration has been chosen, it is necessary to determine in what mode each sequencer will operate. The sequencer operation mode depends on the continuous-run mode bit (CONT RUN) in ADCCTRL1. The ADC's interrupt flag is always set when the ADC completes the number of conversions specified by (MAXCONV + 1) regardless of the CONT RUN bit. The two ADC operation modes which apply to both dual (SEQ1, SEQ2) and cascaded (SEQ) sequencer modes are:

- Start/Stop Auto-Sequencer Mode
- Continuous Auto-Sequencer Mode

Start/Stop Auto-Sequencer Mode

If the CONT RUN bit is not set, upon receiving a trigger, the ADC performs all conversions and halts at the last conversion state (CONVxx) in the corresponding CHSELSEQn. To perform another batch of conversions, the ADC is normally reset to its initial state via the RST SEQn bit in the ADCTRL2 register and reinitialized. After being reinitialized, another trigger is given and the whole process starts over again. Figure 5.3 is a flowchart of the operation of the ADC under start/stop mode.

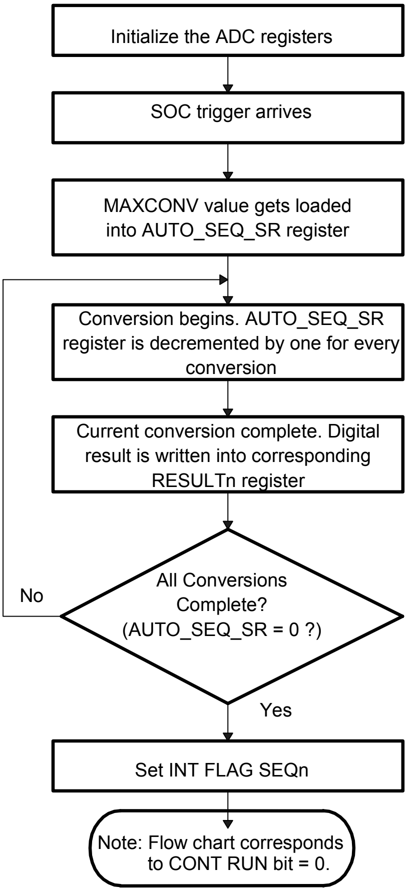

Initialize the ADC registers
SOC trigger arrives
MAXCONV value gets loaded into AUTO_SEQ_SR register
Conversion begins. AUTO_SEQ_SR register is decremented by one for every conversion
Current conversion complete. Digital result is written into corresponding RESULTn register
All Conversions Complete? (AUTO_SEQ_SR = 0 ?)
No
Yes
Set INT FLAG SEQn
Note: Flow chart corresponds to CONT RUN bit = 0.

Figure 5.3 Flowchart for Start/Stop Auto-Sequencer Mode (CONT RUN=0).

In the case when another trigger signal is given and the ADC has not been reset, the ADC performs another specified number of conversions (MAXCONV + 1) from the current conversion state and then halts. Another trigger signal will simply restart the sequencer from the point where it halted. When the ADC is given multiple triggers without being reset in between, this operation is referred to as multiple time-sequenced trigger operation. Example 5.2 illustrates a situation where a multiple time-sequenced trigger operation might be used.

Example 5.2 – The following is a situation where a multiple time-sequenced trigger operation might be used.

An application requires conversions on all 16 channels, but not all at once. The application requires conversions on channels 0 through 3, perform a few calculations, convert channels 4 through 7, do a few more calculations, convert channels 8 through 11, etc. until conversions are performed on all 16 input channels. The four CHSELSEQ registers would be loaded only once with all 16 channels in the desired order. The MAXCONV register would be loaded with the number "3", which configures the auto-sequencer for four conversions. Each time the sequencer pauses, the algorithm would branch to the section of code that performs calculations and retrigger the ADC. This "branching" could either occur as a result of an interrupt or bit polling algorithm.

Continuous Auto-Sequencer Mode

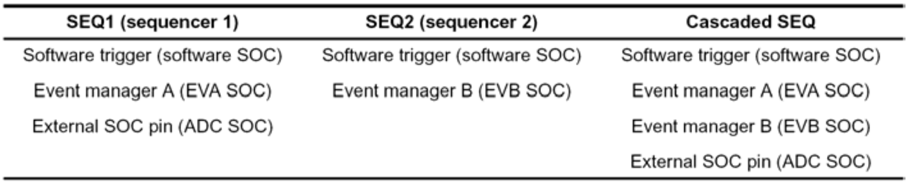
The continuous-run mode bit is set to "1" for this mode of operation. When in this mode, the ADC completes the number of conversions specified, resets itself to the first conversion state (CONV00), and then performs the whole operation over again. This operation is similar to the start/stop mode except that the ADC is put in a continuous "looping" operation.

