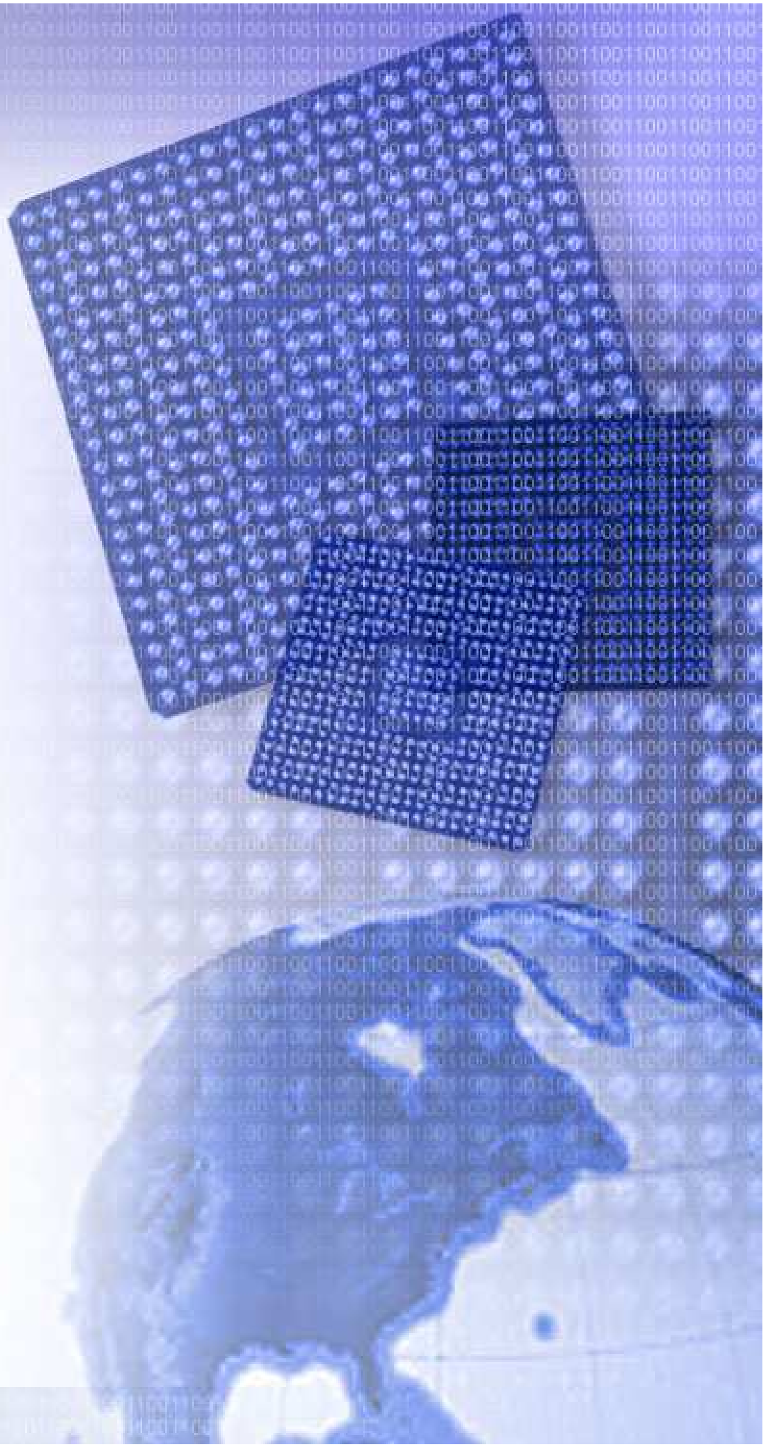
XILINX®

SPARTAN®-3E FPGA

PicoBlaze™

Amplifier and A/D Converter Control
for
Spartan-3E Starter Kit



Ken Chapman
Xilinx Ltd
23rd February 2006

Rev.2

Limitations

Limited Warranty and Disclaimer. These designs are provided to you “as is”. Xilinx and its licensors make and you receive no warranties or conditions, express, implied, statutory or otherwise, and Xilinx specifically disclaims any implied warranties of merchantability, non-infringement, or fitness for a particular purpose. Xilinx does not warrant that the functions contained in these designs will meet your requirements, or that the operation of these designs will be uninterrupted or error free, or that defects in the Designs will be corrected. Furthermore, Xilinx does not warrant or make any representations regarding use or the results of the use of the designs in terms of correctness, accuracy, reliability, or otherwise.

Limitation of Liability. In no event will Xilinx or its licensors be liable for any loss of data, lost profits, cost or procurement of substitute goods or services, or for any special, incidental, consequential, or indirect damages arising from the use or operation of the designs or accompanying documentation, however caused and on any theory of liability. This limitation will apply even if Xilinx has been advised of the possibility of such damage. This limitation shall apply not-withstanding the failure of the essential purpose of any limited remedies herein.

This design module is **not** supported by general Xilinx Technical support as an official Xilinx Product.
Please refer any issues initially to the provider of the module.

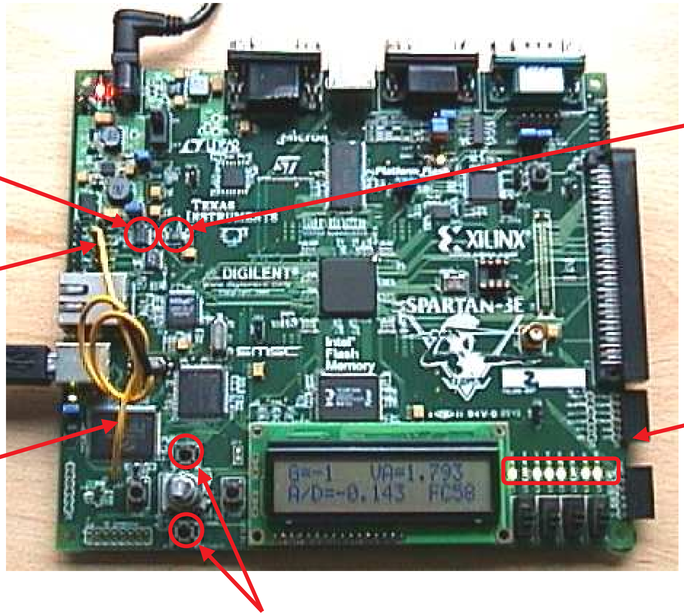
Any problems or items felt of value in the continued improvement of KCPSM3 or this reference design would be gratefully received by the author.

Ken Chapman
Senior Staff Engineer – Spartan Applications Specialist
email: chapman@xilinx.com

The author would also be pleased to hear from anyone using KCPSM3 or the UART macros with information about your application and how these macros have been useful.

Design Overview

This design outlines the fundamental operation of the Linear Technology LTC1407A-1 dual channel Analogue to Digital converter (A/D) and LTC6912-1 dual channel programmable amplifier provided on the Spartan-3E Starter Kit. The design illustrates how PicoBlaze can be used to control these devices as well as present results on the LCD display. It is hoped that the design may form the basis for future PicoBlaze designs as well as provide a general introduction to the analogue inputs. Some exercises are suggested to encourage further self study.



Linear Technology LTC6912-1 dual channel programmable amplifier

Linear Technology LTC1407A-1 dual channel Analogue to Digital converter

The design focuses on the **VINA** analogue input provided on J7. The design contains all the 'hooks' for controlling and reading the VINB channel but information about the VINA channel is presented on the LCD display.

8 LEDs
Simple binary count used to indicate design is working. Advances at 1 second intervals.

In this example a wire link is being used to connect VINA to the 1.8v supply test point J27. There are numerous test points which can be used to exercise the design.

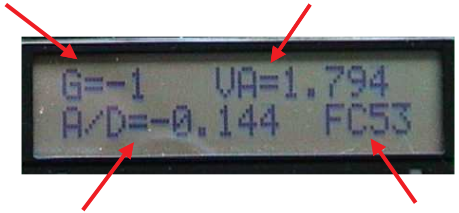
Use 'North' and 'South' buttons to adjust amplifier gain

Try it now – it only takes 30 seconds!

As well as the source design files, a compiled configuration bit file is provided which you can immediately download into the Spartan XC3S500E device on your board. It is recommended that you try this to become familiar with what the design does before continuing to read. To make this task really easy the first time, unzip all the files provided into a directory and then….

double click on 'install_picoblaze_amp_adc_control.bat'.

Assuming you have the Xilinx software installed, your board connected with the USB cable and the board powered (don't forget the switch), then this should open a DOS window and run iMPACT in batch mode to configure the Spartan-3E with the design.



