



OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

ELEKTRO- UND INFORMATIONSTECHNIK

Getting Started with ADC LTC2308 on DE1-SoC Board Using VHDL

Prof. Dr. Martin J. W. Schubert, Electronics Laboratory, OTH Regensburg, Regensburg, Germany

Getting Started with ADC *LTC2308* on *DE1-SoC* Board Using *VHDL*

Abstract. This communication introduces to the usage and operation of the *LTC2308* A/D converter employed on Terasic's *DE1-SoC* board. The *Cyclone V FPGA* on the board is programmed with *VHDL*. Some background information on the operation of *SAR* ADCs is provided.

1 Introduction

1.1 Objectives and Organization of this Document

This document introduces the usage of A/D converter (ADC) *LTC2308* [16] used on *Terasic's* [1] *DE1-SoC* board [2], which is programmed using *Quartus* and *VHDL* [12] hardware description language.

This communication is intended for teaching and may be subject to manifold side effects, for example due to input protection circuits of pins. Therefore:

This educational setup is not suitable for precise characterization or benchmarking of hardware.

The organization of this document is as follows:

- Section 1 is the Introduction,
- Section 2 introduces the hardware and the *Quartus* programming software,
- Section 3 explains the test setup and some backgrounds about A/D conversion,
- Section 4 demonstrates the *LTC2308* with some practical experiments,
- Section 5 draws relevant conclusion and
- Section 6 offers references.

1.2 Tools

1.2.1 DE1-SoC Hardware

This document assumes that you are familiar with the *Terasic's* [1] *DE1-SoC* board using an *Intel Cyclone V FPGA* [2] or a similar *DEx* board with the same general-purpose input/output (GPIO) user header. The version of your *DE1-SoC* board can be identified at [3]. *DE1-SoC* board revisions *F* and *G* differ in a printed company label only. To get it from the internet, go to [4] to find and download *DE1-SoC_v.5.1.2_HWrevF_SystemCD.zip* [5] and download it. It contains amongst other things important documents such as *DE1-SoC User Manual* [6] and *Schematic* [7]. On the computer system of OTH Regensburg, you will also find the CD on drive K:\Sb\[8]. Do not use any other manual revision to follow this documentation. The differences may be considerable.

1.2.2 Quartus [9] and ModelSim [10] Software Tools

It is assumed that you have *Intel's Quartus 13 [9]* and *ModelSim [10]* software available. To download this freeware for your private PC you have to sign in at *Intel* [11]. At OTH Regensburg's PC pools of faculties *EI* and *IM*, this software is installed. At faculty *EI*, also *Quartus 8* is installed supporting some older *DE2* boards with *Cyclone II* FPGAs, because they are not supported for *QuartusI* versions greater than 13.1.

1.2.3 Use of *VHDL*

The *IEEE standard VHDL Language reference manual* [12] is comprehensive and demanding to read. *Qualis VHDL Quick Reference Card* [13] and *1164 Packages Quick Reference Card* [14] are compact but difficult to understand. Feel free to find your own sources.

VHDL is not case sensitive. In the following, **KEYWORDS** will be written in **ALL** CAPITAL LETTERS and user defined names in lowercase letters. Exceptions are capitalized initials used for composed self-made names, e.g. *AddressBus* or *DataBus*. Self-made data types begin with t_{-} , e.g. $t_{-}StateVector$.

1.3 Acknowledgements

The author would like to thank *Terasic Technologies* [1] for admission to use screen copies of *Terasic* documentation for teaching purposes in this lectures.

2 Getting Started with the Hardware and Tool *Quartus*2.1 Using the ADC Connector on the *DE1-SoC* Board



(c) 2x5 header 180° rotated

			_
GND	-0.10	• •	ADC_IN7
	0.0	30	
ADC_IN6		7 0	ADC_IN5
	-0.0	10-	
ADC_IN4	-0.6	5.0	ADC_IN3
	-00	50	
ADC_IN2		3 0	ADC_IN1
	-04	30	
ADC_IN0	2	1 []	VCC5
	-02		

Fig. 2.1: A/D converter LTC2308 and connectors from the DE1-SoC board point of view

Take your *DE1-SoC* board; remove anything connected to the different plugs. Connect power cable and *USB* programming cable. Power the board on.