Note: *If the CONT RUN (continuous run) mode is selected, the user must be sure that the result registers are read before the next conversion sequence begins. This is because every time the ADC runs, the result registers will be overwritten with the most current results.*

5.2.3 Triggering Sources for the LF2407 ADC

In order to start the conversion sequence on the ADC, the sequencer must be triggered. There are several different trigger sources on the LF2407. Triggers may come from a SOC signal from EVA: external pin or software. A software trigger is the trigger thus far used as an example. The software trigger is generated by setting the SOC SEQ1 bit (cascaded mode) or SOC SEQ1,2 bits (dual mode) in the ADCTRL2 register. Other than software triggers, hardware in the form of an external pin or on-chip peripheral can also trigger the ADC. Table 5.2 lists the possible triggering sources which generate a SOC for the ADC. Each trigger input can be enabled /disabled.

Table 5.2 SOC Trigger Sources for the ADC



SEQ1 (sequencer 1)	SEQ2 (sequencer 2)	Cascaded SEQ
Software trigger (software SOC)	Software trigger (software SOC)	Software trigger (software SOC)
Event manager A (EVA SOC)	Event manager B (EVB SOC)	Event manager A (EVA SOC)
External SOC pin (ADC SOC)		Event manager B (EVB SOC)
		External SOC pin (ADC SOC)

The following conditions apply to trigger operation:

a. A SOC trigger can initiate an auto-conversion sequence whenever a sequencer is in an idle state. An idle state is either just after reset (CONV00), or any state where the sequencer has just finished a conversion sequence, i.e., when SEQ CNTR has reached zero.

b. If a SOC trigger occurs while a current conversion sequence is underway, it sets the SOC SEQn bit. If yet another SOC trigger occurs, that trigger is ignored. This basically operates as a SOC trigger “buffer” that will catch a trigger even though the ADC might be currently performing a conversion.

c. Once triggered, the sequencer cannot be stopped/halted in mid sequence. The program must either wait until an End-of-Sequence (EOS) or initiate a sequencer reset, which brings the sequencer immediately back to the idle start state (CONV00 for SEQ1 and cascaded cases; CONV08 for SEQ2).

d. When SEQ1 and SEQ2 are used in **cascaded mode, triggers going to SEQ2 are ignored**, while **SEQ1 triggers are active**. Cascaded mode can be viewed as SEQ1 with 16 conversion states instead of 8.

5.2.4 The ADCTRL1 and ADCTRL2 Control Registers

ADC Control Register 1 (ADCTRL1) — Address 70A0h

15	14	13	12	11	10	9	8
Reserved	RESET	SOFT	FREE	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0
	RS–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
CPS	CONT RUN	INT PRI	SEQ CASC	CAL ENA	BRG ENA	HI/LO	STEST ENA
RW–0	RW–0	RW–0	RW–0	RW-0	RW-0	RW-0	RW-0

Note: *R = read access, W = write access, S = set only, -0 = value after reset.*

Bit 15 Reserved

Bit 14 **RESET**. ADC module software reset. This bit causes a master reset on the entire ADC module. All register bits and sequencer state machines are reset to the initial state as occurs when the device reset pin is pulled low (or after a power-on reset).

0 No effect

1 Resets entire ADC module (bit is then set back to 0 by ADC logic)

Note: *Using the RESET Bit in the ADCTRL1 Register*

The ADC module is reset during a system reset. If an ADC module reset is desired at any other time, you can do so by writing a 1 to this bit. After a NOP, you can then write the appropriate values to the ADCTRL1 register bits:

```
SPLK #01xxxxxxxxxxxxxxb,ADCTRL1 ;Resets the ADC (RESET = 1)
NOP                             ;Provides the required delay between
                                ; two writes to the ADCTRL1
SPLK #00xxxxxxxxxxxxxxb,ADCTRL1 ;Takes ADC out of Reset(RESET= 0)
```

Note: *The second SPLK is not required if the default/power-on configuration of the ADC is sufficient.*

Bits 13, 12 **SOFT and FREE**. Soft and Free bits. These bits determine what happens with the ADC when an emulation-suspend occurs (due to the debugger hitting a breakpoint, for example). In free-run mode, the peripheral can continue with whatever it is doing. In stop mode, the peripheral can either stop immediately or stop when the current operation (i.e., the current conversion) is complete.

