Application Note

Implementing the EV10AQ190A

1. Introduction

This application note aims at providing some recommendations to implement the EV10AQ190A Quad 10-bit 1.25 Gsps ADC in your system.

It first presents the ADC input/output interfaces and then provides some recommendations as regards the device settings and board layout to obtain the best performance of the device.

This document applies to the:

- EV10AQ190A Quad 10-bit 1.25 Gsps ADC

This document should be read with all other applicable documentation related to this part, (datasheet, errata sheet ,….).

For further information and assistance please contact **Hotline-bdc@e2v.com**

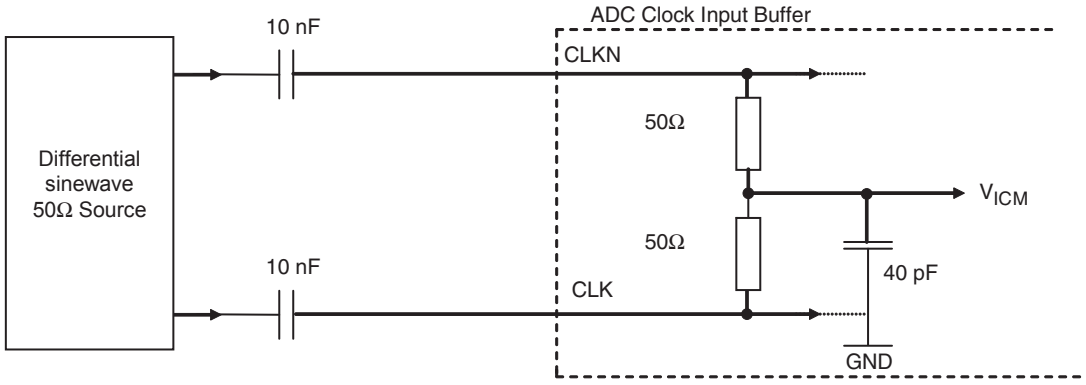
2. EV10AQ190A ADC Input Terminations

2.1 Clock Input

It is recommended to drive the Quad ADC input clock in a differential way in order to optimize the performance of the ADC and minimize the injection of noise in the die ground.

As the clock input common mode is 1.8V, it is recommended to AC couple the clock signal, as illustrated in Figure 2-1 on page 2.

Figure 2-1. EV10AQ190A ADC Clock Input Termination Scheme



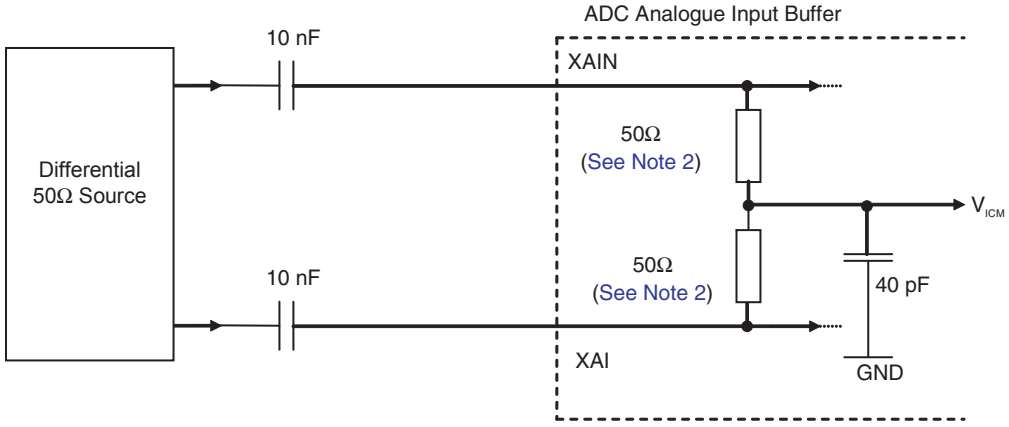
2.2 Analog Input

It is highly recommended to use the analog input (any channel) in differential mode (the performance will be decreased if the ADC is used in single-ended mode).

2.2.1 Differential Analog Input

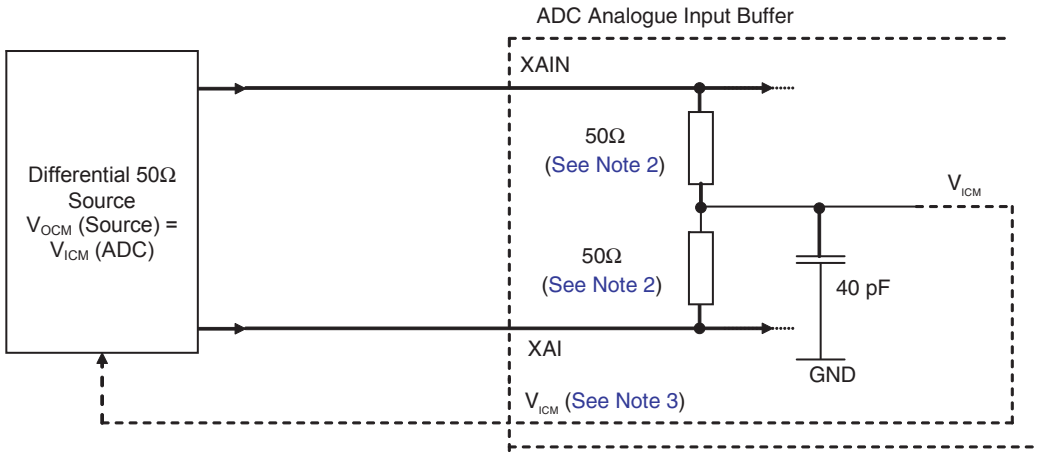
When used in differential mode, the analog input can be either DC or AC coupled as described in Figure 2-2 and Figure 2-3.

Figure 2-2. Differential Analog Input Implementation (AC Coupled)



Notes:
1. X = A, B, C or D.
2. The 50Ω terminations are implemented on-chip and can be fine tuned (TRIMMER register at address 0x13).
3. V_{ICM} = 1.6V.

Figure 2-3. Differential Analog Input Implementation (DC Coupled)



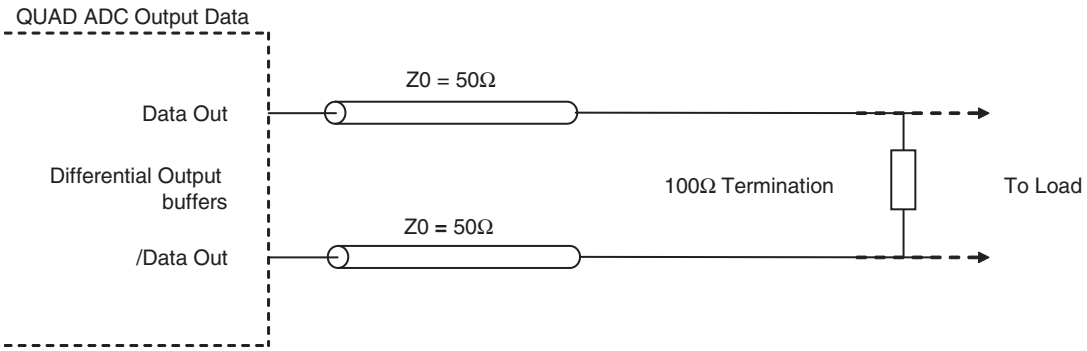
Notes:
1. X = A, B, C or D.
2. The 50Ω terminations are implemented on-chip and can be fine tuned (TRIMMER register at address 0x13).
3. V_{ICM} = 1.6V. The V_{ICM} signal is output from the ADC to provide the common mode to the front-end.

An example of an ADC driver which might be used is the ADA4960-1 from Analog Devices.

3. EV10AQ190A ADC Output Terminations

The digital outputs are LVDS compatible. They have to be 100Ω differentially terminated.