Do not connect any measurement equipment at this point!

Consider Fig. 2.1(a), the 2x5 ADC user header copied from the DE1-SoC User Manual [6]. Then identify it on the *DE1-SoC* board and on the photo of Fig. 2.1(b).

To which pin (label) of the 2x5-user header is the red wire in Fig. 2.1(b) connected?

Observe the notch in the plug guide of Figs. 2.1 (a) and (b). You will notice that Fig. 2.1(a) is misleading and has to be rotated by 180° as shown in Fig. 2.1(c).

Use a voltmeter to check for voltages *GND* and *VCC5* at the 2x5 ADC User Header. If you use a voltmeter with grounded reference potential, such as an oscilloscope's probe,

be careful to not connect a ground line to V_{CC5} !

Comment: We have had that short between V_{CC5} and GND caused by an oscilloscope's probe ground at the ADC's 2x5 *User Header* several times. All boards survived this short due to overcurrent protection of V_{CC5} , but please **do not challenge fate!**

2.2 Getting Started with *Quartus* and *DE1-SoC* Board

This chapter assumes that you are familiar with the authors lab "*Getting Started with DE1-SoC Board Using VHDL*" [15] introducing to the DE1-SoC board programmed with the Quartus software.

 Table 2.2: Files in subdirectory ...\ci_delsoc_adci_ltc2308\ and their significance

File in directory \ci_de1soc_adci_ltc2308	Significance
ci_delsoc_adci_ltc2308.qpf	Quartus project file, ASCII, saves date and
	software version
ci_delsoc_adci_ltc2308.qsf	Quartus specification file, ASCII, saves date
	and assignments, e.g. signal ⇔ pin
ci_delsoc_adci_ltc2308.vhd	configuration interface module instantiating
	delsoc adci ltc2308.vhd with identical ports
output_files\ci_de1soc_adci_ltc2308.cdf	Optional: the *. <i>dcf</i> file is an ASCII file that
	saves programmer configurations
output_files\ci_de1soc_adci_ltc2308.sof	Optional: the *. <i>sof</i> file is a binary file ready
	for download without further compilation

Navigate to directory ...\ci_delsoc_adci_ltc2308\ where you will find the files listed in Tab. 2.2.

Double-click left on *ci_de1soc_adci_ltc2308.qpf* to start *Quartus*.

- If there is no *.sof file you will have to compile the model: *Processing* > *Start Compilation*.
- Start programmer: *Tools > Programmer*
- Within programmer: *Hardware Setup* > click on *DE1-SoC* > *Close*. You should see (i) something like *DE1-SoC* [USB-1] in the *Hardware Setup* window, and (ii) in the lower widow a graphics with devices "SOCVHPS→ 5CSEMA5F31"
- Click on *Start* button and observe the 7-segment displays on the *DE1-SoC* board.

Set all switches of the *DE1-SoC* board to '0'. The 7-segment displays should indicate something like "0 *XXXX*", whereas the '0' is the ADC's input *Set* indicator and *XXXX* the input voltage at pin *ADC_IN0* in mV.

2.3 The VGEN Daughter Board

VGEN **Right:** daughter board photo. Bottom: VGEN daughter board .connected to **DE1-SoC** main board 1551

Fig. 2.3.1: *VGEN* daughter board photos

Fig. 2.3.1 is a photo of the VGEN daughter board. It is available in the electronics laboratory of OTH Regensburg and intended to be connected to the *DE1-SoC* board to feed voltages to the 8 input channels of the *LTC2308* ADC. Fig. 2.3.2 (a) and (b) illustrate schematics and layout of the *VGEN* daughter board, respectively.



(a) Layout of the *VGEN* daughter board

(b) Layout of the VGEN daughter board



Fig. 2.3.2: *VGEN* daughter board: brown top layer and blue bottom layer.

3 Theoretical Background

3.1 LTC2308 Schematic and Operation



Fig. 3.1(a) sketches a behavioral block diagram of the *LTC2308* ADC. Fig. 3.1(b) is the respective view copied from the *DE1-SoC schematic* [7].

Exercises:

From which document and which page originates the schematic copy shown in Fig. 3.1(b).

Which voltage is supplied under which conditions to the *LTC2308*'s *COM* input on the *DE1-SoC* board? Argue!

