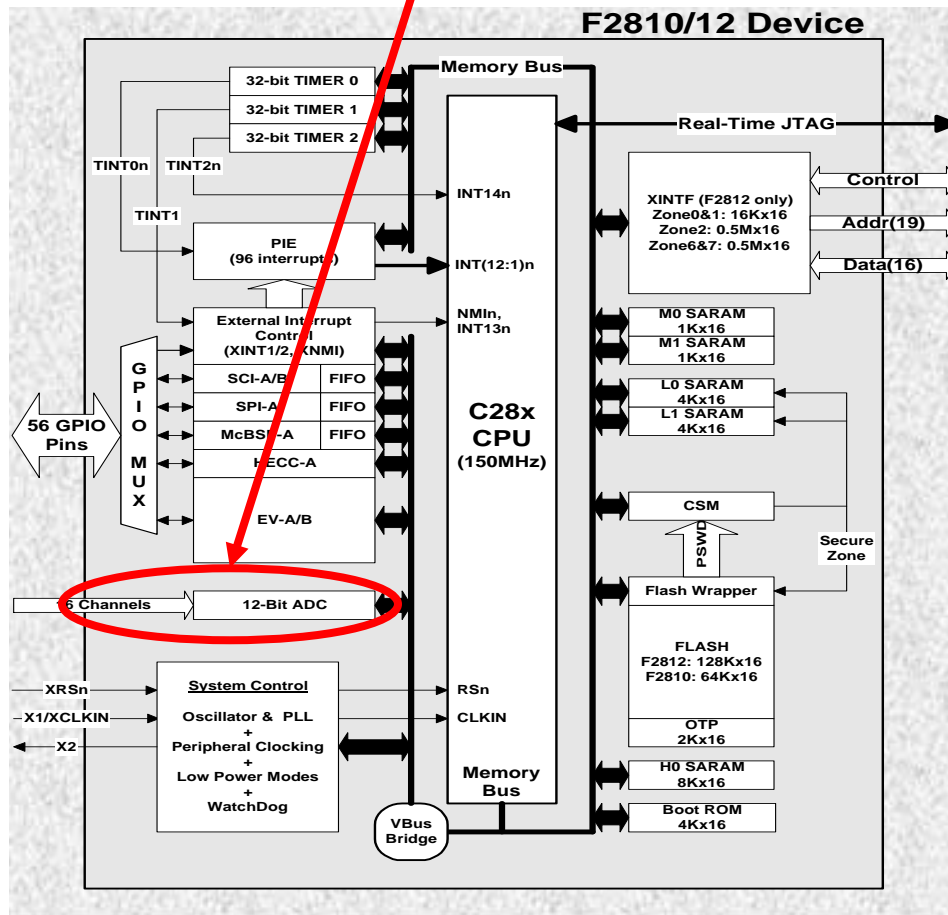


F2810/12 Application Note

ADC Calibration



i. Document History:

Version	Date	Who	Updates
1.0	20 June 2003	A.T./S.D.	Original Release.



1. Introduction

This application note describes a method for improving the absolute accuracy of the 12-bit ADC found on the F2810/12 devices. Due to inherent gain and offset errors, the absolute accuracy of the ADC is impacted. The methods described in this application note can improve the absolute accuracy of the ADC to achieve levels better than 0.5%.

This application note contains the following sections:

2. Gain & Offset Error Definition
3. Gain & Offset Error Impact
4. Calibration
5. Hardware Connectivity
6. ADC Sampling Techniques
7. Example Software Calibration Driver
8. Useful Tips

This application note is accompanied by an example program that executes from RAM on the F2812 EzDSP.

- [ADCcalibration.zip](#)

ATTENTION

The data in this application note is based on observations of ADC behavior on F2810/12 Rev-C TMX devices. More accurate data will be available when full characterization is performed (TMS devices).



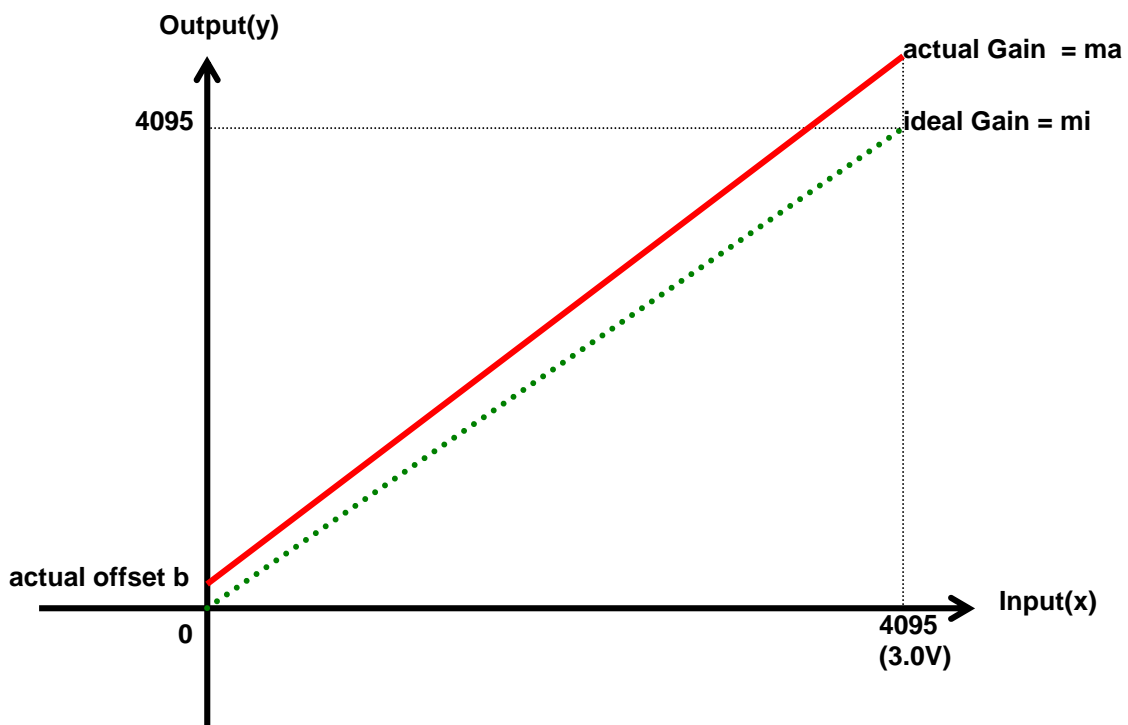
2. Gain & Offset Error Definition

An ideal 12-bit ADC with no gain and offset error would be described by equation (1):