Figure 3-1. Differential Digital Outputs Terminations (100Ω LVDS)



4. EV10AQ190A ADC Hardware Signals

4.1 RESET Signals

4.2 ADC Synchronization Signal (SYNC, SYNCN)

The SYNC signal is used to reset the component and so this should be used in the synchronization scheme.

SYNC resets the internal timers and it also resets the test signal generation circuitry. It should be used in the following situations;

After power up or power configuration:- when switching the ADC from standby (full or partial) to normal mode.

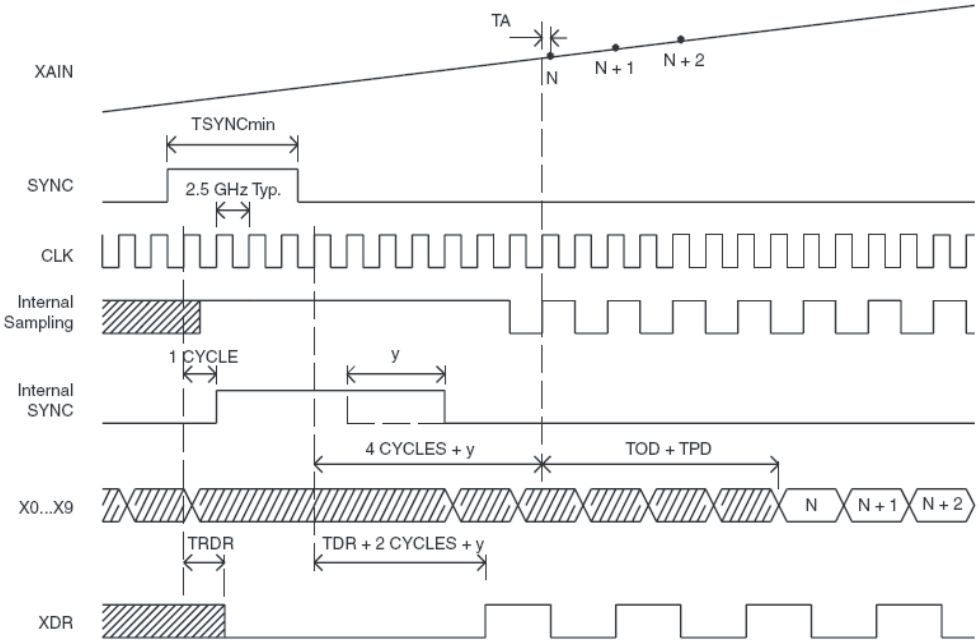
After channel mode configuration:- when switching the ADC from four-channel mode to one-channel mode.

For entering test sequence:- when switching the ADC from normal running mode to ramp or flashing mode. It is not needed when the ADC is switched from test mode (ramp or flashing), to normal running mode.

The timing functionality of the SYNC signal is shown in figure 1. From the rising edge of SYNC the Data Ready clocks return to zero and after the falling edge of SYNC the Data Ready clocks start to toggle, after a fixed delay.

There is a minimum period that the SYNC pulse should be active for correct operation of this function.

Figure 4-1. SYNC Timing in 4-Channel Mode, 1:1 DMUX Mode (For Each Channel)

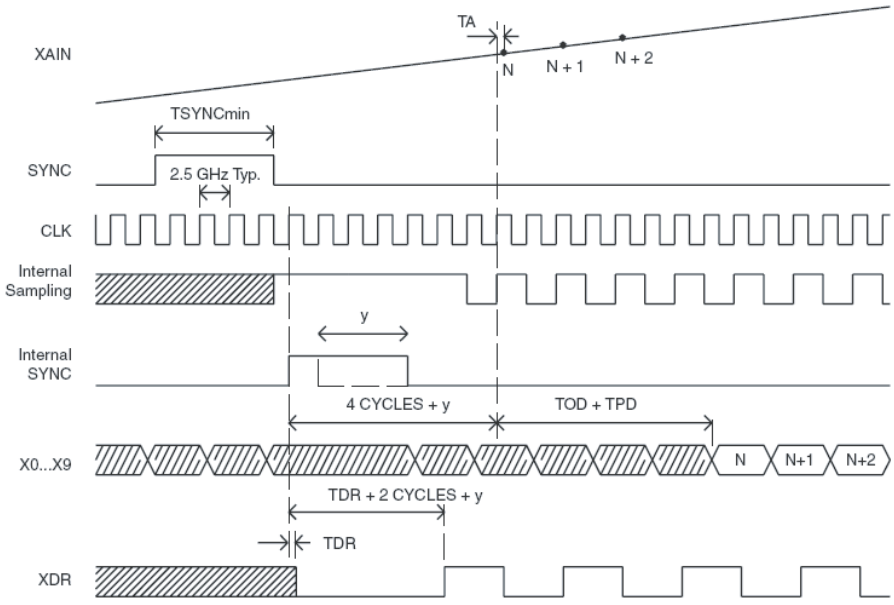


Note: X refers to A, B, C and D.

A new functionality for the version ‘A’ of the silicon which can be selected using the Control register address 0x01 bit 10, available using SPI, will enable a SYNC mode that will be timed to the falling edge of SYNC and will also not stop the Data Ready signal from toggling while SYNC is high.

The timing diagram for this mode (RM = 1) is shown below.

Figure 4-2. SYNC Timing in 4-Channel mode, RM = 1 1:1 DMUX mode (for each channel)



A delay y can be added to extend the time of the reset, this could be used to provide a delay in the output of good data from one ADC to another in a multi-ADC system.

4.3 Digital Reset (RSTN)

This is a global hardware Reset for the SPI core.

It is active Low.

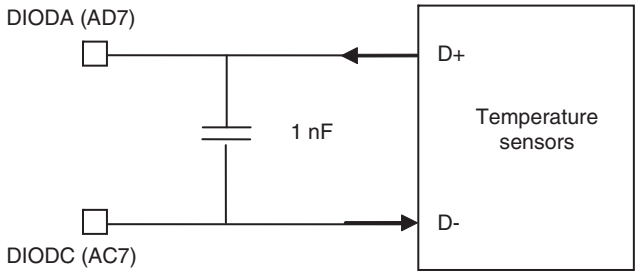
Note: There are 2 ways to reset the Quad 10-bit 1.25 Gsps ADC SPI register:
- by asserting low the RSTN primary pad (hardware reset)
- by writing a '1' in the bit SWRESET of the SWRESET register through the SPI (software reset)

Both ways will clear ALL configuration registers to their reset values.

4.4 Diode

DIODEA, DIODEC: two pins are provided so that the diode can be probed using standard temperature sensors.

Figure 4-3. Junction Temperature Monitoring Diode System



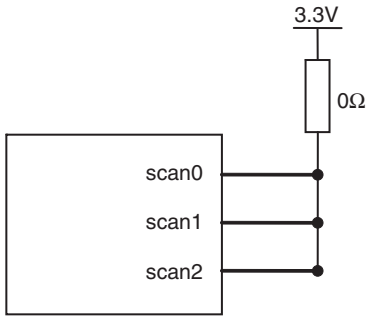
Standard temperature sensor interfaces are ADM1032 from ON SEMI or MAX642 from MAXIM.

Note: The diode ideality constant of 1.008 gives an accuracy of ±2°C if greater accuracy is required a calibration of the sensor interface should be made.

4.5 SCAN Signals

The scan signals (pins AD14, AC14, AD15) have to be connected to 3.3V as illustrated in Figure 5-1 on page 7.

Figure 4-4. Scan Signals Recommended Implementation



5. SPI

Atmel ATmega128L AVR can be used to drive the SPI port of the EV10AQ190A Quad 10-bit 1.25 Gsps ADC.

In this first section, a simple configuration for the interfacing of the AVR with the ADC is provided.

