Modeling of ADS 5547 using VHDL: Serial interface mode

Ammar Armghan[1], Arslan Haider[2], Fawad Naseer[3], Ussama Assad[4]
Department of Electrical Engineering
The University of Faisalabad
Faisalabad, Pakistan
ammar.armghan@gmail.com, arslanhdr@yahoo.com, fawadn.84@gmail.com, osama.asad@hotmail.com

Abstract—**The analogue to digital conversion is very important for every electronic equipments ranging from the bottom level to the top level. Every data which is captured is changed into the digital form. In order to capture the data with proper and efficient ways different ADC devices are designed and manufactured. Before the designing of the device, modelling of the device is checked how its behavior would be in the real time functioning. This helps to simulate the functionality of the device before the fabrication and change it if needed for the future amendments. Thus, modelling allows reducing the error and increasing the performance of the device. Different software is used for these purposes and one of the languages which do is hardware descriptive language. Here, we have used Model Sim for the modelling of the ADC.**

The work is carried by splitting the ADC into different blocks like 14-bit ADC, Digital Serializer, Clock generator and control interface for serial mode configuration and accordingly the modeling of the device is done. The Introduction explains the design requirements and discussed some the methodologies. The Analog to digital converters is described with various features and key terms. Further the VHDL implementation is explained with simulation results. The paper includes the detail of the work researched and carried out.

Keywords: FPGA, VHDL, Modelling, Xilinx, ADS

I. INTRODUCTION

A. Design Requirement

The Mentor Graphics ModelSim SE 6.5 development environment is used for modelling the device for the given design task. The device is a Texas Instruments' ADS5547 high performance 14-bit, 210-MSPS A/D converter [1]. The converter has several operating modes as fully differential LVDS DDR (double data rate). The converter can be configured using the industry standard interface SPI protocol .The Serial Peripheral Interface Bus or SPI bus is a synchronous serial data link standard named by Motorola that operates in full duplex mode.

Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (chip select) lines. Sometimes SPI is called a "four wire" serial bus, contrasting with three, two, and one wire. The ADS5547 acts as a slave SPI device. ADC is configured through SPI port to provide continuous samples on the serial interface. To start with the design of the project, it is divided into two tasks as follows: **Task 1:** In this task we will perform functional modelling of simple data paths with basic operation and write VHDL code along with testbench to simulate and verify it. We will also model the device's serial programming modes. Then we will extend Task 1's VHDL code further including the testbench

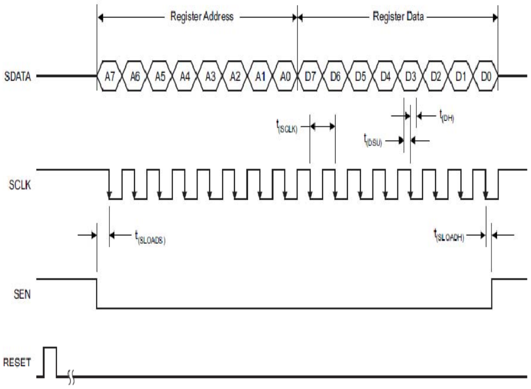


Figure 1: Serial Peripheral Interface timing [1]

Task 2: In this task we will add timing delay to the model from the above tasks considering setup and hold time, and therefore will be extending VHDL code further.

The tasks are achieved by planning the design and sub dividing the tasks. By understanding the device datasheet ADS5547, the design is divided into different categories such as main 14 bit ADC block, Digital Encoder and Serializer block, Control Interface block and Reference block. The basic operation of conversion of analogue data to digital form is done at the first step. Then Serializer block is started which gives the data serially. The serial mode operation is implemented according to the data sheet and appropriate delay of 14 latency is given which can be seen in the simulation results. Different Register configuration is implemented e.g. A-Register, B-register, G-Register, C-Register, E-Register and also serial mode configuration is applied.

II. LITERATURE REVIEW

In order to test and verify the digital device, modelling is undertaken which checks the behavioral working of the device in different modes of operation. Both functional and timing behavior of the device can be simulated and tested by modelling. The behavior of such devices is coded by VHDL which makes a factual device and gives various results. These results can then be modified to increase the quality whilst reaping benefits of device's features to design and

978-0-7695-4806-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICECC.2012.580

implement the whole real FPGA chip through VHDL's cutting edge technology.