Soft	Free	
0	0	Immediate stop on suspend
1	0	Complete current conversion before stopping
X	1	Free run, continue operation regardless of suspend

Bits 11–8 **ACQ PS3 – ACQ PS0**. Acquisition time window – pre-scale bits 3–0 These bits define the ADC clock pre-scale factor applied to the acquisition portion of the conversion and determine over what time period each ADc sample will take place. The pre-scale values are defined in the following table.

#	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0	PRE-SCALER (div. by)	Acquisition Time Window	Source Z (CPS=0)	Source Z (CPS=1)
0	0	0	0	0	1	2 x Tclk	67	385
1	0	0	0	1	2	4 x Tclk	385	1020
2	0	0	1	0	3	6 x Tclk	702	1655
3	0	0	1	1	4	8 x Tclk	1020	2290
4	0	1	0	0	5	10 x Tclk	1337	2925
5	0	1	0	1	6	12 x Tclk	1655	3560
6	0	1	1	0	7	14 x Tclk	1972	4194
7	0	1	1	1	8	16 x Tclk	2290	4829
8	1	0	0	0	9	18 x Tclk	2607	5464
9	1	0	0	1	10	20 x Tclk	2925	6099
A	1	0	1	0	11	22 x Tclk	3242	6734
B	1	0	1	1	12	24 x Tclk	3560	7369
C	1	1	0	0	13	26 x Tclk	3877	8004
D	1	1	0	1	14	28 x Tclk	4194	8639
E	1	1	1	0	15	30 x Tclk	4512	9274
F	1	1	1	1	16	32 x Tclk	4829	9909

Notes:
1) Period of Tclk is dependent on the "Conversion Clock Prescale" bit (Bit 7); i.e.,
CPS = 0: Tclk = 1/CLK (example, for CLK = 30 MHz, Tclk = 33 ns)
CPS = 1: Tclk = 2(1/CLK) (example, for CLK = 30 MHz, Tclk = 66 ns)
2) Source impedance Z is a design estimate only.

Bit 7 **CPS**. Conversion clock prescale. This bit defines the ADC conversion logic clock prescale
0 ^{F}clk = CLK/1
1 ^{F}clk = CLK/2
CLK = CPU clock frequency

Bit 6 **CONT RUN**. Continuous run

This bit determines whether the sequencer operates in continuous conversion mode or start-stop mode. This bit can be written while a current conversion sequence is active. This bit will take effect at the end of the current conversion sequence, i.e., software can set/clear this bit until EOS has occurred for valid action to be taken. In the continuous conversion mode, there is no need to reset the sequencer; however, the sequencer must be reset in the start-stop mode to put the converter in state CONV00.

0 Start-stop mode. Sequencer stops after reaching EOS. This is used for multiple time-sequenced triggers.

1 Continuous conversion mode. After reaching EOS, the sequencer starts all over again from state CONV00 (for SEQ1 and cascaded) or CONV08 (for SEQ2).

Bit 5 **INT PRI**. ADC interrupt request priority
0 High priority
1 Low priority

Bit 4 **SEQ CASC**. Cascaded sequencer operation. This bit determines whether SEQ1 and SEQ2 operate as two 8-state sequencers or as a single 16-state sequencer (SEQ).
0 Dual-sequencer mode. SEQ1 and SEQ2 operate as two 8-state sequencers.
1 Cascaded mode. SEQ1 and SEQ2 operate as a single 16-state sequencer (SEQ).

Bit 3 **CAL ENA**. Offset calibration enable

When set to 1, CAL ENA disables the input channel multiplexer, and connects the calibration reference selected by the bits HI/LO and BRG ENA to the ADC core inputs. The calibration conversion can then be started by setting bit 14 of ADCTRL2 register (STRT CAL) to 1. Note that CAL ENA should be set to 1 first before the STRT CAL bit can be used.