Note: All the information contained in this document concerning the AVR complies with the version available at the date the document was created. It should be checked versus the current version available before design.

5.1 EV10AQ190A 10-bit 1.25 Gsps ADC SPI

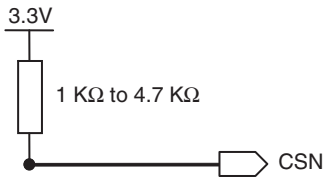
Five signals of the EV10AQ190A Quad 10-bit 1.25 Gsps ADC can be driven via the ATmega128L AVR:

- The CSN signal (pin AC16): used in the ADC to activate the 3-wire serial interface
- The SCLK signal (pin AD16): input clock for the SPI
- The MISO signal (pin AC17): master input slave output of the SPI (output of the ADC)
- The MOSI signal (pin AD17): master output slave input of the SPI (input for the ADC)
- The RSTN signal (pin AC15): external reset for the SPI.

5.1.1 CSN Signal

This signal should be pulled up to V_{CC} = 3.3V via a pull-up resistor (1 KΩ to 4.7 KΩ) if it is not use, so that it is not activated by default.

Figure 5-1. CSN Implementation is not use



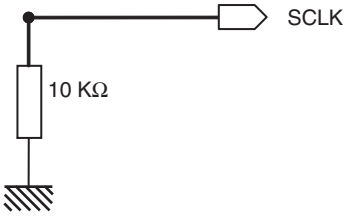
This signal could be connected directly to FPGA or Micro controller like FX2 or AVR if it is use.

Figure 5-2. CSN Implementation is use

5.1.2 SCLK Signal

This signal should be pulled down to ground via a 10 KΩ resistor if not use, so that it is not activated by default.

Figure 5-3. SCLK Implementation is not use



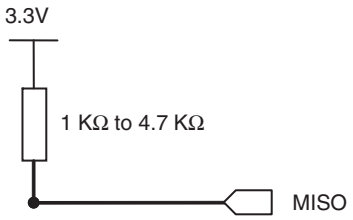
This signal could be connected directly to FPGA or Micro controller if it is used.

Figure 5-4. SCLK Implementation is use

5.1.3 MISO Signal

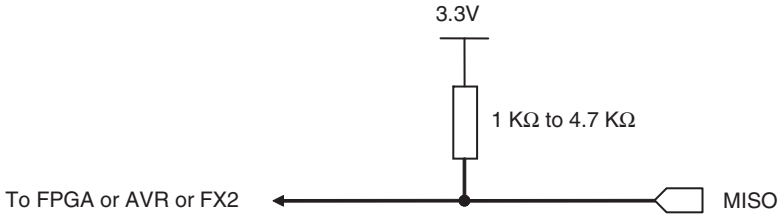
This signal must be pulled up to V_{CC} = 3.3V via a pull-up resistor (1 KΩ to 4.7 KΩ) **if it is used or not** so that it is not activated by default.

Figure 5-5. MISO Implementation if not use



This signal must be pulled up to V_{CC} = 3.3V via a pull-up resistor (1 KΩ to 4.7 KΩ) even if it is use.

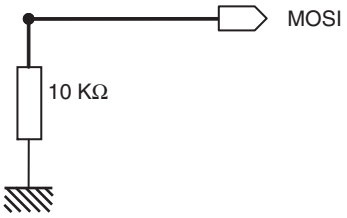
Figure 5-6. MISO Implementation if use



5.1.4 MOSI Signal

This signal should be pulled down to ground via a 10 KΩ resistor if it not used so that it is not activated by default.

Figure 5-7. MOSI Implementation



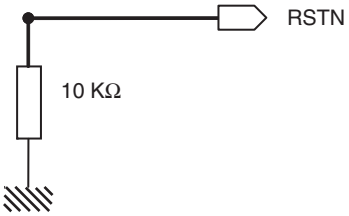
This signal could be connected directly to FPGA or Micro controller like FX2 or AVR if it is used.

Figure 5-8. MOSI Implementation

5.1.5 RSTN Signal

This signal should be pulled down to GND via a 10 KΩ resistor, if it is used so that it is activated by default and the default parameters will apply.

Figure 5-9. RSTN Implementation (Command via the AVR and via a Push Button)

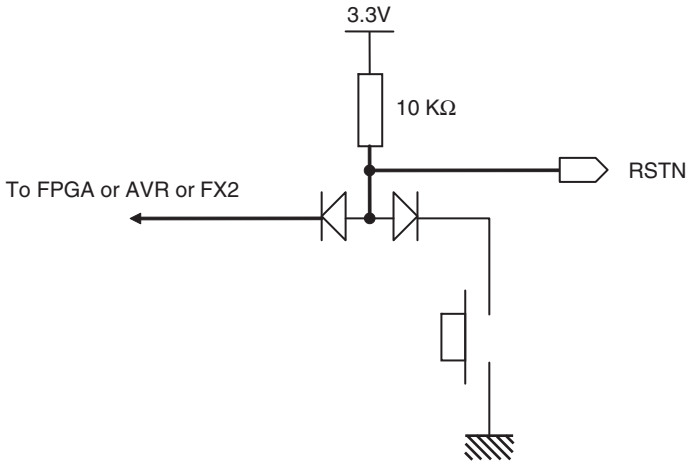


This signal could be connected directly to FPGA or Micro controller like FX2 or AVR if it is used.

Figure 5-10. RSTN Implementation

As this reset is normally a hardware reset, it can be useful to allow an external reset by a push button for example. In order to work out the conflict between the AVR and the push button, diodes can be used as illustrated in Figure 5-1 on page 7.

Figure 5-11. RSTN Implementation (Command via the AVR and via a Push Button)



5.2 ATmega128L 8-bit Microcontroller In-System Programmable Flash

Because Port B provides the pins for the SPI channel, this is the port chosen for the 4 signals of the ADC SPI as well as the RSTN signal:

- CSN -> PB4 (OC0 = Output Compare and PWM Output for Timer/Counter0)
- SCLK -> PB1 (SCK = SPI bus serial clock)
- MOSI -> PB2 (MOSI = SPI bus Master Output/Slave Input)
- MISO -> PB3 (MISO = SPI bus Master Input /Slave Output)
- RSTN -> PB5 (OC1A = Output Compare and PWM Output A for Timer/Counter1)

The other pins PB0 ($\overline{SS}$), PB6 (OC1B) and PB7 (OC2/OC1C) can be left floating (open).

Pin PB3 (MISO = SPI Bus Master Input/Slave Output) needs to be pulled up to 3.3V via a 1 KΩ resistor in order to be forced to a high level and not left open.

Pins SPICLOCK = PB1 and SPIDATA = PB2 need to be pulled down to ground via a 10 KΩ resistor to be forced to low level (inhibition of the SPI during reset of the microcontroller).

Pin SLE = PB4 (OC0 = Output Compare and PWM Output for Timer/Counter0) needs to be pulled up to 3.3V via a 3.3 KΩ (or 1 KΩ if the power consumption is not critical) resistor in order to protect the line during reset of the microcontroller (in which phase the signal becomes an input).

Ports A and C of the AVR can be left floating (open) but have to be internally configured with pull-ups.

For Port D, pins PD7, PD6, PD5 and PD4 can be left unused (open) but have to be internally configured with pull-ups. Pins PD3, PD2, PD1 and PD0 have to be pulled up to 3.3V via a 1 KΩ resistor in order to inhibit external interrupts.

For port E, pins PE3 and PE2 can be left unused (open) but have to be internally configured with pull-ups. Pins PE7, PE6, PE5 and PE4 have to be pulled up to 3.3V via a 1 KΩ resistor in order to inhibit external interrupts.

PE1 and PE0 can be used as the Programming Data Output (TX) and Input (RX) to be connected to the TX and RX of the system (in the case of the EV10AQ190x-EB evaluation board, these signals are sent to the PC via an RS232 port).