The modelling of analogue-to-digital converter in digital aspects has been done before, but the device ADS5547 modelling is not done yet. Therefore, it would be a new research development. By modelling this device we will be able to check simulation features in order to implement the chip for real application. Modelling helps in correcting errors and adding up some advance features if needed. It manipulates the device efficiency with accuracy and usability whilst simulating and verifying the functionality. Modelling improves the device's performance and quality. After modelling the device, it can also be implemented for chip manufacturing.

In modelling an ADC, detailed aspects of the device features are considered. In an ADC, input and output logic are taken well into account for the processes. The clock signals are used to control the synchronous process. The analogue signals are used as vector signals. All digital signals are synchronized by clock signals. The delays are fitted to get a strong control for the transition purposes. This methodology for modelling has been achieved for various ADC devices. VHDL based behavioral modelling of the analogue and digital part can be synthesized using different modes of operation from the device features and predictabilities. We can test and calibrate the device by obtaining high accurate approach for simulation. The digital circuit of the device assists calibration, control and correction of different testing modes for the ADC. Modelling is being carried out using Mentor Graphics and synthesized by VHDL code schematically and behavior is analyzed for area and timing [3]. ADC modelling is an asset to an electronic world as it allows better transition from factual to a real device through easily carried adjustments to achieve better results in a factual device which are then optimized and applied to a real device for different applications. The modelling guidelines approved by European Space Agency (ESA) gives an outline about configuring, packaging,

A. *ADS5547 Block Diagram*

Below in is the ADC5547 block diagram. The main blocks in ADC5547 are 14-bit ADC that take the input sample signal from sample-hold circuit and convert it is 14 bit discrete digital output, Reference Voltage block that take the input Interfacing the appropriate signals and other set rules for verification [4].

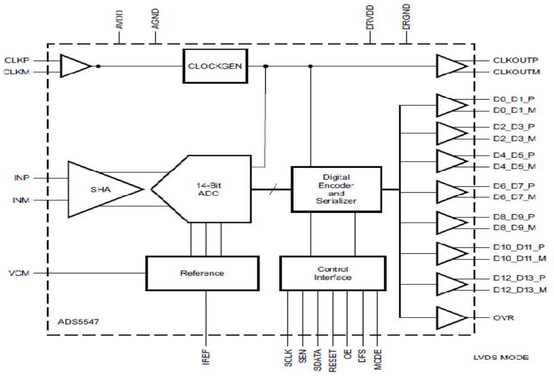


Figure 2: Block Diagram of ADS5547 [1]

reference voltage/current signal, Clock Generator that take input clock and generate output clock, Control interface that take control signals to configure device in different modes and work both as serial and parallel interface, Digital Encoder and Serializer that serialize the output data from ADC and encode it based on the control signals [1].

B. *Sub Blocks of ADS5547*

The 14-bit ADC block is main block of ADC5547 device which actually performs the analog to digital (A/D) conversion. It takes the sampled continuous input values and after conversion converts it in 14-bit discrete digital form.

The general ADC block diagram is shown in the figure 3. Here to the Prefilter, input signal is sent which filters the input wherever there is aliasing effects of higher frequency is reduced. This filter is called anti-aliasing filter. This filter is implemented with regards to band limiting behavior of ADC. Then after signal is sent to Sample/Hold circuit which keeps the input analogue signal to constant until it is converted in digital form. The time period for the conversion is called conversion time. The Quantizer is used to step by step quantize the signal into segmented sub ranges which is denoted by 2N where N represents the number of bits. Corresponding ranges the sampled input. These sub ranges is understood by the digital processor to encode the digital bits. Thus within the conversion time sampled analog input is converted to digital output code [7].

C. *Reference Block*

The Reference block is used to take the input reference voltage/current signal. There are built in two internal references as REFP and REFM [1]. This method the converter load linearly with the references [1]. There is no need of external component therefore, the integration of the necessary reference capacitors on-chip remove the external decoupling [1].

The external reference is used to control the full-scale input range of the converter [1]. The MODE pin 23 is used for selecting the internal or external reference modes by programming the serial interface register H bit [1].