Note: *This bit should not be set to 1 if STEST ENA = 1*
0 Calibration mode disabled
1 Calibration mode enabled

Bit 2 **BRG ENA**. Bridge enable

Together with the HI/LO bit, BRG ENA allows a reference voltage to be converted in calibration mode. See the description of the HI/LO bit for reference voltage selections during calibration.
0 Full reference voltage is applied to the ADC input
1 A reference midpoint voltage is applied to the ADC input

Bit 1 **HI/LO**. ^{V}REFHI$^{/V}$REFLO selection

When the fail self-test mode is enabled (STEST ENA = 1), HI/LO defines the test voltage to be connected. In calibration mode, HI/LO defines the reference source polarity; see Table 7–5. In normal operating mode, HI/LO has no effect.
0 ^{V}REFLO is used as precharge value at ADC input
1 ^{V}REFHI is used as precharge value at ADC input

Reference Bit Voltage Selection

BRG ENA	HI/LO	CAL ENA = 1 Reference voltage (V)	STEST ENA = 1 Reference voltage (V)
0	0	V_{REFLO}	V_{REFLO}
0	1	V_{REFHI}	V_{REFHI}
1	0	$\|(V_{REFHI} - V_{REFLO}) / 2\|$	V_{REFLO}
1	1	$\|(V_{REFLO} - V_{REFHI}) / 2\|$	V_{REFHI}

Bit 0 **STEST ENA**. Self-test function enable

0 Self-test mode disabled

1 Self-test mode enabled

ADC Control Register 2 (ADCTRL2) — Address 70A1h

15	14	13	12	11	10	9	8
EVB SOC SEQ	RST SEQ1/ STRT CAL	SOC SEQ1	SEQ1 BSY	INT ENA SEQ1 (Mode 1)	INT ENA SEQ1 (Mode 0)	INT FLAG SEQ1	EVA SOC SEQ1
RW–0	RS–0	RW–0	R–0	RW–0	RW–0	RC–0	RW–0
7	6	5	4	3	2	1	0
EXT SOC SEQ1	RST SEQ2	SOC SEQ2	SEQ2 BSY	INT ENA SEQ2 (Mode 1)	INT ENA SEQ2 (Mode 0)	INT FLAG SEQ2	EVB SOC SEQ2
RW–0	RS–0	RW–0	R–0	RW–0	RW–0	RC–0	RW–0

Note: *R = read access, W = write access, S = set only, C = clear, -0 = value after reset.*

Bit 15 **EVB SOC SEQ**. EVB SOC enable for cascaded sequencer *(Note: This bit is active only in cascaded mode.)*

0 No action

1 Setting this bit allows the cascaded sequencer to be started by an Event Manager B signal. The Event Manager can be programmed to start a conversion on various events. See Chapter 6 for details.

Bit 14 **RST SEQ1 / STRT CAL**. Reset Sequencer1/Start Calibration

Case: Calibration Disabled (Bit 3 of ADCTRL1) = 0

Writing a 1 to this bit will reset the sequencer immediately to an initial “pre-triggered” state, i.e., waiting for a trigger at CONV00. A currently active conversion sequence will be aborted.

0 No action

1 Immediately reset sequencer to state CONV00

Case: Calibration Enabled (Bit 3 of ADCTRL1) = 1

Writing a 1 to this bit will begin the converter calibration process.

0 No action

1 Immediately start calibration process

Bit 13 **SOC SEQ1**. SOC trigger for Sequencer 1 (SEQ1). This bit can be set by the following triggers:

S/W – Software writing a 1 to this bit
EVA – Event Manager A
EVB – Event Manager B (only in cascaded mode)
EXT – External pin (i.e., the ADCSOC pin)

When a trigger occurs, there are three possibilities:

Case 1: SEQ1 idle and SOC bit clear. SEQ1 starts immediately (under arbiter control). This bit is set and cleared, allowing for any "pending" trigger requests.
Case 2: SEQ1 busy and SOC bit clear. Bit is set signifying a trigger request is pending. When SEQ1 finally starts after completing current conversion, this bit will be cleared.
Case 3: SEQ1 busy and SOC bit set. Any trigger occurring in this case will be ignored (lost).

0 Clears a pending SOC trigger.

Note: *If the sequencer has already started, this bit will automatically be cleared, and, hence, writing a zero will have no effect; i.e., an already started sequencer cannot be stopped by clearing this bit.*

1 Software trigger – Start SEQ1 from currently stopped position (i.e., idle mode)

Note: *The RST SEQ1 (ADCTRL2.14) and the SOC SEQ1 (ADCTRL2.13) bits should not be set in the same instruction. This will reset the sequencer, but will not start the sequence. The correct sequence of operation is to set the RST SEQ1 bit first, and the SOC SEQ1 bit in the following instruction. This ensures that the sequencer is reset and a new sequence started. This sequence applies to the RST SEQ2 (ADCTRL2.6) and SOC SEQ2 (ADCTRL2.5) bits also.*