.....

3.2 Configuring the *LTC2308*

3.2.1 Significance of Configuration Bits *S/D*, *O/S*, *S1*, *S0*

The *LTC2308* is configured with the six bits (*S/D*, *O/S*, *S1*, *S0*, *UNI*, *SLP*) are serially clocked in beginning with *S/D* and ending with *SLP*. While the 6 configuration bits for the next conversion are serially clocked from register $sdi_reg(5:0)$ into the *LTC2308* on port *SDI*, the ADC's last conversion result is serially clocked out on port *SDO* to register $sdo_reg(11:0)$.

Within the author's *VHDL* coded *ADC* interface module $de_{1soc_adci_ltc2308}$, the six configuration bits are stored on register $sdi_reg(5:0) = (S/D, O/S, S1, S0, UNI, SLP)$, which is set by switches sw(5:0) of the *DE1-SoC* board according to listing 3.2.1 and table 3.2.1.

The effect of listing 3.2.1 is detailed in table 3.2.1, whose part (a) is ordered according to *Table* 1 in the data sheet of *LC2308* [16, p.11], and part (b) is ordered with ascending variable Set = hex(sw(3:0)).

Listing 3.2.1: VHDL source code in module *delsoc_adci_ltc2308* assigning *sdi_reg* from *sw*. p program ltc2308: PROCESS (sw (5 DOWNTO 0))

```
BEGIN
 CASE sw(3 DOWNTO 0) IS
                                                 -- set:
 WHEN "0000" => sdi reg(5 DOWNTO 2) <= "1000"; -- 0 : INO-COM
 WHEN "0001" => sdi_reg(5 DOWNTO 2) <= "1100"; -- 1 : IN1-COM
 WHEN "0010" => sdi reg(5 DOWNTO 2) <= "1001"; -- 2 : IN2-COM
 WHEN "0011" => sdi_reg(5 DOWNTO 2) <= "1101"; --
                                                    3 : IN3-COM
 WHEN "0100" => sdi_reg(5 DOWNTO 2) <= "1010"; --
                                                    4 : IN4-COM
 WHEN "0101" => sdi reg(5 DOWNTO 2) <= "1110"; --
                                                    5 :
                                                        IN5-COM
 WHEN "0110" => sdi_reg(5 DOWNTO 2) <= "1011"; --
                                                    6 : IN6-COM
 WHEN "0111" => sdi reg(5 DOWNTO 2) <= "1111"; --
                                                    7 : IN7-COM
 WHEN "1000" => sdi reg(5 DOWNTO 2) <= "0000"; --
                                                    8 : IN0-IN1
 WHEN "1001" => sdi_reg(5 DOWNTO 2) <= "0100"; --
                                                    9 : IN1-IN0
 WHEN "1010" => sdi_reg(5 DOWNTO 2) <= "0001"; --
                                                    A : IN2-IN3
 WHEN "1011" => sdi_reg(5 DOWNTO 2) <= "0101"; --
                                                    B : IN3-IN2
 WHEN "1100" => sdi reg(5 DOWNTO 2) <= "0010"; --
                                                    C : IN4-IN5
 WHEN "1101" => sdi reg(5 DOWNTO 2) <= "0110"; --
                                                    D : IN5-IN4
 WHEN "1110" => sdi_reg(5 DOWNTO 2) <= "0011"; --
                                                    E : IN6-IN7
 WHEN "1111" => sdi_reg(5 DOWNTO 2) <= "0111"; --
                                                    F : IN7-IN6
 WHEN OTHERS => sdi_reg(5 DOWNTO 2) <= "1000"; --
                                                    0 : IN0-IN1
 END CASE:
     sdi reg(1) <= NOT sw(4); -- UNI bit: sw(4) = 0': unipolar, sw(4) = 1': bipolar
     sdi reg(0) <= sw(5);</pre>
                              -- SLP bit: '0': NAP mode after conv. if CONVST='1'
                              ___
                                    '1': SLEEP mode after conversion if CONVST='1'
```

END PROCESS p_program_ltc2308;