III. Output Clock Position Programmability:

About default position the output clock is stimulated in the LVDS and CMOS modes [1]. A-register bits (with serial interface)

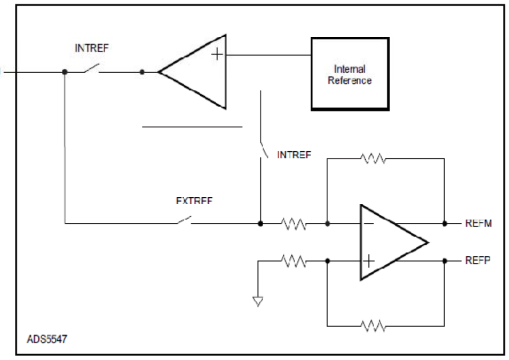


Figure 3: Reference Block [1]

or SEN pin (with parallel interface) is used to complete this functionality [1]. The transaction in the setup and hold timing which directs to get consistent data thus sanctioning it [1]. The different timing is set to transfer the clock edge afterwards or before than input clock edge or is set to default [1]. When data is changed, we can arrange the output clock edge as needed [1].

IV. OUTPUT DATA FORMAT:

We can assist two output data arrangements that are two's complement and also the offset binary [1]. We can choose them by means of B-register bit in the serial interface [1].

In offset binary format data comes out of ADC device in normal binary form of 14 bits while in 2's complement format data available at the output is 2's complement of the converted data from ADC block [1].

V. SIMULATION AND RESULTS

The ADS5547 Modeling can be tested by giving different input samples on ADC input pins and then checking the digital output on ADC output pins and converting it back to see whether output is same as differential input or not. By giving different values on control signals ADC can be configured to generate normal binary output, 2's complement output, output clock timing can be changed. The code for ADC is written in VHDL languages and same is tested using test bench that is also written in VHDL using Model Sim.

The basic thing in ADC modeling is to generate correct digital output bits as per the differential input signal and reference voltage signal. Now we will check all the test case in serial mode and view the waveform to verify the ADC functionality.

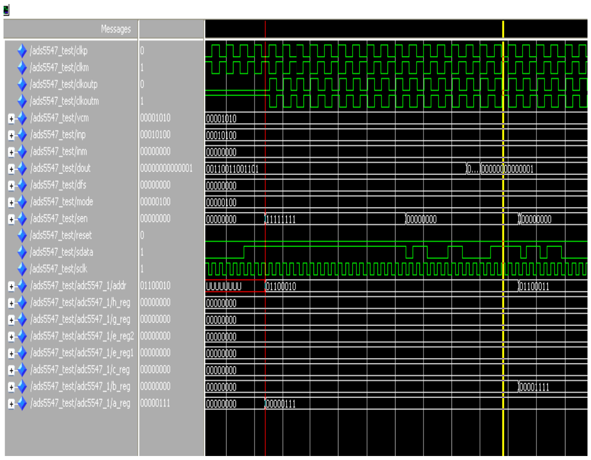
Serial mode Configuration and the ADC Operation:

Since in serial mode the ADC operation will be controlled by different registers values we have to actually write values on ADC register with appropriate address. For this mode RESET signal should be tied to low during write operation on ADC register. First there will be active high pulse on RESET pins to reset all register in the starting as we did in our test bench and then RESET should be tied to 0 only. SEN should be also low if we have to write any values on register and so in our test bench we are driving 0v ("00000000") on it. The time difference between RESET goes low and SEN become enable is as per RESET timing and so we have to take 30 ns.

Now on every negative edge of SCLK, SDATA will be latched into ADC device and after 16 bit are latched; first 8bits will decide about the address and rest 8 bit will write into that address register. So we have to the send bit in appropriate technique if we want to write any particular value in any of the ADC register. After we write a value to the ADC register, we disable the serial interface by putting high values on SEN ("11111111") [1].Now we will see how we have written the values on different ADC register to configure it in different modes.

A-Register Write:

A-register controls the output clock edge and like in our test bench to shift the clock edge by 2/12 clock cycle we have given the values on the SDATA input pin accordingly first to generate the address for A-register that is h'62 ("01100010") and then we are writing "00111" on Clock edge position programming bits.