Bit 12 **SEQ1 BSY**. SEQ1 Busy
This bit is set to a 1 while the ADC auto-conversion sequence is in progress. It is cleared when the conversion sequence is complete.
0 Sequencer is idle (i.e., waiting for trigger)
1 Conversion sequence is in progress

Bits 11–10 INT ENA SEQ1. Interrupt-mode-enable control for SEQ1

Bit 11	Bit 10	Operation Description
0	0	Interrupt is disabled
0	1	Interrupt **Mode 1.** Interrupt requested immediately when INT FLAG SEQ1 flag is set
1	0	Interrupt **Mode 2** Interrupt requested only if INT FLAG SEQ1 flag is already set. If clear,† INT FLAG SEQ1 flag is set and INT request is suppressed. (This mode allows interrupt requests to be generated for every other EOS.)
1	1	Reserved

† This means that the last completed sequence is the first of the two sequences needed to assert an interrupt.

Bit 9 **INT FLAG SEQ1.** ADC interrupt flag bit for SEQ1

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event

1 An interrupt event has occurred

Checking for ADC Peripheral Interrupt Flag

After a SOC is initiated, we can check the INT FLAG SEQn bit to see if the results are in the result registers.

Example code:

```
ADC_LOOP1:
    LDP   #0E1h                         ;data page - ADCTRL2
    SPLK  #0100000000000000b,ADCTRL2    ;Reset for SEQ1
    NOP
    NOP
    NOP
    NOP
    SPLK  #0010000000000000b,ADCTRL2    ;SOC for SEQ1
CHK_INTFLAG:
    BIT   ADCTRL2, 6          ;Wait for INT Flag to set
    BCND  CHK_INTFLAG, NTC    ;If TC=0, keep looping.
```

Bit 8 **EVA SOC SEQ1.** Event Manager A SOC mask bit for SEQ1

0 SEQ1 cannot be started by EVA trigger.

1 Allows SEQ1/SEQ to be started by Event Manager A trigger. The Event Manager can be programmed to start a conversion on various events. See Chapter 6 for details.

Bit 7 **EXT SOC SEQ1.** External signal SOC bit for SEQ1

0 No action

1 Setting this bit enables an ADC auto-conversion sequence to be started by a signal from the ADCSOC device pin.

Bit 6 **RST SEQ2.** Reset SEQ2

0 No action

1 Immediately resets SEQ2 to an initial "pre-triggered" state, i.e., waiting for a trigger at CONV08. A currently active conversion sequence will be aborted.

Bit 5 **SOC SEQ2.** SOC trigger for Sequencer 2 (SEQ2)

(Only applicable in dual-sequencer mode; ignored in cascaded mode.)

This bit can be set by the following triggers:

S/W – Software writing of 1 to this bit
EVB – Event Manager B

When a trigger occurs, there are three possibilities:

Case 1: SEQ2 idle and SOC bit clear
SEQ2 starts immediately (under arbiter control) and the bit is cleared, allowing for any pending trigger requests.

Case 2: SEQ2 busy and SOC bit clear
Bit is set signifying a trigger request is pending. When SEQ2 finally starts after completing current conversion, this bit will be cleared.

Case 3: SEQ2 busy and SOC bit set
Any trigger occurring in this case will be ignored (lost).

0 Clears a pending SOC trigger.
Note: If the sequencer has already started, this bit will automatically be cleared, and hence, writing a zero will have no effect; i.e., an already started sequencer cannot be stopped by clearing this bit.

1 Software trigger – Start SEQ2 from currently stopped position (i.e., idle mode)

Bit 4 **SEQ2 BSY**. SEQ2 Busy

This bit is set to a 1 while the ADC auto-conversion sequence is in progress. It is cleared when the conversion sequence is complete.

0 Sequencer is idle (i.e., waiting for trigger).
1 Conversion sequence is in progress.

Bits 3–2 INT ENA SEQ2. Interrupt-mode-enable control for SEQ2

Bit 3	Bit 2	Operation Description
0	0	Interrupt is disabled
0	1	Interrupt **Mode 1** Interrupt requested immediate on INT FLAG SEQ2 flag set
1	0	Interrupt **Mode 2** Interrupt requested only if INT FLAG SEQ2 flag is already set. If clear,[†] INT FLAG SEQ2 flag is set and INT request is suppressed. (This mode allows interrupt requests to be generated for every other EOS.)
1	1	Reserved

[†] This means that the last completed sequence is the first of the two sequences needed to assert an interrupt.