All the pins of Port F have to be connected to ground so that they are in a known fixed state (no internal pull-down available for these pins).

All pins of Port G can be left floating (open).

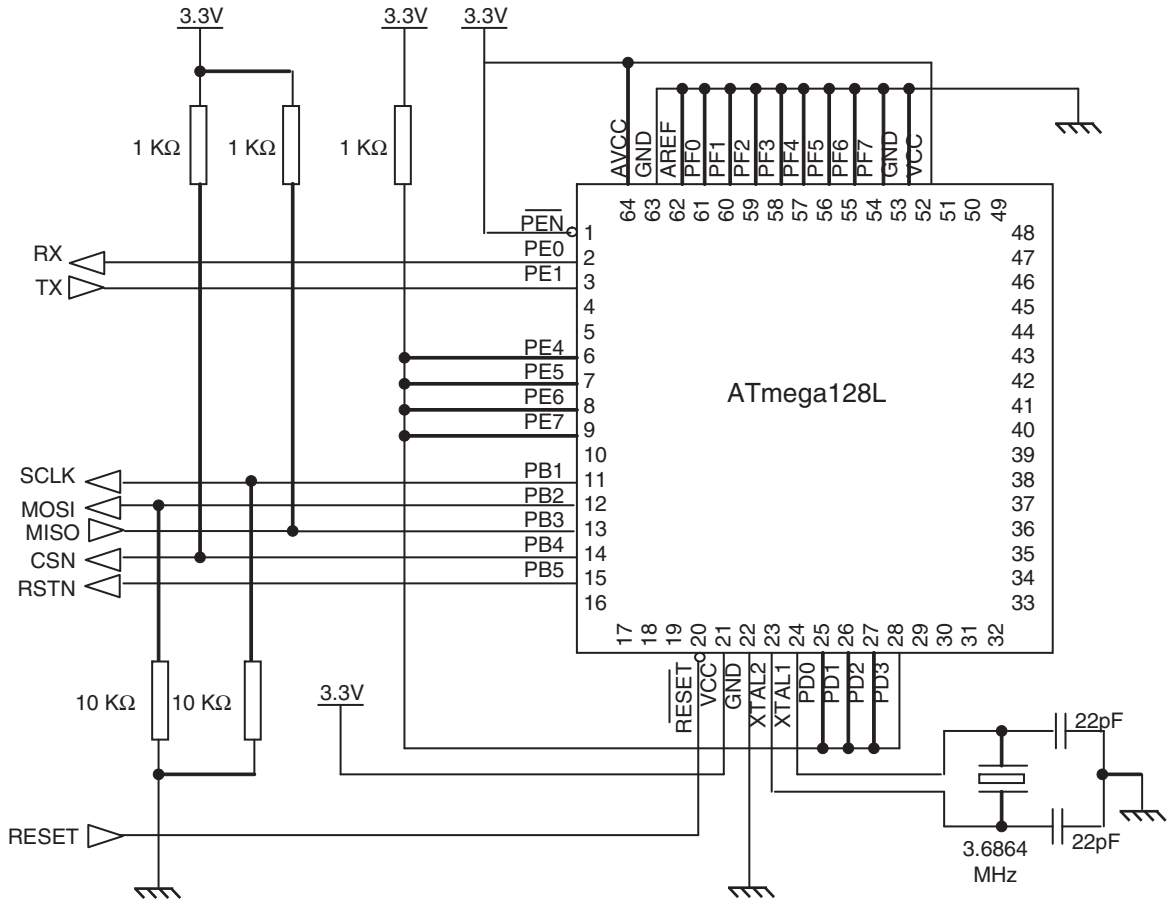
Finally, the five remaining signal pins are to be connected as follows:

- $\overline{PEN}$ = programming enable pin for the SPI serial programming mode, to be connected to V_{CC} = 3.3V to activate the SPI programming mode
- $\overline{RESET}$ = Master reset of the AVR, to be connected to a microcontroller supervisory circuit (for example and for information only: MCP809 from Microchip™; one possible configuration is given in the next section)
- XTAL1 and XTAL2: input and output to/from the inverting Oscillator amplifier
- AREF = analog reference for the A/D internal converter

Finally, V_{CC} and AV_{CC} have to be connected to a 3.3V source and GND, to ground.

This gives the following configuration (AVR only):

Figure 5-12. ATmega128L Application Diagram (for use with Atmel EV10AQ190A Quad 10-bit 1.25 Gsps ADC)



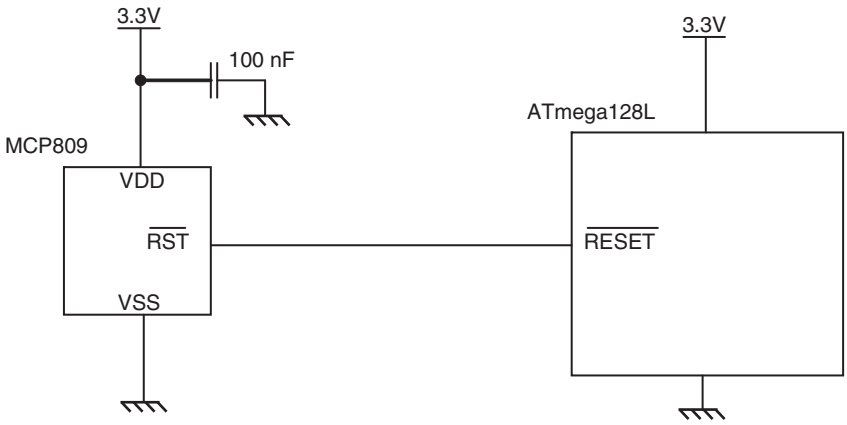
Note: Only the connected pins are shown (the unused pins are left open).

The reset of the ATmega128L AVR can be controlled thanks to a "voltage supervisory circuit" comparable to the MCP809 device from Microchip™ (for information only). Such a device allows you to keep the microcontroller in reset until the system voltage has reached its final level. It also ensures that the microcontroller will be reset whenever a power drop occurs.

Any voltage supervisory circuit compliant with V_{CC} = 3.3V and with a reset pulse longer than 50 ns minimum width (active low) would work.

Here the supervisory device from Microchip™ reset voltage level is set to 3.0V with a pulse of 350 ms.

Figure 5-13. Typical Application Diagram for the $\overline{\text{RESET}}$ circuit



5.2.1 Programming of Atmel ATmega128L AVR

Atmel ATmega128L AVR can be programmed thanks to the AVR ISP (In-System Programmer) tool using AVR Studio , Atmel's Integrated Development Environment (IDE) for code writing and debugging. The programming software can be controlled from both Windows environment and a DOS command-line interface.

For more information on the AVR Studio programming software, please refer to Atmel Website.

The programming of the AVR requires the use of a 6- or 10-pin ISP connector.

In our case, an HE10 6-pin connector is chosen:

- pin 1 = PDO, AVR Programming Data Out
- pin 2 = AVR Target application card power supply (= 3.3V)
- pin 3 = SCK, AVR programming clock
- pin 4 = PDI, AVR programming Data In
- pin 5 = RST_ISP, AVR programming Reset
- pin 6 = ground

Notes: 1. The ISP card power supply comes from the AVR card (3.3V). There is no need for an additional power supply.
2. The mode used to program the AVR is a serial mode.

The RST_ISP signal is used to manage the AVR mode: programming mode or SPI mode.