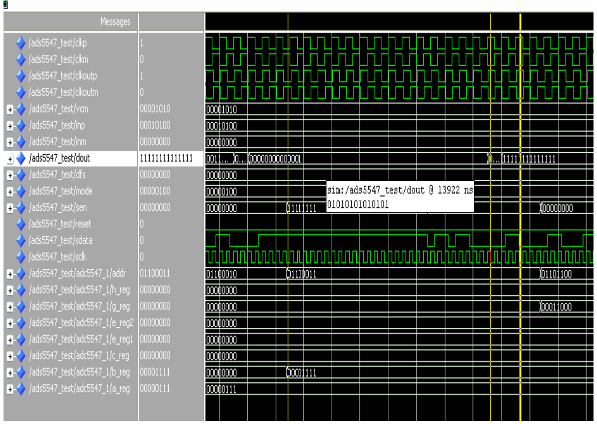
```
Messages
/ads5547_test/clkp                 0
/ads5547_test/clkm                 1
/ads5547_test/clkoutp              0
/ads5547_test/clkoutm              1
/ads5547_test/vcm                  00001010
/ads5547_test/inp                  00010100
/ads5547_test/inm                  00000000
/ads5547_test/dout                 00000000000001
/ads5547_test/dfs                  00000000
/ads5547_test/mode                 00000100
/ads5547_test/sen                  00000000
/ads5547_test/reset                0
/ads5547_test/sdata                1
/ads5547_test/sclk                 1
/ads5547_test/adc5547_1/addr       01100010
/ads5547_test/adc5547_1/h_reg      00000000
/ads5547_test/adc5547_1/g_reg      00000000
/ads5547_test/adc5547_1/e_reg2     00000000
/ads5547_test/adc5547_1/e_reg1     00000000
/ads5547_test/adc5547_1/c_reg      00000000
/ads5547_test/adc5547_1/b_reg      00000000
/ads5547_test/adc5547_1/a_reg      00000111
```

Figure 4: Simulation result of A Register

By default ADC will be in DDR LVDS mode with 2's complement output. Since we have still given same input signal of 5v ("00010100") and 0v ("00000000") on positive and negative pins respectively the ADC block will generate output of all 14 1's and with 2's complement the output from ADC device would be "00000000000001". The clock edge in this case would be at default position as you can verify from the waveform as shown in the figure 4.

A. B-Register Write:

B-register controls the output data format (2's complement or straight binary). Now to configure the output format as straight binary we have to write 1 on data format bit of B-register . For this first we need to send first 8 bits on SDATA such that it makes the address of B-register that is h'63 ("01100011") and then we will write on the data format bit. Since ADC inputs voltage are still same just that output format is changes so the output from the ADC device will be all 1's as offset binary format as shown in the figure 10. The clock position will still remain the same [1].



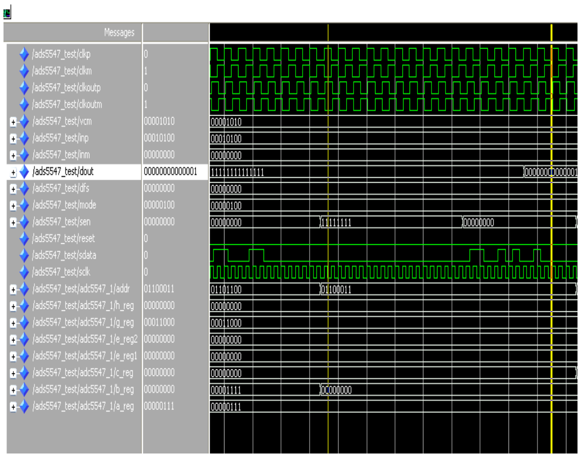
```
Messages
/ads5547_test/clkp                 1
/ads5547_test/clkm                 0
/ads5547_test/clkoutp              1
/ads5547_test/clkoutm              0
/ads5547_test/vcm                  00001010
/ads5547_test/inp                  00010100
/ads5547_test/inm                  00000000
/ads5547_test/dout                 11111111111111
/ads5547_test/dfs                  00000000
/ads5547_test/mode                 00000100
/ads5547_test/sen                  00000000
/ads5547_test/reset                0
/ads5547_test/sdata                0
/ads5547_test/sclk                 0
/ads5547_test/adc5547_1/addr       01100011
/ads5547_test/adc5547_1/h_reg      00000000
/ads5547_test/adc5547_1/g_reg      00000000
/ads5547_test/adc5547_1/e_reg2     00000000
/ads5547_test/adc5547_1/e_reg1     00000000
/ads5547_test/adc5547_1/c_reg      00000000
/ads5547_test/adc5547_1/b_reg      00001111
/ads5547_test/adc5547_1/a_reg      00000111

sim:/ads5547_test/dout @ 13922 ns
01010101010101
```