Bit 1 **INT FLAG SEQ2.** ADC interrupt flag bit for SEQ2

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event.
1 An interrupt event has occurred.

Note: *The bit polling algorithm discussed after the bit 9 description is also valid for the INT FLAG SEQ2 bit.*

Bit 0 **EVB SOC SEQ2.** Event Manager B SOC mask bit for SEQ2

0 SEQ2 cannot be started by EVB trigger.
1 Allows SEQ2 to be started by Event Manager B trigger. The Event Manager can be programmed to start a conversion on various events. See Chapter 6 for details.

This concludes the main operating modes of the ADC sequencers. Now that the reader has a general idea of the basic modes of operation (necessary for the initialization of registers ADCTRL1 and ADCTRL2), we will now discuss the configuration of the other ADC registers.

5.2.5 Specifying the Maximum Number of Auto-Conversions

The MAXCONV register is used to specify the maximum number of conversions that the ADC will automatically perform once triggered. The MAXCONV register should be loaded with the maximum number of desired auto-conversions minus 1. In this case, since 16 is the maximum number of conversions that the ADC can perform, the maximum value that should be loaded in the MAXCONV register is “0Fh”.

When the ADC is in dual sequencer mode, the MAXCONV register is “split” and serves both SEQ1 and SEQ2. The lower half of the register serves SEQ1, while the upper half serves SEQ2. See the bit description of MAXCONV below.

Maximum Conversion Channels Register (MAXCONV) — Address 70A2h

15-8							
Reserved							
R–x							
7	6	5	4	3	2	1	0
Reserved	MAX CONV2_2	MAX CONV2_1	MAX CONV2_0	MAX CONV1_3	MAX CONV1_2	MAX CONV1_1	MAX CONV1_0
R–x	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0

Note: *R = read access, W = write access, x = undefined, -0 = value after reset.*

Bits 15–7 Reserved

Bits 6–0 MAX CONVn. MAX CONVn bit field defines the maximum number of conversions executed in an auto-conversion session. The bit fields and their operation vary according to the sequencer modes (dual/cascaded).

- For SEQ1 operation, bits MAX CONV1_2 – 0 are used.
- For SEQ2 operation, bits MAX CONV2_2 – 0 are used.
- For SEQ operation, bits MAX CONV1_3 – 0 are used.

An auto-conversion session always starts with the initial state and continues sequentially until the end state if allowed. The result registers are filled in a sequential order. Any number of conversions between 1 and (MAX CONVn +1) can be programmed for a session.

Example: *MAXCONV Register Bit Programming*

If only five conversions are required, then MAX CONVn is set to four.

Case 1: Dual mode SEQ1 and cascaded mode
Sequencer goes from CONV00 to CONV04, and the five conversion results are stored in the registers Result 00 to Result 04 of the Conversion Result Buffer.

Case 2: Dual mode SEQ2
Sequencer goes from CONV08 to CONV12, and the five conversion results are stored in the registers Result 08 to Result 12 of the Conversion Result Buffer.

MAX CONV1 Value >7 for Dual-Sequencer Mode

If a value for MAX CONV1, which is greater than 7, is chosen for the dual-sequencer mode (i.e., two separate 8-state sequencers), then SEQ CNTR n will continue counting past seven, causing the sequencer to wrap around to CONV00 and continue counting.

5.2.6 Specifying ADC Input Channels and Conversion Order

The ADC input channels and conversion order are specified by the four Channel Select and Sequencing registers (CHSELSEQ1 through CHSELSEQ4). Each register selects four channels, which must be loaded in reverse order (from the least significant hex number to the most significant).

The ADC will perform conversions on the 16 channels in the order that is specified by the channel select sequence registers (CHSELSEQn). Channels must be written to the CHSELSEQ registers in reverse order (see Example 5.1).

CHSELSEQ1 controls and specifies conversions CONV00 through CONV03.
CHSELSEQ2 controls and specifies conversions CONV04 through CONV07.
CHSELSEQ3 controls and specifies conversions CONV08 through CONV11.
CHSELSEQ4 controls and specifies conversions CONV12 through CONV15.

Example 5.1: We want to perform conversions on channels 2, 4, 1, 5, 7, 1, and 4 in this order. We would load CHSELSEQ1 with "5142 h" and CHSELSEQ2 with "417 h". Since 7 conversions are needed, we would load MAXCONV with "6h".