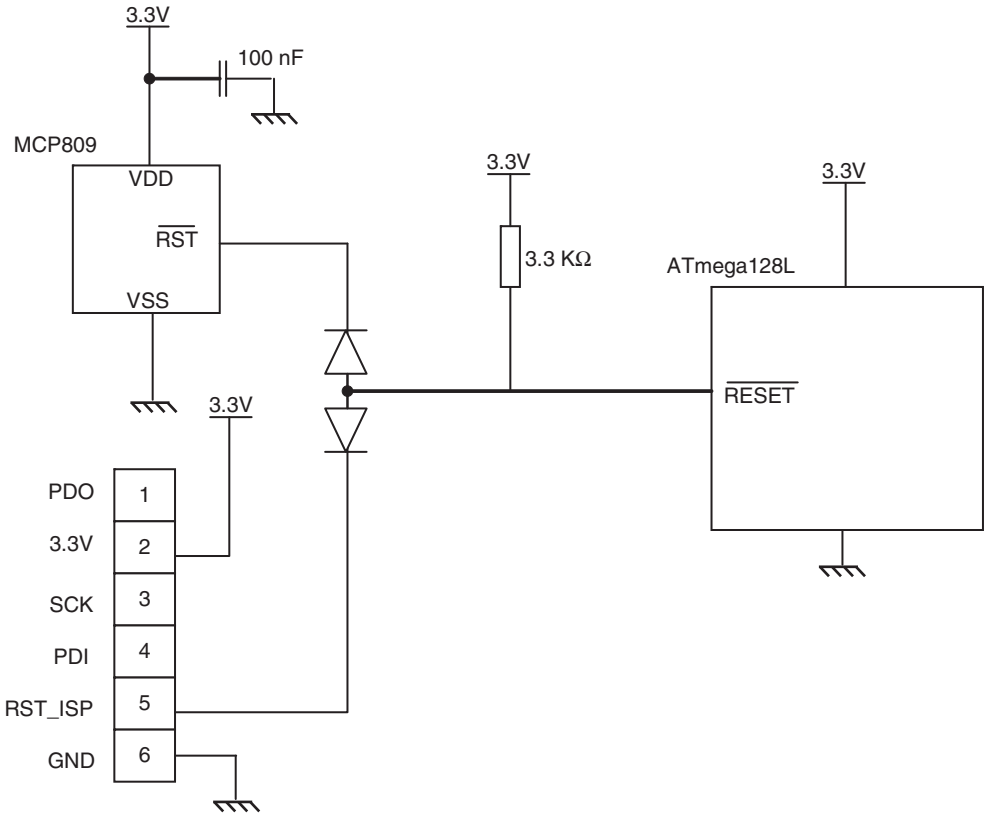
This signal is sent to the $\overline{\text{RESET}}$ of the AVR so that:

- when RST_ISP = 0, $\overline{\text{RESET}}$ = 0 and the AVR is in reset (ISP mode), PE0 is used as the Data In for the programming of the AVR, PE1 is the Data Out and PB1 is the programming clock,
- when RST_ISP = 1, $\overline{\text{RESET}}$ = 1 and the AVR is in normal mode, PE0 = RX, PE1 = TX, PB1 = SCLK.

The three AVR signals mentioned previously (PE0, PE1 and PB1) thus have 2 functions, controlled by RST_ISP. Be careful when implementing these signals (series resistors on the SCK, PDO and PDI data may be needed to manage possible conflicts - see next section).

Similarly, the $\overline{\text{RESET}}$ signal has two possible sources: the signal generated by the microcontroller supervisory device and the RST_ISP signal from the ISP. In order to manage this signal and in case the microcontroller supervisory device is not with open collector (as for the MCP809 device), two head-to-tail diodes are required, as illustrated in Figure 5-10 on page 9. The line going to the $\overline{\text{RESET}}$ signal of the AVR is then in open-collector and a pull-p resistor (3.3 KΩ) to 3.3V is required.

Figure 5-14. Typical Application Diagram for the $\overline{\text{RESET}}$ Circuit with the ISP Connector

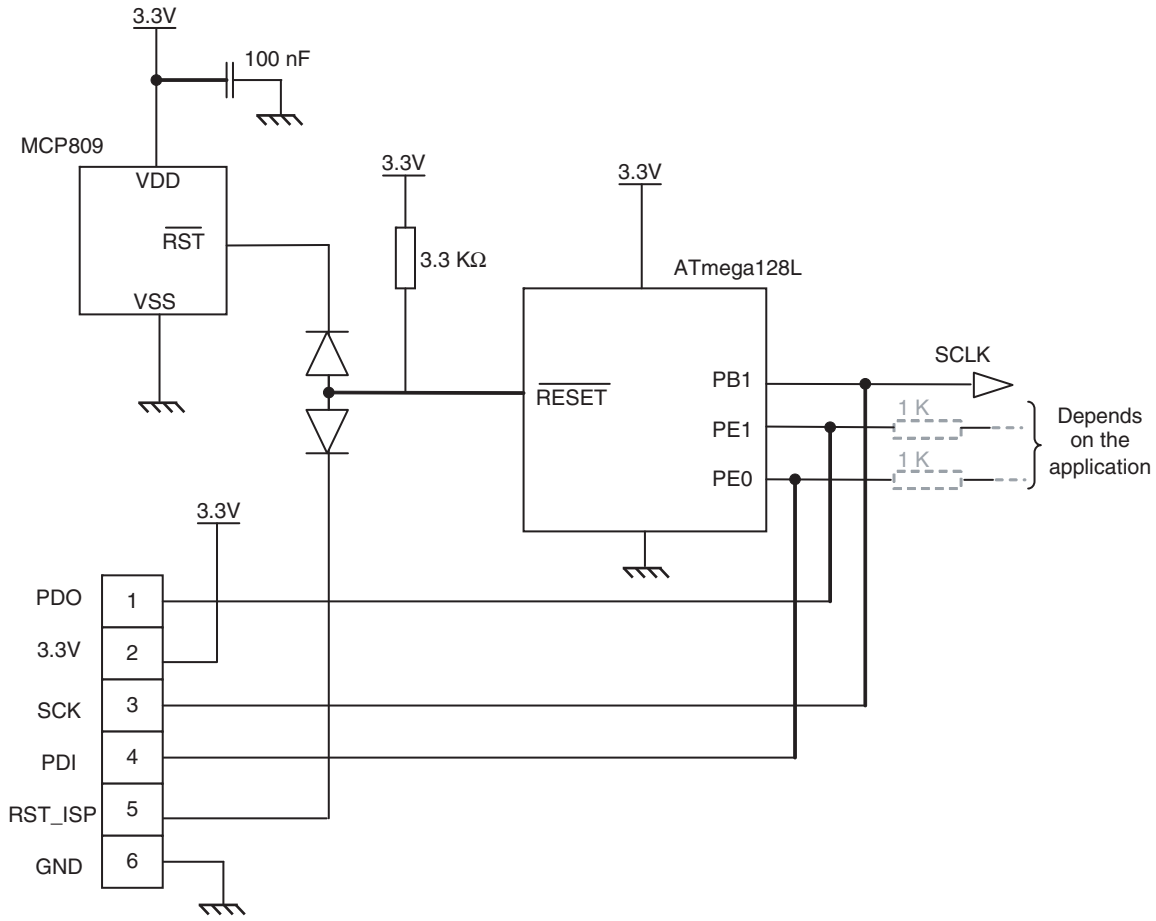


A basic diagram illustrating the interface between the ISP connector and the AVR is depicted in Figure 5-11 on page 9.

In this general case, PE1 and PE0 interconnections are left to the user's responsibility. In case of possible conflict on these signals (example PE1 could be driven by both PDO and another signal), it may be necessary to add a 1 KΩ resistor in series so that any voltage difference will be dissipated in this resistor.

No additional protection is required on the AVR PB1 signal as there is no conflict between SCLK and SCK. It is nevertheless recommended to set the ADC in standby mode or disable the SPI thanks to the CSN bit during programming of the AVR.

Figure 5-15. General Application Diagram for the ISP Connector and the AVR



In case the RX and TX signals are to be connected to a transceiver (RS232 connector to a PC for example), in order to multiplex the signals of the AVR (PE0, PE1 and PB1) between the ISP and the RX, TX and SCLK signals, a low voltage buffer/line driver with 3-state outputs device can be used. The 74LVQ241 devices are well-suited for this application (clock driver and bus oriented transmitter or receiver).

The 74LVQ241 device has 8 inputs and 8 corresponding outputs and two 3-state Output enable inputs. These two 3-state output enable inputs can be managed by the RST_ISP signal:

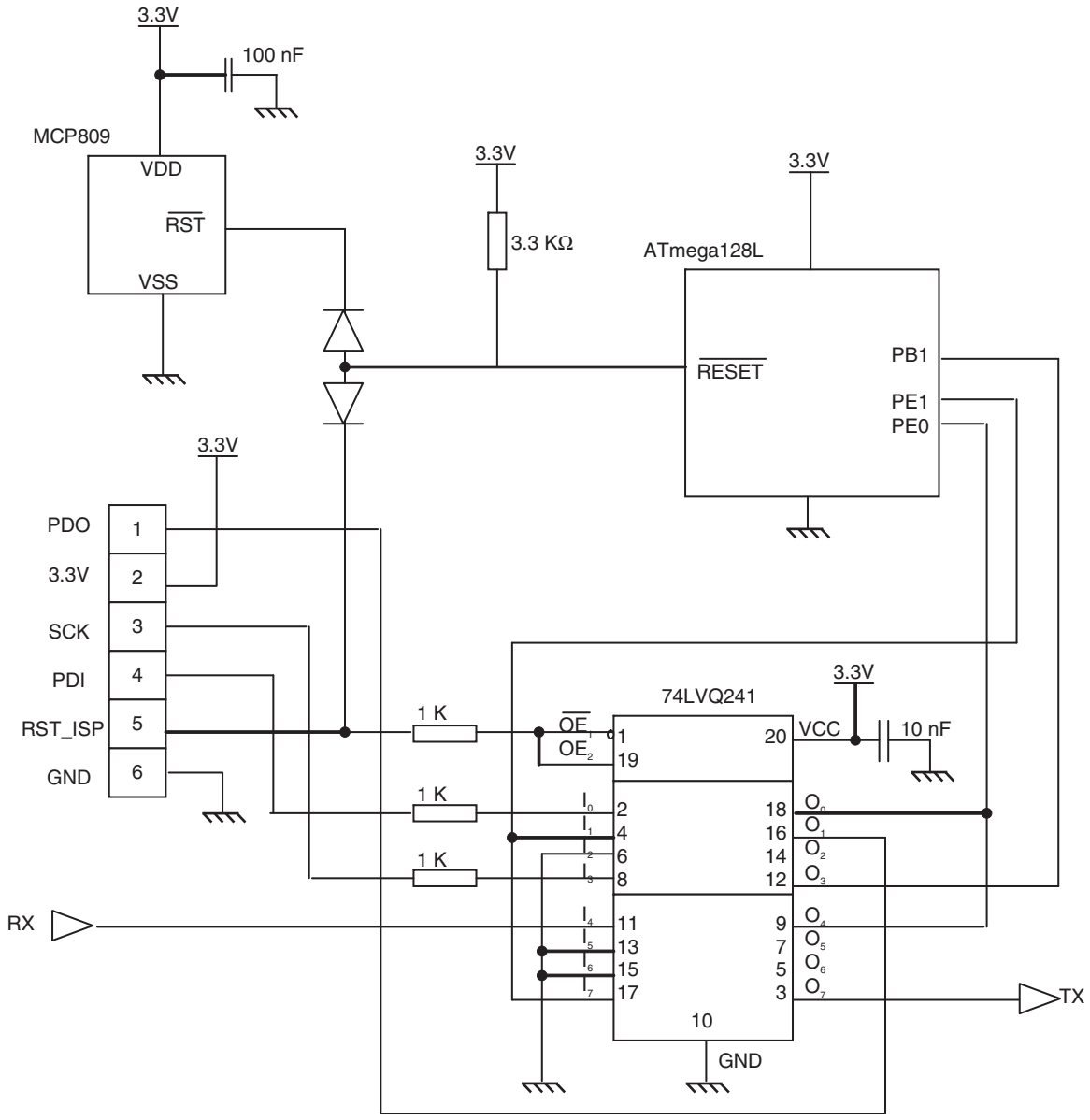
- when RST_ISP = 0, $\overline{OE1}$ = OE2 = 0 and then O_0 to O_3 are low and O_4 to O_7 are in high impedance
- when RST_ISP = 1, $\overline{OE1}$ = OE2 = 1 and then O_0 to O_3 are in high impedance and O_4 to O_7 are low.

The truth table of the 74LVQ241 device is shown in Table 5-1.

Table 5-1. 74LVQ241Truth Table

Inputs		**Outputs**
$\overline{OE1}$	I_n	**(O_0, O_1, O_2, O_3)**
L	L	L
L	H	H
H	X	Z
Inputs		**Outputs**
OE2	I_n	**(O_4, O_5, O_6, O_7)**
L	X	Z
H	L	L
H	H	H

Figure 5-16. Typical Application Diagram with the 74LVQ241



Notes:
1. The unused inputs are connected to ground to prevent them from toggling.
2. $\overline{OE1}$ and OE2 are connected together and to RST_ISP via a 1 KΩ resistor.
3. SCK, RST-ISP and PDI are connected to I_3, $\overline{OE1}$ and OE2 and I0 respectively via 1 KΩ resistors in order to manage the possible conflicts on the signals in case the connector is used to program several AVRs.
4. PE0 is connected to both O_0 and O_4, which are respectively the inputs corresponding to SCK and RX: PE0 will be either generated by SCK or RX depending on the mode.
5. PE1 is connected to both I_1 and I_7, which are respectively the outputs corresponding to PDO and TX: PE1 will either generate by PDO or TX depending on the mode.

The programming of the AVR itself as well as the connections of the RX and TX signals is not described in this application note as they depend on the final application.

For more information on the AVR, please contact the AVR hotline at avr@atmel.com.

6. Grounding and Power Supplies

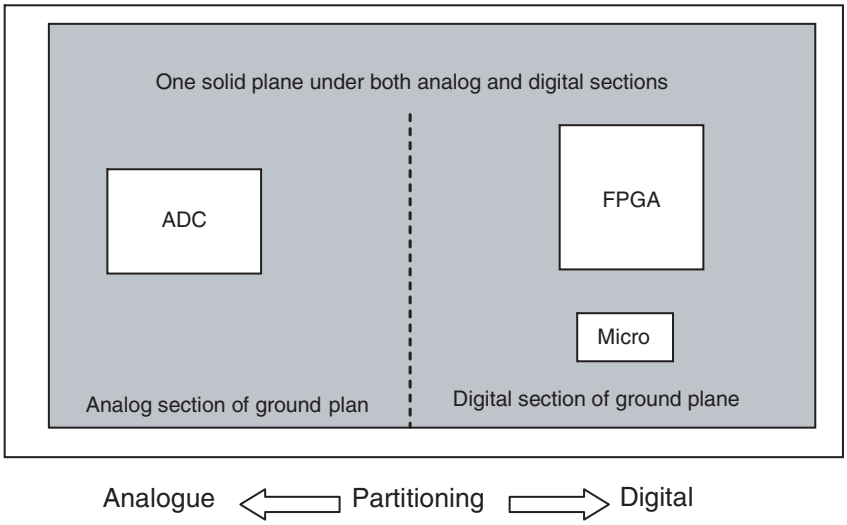
6.1 Ground Plane

One ground plane is necessary for the ADC and ideally this would be separated from the ground plane for the digital part (apart from one star point).