Figure 5: Simulation result of B Register

B. G-Register Write:

G-register select between DDR LVDS or Parallel CMOS output, so now to check the Parallel CMOS mode we have



Messages	
/ads5547_test/clkp	0
/ads5547_test/clkm	1
/ads5547_test/clkoutp	0
/ads5547_test/clkoutm	1
/ads5547_test/vcm	00001010
/ads5547_test/inp	00010100
/ads5547_test/inm	00000000
/ads5547_test/dout	00000000000001
/ads5547_test/dfs	00000000
/ads5547_test/mode	00000100
/ads5547_test/sen	00000000
/ads5547_test/reset	0
/ads5547_test/sdata	0
/ads5547_test/sclk	0
/ads5547_test/adc5547_1/addr	01100011
/ads5547_test/adc5547_1/h_reg	00000000
/ads5547_test/adc5547_1/g_reg	00011000
/ads5547_test/adc5547_1/e_reg2	00000000
/ads5547_test/adc5547_1/e_reg1	00000000
/ads5547_test/adc5547_1/c_reg	00000000
/ads5547_test/adc5547_1/b_reg	00000000
/ads5547_test/adc5547_1/a_reg	00000111

Figure 6: Simulation result of G Register

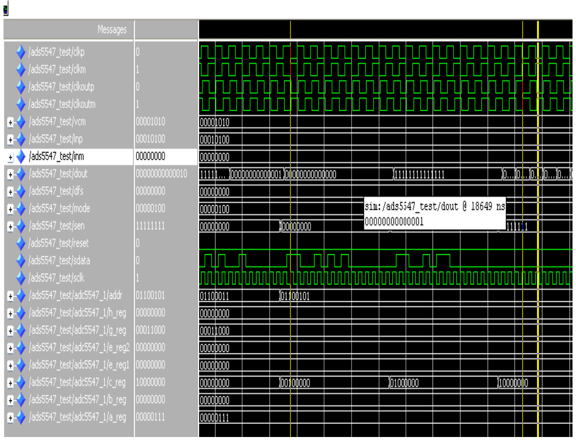
written the value in G-register accordingly. First we have given the 8 bits on SDATA such that it make address of G-register that is h'6C("01101100") and then write "11" on output interface bit to set it in Parallel CMOS mode as shown in the figure -6 [1]. Since input signal voltage are same digital output data will also remain same just the only difference will come in output clock edge, now it will shift by 2/12 clock cycle time. Again we have configured ADC such that now it again generates the output in 2's complement format and for this we again have to clear the data format bit of B-register as we did in test bench [1].

C. C-Register Write:

C-register is used for giving different test patterns on output of ADC. So in the test bench we have written different values on C-register to verify different test pattern like all 0's, all 1's, toggle etc. First we have sent 8 bits on SDATA as h'65, one bit on each serial clock cycle to make the address for C-register and after that we have written different values on 3 MSB bits of C-register for different test pattern [1]. The output of ADC device would be one of the test pattern regardless of whatever values coming from the ADC module as shown in the figure 7. In the test bench we have verified all 0's, all 1's and ramp test pattern.

D. E-Register Write:

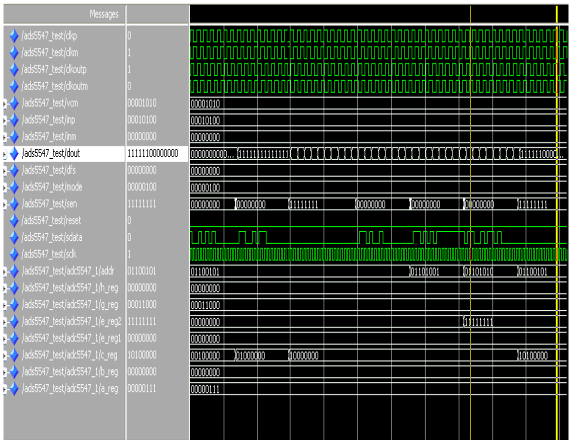
E-register is used to give custom pattern on the ADC output while the ADC is configured for custom test pattern on the ADC output using C-register. For this we have written custom pattern on register-E (E1 & E2) and checked that same custom pattern is coming from the ADC output. For this we send the SDATA such like that to make address h'69 and h'6A for E-register and then we have written "101" on C-register test pattern select bit as shown in the figure 8[1]. The output of the ADC device would be "11111100000000" as we given the same custom pattern in E-register.



Messages	
/ads5547_test/clkp	0
/ads5547_test/clkm	1
/ads5547_test/clkoutp	0
/ads5547_test/clkoutm	1
/ads5547_test/vcm	00001010
/ads5547_test/inp	00010100
/ads5547_test/inm	00000000
/ads5547_test/dout	00000000000010
/ads5547_test/dfs	00000000
/ads5547_test/mode	00000100
/ads5547_test/sen	11111111
/ads5547_test/reset	0
/ads5547_test/sdata	0
/ads5547_test/sclk	1
/ads5547_test/adc5547_1/addr	01100101
/ads5547_test/adc5547_1/h_reg	00000000
/ads5547_test/adc5547_1/g_reg	00011000
/ads5547_test/adc5547_1/e_reg2	00000000
/ads5547_test/adc5547_1/e_reg1	00000000
/ads5547_test/adc5547_1/c_reg	10000000
/ads5547_test/adc5547_1/b_reg	00000000
/ads5547_test/adc5547_1/a_reg	00000111

sim:/ads5547_test/dout @ 18649 ns
00000000000001

Figure 7: Simulation result of C Register



Messages	
/ads5547_test/clkp	0
/ads5547_test/clkm	1
/ads5547_test/clkoutp	1
/ads5547_test/clkoutm	0
/ads5547_test/vcm	00001010
/ads5547_test/inp	00010100
/ads5547_test/inm	00000000
/ads5547_test/dout	11111100000000
/ads5547_test/dfs	00000000
/ads5547_test/mode	00000100
/ads5547_test/sen	11111111
/ads5547_test/reset	0
/ads5547_test/sdata	0
/ads5547_test/sclk	1
/ads5547_test/adc5547_1/addr	01100101
/ads5547_test/adc5547_1/h_reg	00000000
/ads5547_test/adc5547_1/g_reg	00011000
/ads5547_test/adc5547_1/e_reg2	11111111
/ads5547_test/adc5547_1/e_reg1	00000000
/ads5547_test/adc5547_1/c_reg	10100000
/ads5547_test/adc5547_1/b_reg	00000000
/ads5547_test/adc5547_1/a_reg	00000111

Figure 8: Simulation result of E Register

VI. Conclusion

The ADC modeling is carried out in Model Sim. The functionality of the ADS5547 is achieved. The timing specifications such as RESET, LVDS are taken into account and the data output which is achieved is according to the specifications. The operation of serial mode configuration is accomplished by doing the various register configurations like A-Register, B-register, G-Register, C-Register, E-Register. All the implementation specified is verified by writing down the test bench for each operation as shown in the waveform

References

[1] ADS5547 data sheet. http://www.ti.com

[2] J Bhasker. A VHDL Primer. Third Edition, Pearson Prentice Hall 2005.

[3] Science Direct, VHDL behavioural modeling of pipeline analog to digital converts by Antonio J. Acosta. http://www.sciencedirect.com/science

[4] VHDL Org, Model Guide. http://www.vhdl.org/rassp/vhdl/guidelines/ModelGuide.pdf

[5] Analogue-to-Digital Converters. http://www.microlink.co.uk/a-d.html

[6] Kenneth L. Short. VHDL For Engineers. Pearson International Edition, Pearson Prentice Hall 2009, p445.

[7] Philips E. Allen and Douglas R. Holberg. CMOS Analog Circuit Design. 2nd Edition, Oxford University Press 2004.

[8] G. Chioboli, C.Morandi "ADC modeling and testing" IEEE Instrumentation and Measurement Technology Conference Budapest, Hungary, pp 1992-1999 May 21-23, 2001.

[9] Peralias, E.; Acosta, A.J.; Rueda, A.; Huertas, J.L "VHDL- based behavioural description of pipeline ADCs" Circuits and Systems,200. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE international Symposium, Geneva, Switzerland pp 681-684 vol4.

[10] Balazs Fodor and Istvan Kollar, Fellow, IEEE "ADC TESTING WITH VERIFIACTION" IEEE Transaction on Instrumentation And Measurement, VOL. 57, NO. 12, DECEMBER 2008.