		U		U						L / I	-			
0	1	2	3	4	5	6	7	8	9	10	101	12	13	14
sw(3:0)	Set	S/D	O/S	S1	S0	0	1	2	3	4	5	6	7	СОМ
1000	8	0	0	0	0	+	-							
1010	А	0	0	0	1			+	-					
1100	С	0	0	1	0					+	-			
1110	Е	0	0	1	1							+	-	
1001	9	0	1	0	0	-	+							
1011	В	0	1	0	1			-	+					
1101	D	0	1	1	0					-	+			
1111	F	0	1	1	1							-	+	
0000	0	1	0	0	0	+								-
0010	2	1	0	0	1			+						-
0100	4	1	0	1	0					+				-
0110	6	1	0	1	1							+		-
0001	1	1	1	0	0		+							-
0011	3	1	1	0	1				+					-
0101	5	1	1	1	0						+			-
0111	7	1	1	1	1								+	-

Table 3.2.1: Setting configuration bits of *LTC2308* with switches *sw*(5:0) of *DE1-SOC* board.(a): Line ordering according to *Table 1* of *LTC2308* data sheet [16, p 11].

(b): Line ordering with ascending variable Set = hex(sw(3:0)).

0	1	2	3	4	5	6	7	8	9	10	101	12	13	14
sw(3:0)	Set	S/D	O/S	SI	S0	0	1	2	3	4	5	6	7	СОМ
0000	0	1	0	0	0	+								-
0001	1	1	1	0	0		+							-
0010	2	1	0	0	1			+						-
0011	3	1	1	0	1				+					-
0100	4	1	0	1	0					+				-
0101	5	1	1	1	0						+			-
0110	6	1	0	1	1							+		-
0111	7	1	1	1	1								+	-
1000	8	0	0	0	0	+	-							
1001	9	0	1	0	0	I	+							
1010	А	0	0	0	1			+	1					
1011	В	0	1	0	1			-	+					
1100	С	0	0	1	0					+	1			
1101	D	0	1	1	0					-	+			
1110	Е	0	0	1	1							+	-	
1111	F	0	1	1	1							-	+	

3.2.2 Significance of Configuration Bit UNI

The UNI bit affects signed/unsigned coding of the digital output delivered by the LTC2308:

- UNI = '0': Bipolar: Output bits are delivered in 12 bit signed format, range $-2^{11}...2^{11}-1$, corresponding to decimal range -2048 ... 2047 and input voltage range -2.048...2.047V.
- UNI = '1': Unipolar: Output bits are delivered in 12 bit unsigned format, range $0...2^{12}$ -1 corresponding to decimal range 0...4095 and input voltage of 0...4095V.

Listing 3.7 details the impact of the UNI flag: Function conv_integer is taken from package std_logic_signed (1st line), so that conv_integer(sdo_reg) interprets sdo_reg as signed. This is avoided by pre-concatenation of a '0' in expression conv_integer('0'&sdo_reg) when UNI='1'.

Listing 3.2.2: VHDL source code in module adci ltc2308 interpreting the UNI flag.

```
LIBRARY ieee; USE ieee.std_logic_1164.ALL, ieee.std_logic_signed.conv_integer;
```

```
-- translate bit vector sdo_reg to integer Nout after transmission
UNI <= sdi_reg(1); -- '0': bipolar, '1': Unipolar integer number
p_out:PROCESS(reset,sclk)
BEGIN
IF reset='0' THEN
Nout <= 0;
ELSIF sclk'EVENT AND sclk='0' AND sclk_count=12 THEN
IF UNI='1' THEN
Nout <= conv_integer('0'&sdo_reg);
ELSE
Nout <= conv_integer('0'&sdo_reg);
END IF;
END IF;
END IF;
END PROCESS p_out;
```

3.2.3 Significance of Configuration Bit SLP

The *LTC2308*'s serial interface is SPI compatible, but may have some deviations of the official SPI protocol. A rising edge at signal *CONVST* starts the conversion. When *CONVST* is held high after the conversion ends the ADC goes into a NAP mode (SPL='0') or a *SLEEP* mode (*SLP*='1'). Power consumption in this mode is minimal, but wakeup from *SLEEP* mode \geq 200ms. In this communication we will not consider power consumption and work within *NAP* mode only (*SLP*='0').