Voltage level at the VINA input

Relative voltage level at the input to the A/D converter

Actual 14-bit value from A/D converter

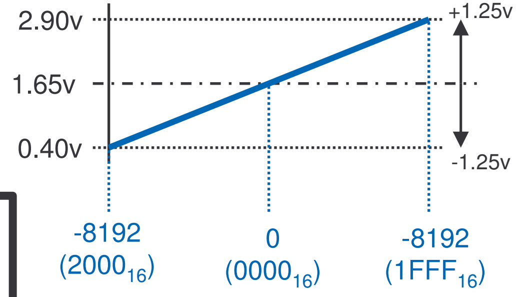
Analogue Inputs Overview

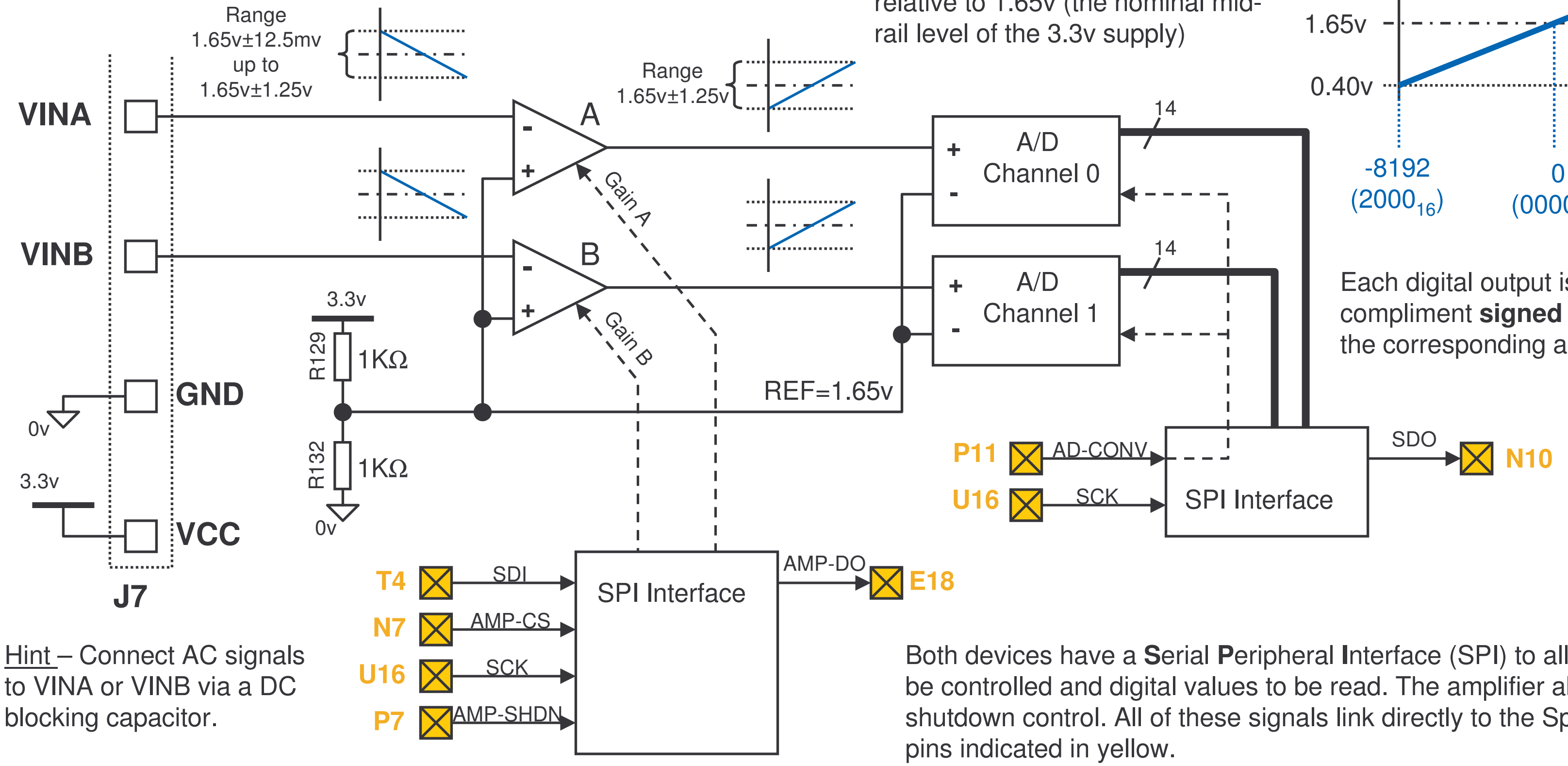
This overview is not intended to replace the Linear Technology data sheets and you are recommended to consult the LTC1407A-1 and LTC6912-1 data sheets to review the full range of features these devices offer as well as check the operating specifications particularly in relation to analogue inputs.

The LTC6912-1 provides two independent **inverting** amplifiers. Signals are amplified relative to 1.65v. The gain of each amplifier is programmable from -1 to -100 which enables signals as small as ±12.5mV to apply full scale inputs to the A/D converters.

The LTC1407A-1 provides two analogue to digital converters. Both analogue inputs are sampled simultaneously when the AD-CONV signal is applied.

The analogue input range is ±1.25v relative to 1.65v (the nominal mid-rail level of the 3.3v supply)





Each digital output is a **14-bit** twos compliment **signed** representation of the corresponding analogue input.

Hint – Connect AC signals to VINA or VINB via a DC blocking capacitor.

Both devices have a **S**erial **P**eripheral **I**nterface (SPI) to allow the devices to be controlled and digital values to be read. The amplifier also has a shutdown control. All of these signals link directly to the Spartan-3E using the pins indicated in yellow.

Important hint - The SPI bus signals (SDI, SDO and SCK) are shared by other devices on the board.
It is vital that other devices are disabled when communicating with the Amplifier or A/D converter.

Analogue Input Conversion

These two worked examples indicate the ideal response of the analogue inputs and derive the formula for this board. In practice, there will be slight variations in reference voltages and system noise which will make such precise results impossible to obtain.

Hint – 1.2v and 1,8v are nominal supply voltages available on the board at test points J24 (near on/off switch) and J27 (near BTN west). Use these with the design provided to compare results.

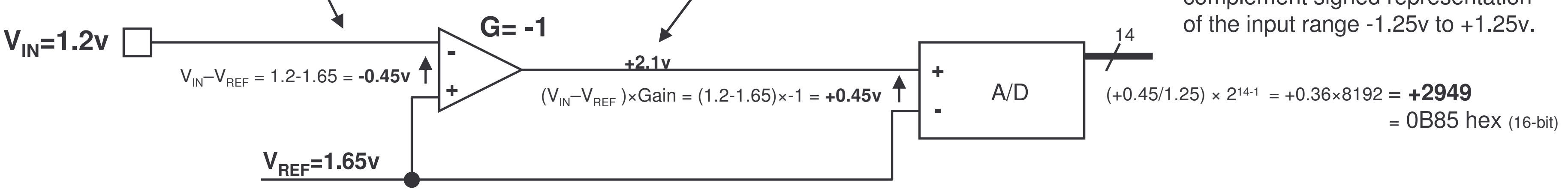
$$\text{Digital Output} = \frac{(V_{IN} - 1.65) \times \text{Gain}}{1.25} \times 8192$$

Where
Digital Output is 14-bit twos complement.
Gain is negative (inverting).
Available Gain figures are -1, -2, -5, -10, -20, -50 and -100.

Although the actual voltage being applied to the input is V_{IN} the amplifier observes this relative to the reference voltage. Therefore the input, as far as the amplifier is concerned, is $V_{IN}–V_{REF}$

As far as the input to the A/D converter is concerned, the output from the amplifier is $(V_{IN}–V_{REF})\times$Gain. However, do not allow yourself to be confused. If you measure this signal, you will find that it is higher by V_{REF}.

Digital value is a 14-bit twos complement signed representation of the input range -1.25v to +1.25v.