However in complex systems with multiple ADCs it may be difficult to maintain these separate ground planes due to common signals being distributed to each ADC. In these cases the use of a continuous ground plane might be more appropriate. For more details on this consult the applications note 'Design Considerations for Mixed Signal PCB layout.' available from the website www.e2v.com or by contacting Hotline-BDC@e2v.com.

We recommend a distance of at least 2cm between analog and digital sections.

Figure 6-1. Ground Plane



6.2 Power Supply Planes

The Quad ADC requires 3 distinct power supplies:

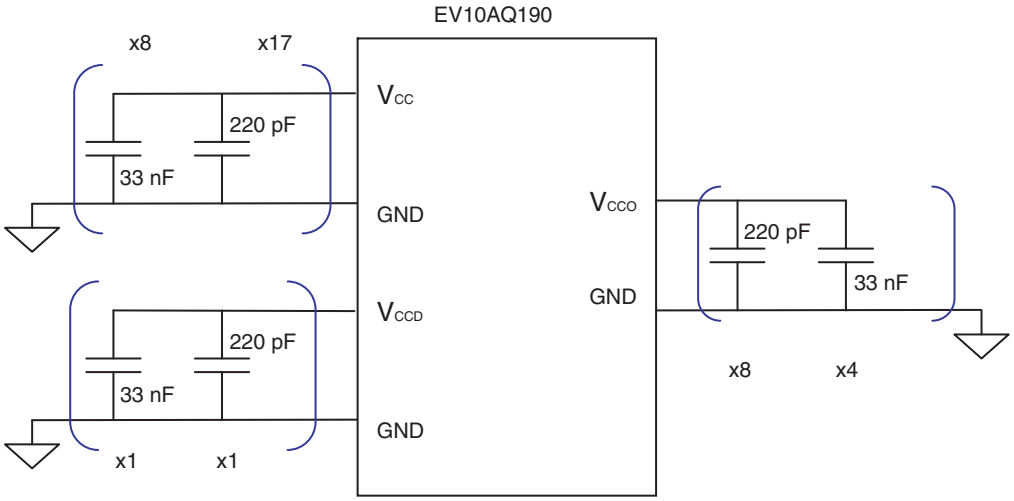
V_{CC} = 3.3V (for the analog parts and the SPI pads)

V_{CCD} = 1.8V (for the digital parts)

V_{CCO} = 1.8V (for the output buffers)

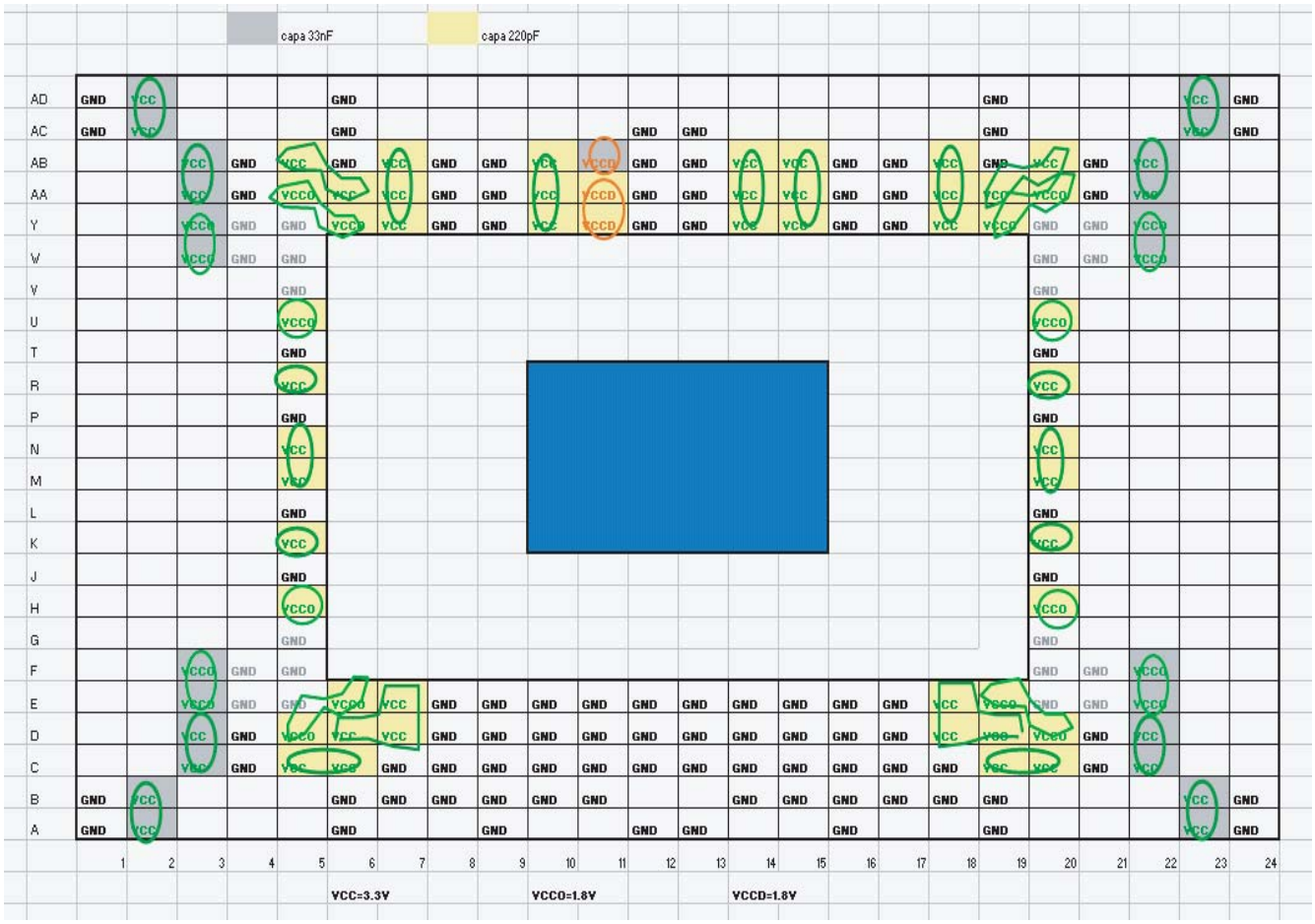
It is recommended to decouple all power supplies to ground as close as possible to the device balls with 220 pF in parallel to 33 nF capacitors. The minimum number of decoupling pairs of capacitors can be calculated as the minimum number of groups of neighboring pins. 17 capacitors of 220 pF and 8 capacitors of 33 nF for Vcc; 8 capacitors of 220 pF and 4 capacitors of 33 nF for Vcco and 1 220 pF capacitor with 1 ""nF capacitor for Vccd.

Figure 6-2. Power Supplies Decoupling Scheme



If the PCB space for decoupling capacitors is limited, adjustments to the value and number of capacitors can be made. The diagram below shows one such arrangement.

Figure 6-3. Suggested Power Supplies Decoupling Scheme



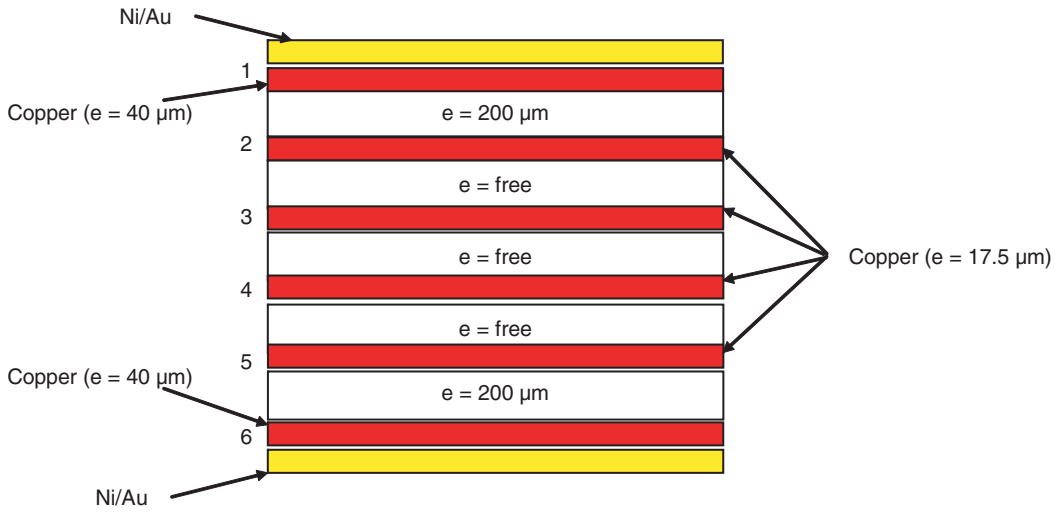
6.3 Board Layout Recommendations

6.3.1 PCB Stack up

The recommended board stack up is described in Figure 6-4 on page 19.

It applies to ISOLA410 PCB board material.

Figure 6-4. Board Recommended Stack up Using ISOLA 410 Board Material



6.3.2 Clock, Analog Input and Output Data Signals

It is recommended to route the clock, analog input signals and output data signals as differential signals.

In the case of the use of a PCB with dielectric material such as FR4 HTG (ISOLA IS410 with 42% resin content and $\varepsilon_r = 4$), the recommended board layout for the differential signals (clock and analogue inputs) is described in Figure 6-5 and Figure 6-6. This recommended layout only applies if the board stack up described in Section 6.3.1 ”PCB Stack up” on page 19 is satisfied.

Note: In the case of the digital output signals, the recommended differential routing is provided in Figure 6-6.

Figure 6-5 shows the standard routing with 1.27 mm pitch between signals of the same differential pair (clock and analogue inputs only).

Figure 6-6 shows another possible routing for differential signals (clock analogue inputs and digital outputs), preferred in case of high board size constraints.

Both configurations satisfy the impedance matching required for differential signals.

Figure 6-5. Differential Board Routing for the Clock and Analog Input Signals on FR4 HTG (IS410)

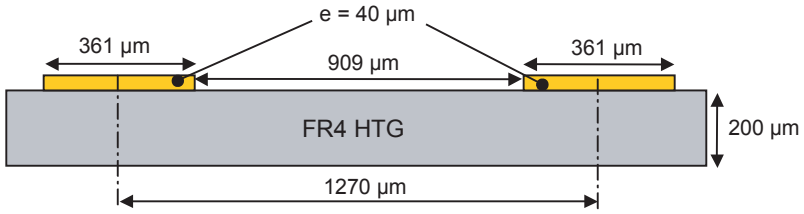
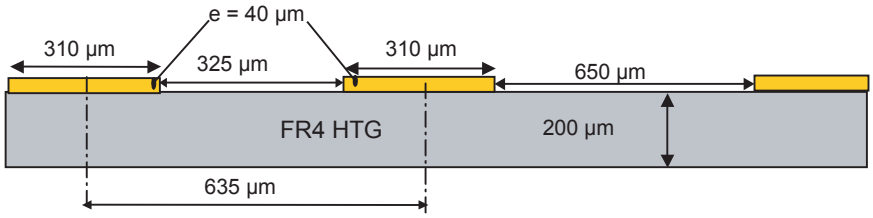
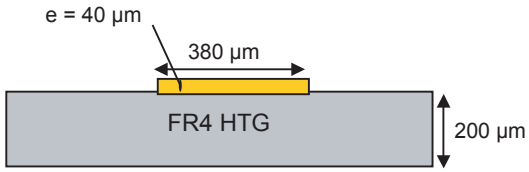


Figure 6-6. Differential Board Routing for the Clock, Analog Input and Digital Output Signals on FR4 HTG (IS410)



In the case of single-ended signals, the board layout should be as illustrated in Figure 5-16 on page 16.

Figure 6-7. Single-ended Routing on FR4 HTG (IS410)



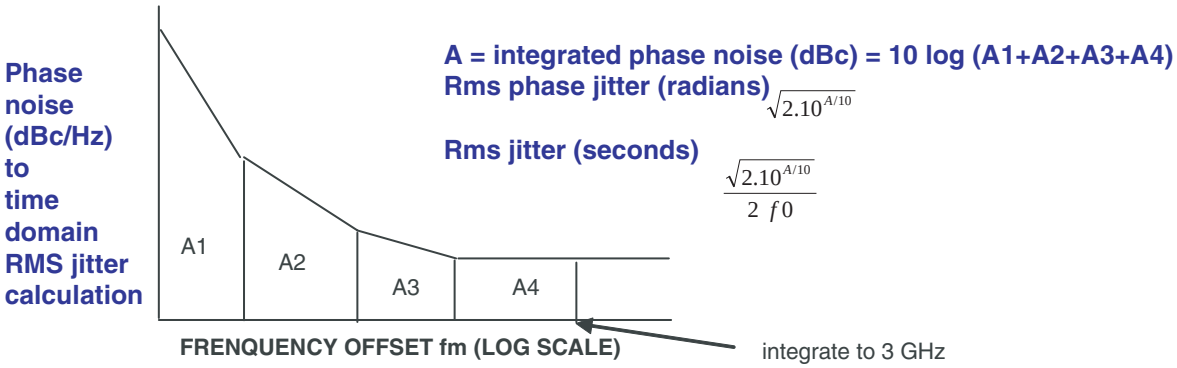
7. Choice of PLL

For a 10 bit ADC running at such high sampling rates, clock jitter becomes a critical parameter.

The effect of the clock jitter on the sampled signal noise floor can be calculated using the phase noise figures given by the PLL manufacturer.

In fact for large bandwidth converters the phase noise close to the carrier is not the dominant source and the integral of the phase noise floor becomes more important.

Figure 7-1. Phase Noise Calculation



Given the recommendation for 150 fs maximum clock jitter to achieve optimum performance; a PLL with phase noise floor approaching 150 dBc /Hz (at 10 MHz of carrier) should be considered.

For further details regarding the phase noise to jitter calculations refer to application note 'High Speed ADC Input Clock Issues' available from the website www.e2v.com or by contacting Hotline-BDC@e2v.com.

How to reach us

Home page: www.e2v.com

Sales offices:

Europe Regional sales office

e2v ltd

106 Waterhouse Lane

Chelmsford Essex CM1 2QU

England

Tel: +44 (0)1245 493493

Fax: +44 (0)1245 492492

mailto: enquiries@e2v.com

e2v sas

16 Burospace

F-91572 Bièvres Cedex

France

Tel: +33 (0) 16019 5500

Fax: +33 (0) 16019 5529

mailto: enquiries-fr@e2v.com

e2v Aerospace and defense inc

765 Sycamore Drive

Milpitas

California 95035

USA

Tel: +33 (0) 1 408 737 0992

Fax: +33 (0) 1 408 736 8708

mailto: e2v-us.com

Americas

e2v inc

520 White Plains Road

Suite 450 Tarrytown, NY 10591

USA

Tel: +1 (914) 592 6050 or 1-800-342-5338,

Fax: +1 (914) 592-5148

mailto: enquiries-na@e2v.com

Asia Pacific

e2v ltd

11/F.,

Onfem Tower,

29 Wyndham Street,

Central, Hong Kong

Tel: +852 3679 364 8/9

Fax: +852 3583 1084

mailto: enquiries-ap@e2v.com

Product Contact:

e2v

Avenue de Rochepleine

BP 123 - 38521 Saint-Egrève Cedex

France

Tel: +33 (0)4 76 58 30 00

Hotline:

mailto: hotline-bdc@e2v.com