3.3 Resolution, Levels and Steps

The minimum step that can be resolved is 1 digital least significant bit (*LSB*). It corresponds to a Δ the analog side, which is also referred to as resolution. Fig. 3.3 illustrates, that the representation of *K* steps Δ requires *K*+1 levels. For this reasons, most ADCs cannot represent the nominal maximum voltage, but remain $1\Delta = 1$ *LSB* below it, because the maximum number representable with *m* bits is 2^m -1.

(a) Weighted summation of charges



Fig. 3.3: Representation of *K* steps of height Δ requires *K*+1 levels

3.4 Interfacing the *LTC2308*

The *LTC2308* operates as (*SPI* compatible) slave and gets its clock signal on *SCLK* with a maximum frequency of 40MHz from the master. In the author's code, the master is *VHDL* module *adci_ltc2308*. The relationship between official *SPI* signal names and *LTC2308* I/O signals is listed in table 3.4.

According to the LTC2308 data sheet [16], the

falling edge of SCLKforces:master and slave write to lines SDI and SDO, respectively.rising edge of SCLKreads:master and slave read from lines SDO and SDI, respectively.

LTC2308	SPI	Comment
SDI	MOSI or SIMO	Master Out / Slave In
SDO	MISO or SOMI	Master IN / Slave Out
CONVST	CS	Chip Select, low active.
		LTC2308: rising edge begins A/D conversion

Table 3.4: Signal mapping: LTC2308 versus Serial Peripheral Interface (SPI).

3.5 Successive Approximation Register (SAR) ADC

This is the most probably mostly used ADC type due to its good price/performance relationship. The acronym SAR stands for "successive approximation register", which is contained in the successive logic. The procedure of this conversion type is illustrated in Fig. 3.5:

Conversion procedure

- 1. Initialize: Set *j* to its maximum value j=M-1, whereas *M* is the number of output bits.
- 2. Set $a_j = '1'$ and check for U_{DACout} .
- 3. If $U_{DACout} > U_{in}$ reset $a_j = 0$, else let $a_j = 1$.
- 4. Decrement *j* if j > 0 and go to 2., otherwise finish conversion.

Comments

- Phase (b) in Fig. 3.5(b) was introduced for educational purposes. Resetting a_j and setting a_{j-1} can be made in the same step.
- This kind of ADC seeks to keep $U_{DACout} \leq U_{ADCin}$ yielding a quantization error offset of $-\Delta/2$.



Fig. 3.5: Principle of successive approximation register (SAR) A/D conversion

3.6 Possible Physical Realization of the DAC

An A/D converter (ADC) cannot be more accurate than its inner D/A converter (DAC), which is placed into the ADC's feedback loop, except for flash ADCs.

Fig. 3.6 illustrates a frequent D/A conversion principle: binary weighted summation. We find it also in the binary number $a_3 a_2 a_1 a_0 = a_3 \cdot 2^3 a_2 \cdot 2^2 a_1 \cdot 2^1 a_0 \cdot 2^0$. Figs. 3.6(a) and (b) illustrate how it can be done with conductors and capacitors, respectively, whereas it is assumed that $U_{\#} = a_{\#} \cdot U_{ref}$ with $a_{\#} = 0, 1$, $G_{\#+1} = 2 \cdot G_{\#}$ and $C_{\#+1} = 2 \cdot C_{\#}$.

For the resistive network we have with conductor = 1/resistor or $G_{\#} = 1/R_{\#}$, the reverse output impedance and output voltage computes as

$$Z_{out}^{-1} = G_{sum} = \sum_{j=0}^{M-1} G_j$$
 and $U_{DACout} = \sum_{j=0}^{M-1} \frac{G_j}{G_{sum}} U_j$, respectively.

For the capacitive network, the output impedance and output voltage computes as

$$Z_{out}^{-1} = C_{sum} = \sum_{j=0}^{M-1} C_j \quad \text{and} \quad U_{DACout} = \sum_{j=0}^{M-1} \frac{C_j}{C_{sum}} U_j, \text{ respectively.}$$

When $U_B = 0$ V, the output voltage range is $U_{DACout} = 0...U_{ref} - 1$ LSB; when $U_B = U_{ref}$, then the output voltage range is $U_{DACout} = 1$ LSB...U_{ref}.

(a) Resistive network

(b) Capacitive network.