ADC Input Channel Select Sequencing Control Registers (CHSELSEQn)

Each of the 4-bit fields, CONVnn, selects one of the 16 multiplexed analog input ADC channels for an auto-sequenced conversion.

	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0	
70A3h	CONV03	CONV02	CONV01	CONV00	CHSELSEQ1
	RW–0	RW–0	RW–0	RW–0	

***Note:** R = read access, W = write access, -0 = value after reset.*

	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0	
70A4h	CONV07	CONV06	CONV05	CONV04	CHSELSEQ2
	RW–0	RW–0	RW–0	RW–0	

Note: *R = read access, W = write access, -0 = value after reset.*

	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0	
70A5h	CONV11	CONV10	CONV09	CONV08	CHSELSEQ3
	RW–0	RW–0	RW–0	RW–0	

Note: *R = read access, W = write access, -0 = value after reset.*

	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0	
70A6h	CONV15	CONV14	CONV13	CONV12	CHSELSEQ4
	RW–0	RW–0	RW–0	RW–0	

Note: *R = read access, W = write access, -0 = value after reset.*

5.2.7 Results of the ADC Conversion

After the ADC has finished performing the number of conversions specified by the MAXCONV register, the RESULTn registers can be read. Each result register contains a 10-bit conversion result in the 10 most significant bits (MSB) of the register. There are 16 total result registers, RESULT0 through RESULT15. These registers contain the conversion results in the sequential order that the conversions take place. For example, the result of the first conversion performed will be stored in RESULT0, the second in RESULT1 etc.

It is usually desired to right shift the contents of the result register by six places in order to truncate the extra zeros. This right shift can be performed easily by the SFR command. Once the ADC result has been shifted, it may be used in calculations or other purposes. The bit descriptions of the RESULT registers are given below.

ADC Conversion Result Buffer Registers (RESULTn)

Note: In the cascaded sequencer mode, registers RESULT8 through RESULT15 will hold the results of the ninth through sixteenth conversions.

15	14	13	12	11	10	9	8
D9	D8	D7	D6	D5	D4	D3	D2
7	6	5	4	3	2	1	0
D1	D0	0	0	0	0	0	0

Notes:
1) Buffer addresses = 70A8h to 70B7h (i.e., 16 registers)
2) The 10-bit conversion result (D9–D0) is left-justified

5.2.8 The Auto-Sequence Status Register

The Auto-Sequence Status Register contains information on the current state of the sequencer when running conversions. Its bits can be polled (read) to determine, for example, if the sequencer is near or closer to the end number of conversions.

Auto-sequence Status Register (AUTO_SEQ_SR) — Address 70A7h

15-12				11	10	9	8
Reserved				SEQ CNTR 3	SEQ CNTR 2	SEQ CNTR 1	SEQ CNTR 0
R–x				R–0	R–0	R–0	R–0
7	6	5	4	3	2	1	0
Reserved	SEQ2–State2	SEQ2–State1	SEQ2–State0	SEQ1–State3	SEQ1–State2	SEQ1–State1	SEQ1–State0
R–x	R–0	R–0	R–0	R–0	R–0	R–0	R–0

***Note:** R = read access, x = undefined, -0 = value after reset.*

Bits 15–12 Reserved

Bits 11–8 SEQ CNTR 3 – SEQ CNTR 0. Sequencing counter status bits

The SEQ CNTR n 4-bit status field is used by SEQ1, SEQ2, and the cascaded sequencer. SEQ2 is irrelevant in cascaded mode.

At the start of an auto-sequenced session, SEQ CNTR n is loaded with the value from MAX CONVn. The SEQ CNTR n bits can be read at any time during the countdown process to check the status of the sequencer. This value, together with the SEQ1 and SEQ2 busy bits, uniquely identifies the progress or state of the active sequencer at any point in time.

SEQ CNTR n (read only)	Number of conversions remaining
0000	1
0001	2
0010	3
0011	4
0100	5
0101	6
0110	7
0111	8
1000	9
1001	10
1010	11
1011	12
1100	13
1101	14
1110	15
1111	16

Bit 7 Reserved

Bits 6–4 SEQ2-State2 through SEQ2-State0

Reflects the state of SEQ2 sequencer at any point of time. If need be, user can poll these bits to read interim results before an EOS. SEQ2 is irrelevant in cascaded mode.

Bits 3–0 SEQ1-State3 through SEQ1-State0

Reflects the state of SEQ1 sequencer at any point of time. If need be, user can poll these bits to read interim results before an EOS.

ADC Register Addresses Summary (Mapped in Data Memory)

Address	Register	Name
70A0h	ADCTRL1	ADC control register 1
70A1h	ADCTRL2	ADC control register 2
70A2h	MAXCONV	Maximum conversion channels register
70A3h	CHSELSEQ1	Channel select sequencing control register 1
70A4h	CHSELSEQ2	Channel select sequencing control register 2
70A5h	CHSELSEQ3	Channel select sequencing control register 3
70A6h	CHSELSEQ4	Channel select sequencing control register 4
70A7h	AUTO_SEQ_SR	Autosequence status register
70A8h	RESULT0	Conversion result buffer register 0
70A9h	RESULT1	Conversion result buffer register 1
70AAh	RESULT2	Conversion result buffer register 2
70ABh	RESULT3	Conversion result buffer register 3
70ACh	RESULT4	Conversion result buffer register 4
70ADh	RESULT5	Conversion result buffer register 5
70AEh	RESULT6	Conversion result buffer register 6
70AFh	RESULT7	Conversion result buffer register 7
70B0h	RESULT8	Conversion result buffer register 8
70B1h	RESULT9	Conversion result buffer register 9
70B2h	RESULT10	Conversion result buffer register 10
70B3h	RESULT11	Conversion result buffer register 11
70B4h	RESULT12	Conversion result buffer register 12
70B5h	RESULT13	Conversion result buffer register 13
70B6h	RESULT14	Conversion result buffer register 14
70B7h	RESULT15	Conversion result buffer register 15
70B8h	CALIBRATION	Calibration result, used to correct subsequent conversions

5.3 Analog to Digital Converter Usage Exercise

The purpose of this exercise is to familiarize the reader with the practical usage of the ADC. As stated earlier, the ADC on the LF2407 produces a 10-bit binary number which represents the voltage of the sampled analog signal.

This 10-bit number is stored in a 16-bit register "RESULTn" n=0..15. When reading from the register, the least significant 6 bits (bits 0-5) need to be disregarded because the 10 ADC result bits are bits 15-6. This can be done with the repeat command and SFR command.

For example, after the initial value is loaded into the accumulator:

```
RPT  #5 SFR      ; the accumulator will be shifted right 6 (RPT+1) times
```

Create an assembly source file and project file called "lab5"which:

a. Turns on the ADC clock in SCSR1 during the general initialization.
b. Puts the ADC in Reset.
c. Configures the ADC for Cascaded Mode; Continuous Mode = OFF; brings ADC out of Reset mode.
d. Sets maximum conversions to one and selects channel 0 for conversion.
e. Triggers a SOC via BIT 13 in ADCTRL2.
f. Checks if the ADC is finished via BIT 12 in ADCCTRL2.
g. If the ADC is finished with the conversion, the accumulator is loaded with the value in RESULT0 (the ADC conversion).
h. Continuously loops in an endless loop, i.e., does nothing.

1. Run the code on the LF2407 EVM with a 1.5V battery connected to channel 0 of the ADC and analog ground.
2. Record the value from the accumulator (the accumulator has been loaded with the value from the first ADC RESULT register). The value should be approximately half of the full voltage value 0x3FFF or approximately "1FF" = (1.5V).
3. Modify the program to output the result data from address 60h to DAC channel 1 on the LF2407 EVM. The DAC onboard is a 12-bit DAC, so in order to get the correct voltage output, left shift the accumulator (ADC data) two places to account for the extra two least significant bits. Also, the EVM DAC has a voltage reference of 5V rather then 3.3V like the ADC. This will cause a voltage slightly higher than what the ADC sampled to be outputted on the DAC channel. For this academic exercise, this can be ignored because the voltage difference is somewhat small. When writing to the DAC Channel 1, the data must first be written to IO space address 0000h. Then, in order for the data to actually be "sent out" on the DAC, the IO space address 0004h must be written to. It does not matter what value is written to IO address 0004h, just as long as it is written to.
4. The data will then be sent from the buffers to the DAC outputs. (See the OUT command and the Spectrum Digital LF2407 EVM manual for more information on programming the on-board DAC.)
5. Rebuild the project, reset the DSP, and run the new code.
6. Measure the voltage output of the DAC. It should reflect the ADC input voltage.
7. Modify the program to have the ADC continuously sample and continuously output the sampled data on the DAC.
8. Rebuild the program, reset, and run the DSP.

This concludes the ADC usage exercise.