V_{IN}=1.2v

$V_{IN}–V_{REF}$ = 1.2-1.65 = **-0.45v**

G= -1

+2.1v

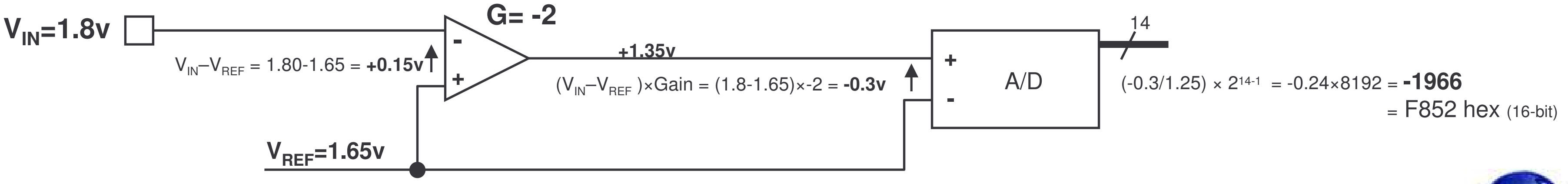
$(V_{IN}–V_{REF})\times$Gain = (1.2-1.65)×-1 = **+0.45v**

V_{REF}=1.65v

A/D

14

(+0.45/1.25) × 2^{14-1} = +0.36×8192 = **+2949**
= 0B85 hex (16-bit)



V_{IN}=1.8v

$V_{IN}–V_{REF}$ = 1.80-1.65 = **+0.15v**

G= -2

+1.35v

$(V_{IN}–V_{REF})\times$Gain = (1.8-1.65)×-2 = **-0.3v**

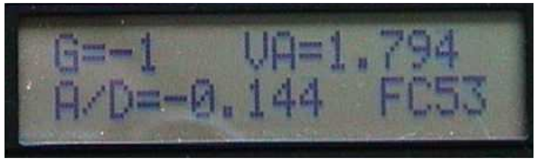
V_{REF}=1.65v

A/D

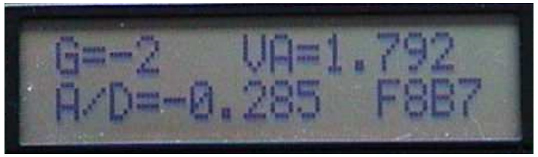
14

(-0.3/1.25) × 2^{14-1} = -0.24×8192 = **-1966**
= F852 hex (16-bit)

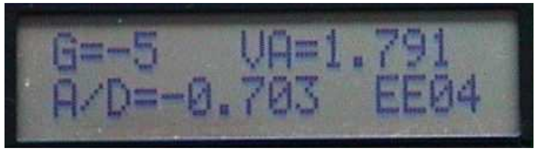
Typical Results



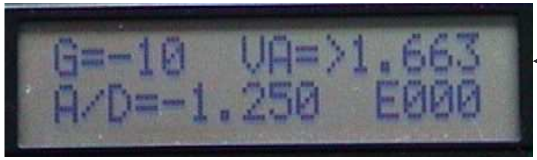
```
G=-1    VA=1.794
A/D=-0.144  FC53
```



```
G=-2    VA=1.792
A/D=-0.285  F8B7
```



```
G=-5    VA=1.791
A/D=-0.703  EE04
```



```
G=-10   VA=>1.663
A/D=-1.250  E000
```

These photographs of the LCD display were all taken whilst the VINA input was connected to the 1.8V test point at J27.

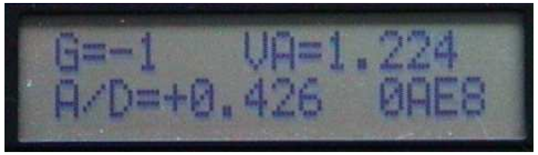
Since 1.8v is larger than the 1.65v reference level the input to the amplifier is positive (0.15v). Therefore the input to the A/D converter is negative relative to the 1.65v reference level. This negative value obviously becomes larger as the gain is increased.

Gain=-1 A/D input = 0.15v × -1 = -0.15v
Gain=-2 A/D input = 0.15v × -2 = -0.30v
Gain=-5 A/D input = 0.15v × -5 = -0.75v
Gain=-10 A/D input = 0.15v × -10 = -1.50v – Too big!

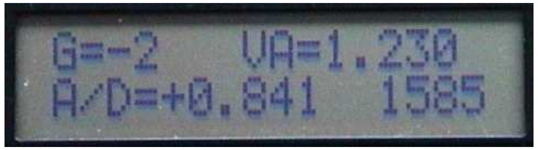
In practice we can see the values are not perfect. This will be due to the actual 1.8v supply and the 3.3v supply combined with R129 and R132 which form the 1.65v reference.

Exercise – Use a calibrated voltmeter to measure the reference level on your board (R129-R132) and modify the constants VREF_lsb and VREF_msb in the PicoBlaze program to increase accuracy.

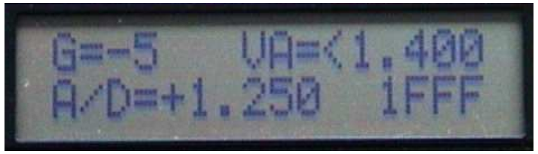
With the gain set to 10, the A/D input has exceed the negative limit of -1.25v and it is no longer possible to work out how high the actual VINA input is above the maximum it can measure. The display indicates this with the > sign.



```
G=-1    VA=1.224
A/D=+0.426  0AE8
```



```
G=-2    VA=1.230
A/D=+0.841  1585
```



```
G=-5    VA=<1.400
A/D=+1.250  1FFF
```

This sequence show the LCD display whilst the VINA input was connected to the 1.2V test point at J24.

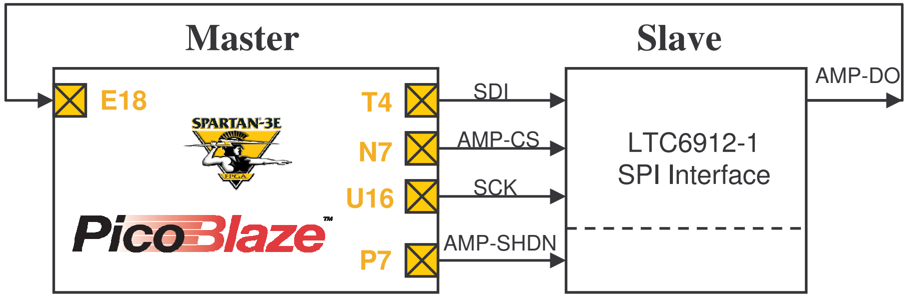
Since 1.2v is smaller than the 1.65v reference level the input to the amplifier is negative (-0.45v). Therefore the input to the A/D converter is positive relative to the 1.65v reference level and again becomes larger with increased gain.

Gain=-1 A/D input = -0.45v × -1 = 0.45v
Gain=-2 A/D input = -0.45v × -2 = 0.90v
Gain=-5 A/D input = -0.45v × -5 = 2.25v – Too big!

Exercise – The reference value on the board used in these photographs was measured and found to be 1.72v. Recalculate the values expected with a 1.2v input and also determine what the actual 1.2v supply was on this board.

Amplifier SPI Control

The **S**erial **P**eripheral **I**nterface (SPI) is formally described as being a full-duplex, synchronous, character-oriented channel employing a 4-wire interface. As each bit is transmitted by the master, the slave also transmits a bit allowing one byte to be passed in each direction at the same time. In this case the PicoBlaze in the Spartan-3E is the master and the SPI Amplifier is the slave.

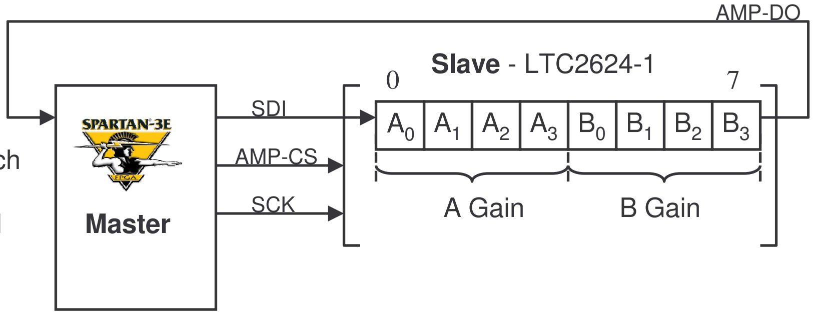


AMP-CS

Communication is only possible with the LTC6912-1 device when the select signal (AMP-CS) is Low. Therefore the PicoBlaze master is responsible for driving AMP-CS Low before transmitting and receiving a command byte and then driving AMP-CS High. It is the act of driving AMP-CS High which actually causes the amplifier to use the new gain setting for both channels.

Looking specifically at the LTC6912-1 Amplifier, each communication is formed of 1 byte or 8-bits. Inside the Amplifier, the SPI interface is formed by an 8-bit shift register. As a new 8-bit command byte is transmitted to it, the byte previously sent is echoed back to the master. In order to use the amplifier this response can be ignored, however, it is a useful to confirm correct communication is taking place and it is read back in the supplied PicoBlaze code.

PicoBlaze needs to transmit each byte most significant bit first. So bit 7 is transmitted and received first and bit 0 is transmitted and received last.



Note: AMP-DO is always active and therefore can not share the same SDO signals as other devices. It therefore has a separate input to the Spartan-3E which PicoBlaze reads.

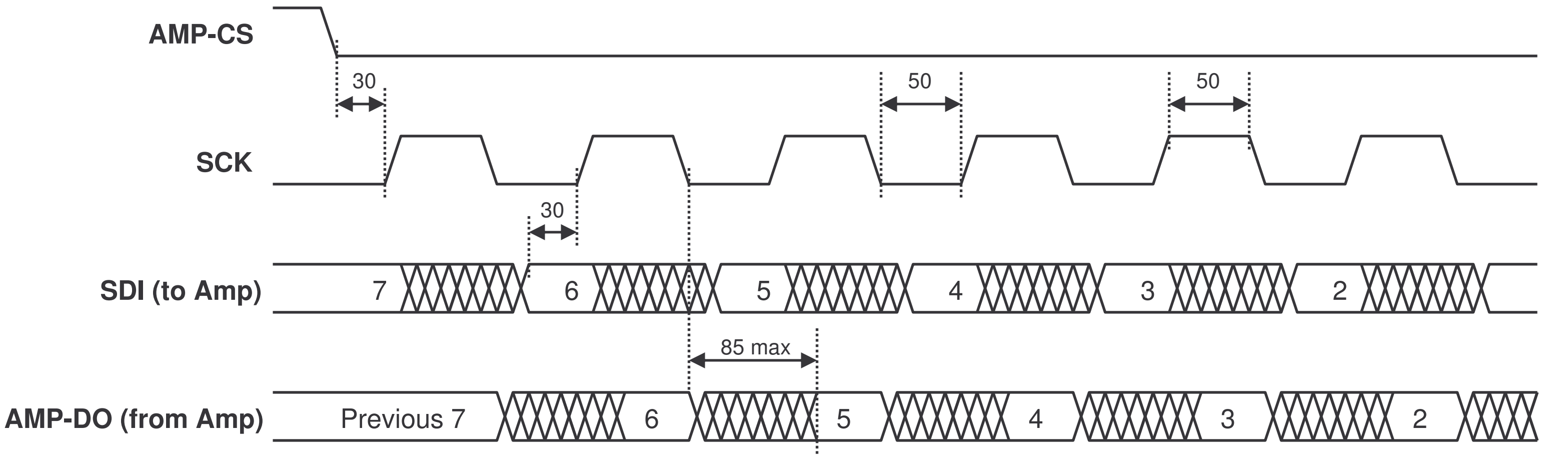
When all bits have been transmitted and the AMP-CS driven High, the amplifier interprets the bits in the shift register to set the gain of both amplifiers as described here.

A_3	A_2	A_1	A_0	
B_3	B_2	B_1	B_0	Gain
0	0	0	1	-1
0	0	1	0	-2
0	0	1	1	-5
0	1	0	0	-10
0	1	0	1	-20
0	1	1	0	-50
0	1	1	1	-100

Amplifier SPI Detail

Each bit is transmitted or received relative to the SCK clock. The system is fully static and any clock rate up to the maximum of ~10MHz supported by the LTC6912-1 is possible. Remember to check all timing parameters in the LTC2624-1 data sheet if you intend working at or close to the maximum speed.

Hint - As SPI interfaces go, it should be noted that the LTC2624-1 SPI interface is relatively slow. Even though the supplied PicoBlaze software based communication is also relatively slow, some care was required to ensure none of the timing specifications were violated. It would be very easy to violate them with a hardware based implementation. To emphasize the point, some key timing specification have been annotated on the timing diagram below.



All times are minimums in nano-seconds unless otherwise stated.

This timing diagram has been created approximately to scale assuming that the highest speed SCK is being used (minimum of 50ns Low and 50ns High). The LTC6912-1 captures data (SDI) on the rising edge of SCK, so the data needs to be valid for at least 30ns before the rising edge. The LTC6912-1 outputs data (AMP-DO) on the falling edge of SCK. This output may take up to 85ns, so if the AMP-D0 value needs to be read, then it is advisable to delay the reading of this signal as long as possible or operate at a slower clock speed. As the above diagram indicates, it is definitely not possible for the master to read AMP-DO using the next rising edge of SCK if the maximum clock rate is being used.

Note that bit 7, the MSB, is the first to be transmitted after AMP-CS is driven Low.
The previous value of bit 7 must be read before the first falling edge of SCLK is transmitted otherwise it will be missed.

Software Amplifier Communication

PicoBlaze is used to implement the SPI communication 100% in software. The Amplifier setting routine is shown below. The s2 register is used to set both the A and B amplifiers. On return, the s2 register contains the previous command byte value which can be used to verify the setting.

```
         set_amp: CALL SPI_init                  ;ensure known state of bus and s0 register
                  XOR s0, SPI_amp_cs             ;select Low on Amplifier chip select
                  OUTPUT s0, SPI_control_port
                  LOAD s1, 08                    ;8-bits to transmit and receive
next_amp_SPI_bit: OUTPUT s2, SPI_output_port     ;output data bit
                  XOR s0, SPI_sck                ;clock High (bit0)
                  OUTPUT s0, SPI_control_port    ;drive clock High
                  INPUT s3, SPI_input_port       ;read input bit
                  TEST s3, SPI_amp_sdi           ;detect state of received bit
                  SLA s2                         ;shift new data into result and move to next transmit bit
                  XOR s0, SPI_sck                ;clock Low (bit0)
                  OUTPUT s0, SPI_control_port    ;drive clock Low
                  SUB s1, 01                     ;count bits
                  JUMP NZ, next_amp_SPI_bit      ;repeat until finished
                  XOR s0, SPI_amp_cs             ;deselect the amplifier
                  OUTPUT s0, SPI_control_port
                  RETURN
```

AMP-CS Low

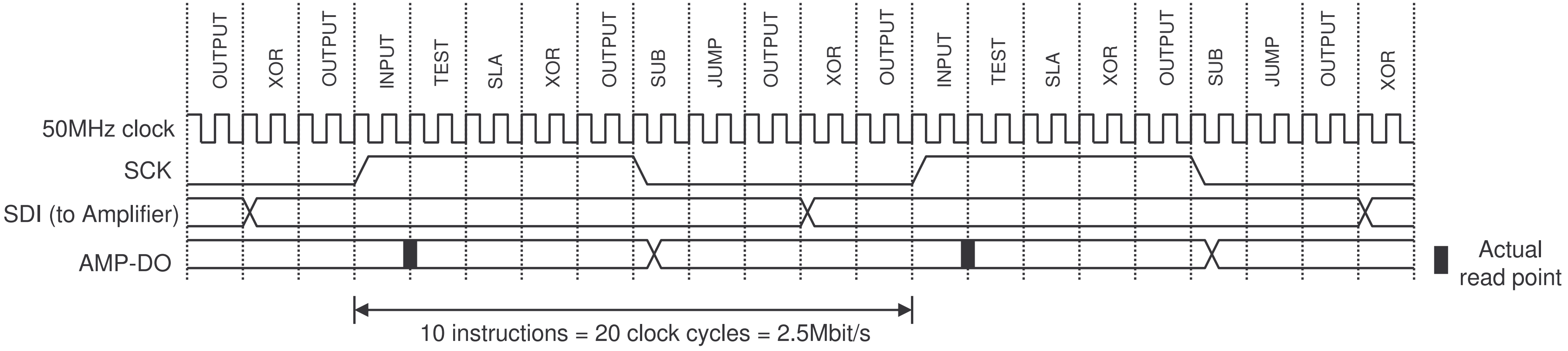
Serial Communication

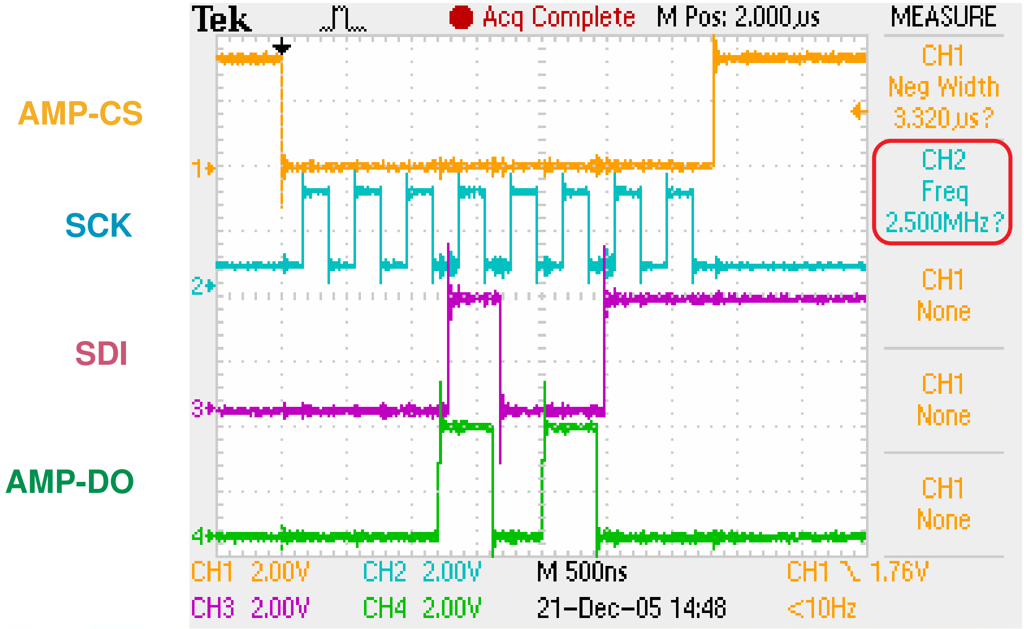
AMP-CS High

Note that the routine is reading and testing the AMD-D0 signal.

Amplifier Communication Timing

The routine generates SCK. Since every PicoBlaze instruction executes in 2 clock cycles and the design uses the 50MHz clock source on the board, the actual SPI bit rate can be determined. Although this is not as fast as the hardware can support, it keeps the design small and flexible.





Acq Complete   M Pos: 2.000µs

MEASURE

CH1 Neg Width 3.320µs?

CH2 Freq 2.500MHz?

CH1 None

CH1 None

CH1 None

CH1 2.00V   CH2 2.00V   M 500ns   CH1 ∖ 1.76V

CH3 2.00V   CH4 2.00V   21-Dec-05 14:48   <10Hz

This oscilloscope screen shot shows the SPI communication with the Amplifier. SDI and SCK were observed at the J12 connector and AMP-DO and AMP-CS were duplicated to J12 pins IO9 and IO10 to allow convenient connection of probes.

Hint – Always exploit programmable devices during development.

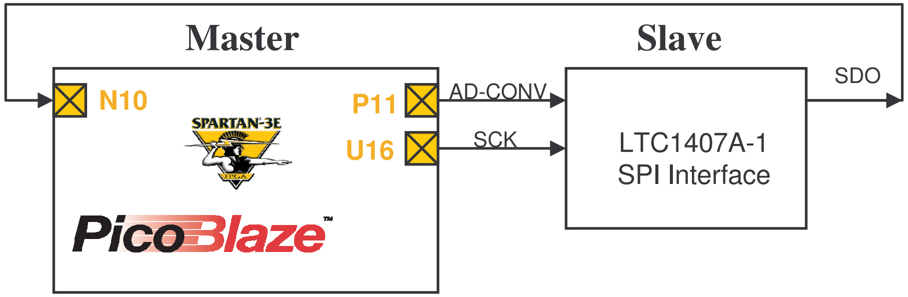
It can be seen that the SCK rate is the predicted 2.5MHz and the total communication time is approximately 3.3µs.

The B channel is defined by the first 4 bits transmitted and is being set consistently to '0001'. Therefore the first 4-bits of AMP-DO are also '0001'.

The A channel is being modified. '0011' is being set into the lower four bits and the previously set value of '0100' is being read back.

A/D SPI Control

It is probably fair to say that whilst the LTC1407A-1 device has a serial interface with a couple of similarly named pins to other SPI devices, it really is a little different to work with and it is useful that PicoBlaze can be programmed to accommodate the special requirements.

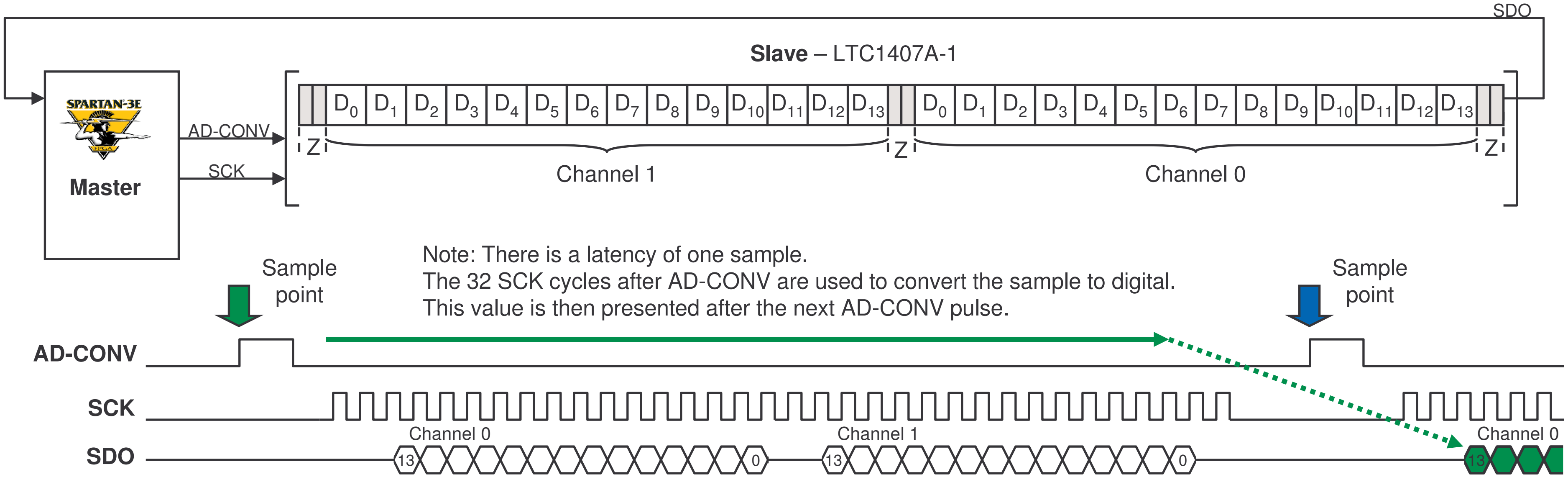


AD-CONV

The rising edge of this signal is used to trigger the sampling of the analogue inputs, to start the conversion and start the serial data transfer. This is really quite different to the chip select found on most SPI devices.

The accurate timing of regularly spaced AD-CONV pulses would be fundamental to any digital signal processing (DSP) algorithms but more sporadic sampling can be useful in monitoring applications.

A typical communication requires 34 cycles to be provided by the master following the rising edge of AD-CONV. Of these 34 cycles, 6 cycles result in SDO being driven tri-state (high impedance) as indicated by the grey boxes and timing diagram below. The remaining cycles transfer the two 14-bit signed values most significant bit first.

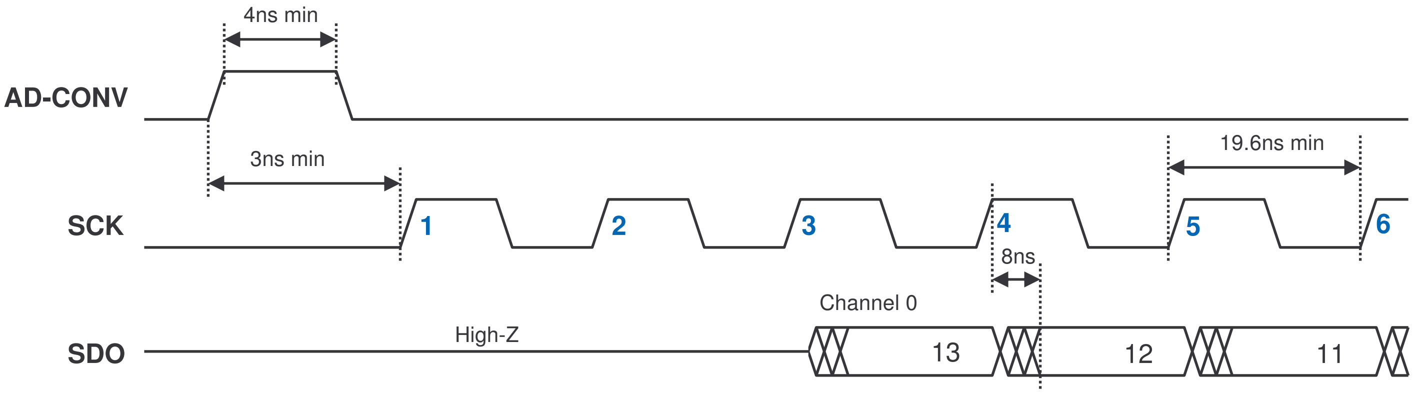


Note: There is a latency of one sample.
The 32 SCK cycles after AD-CONV are used to convert the sample to digital.
This value is then presented after the next AD-CONV pulse.

A/D SPI Detail

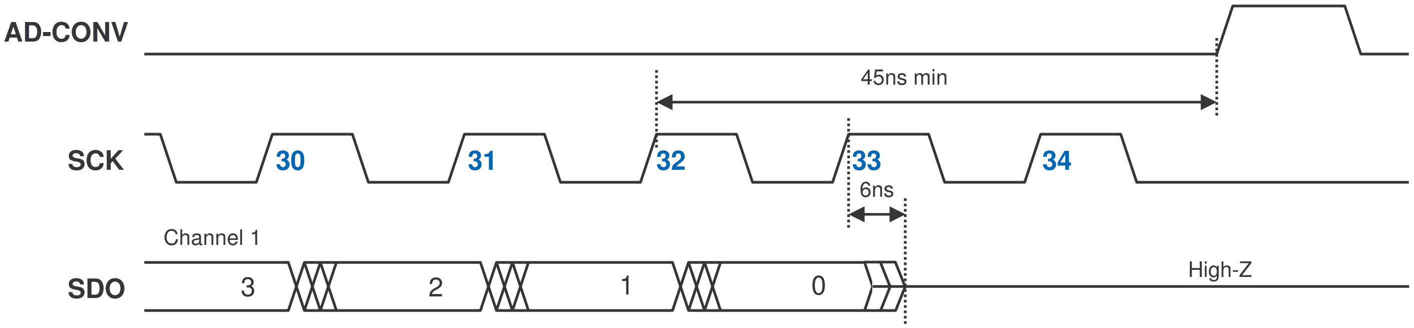
The SDO output of the LTC1407A-1 device changes as a result of the rising edge of the applied SCK clock. This again is different to most slave devices which change their output on the falling edge of SCK.

Since the device does not have a conventional chip select control, it is also vital that adequate SCK cycles are applied to ensure the SDO output is left in tri-state (high impedance) such that it can not interfere with other devices sharing the SPI bus. It is therefore advisable to use the typical 34 cycle sequence.



Note that SCK should also have a maximum period of 10µs otherwise the analogue to digital conversion process can not be guaranteed.

The 33rd SCK cycle after AD-CONV drives the SDO line back to high impedance state.



The maximum sample rate supported by the LTC1407A-1 is 1.5MHz. This is only possible if the maximum rate of SCK is used for conversion and communication.

Software A/D Communication

PicoBlaze is used to implement the SPI communication 100% in software. The A/D reading routine is shown below. The reading always results in samples being acquired for both channels. On return, the register pair [s9,s8] contains the value for Channel0 and the register pair [s7,s6] contains the value for Channel1.

```
    adc_read: CALL SPI_init                ;ensure known state of bus and s0 register
              XOR s0, SPI_adc_conv         ;Pulse AD-CONV High to take sample and start
              OUTPUT s0, SPI_control_port  ;  conversion and transmission of data.
              XOR s0, SPI_adc_conv         ;AD-CONV Low
              OUTPUT s0, SPI_control_port
              LOAD s1, 22                  ;34 clocks to read all data
next_adc_bit: XOR s0, SPI_sck              ;clock High (bit0)
              OUTPUT s0, SPI_control_port  ;drive clock High
              XOR s0, SPI_sck              ;clock Low (bit0)
              OUTPUT s0, SPI_control_port  ;drive clock Low
              INPUT s3, SPI_input_port     ;read input bit
              TEST s3, SPI_sdi             ;detect state of received bit
              SLA s6                       ;shift new data into result registers
              SLA s7
              SLA s8
              SLA s9
              SUB s1, 01                   ;count bits
              JUMP NZ, next_adc_bit        ;repeat until finished
              SRX s9                       ;sign extend 14-bit result in [s9,s8]
              SRA s8
              SRX s9
              SRA s8
              SRX s7                       ;sign extend 14-bit result in [s7,s6]
              SRA s6
              SRX s7
              SRA s6
              RETURN
```

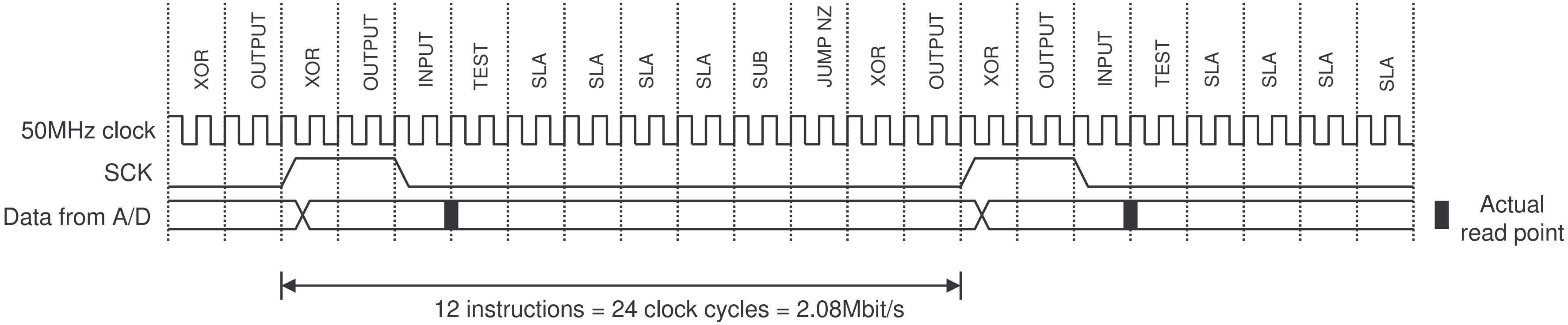
AD-CONV pulse High

Serial Communication

Right justify with sign extension the 14-bit values in the 16-bits provided by each register pair.

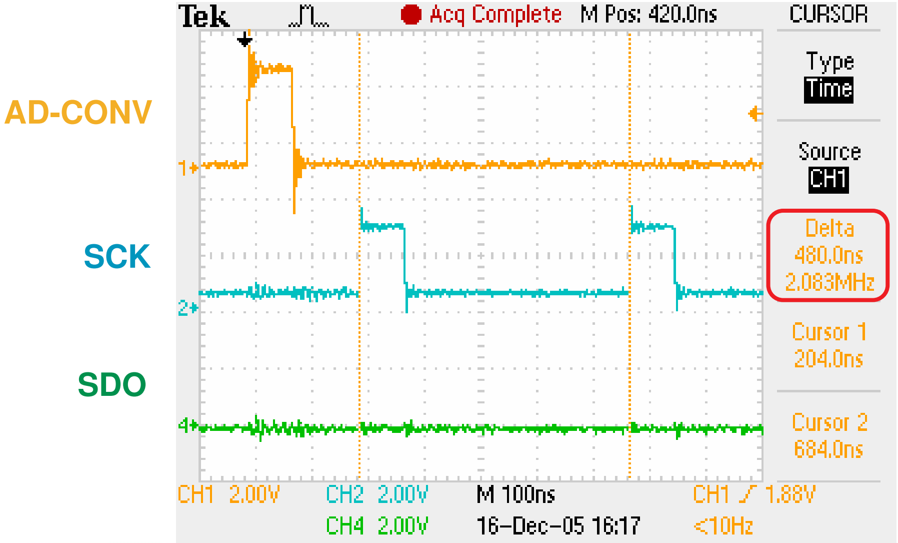
A/D Communication Timing

Since every PicoBlaze instruction executes in 2 clock cycles and the design uses the 50MHz clock source on the board, the actual SPI bit rate can be determined. Although this is not as fast as the hardware can support, it keeps the design small and flexible.

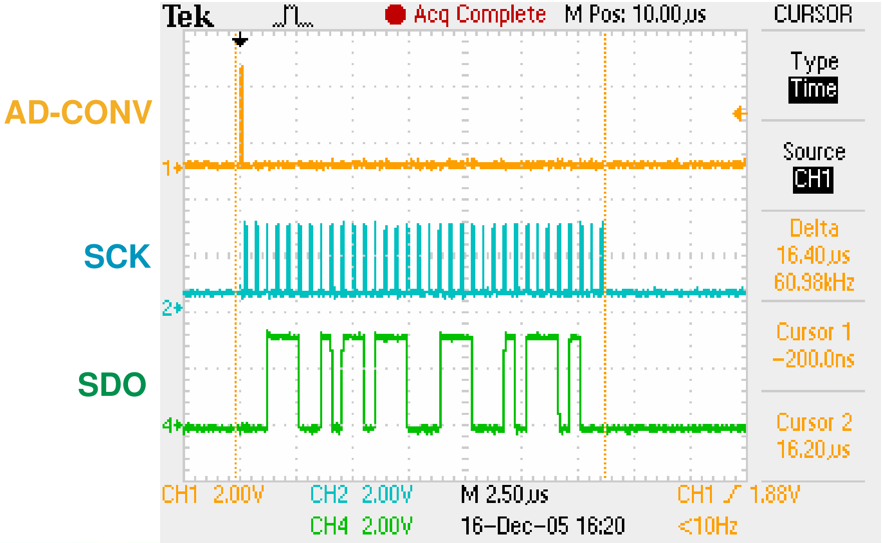


The following oscilloscope traces of the SPI signals observed at the J12 and J4-IO11 connectors shows the timing of the SCK and data signals.

SCK timing matches prediction.

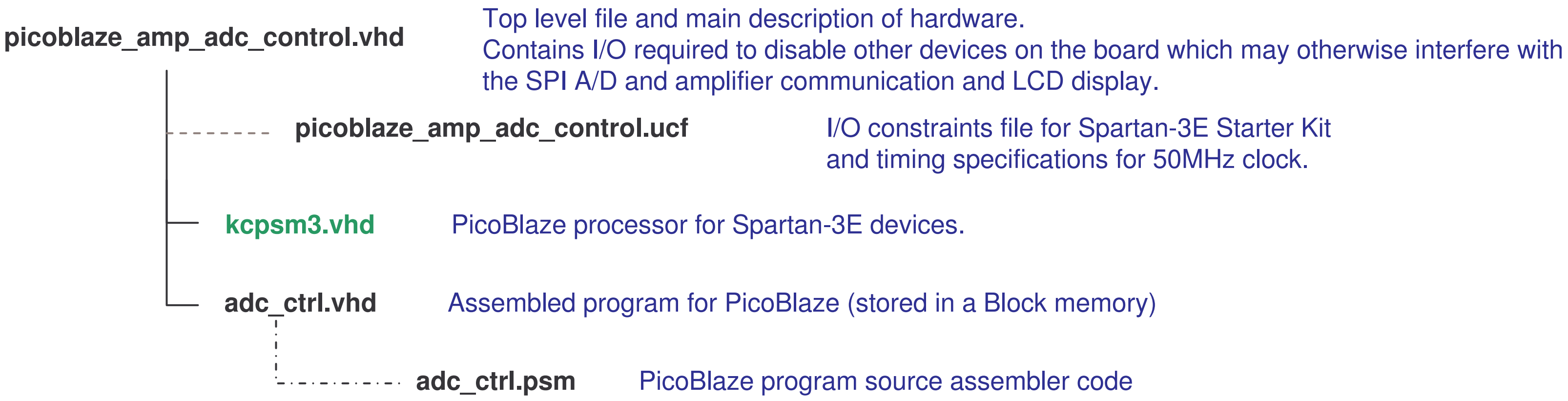


Complete read cycle takes just over 16µs.



Design Files

The source files provided for the reference design are…..



- **picoblaze_amp_adc_control.vhd** Top level file and main description of hardware. Contains I/O required to disable other devices on the board which may otherwise interfere with the SPI A/D and amplifier communication and LCD display.
  - **picoblaze_amp_adc_control.ucf** I/O constraints file for Spartan-3E Starter Kit and timing specifications for 50MHz clock.
  - **kcpsm3.vhd** PicoBlaze processor for Spartan-3E devices.
  - **adc_ctrl.vhd** Assembled program for PicoBlaze (stored in a Block memory)
    - **adc_ctrl.psm** PicoBlaze program source assembler code

Note: The file shown in **green** is not included with the reference design as it is provided with PicoBlaze download. Please visit the PicoBlaze Web site for your free copy of PicoBlaze, assembler, JTAG_loader and documentation.

www.xilinx.com/picoblaze

PicoBlaze Design Size

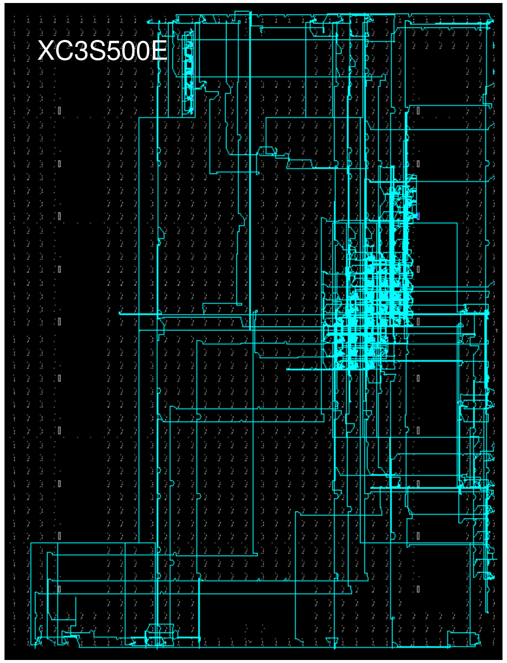
The images and statistics on this page show that the design occupies just 128 slices and 1 BRAM. This is only 2.8% of the slices and 5% of the BRAMs available in an XC3S500E device and would still be less than 14% of the slices in the smallest XC3S100E device.

MAP report

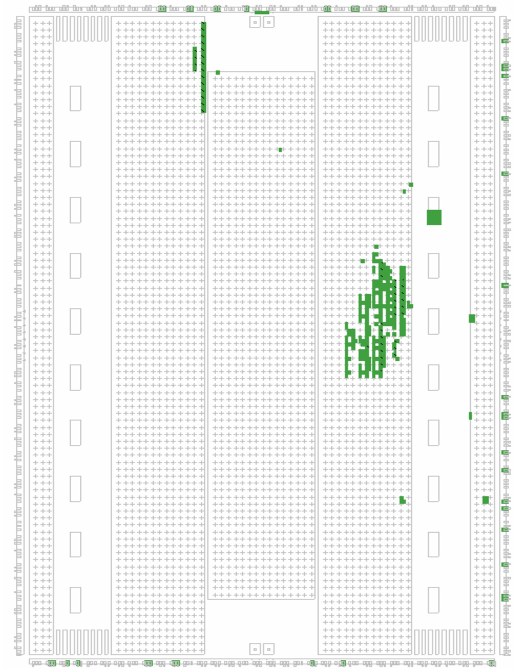
```
Number of occupied Slices:     128 out of    4,656     2%
Number of Block RAMs:            1 out of       20     5%

Total equivalent gate count for design:  76,260
```

PicoBlaze makes extensive use of the distributed memory features of the Spartan-3E device leading to very high design efficiency. If this design was replicated to fill the XC3S500E device, it would represent the equivalent of over 1.5 million gates. Not bad for a device even marketing claims to be 500 thousand gates ☺

FPGA Editor view

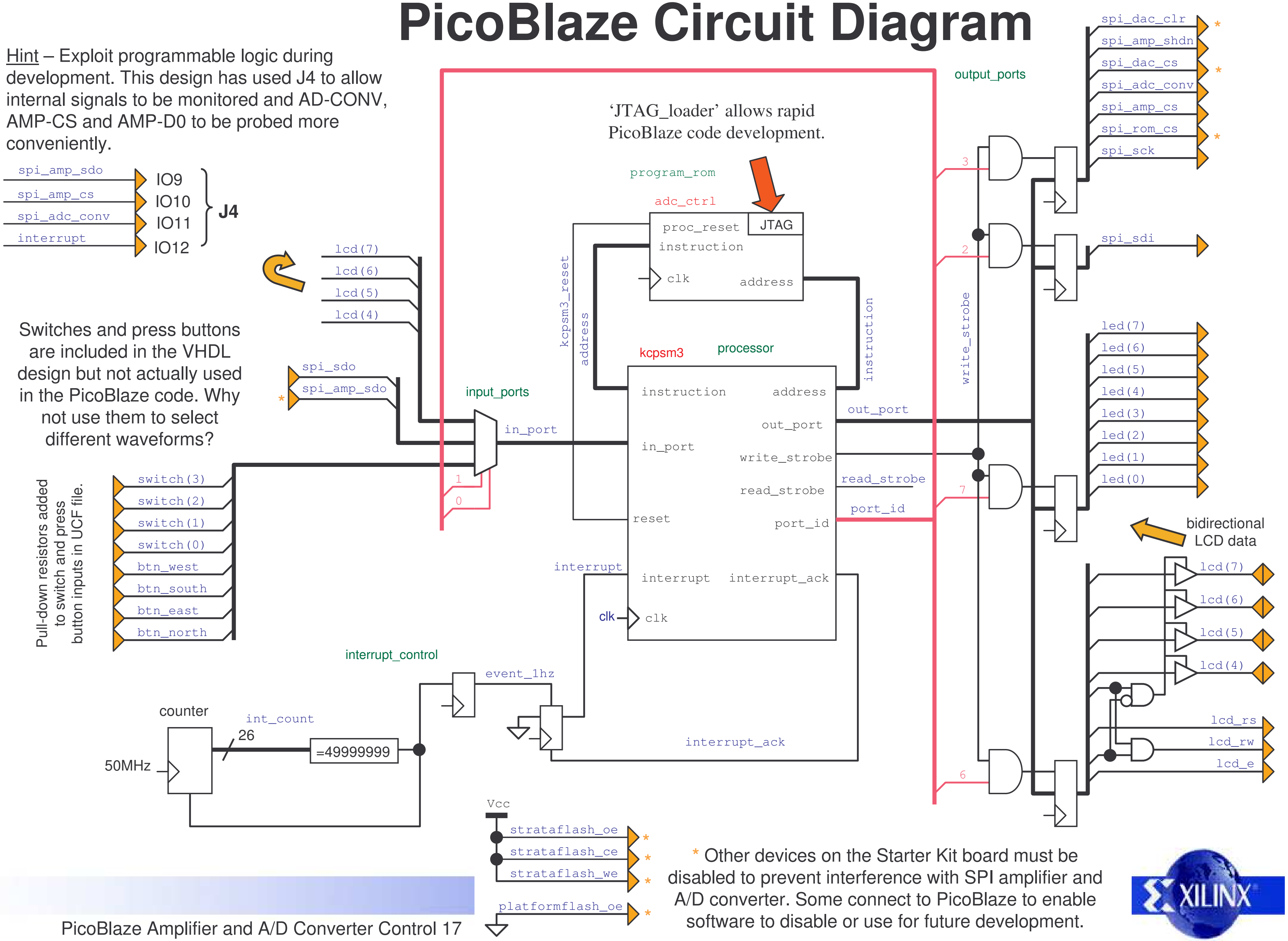


Floorplanner view



PicoBlaze Circuit Diagram

Hint – Exploit programmable logic during development. This design has used J4 to allow internal signals to be monitored and AD-CONV, AMP-CS and AMP-D0 to be probed more conveniently.



'JTAG_loader' allows rapid PicoBlaze code development.

Switches and press buttons are included in the VHDL design but not actually used in the PicoBlaze code. Why not use them to select different waveforms?

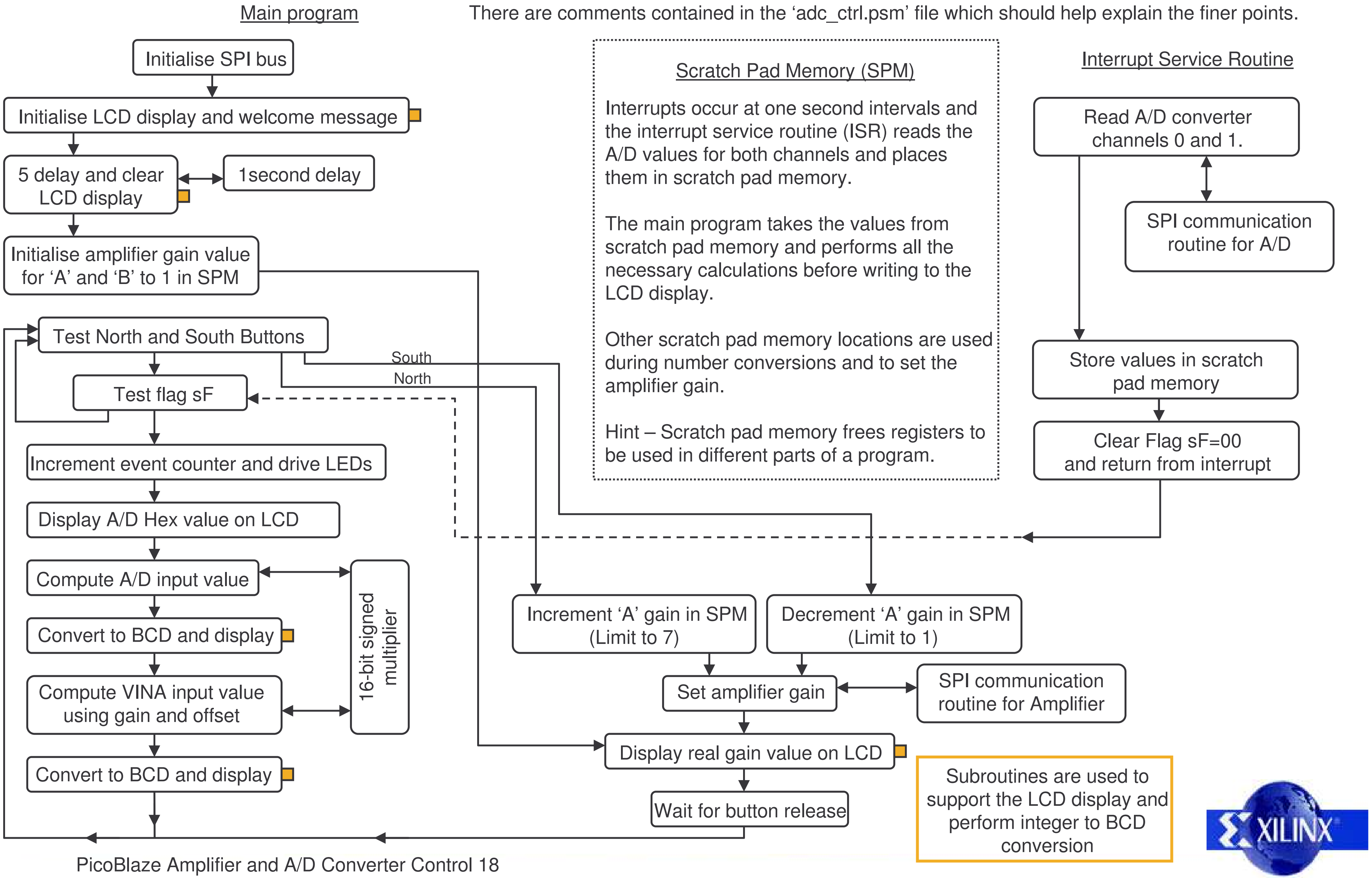
Pull-down resistors added to switch and press button inputs in UCF file.

* Other devices on the Starter Kit board must be disabled to prevent interference with SPI amplifier and A/D converter. Some connect to PicoBlaze to enable software to disable or use for future development.

PicoBlaze Program

This information is intended to give a guide to the way in which the PicoBlaze assembler code is organised. It is not intended to be a lesson in how to write assembler code or explain how PicoBlaze works. Please refer to the documentation for PicoBlaze (KCPSM3).

There are comments contained in the 'adc_ctrl.psm' file which should help explain the finer points.



Main program

Initialise SPI bus

Initialise LCD display and welcome message

5 delay and clear LCD display

1second delay

Initialise amplifier gain value for 'A' and 'B' to 1 in SPM

Test North and South Buttons

Test flag sF

Increment event counter and drive LEDs

Display A/D Hex value on LCD

Compute A/D input value

Convert to BCD and display

Compute VINA input value using gain and offset

Convert to BCD and display

16-bit signed multiplier

South

North

Increment 'A' gain in SPM (Limit to 7)

Decrement 'A' gain in SPM (Limit to 1)

Set amplifier gain

SPI communication routine for Amplifier

Display real gain value on LCD

Wait for button release

Scratch Pad Memory (SPM)

Interrupts occur at one second intervals and the interrupt service routine (ISR) reads the A/D values for both channels and places them in scratch pad memory.

The main program takes the values from scratch pad memory and performs all the necessary calculations before writing to the LCD display.

Other scratch pad memory locations are used during number conversions and to set the amplifier gain.

Hint – Scratch pad memory frees registers to be used in different parts of a program.

Interrupt Service Routine

Read A/D converter channels 0 and 1.

SPI communication routine for A/D

Store values in scratch pad memory

Clear Flag sF=00 and return from interrupt

Subroutines are used to support the LCD display and perform integer to BCD conversion