(1)..... $y = x * m_i$ $x = \text{input count} = \text{input voltage} * 4095/3.0V$
 $y = \text{output count}$
 $m_i = \text{ideal gain} = 1.000$

The F2810/12 ADC will exhibit gain and offset error defined by equation (2):

(2)..... $y = x * m_a + b$ $m_a = \text{actual gain}$
 $b = \text{actual offset (relative to 0 input)}$



Gain and offset errors measured on the F2810/12 (Rev-C, TMX) ADC are:

Gain Error (m_a):	< +5% max	$1.00 < m_a < 1.05$
Offset Error (b):	< +/-2% max	$-80 < b < 80$

Note: Most F2810/12 Rev-C TMX devices exhibit positive gain errors of <+3.5% and offset errors of <+/-1%. However, some devices exhibit errors above these values reaching the maximum values outlined above. More accurate data will be available when full characterization is performed (TMS devices). To be safe, designers should assume worst case scenario.

3. Gain & Offset Error Impact

Gain and offset errors contribute to errors in the control system. Understanding these errors will enable the user to compensate for these errors in the design.

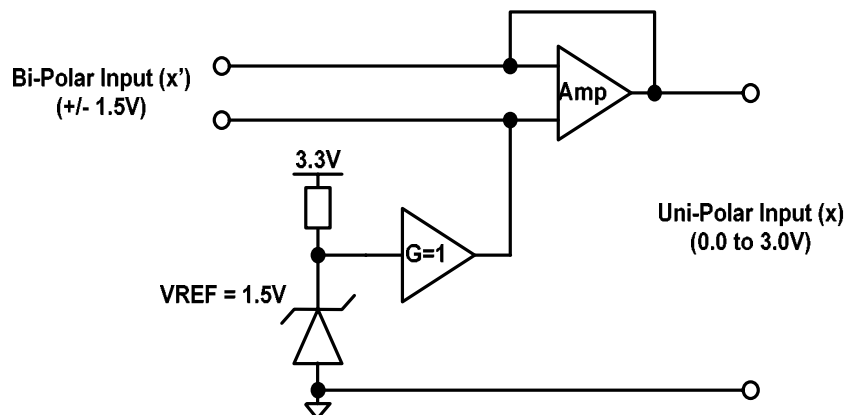
Linear Input Range: The available input voltage range is impacted by the gain and offset errors and the effective resolution is also reduced. The table below summarizes the worst case scenarios:

	Linear Input Range (Volts)	Linear Output Range (counts)	Input Swing (Volts)	Effective Number Of Bits	mV/Count Resolution
$y = x * 1.00$	0.000 to 3.000	0 to 4095	1.500 +/-1.500	12.00	0.732
$y = x * 1.05 + 80$	0.000 to 2.801	80 to 4095	1.415 +/-1.415	11.97	0.697
$y = x * 1.05 - 80$	0.056 to 2.913	0 to 4095	1.457 +/-1.428	12.00	0.698
Safe Range	0.056 to 2.801	80 to 4095	1.429 +/- 1.372	11.97	0.684

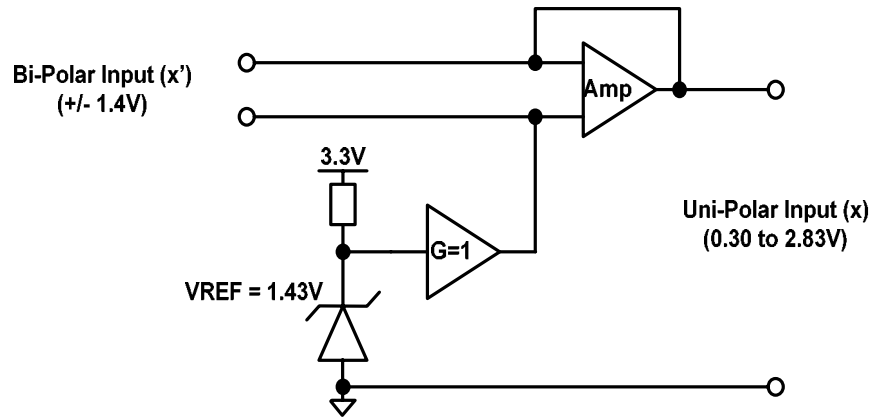
The last row in the table, shows the “safe” parameters with which all devices can be guaranteed to operate under. The effective number of bits of the ADC is only slightly reduced (11.9 bits). The reduction in voltage range has a slight impact on external noise sensitivity. The mV/count is reduced from 0.732 to 0.684 which is a 6.6% reduction (or a 6.6% increase in sensitivity over the ideal case).

Note: The calibration technique described in this paper will not improve the input range. On future revisions of the devices, an external reference option will be supported which will enable a tighter gain tolerance (< +/-1%). This will significantly reduce the impact of gain error on the input range.

Bi-Polar Offset Error: In many applications, the input sensor is a bi-polar input and this needs to be converted to a uni-polar signal before being fed to the ADC. A typical simplified circuit used for this purpose is shown below (ideal ADC case):



Taking into account gain and offset errors and the impact on the input range, the circuit would need to be modified as follows to cater for all possible device characteristics:



The frame of reference for the input offset error has changed. The offset error is measured relative to the bi-polar input when the input value is zero ($x' = 0$). This corresponds to a uni-polar input value of $x = 1.43V$. The inherent ADC gain and offset errors will tend to magnify the error relative to the ideal value.

For example: If we assume the ADC has a +5% gain error and +1% offset error, then the bi-polar offset error count is:

Bi-Polar Input:	$x' = 0.00V$
Uni-Polar ADC Input:	$x = 1.43V$
Expected Count:	$ye = 1.43 * 4095/3 = 1952$
Actual Count:	$ya = 1952 * 1.05 + 40 = 2090$
Bi-Polar Offset Error:	$ya - ye = 2090 - 1952 = 138 \text{ counts (3.3\% error)}$

This is a much higher error than a user would otherwise expect. This error can effectively be removed by calibration.

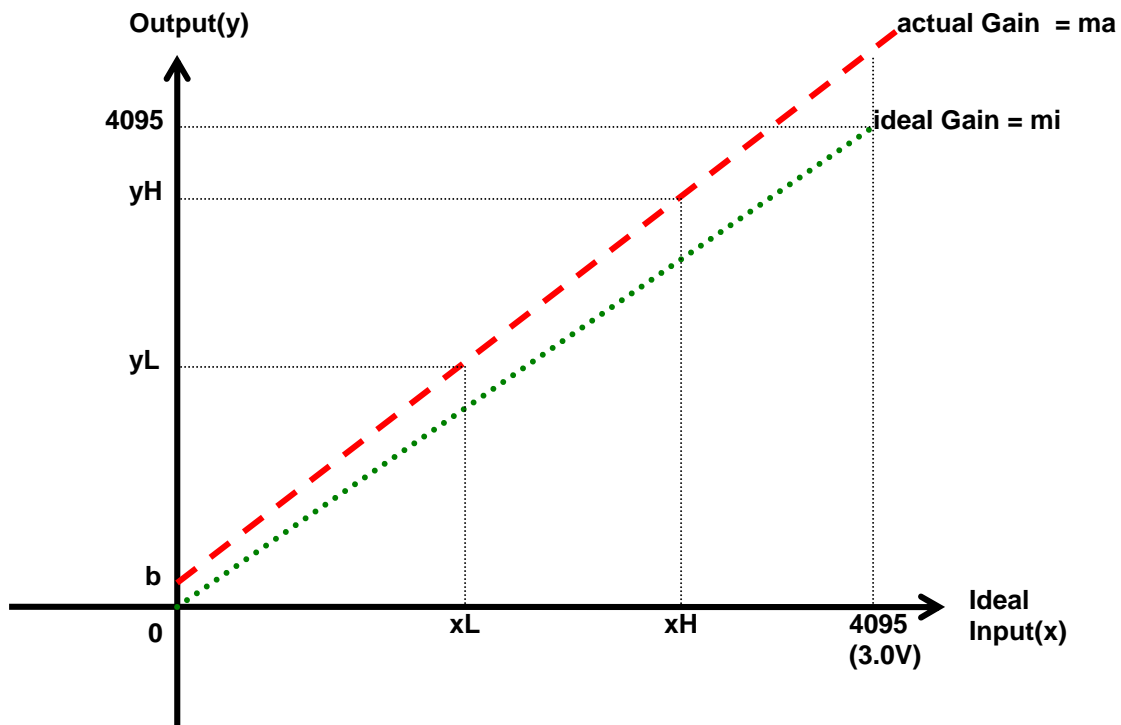


4. Calibration

Calibration is performed by feeding two known reference values into two ADC channels and calculating a calibration gain and offset to compensate the input readings from the other channels. This is possible because the channel-to-channel errors are small. The achievable accuracy using calibration is largely dependant on the accuracy of the know references fed into the ADC. The best possible accuracy achievable is limited by the channel-to-channel gain and offset errors of the ADC.

Note: On Rev-C TMX devices, channel-to-channel gain and offset errors in the order of +/-0.1% have been observed. The exact characteristics will only be available when the devices are fully characterized (TMS devices).

The equations to measure the ADC actual gain and offset and calculate the calibration gain and offset are derived below:



$$y = x * ma + b$$

ma = actual gain
b = actual offset (relative to 0 input)

(3)..... $ma = (yH - yL)/(xH - xL)$

xL = known reference low input
xH = known reference high input
yL = reference low ADC output
yH = reference high ADC output

(4)..... $b = yL - xL * ma$



The calibration equation is derived by inverting the input and output of equation (2) describing the ADC actual gain and offset:

$$y = x * ma + b$$

$$x = (y - b)/ma$$

$$x = y/ma - b/ma$$

$$(5) \dots\dots\dots x = y * \text{CalGain} - \text{CalOffset} \qquad \text{CalGain} = 1/ma$$

$$\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{CalOffset} = b/ma$$

$$(6) \dots\dots\dots \text{CalGain} = (xH - xL)/(yH - yL)$$

$$\text{CalOffset} = (yL - xL * ma)/ma$$

$$\text{CalOffset} = yL/ma - xL$$

$$(7) \dots\dots\dots \text{CalOffset} = yL * \text{CalGain} - xL$$

In summary, using two known references (x_L , y_L) and (x_H , y_H) we can calculate the actual offset and gain error and calculate the calibration gain and offset using the following formulas:

$$(2) \dots\dots\dots y = x * ma + b \qquad \text{ADC actual equation}$$

$$(3) \dots\dots\dots ma = (yH - yL)/(xH - xL) \qquad \text{ADC actual gain}$$

$$(4) \dots\dots\dots b = yL - xL * ma \qquad \text{ADC actual offset}$$

$$(5) \dots\dots\dots x = y * \text{CalGain} - \text{CalOffset} \qquad \text{ADC calibration equation}$$

$$(6) \dots\dots\dots \text{CalGain} = (xH - xL)/(yH - yL) \qquad \text{ADC calibration gain}$$

$$(7) \dots\dots\dots \text{CalOffset} = yL * \text{CalGain} - xL \qquad \text{ADC calibration}$$

The calibration process involves the following four basic steps:

Step 1) Read the known reference values input channels (y_L and y_H).

Step 2) Calculate the calibration gain (CalGain) using equation (6)

Step 3) Calculate the calibration offset (CalOffset) using equation (7)

Step 4) Cycle through all channels applying the calibration equation (5)



5. Hardware Connectivity

Calibration requires that two ADC channels be dedicated to supply two known reference inputs and this leaves 14 user channels. Recommended connection is shown in Figure 5.1 below:

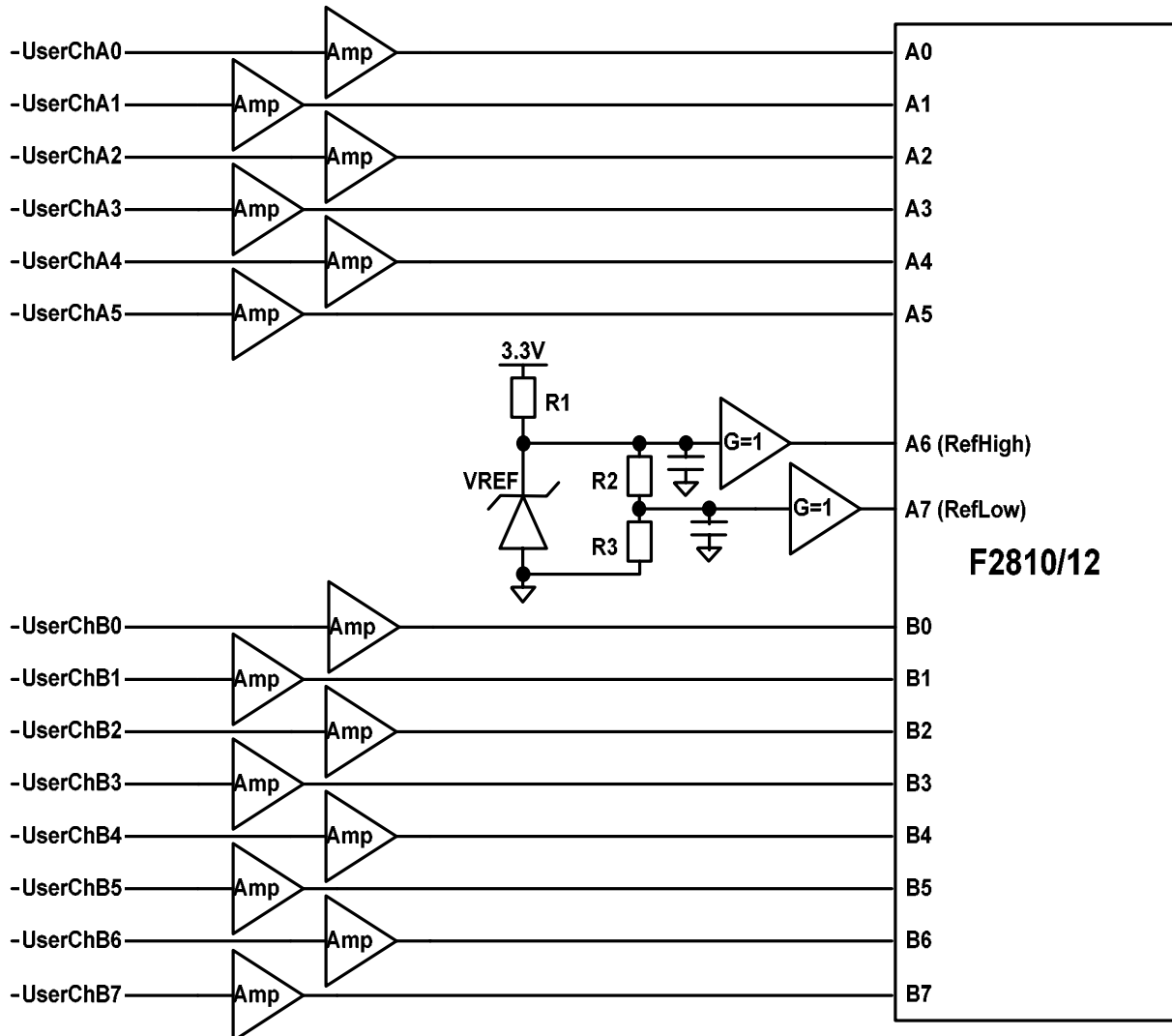


Figure 5.1: 2 Channels For Calibration, 14 User Channels

Note: The choice of channels is user selectable. For improved accuracy, references should be connected to channels belonging to the same group. The accuracy of calibration will be largely dependant on the accuracy of the reference voltages and hence the external components used.

If converting a bi-polar input to uni-polar, then a mid-point reference (~1.5V) would be a good choice as one reference input. The other reference input could either be a higher value (~2.5V) or lower value (~0.5V) reference.



For users wishing to retain 16 user channels, the system shown in Figure 5.2 below is recommended. In this scenario, an external analog switch is added which expands the user channels to 16 and is controlled by a GPIO pin using software. In a simple software implementation, the multiplexed channels are sampled on every alternate cycle, relative to the non-multiplexed channels. This means that the multiplexed channels should be used for slower, supervisory type functions. This system leaves the user with 6 channel pairs (if using simultaneous sampling mode) that can be used for critical functions:

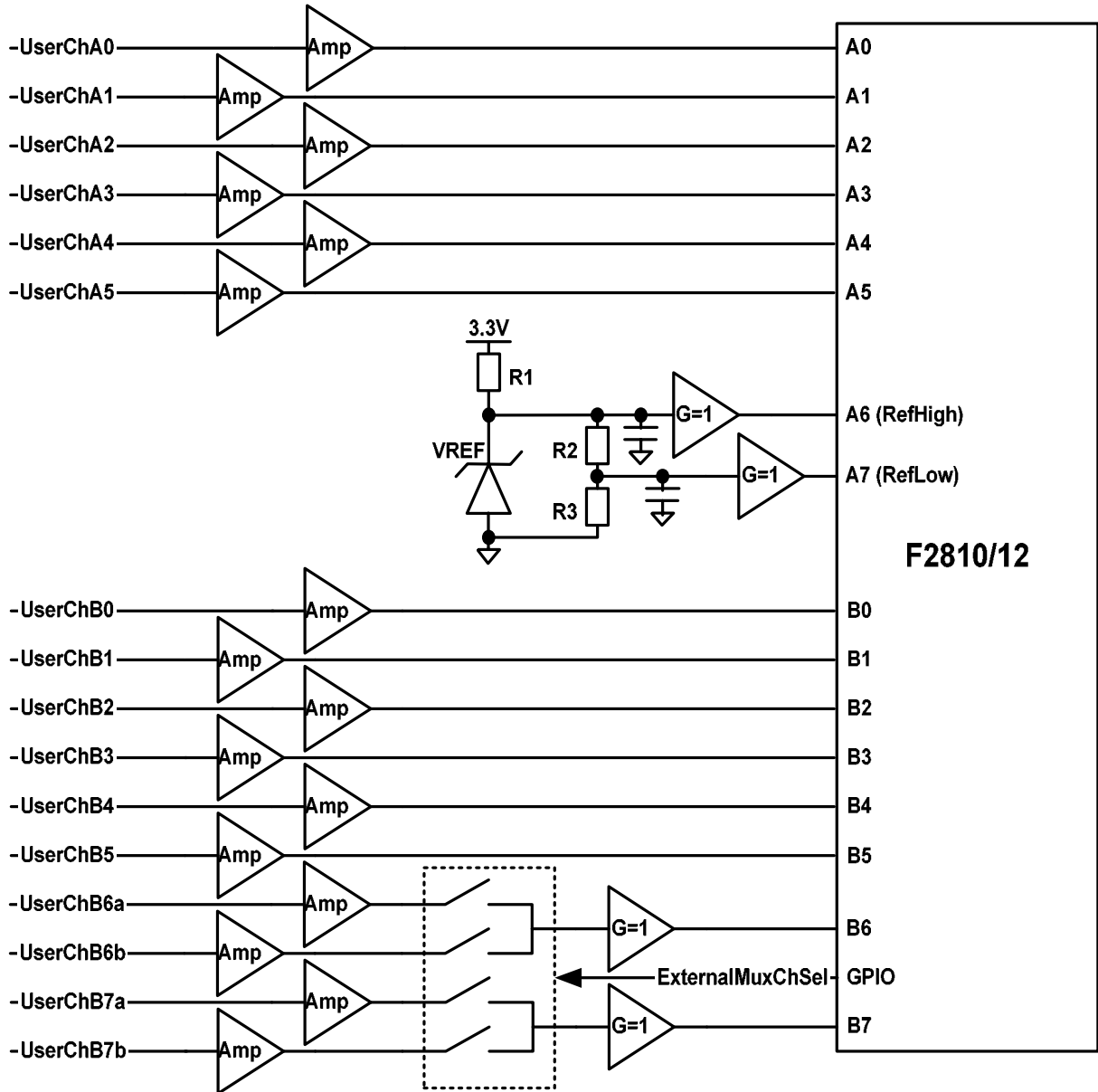


Figure 5.2: 2 Channels For Calibration, 16 User Channels

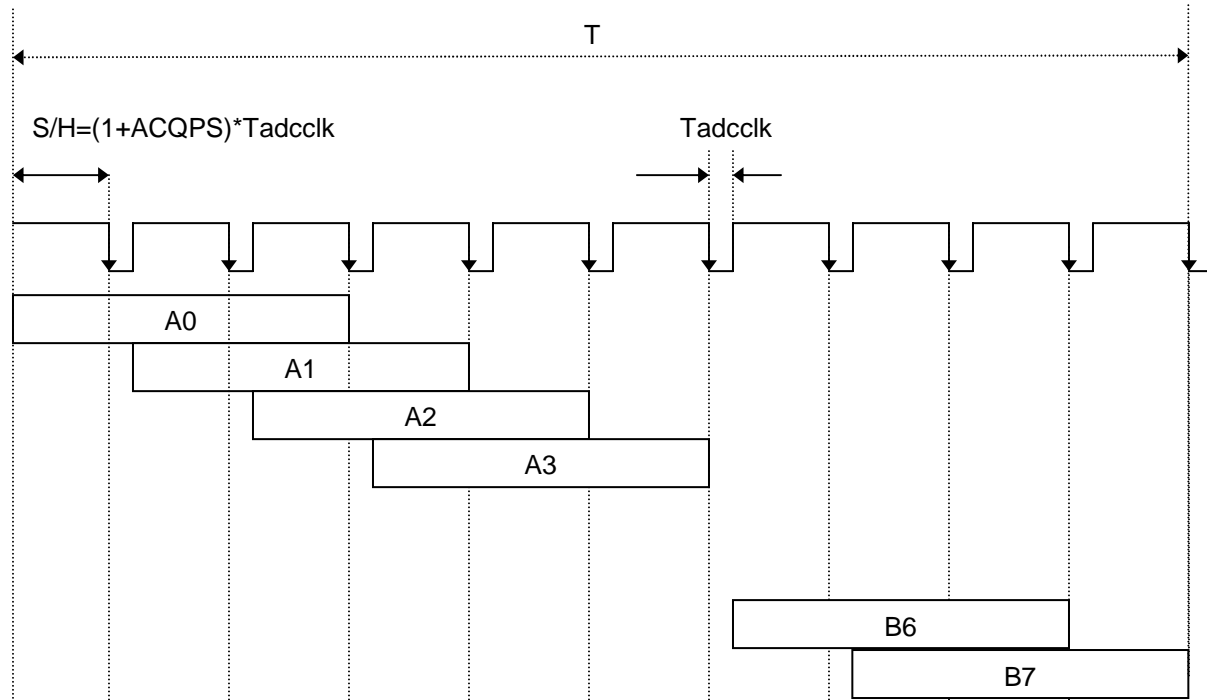
Note: A buffer is required at the output of the mux (or any high resistance source) to prevent errors due to high source impedance when sampling of the ADC channels.

6. ADC Sampling Techniques

6.1. Sequential Sampling mode

ADC converter on F2810/12 device can operate in sequential sampling mode or simultaneous sampling mode. In sequential sampling mode, ADC samples one channel at a time & then the sampled signal is passed through four stages of ADC pipeline for conversion.

Following diagram shows the timing details of 16-channel conversion (A0-A7 & B0-B7) in sequential mode based on event trigger:



Total Time for converting 16-channels in sequential mode:

$$T = 17 * T_{adcclk} + 18 * (1 + ACQPS) * T_{adcclk}$$

Following table shows the conversion time required to convert all 16-channels in sequential sampling mode under different ADC clock frequency & S/H window:

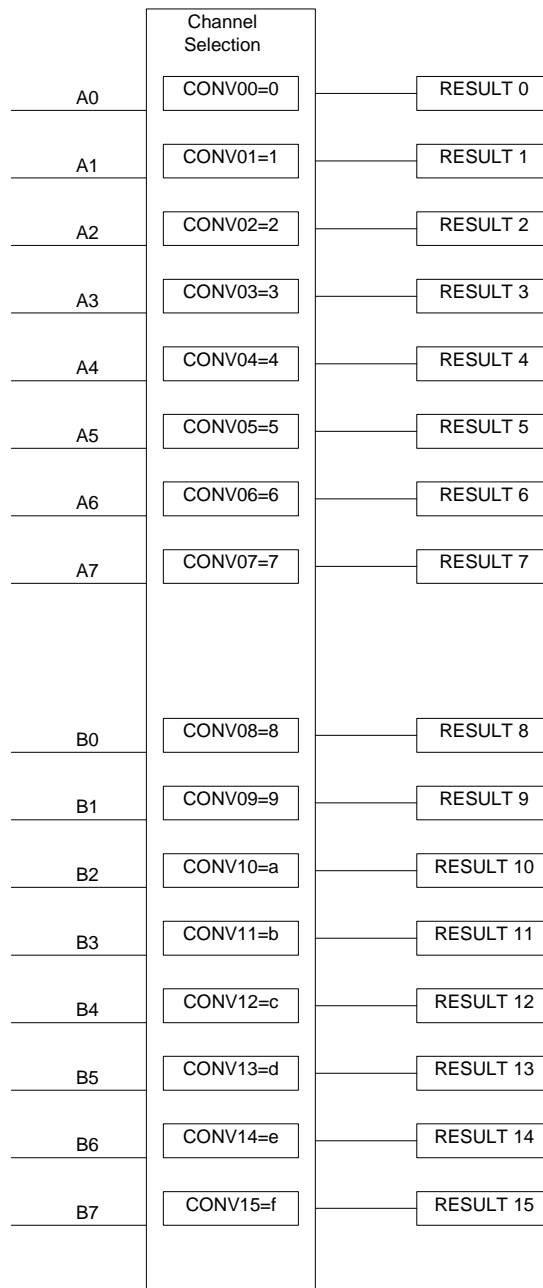
ACQPS	Number Of Tadcclk Periods	T in us (ADCCLK=25Mhz)	T in us (ADCCLK=12.5Mhz)	T in us (ADCCLK=6.25Mhz)
0	35	1.4	2.8	5.6
3	89	3.56	7.12	14.24
7	161	6.44	12.88	25.76
11	233	9.32	18.64	37.28
15	305	12.2	24.4	48.8

Note: ACQSPS is the acquisition window width. Value of 0 is equal to one ADCCLK period.



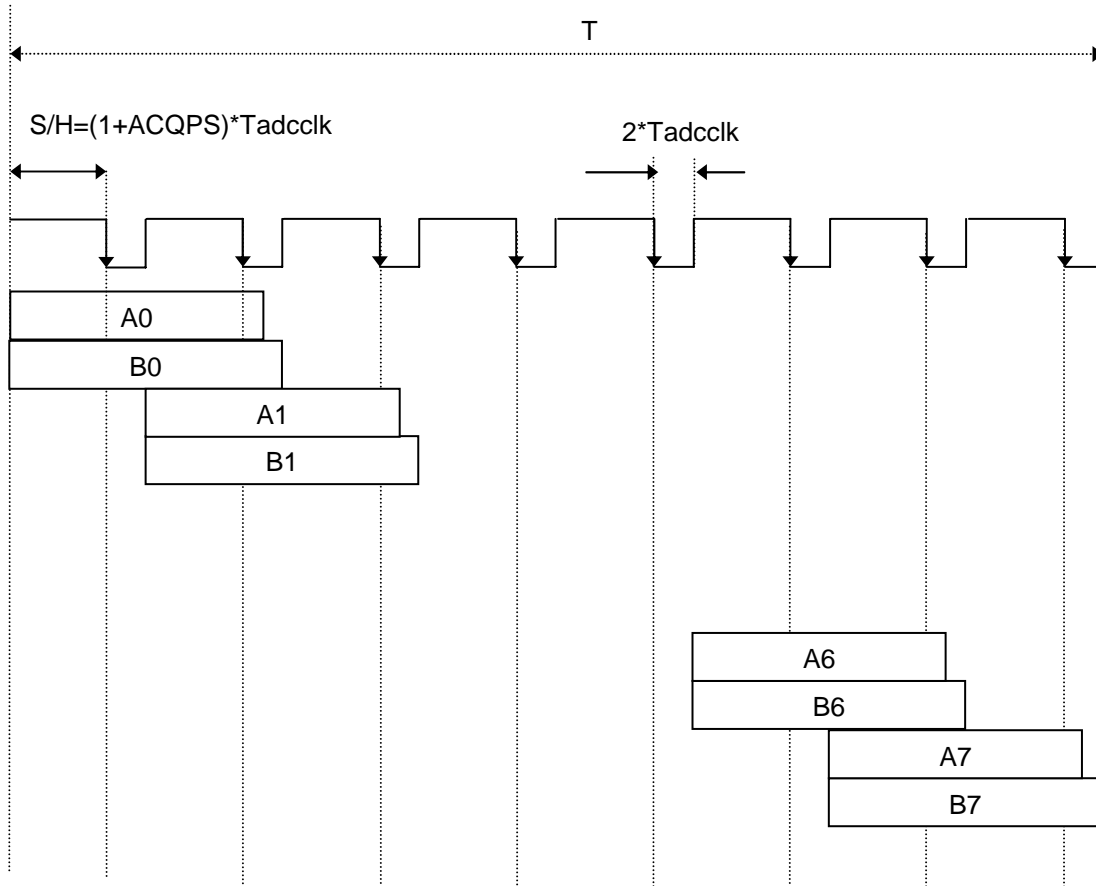
In sequential sampling mode, ADC can be configured in cascade mode or dual sequencer mode. In cascade mode, we can schedule 16-ADC conversions sequentially based on an even trigger. The order in which the channels are converted & stored in result register is controlled by ADC Channel selection control register (CHSELSEQ1, CHSELSEQ2, CHSELSEQ3, CHSELSEQ4). For direct mapping of channels to corresponding result register, we need to program the channel selection control register to the values shown below:

CHSELSEQ1 = 0x3210
CHSELSEQ2 = 0x7654
CHSELSEQ3 = 0xba98
CHSELSEQ4 = 0xfedc



6.2. Simultaneous Sampling mode

In simultaneous sampling mode, the ADC can convert input signals on any pair of channels (A0/B0 to A7/B7). Basically two channels are sampled simultaneously & passed through four stages of ADC pipeline for conversion. Following diagram shows the timing details of 8-pairs of channels (A0/B0 to A7/B7) converted in simultaneous sampling mode based on event trigger:



Total Time for converting 16-channels in simultaneous mode:

$$T = 9 * 2 * T_{adccclk} + 9 * (1 + ACQPS) * T_{adccclk}$$

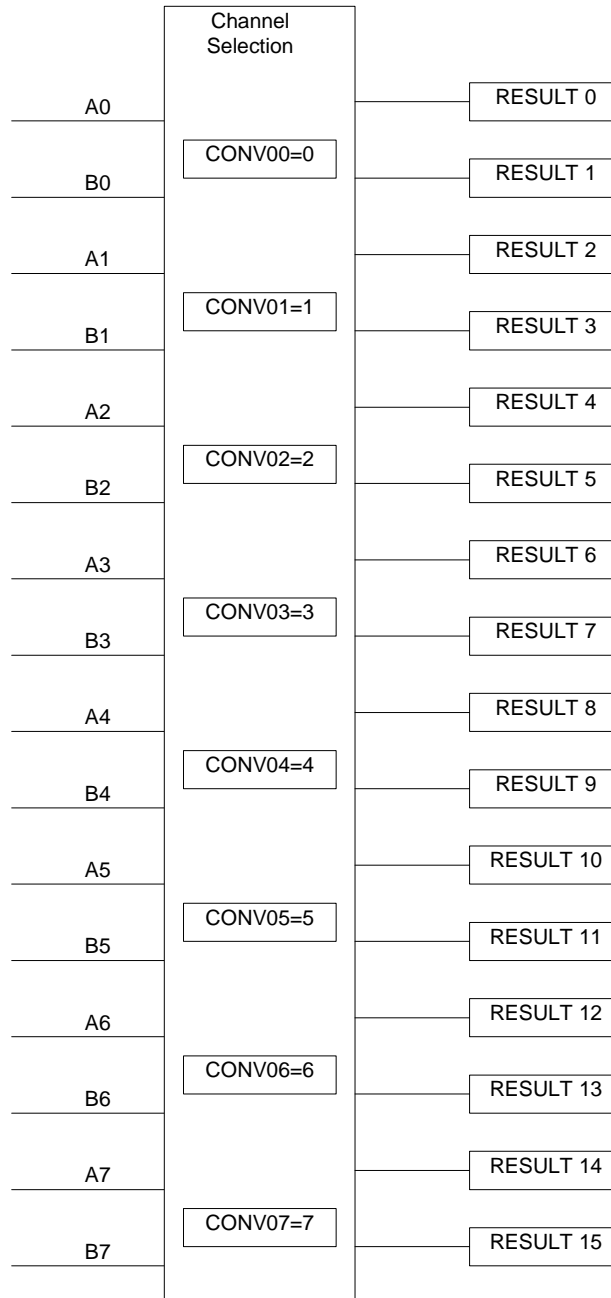
Following tables shows the conversion time required to convert all 16-channels in simultaneous sampling mode under different ADC clock frequency & S/H window.

ACQPS	Number Of T _{adccclk} Periods	T in us (ADCCLK=25Mhz)	T in us (ADCCLK=12.5Mhz)	T in us (ADCCLK=6.25Mhz)
0	27	1.08	2.16	4.32
3	54	2.16	4.32	8.64
7	90	3.6	7.2	14.4
11	126	5.04	10.08	20.16
15	162	6.48	12.96	25.92



In simultaneous sampling mode, we can schedule 8-pair of simultaneous ADC conversions based on an even trigger (A0/B0, A1/B1,...A7/B7). The order in which the channel-pairs are converted & stored in result register is controlled by ADC Channel selection control register (CHSELSEQ1, CHSELSEQ2). To schedule the channels pairs in normal order A0/B0, A1/B1,...A7/B7, channel selection register must be programmed to the following values:

CHSELSEQ1 = 0x3210
CHSELSEQ2 = 0x7654



7. Example Software Calibration Driver

This application note comes with an example C program for simultaneous sampling mode & sequential sampling mode that executes from RAM on the F2812 EzDSP. The example program configures the Event Manager to generate a periodic start of conversion pulse to the ADC. The ADC is configured to process all 16-input channels and then generate an interrupt. In the interrupt service routine, a call is made to an optimized assembly driver which will read the user selected ADC channel references, calculate the calibration gain and offset and calibrate all other channels and store the information in a RAM structure.

The program supports configuring the ADC for simultaneous or sequential conversion modes:

SEQUENTIAL: ADC channels are converted one at a time:
A0 -> A1 -> A2 -> ... B0 -> B1 -> B2 ->

SIMULTANEOUS: ADC channels are converted in pairs:
A0,B0 -> A1,B1 -> A2,B2 ->

The calibrated and converted channels are stored in a RAM structure which contains the following information:

```
typedef struct {
    Uint16 *RefHighChAddr;    // Channel Address of RefHigh
    Uint16 *RefLowChAddr;    // Channel Address of RefLow
    Uint16 *Ch0Addr;         // Channel 0 Address
    Uint16 Avg_RefHighActualCount; // Ideal RefHigh Count (Q4)
    Uint16 Avg_RefLowActualCount;  // Ideal RefLow Count (Q4)
    Uint16 RefHighIdealCount;     // Ideal RefHigh Count (Q0)
    Uint16 RefLowIdealCount;      // Ideal RefLow Count (Q0)
    Uint16 CalGain;               // Calibration Gain (Q12)
    Uint16 CalOffset;            // Calibration Offset (Q0)
    // Store Calibrated ADC Data (Q0):
    // Simultaneous Sequential
    // =====
    Uint16 ch0;                  // A0 A0
    Uint16 ch1;                  // B0 A1
    Uint16 ch2;                  // A1 A2
    Uint16 ch3;                  // B1 A3
    Uint16 ch4;                  // A2 A4
    Uint16 ch5;                  // B2 A5
    Uint16 ch6;                  // A3 A6
    Uint16 ch7;                  // B3 A7
    Uint16 ch8;                  // A4 B0
    Uint16 ch9;                  // B4 B1
    Uint16 ch10;                 // A5 B2
    Uint16 ch11;                 // B5 B3
    Uint16 ch12;                 // A6 B4
    Uint16 ch13;                 // B6 B5
    Uint16 ch14;                 // A7 B6
    Uint16 ch15;                 // B7 B7
    Uint16 StatusExtMux;         // Indicates Status Of External Mux For
    // Current Conversion
}ADC_CALIBRATION_DRIVER_VARS;
```



This program also supports the user toggling a GPIO pin for toggling an external analog mux to expand the usable channels.

To adapt to the user system needs, the user needs to configure assembly time switches and settings contained in the header file:

ADCcalibrationDriver.h

The user needs to select simultaneous or sequential sampling mode of operation. For example:

```
#define SEQUENTIAL 1
#define SIMULTANEOUS 0
#define ADC_SAMPLING_MODE SIMULTANEOUS
```

And needs to select which ADC channels are connected to reference high and reference low and the ideal count value. For example:

```
A6 = RefHigh = 2.5V ( 2.5*4095/3.0 = 3413 ideal count)
A7 = RefLow = 1.25V (1.25*4095/3.0 = 1707 ideal count)
```

```
#define REF_HIGH_CH A6
#define REF_LOW_CH A7
#define REF_HIGH IDEAL_COUNT 3413
#define REF_LOW IDEAL_COUNT 1707
```

Number of cycles required to execute the calibration driver is:

137cycles or 0.91uS @150MHz (9.7 cycles per user channel)

Without calibration, the driver would take approximately 4.7 cycles per user channel to read and store the ADC input. The calibration overhead is therefore approximately an additional 5 cycles per channel.



8. Useful Tips

This is a collection of tips or options that the user can or should consider to improve or guarantee accuracy:

- **Provide Low Resistance Path For ADCLO Pin.** Always make sure that the ADCLO pin on the F2810/12 device is connected directly to analog ground. Any resistance on this pin will further degrade offset and gain errors.
- **Use Bi-Polar Input Conversion Reference As Calibration Input.** When converting bi-polar to uni-polar signals, make sure that the reference voltage used as the “mid-point” of the ADC range is fed as an input calibration channel. This will remove any bi-polar offset error (as discussed earlier in this app-note).
- **Reduce ADCCLK Frequency To Minimum.** At higher sampling frequencies (above 10MHz), the channel-to-channel errors begin to increase. To improve accuracy, users should try to use the lowest frequency that their control system will tolerate. Increasing the sampling window will not have much of an effect on this error at high frequencies.
- **Digital & Analog Grounds Connected At One Point:** To avoid noise created by digital current loops, connect the digital and analog grounds at one point, making sure that any analog or digital current loops do not cross through this point. This is common practice when mixing digital and analog on a single board/device.

The End.