Fig. 3.6: D/A conversion scheme (among many others) with .binary weighted summation

3.7 Questions

Where do we find hints about ADC's architecture (e.g. flash, pipelined, SAR, delta-sigma,...) and the conversion principle of its inner DAC (e.g. resistive, capacitive, inductive, current summation,...) in the *LTC2308* data sheet [16]?

4 Operating the *LTC2308*

4.1 Getting Started with Reading the *LTC2308*

How should we set switch sw(4) when we measure voltage $IN_{\#}$ - COM (#=0...7) with the *DE1-SoC* board? Argue!

.....

What is the valid input voltage and output number range then?

.....

.....

How should we set switch sw(4) when we measure voltage $IN_{\#+1} - IN_{\#}$ (# = 0, 2, 4, 6) with *DE1-SoC* board? Argue!

.....

What is the valid differential input voltage and output number range then?

.....

Measurements:

Table 4.1 lists the 16 possible measurement configurations to be set with Set=sw(3:0), whereas *Set* is indicated by *digit5*. The *SLP* bit is held at zero with sw(5)='0'.

Hints:

- Reset signal is active when pushing *key*(0).
- Enable signal is active when not pushing key(1), i.e. pushing key(1) holds the 7-seg display.

	IN ₀ - COM	IN ₁ - COM	IN ₂ - COM	IN ₃ - COM	IN ₄ - COM	IN ₅ - COM	IN ₆ - COM	IN ₇ - COM
	sw(4)='0'	sw(4)='0'						
Set:	0							
V(IN#):								
	-							
	IN ₀ -IN ₁	IN ₁ -IN ₀	IN ₂ -IN ₃	IN ₃ -IN ₂	IN ₄ -IN ₅	IN ₅ -IN ₄	IN ₆ -IN ₇	IN7-IN6
	sw(4)='1'	sw(4)='1'						
Set:								F
V(IN#):								

Table 4.1: Output at floating input voltages of *LTC2308*. *SLP* bit held at '0' with *sw*(5)='0'.

Remark: The voltages are typically floating somewhere in the valid input range with low differences between them.

4.2 Measuring a Single Input Channel

Connect the *VGEN* daughterboard to the 2x5 ADC *User Header* of *DE1-SoC* board. If you do not have the *VGEN* board you may use a poti as shown in Fig. 4.2.

Set all configuration switches of *DE1-SoC* board to '0', i.e. sw(5:0) = "000000". Which channel is measured in which mode (*BIB/UNI*)? (Switches sw(9:6) do not matter.)

.....

Use poti to supply 0...5V to CH₀. Which number range is observed on the 7-seg display?

Use poti to supply 0...5V to CH_0 . Set bipolar mode, i.e. sw(4)='1'. Which number range is observed on the 7-seg display?



Fig. 4.2: Poti for generation of a voltage in the range 0...Vccs, if VGEN board is not available

4.3 Measurements With Input Channels IN0 ... IN7

Connect the VGEN daughterboard or find another possibility to supply Hint: Pressing $key(1) = enable$ holds the display, while $key(0) = reset$ of	y several voltages. Elears it.
Measure input voltages at $IN0 \dots IN7$ in unipolar mode. What switch measure $IN\#, \#=0\dots7$. Indicate unknown switch settings with 'X'. sw(2)	ch setting is required to 5:0) =
Can you set and measure any input channel in the full range 04095?	
	• • • • • • • • • • • • • • • •
Set switches to unipolar measurement of <i>INO – IN1</i> . sw	e(5:0) =
Set <i>IN1</i> to 2.5V and sweep <i>IN0</i> from 05V. Number range on display	r?
	•••••
Set switches to bipolar measurement of <i>IN0 – IN1</i> . sw	v(5:0) =
Set <i>IN1</i> to 2.5V and sweep <i>IN0</i> from 05V. Number range on display	r?

4.4 Offset Measurements

Connect *IN0...IN7* to ground. (Check Figs. 2.3.1 and 2.3.2 for the connector with 8 *GND* pins.) Note measured offsets at in bipolar mode IN# = 0V, #=0...7 in table 4.4.

Hint. You might have to press key(1) = enable_clocking to get a stable display snapshot

This educational setup is not suitable for precise characterization or benchmarking of hardware.

Table 4.4: Measured offset voltages V(IN#) = 0V, #=0...7.

	INO	INI	IN2	IN3	IN4	IN5	IN6	IN7
Offset								
mV								

Set poti for *CH1* to V(CH1) = 2.5V. Set a short between *IN0* and *IN1*. Turn poti for channel 0. Which voltages against ground do we get at V(IN0) = V(IN1) with poti of *CH0*?

Set switches to measurement von V(IN0) - V(IN1) in the bipolar mode and sweep voltages V(IN0) = V(IN1) in the range 0.1 ...4V. Rang of offset voltages measured?

.

.

4.5 Chopping Techniques

(a) Non-inverted input voltage



Fig. 4.5: Offset measurement techniques.

Differential voltage inputs allow to remove a part of the input offset voltage by chopping. Chopping is flipping the input voltage with preferably high frequency with subsequent averaging or lowpass filtering.

In Fig. 4.5(a) the voltage to be converted is $U_{conv+} = U_{in} + U_{off}$ while after flipping we get in Fig. 4.3(b) $U_{conv-} = -U_{in} + U_{off}$. Averaging delivers

 $U_{conv} = \frac{1}{2} (U_{conv+} - U_{conv-}) = \frac{1}{2} (U_{in} + U_{off} - (-U_{in} + U_{off})) = U_{in}$

The offset voltage is obtained from

 $U_{conv} = \frac{1}{2} (U_{conv+} + U_{conv-}) = \frac{1}{2} (U_{in} + U_{off} + (-U_{in} + U_{off})) = U_{off}$

This educational setup is not suitable for precise characterization or benchmarking of hardware.

Measurements:

Remove any shorts between channels and set bipolar mode.

Set *IN1* to 2.5V and sweep *IN0* from 0...5V. When you measure a voltage of #V, #=0...4.095, and then switch sw(0), you should obtain -#V. The difference you see is 2 U_{off} . Make a little table of offset voltages dependent of several differential input voltages. Note your results in table 4.5. What tendency do you observe?

Table 4.5: Measured offset voltages between V(IN0) and V(IN1) at V(IN1)=2.5V.

V(<i>IN0</i>) -V(<i>IN1</i>)	-2V	-1.5V	-1.0	-0.1V	0.1V	1V	1.5 V	2V
<u>Offset </u> mV								

5 Conclusion

The *LTC2308* [16] is an A/D converter available with its 8 multiplexed input channels on the *DE1-SoC* board [2] from *Terasic* [1]. Practical exercises prove that it features an amplification of 1bit/mV. In the unipolar mode of operation we can output a number range of -2048...2047 and in the unipolar mode a range of 0...4095.

6 References

- [1] Available: <u>https://www.terasic.com.tw</u>.
- [2] Available: <u>https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836</u>
- [3] Available: <u>http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=886</u>
 [4] Available: <u>https://www.terasic.com.tw/cgi-</u>
- bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=4
- [5] CD ROM DE1-SoC CD-ROM (rev.F Board) Version 5.1.2 of 2910-01-28 from [4]
- [6] DE1-SoC User Manual, Ref. F, taken from [5], available:
- https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_UserManual.pdf [7] DE1-SoC Schematic, Ref. F, taken from [5], available:
- https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_Schematic_revF.pdf [8] K:\SB\Sources\EDA\Terasic\Hardware\
- [9] Available: https://en.wikipedia.org/wiki/Intel Quartus Prime
- [10] Available: https://en.wikipedia.org/wiki/ModelSim
- [11] Av. https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
- [12] Available: https://edg.uchicago.edu/~tang/VHDLref.pdf
- [13] Available: https://www.mimuw.edu.pl/~marpe/pul/card_vhdl.pdf
- [14] Available: <u>https://www.mimuw.edu.pl/~marpe/pul/card_1164.pdf</u>
- [15] M. J. W. Schubert, Getting Started With DE1-SoC Board Using VHDL, Available: <u>https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/Lab_ElectronicBoards.htm</u>, OTH Regensburg, Regensburg, Bavaria, Germany
- [16] Analog Devices, *LTC2308* Data Sheet, Availble: <u>https://www.analog.com/media/en/technical-documentation/data-sheets/2308fc.pdf</u>.
- [17] Available: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface