Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
ESPECIALIDAD EN DISEÑO DE SISTEMAS EN CHIP

A predictive control unit for a Low Power 10-bit 200kS/s SAR ADC with Adaptive Conversion Cycle oriented to Audio Applications

---

Trabajo recepcional que para obtener el grado de
Especialista en Diseño de Sistemas en Chip

Presenta: **MARIO ALBERTO MORENO CONTRERAS**

Director **CUAUHTÉMOC RAFAEL AGUILERA GALICIA**

Tlaquepaque, Jalisco. Octubre de 2020.

This work is dedicated to my family, for their patience and the support they gave me through the time of this Project.

Acknowledgments

I am incredibly grateful to the ITESO for give me the opportunity of study this specialty. To Conacyt for providing the economic support and the opportunity to study this specialty.

To my coordinator, Omar Longoria Gándara for the invitation to participate in this fabulous project. Also, I am grateful with Iván Martínez, my colleague in this project. Thanks, Iván for your support in difficult moments and the funny moments across this year of collaboration and job.

I am especially grateful to Dr. Cuauhtémoc Aguilera. Thank you for your advice, your incredible patience, the lessons in this design process and the fabulous help when I was in problems. This design is not mine only, then is yours.

Finally, I am grateful with the life because since last October 2019 I have been sick, but the life gives me the opportunity to continue this project with courage and dedication.

List of acronyms and abbreviations

ADC	Analog to Digital Converter
SAR	Successive Approximation Register
PSAR	Predictive SAR control unit
Vin	Analog input voltaje
Vref	Analog reference voltage
S&H	Sample and Hold
DAC	Digital to Analog Converter
Ptin	Pulse time in
Ptref	Pulse time reference
TAo1	Time amplifier out 1
TAo2	Time amplifier out 2
VTC	Voltage to time converter
eoc	End of conversion
CNT	Count time amplifier output differences
TA	Time amplifier
din	Digital data in
clk	clock
FSM	Finite State Machine
PB	Predictive Bit Register
rst	reset
rst_DAC	Reset DAC
AgtB	A signal greater than B signal
prst	Preset
A_B	A signal less B signal
neg_flag	Negative value flag
zero_flag	Counter zero value flag

List of Figures

Contents

Abstract

This document presents the implementation of a new predictive algorithm for a 10-bit analog-to-digital converter based on the successive approximation register (SAR), technique. The predictive algorithm (PSAR), is designed to predict a specific number of consecutive bits depending on the difference between the analog input voltage (Vin) and the reference voltage (Vref), provided by the DAC used in the successive approximation conversion technique. The PSAR is able to predict from 3 bits to 8 bits in a single conversion cycle.

The typical SAR conversion employs a comparison between Vin and Vref on the voltage domain. In the case of the proposed PSAR, the Vin and Vref are converted to a time pulse width proportional to the magnitude of the inputs and the comparison of both is made in the time domain. The time difference between these two pulses is compared and registered by a counter to determine the total number of consecutive bits to predict.

The proposed PSAR requires 13 clock cycles to perform conversion if there are not more than 2 consecutive bits to predict. The PSAR reduce 21% of the average conversion time when an ascendant ramp with the 1024 possible input values is applied.

Introduction

An ADC is a circuit that converts a continuous analog input voltage to a discrete binary word. ADCs are high in demand due to the increase of numerous mixed-signals systems. There are several conversion techniques in which analog-to-digital conversion can be done such as pipeline, delta-sigma, and flash. Nevertheless, Successive-Approximation technique is one of the most popular because of its accuracy, moderate conversion speed as well as low power consumption [1]. It is generally a feedback system that applies a trial-and-error algorithm to obtain a proportional digital word to an analog input voltage.

The SAR ADC

A SAR ADC (Figure 1) consists of the following design blocks:

i. Sample and Hold circuit
ii. Comparator
iii. N-bit SAR logic
iv. Digital-to-Analog Converter (DAC)

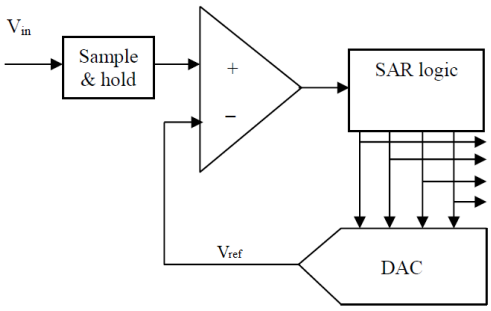

Figure 1. Block diagram of typical SAR-ADC.

The conversion sequence of a SAR ADC is listed below:

i. The system samples the analog input V_{in} in a sample and hold circuit. Simultaneously, the SAR Logic resets the DAC.
ii. The MSB bit is set in the DAC by the SAR logic. The DAC output is known as Voltage Reference (V_{ref}).
iii. The sampled V_{in} and V_{ref} are compared by the comparator circuit. If V_{in} is greater than V_{ref} the Comparator output is '1' logic, else the output is '0'logic.
iv. The SAR logic saves the comparator output in the MSB position of the SAR output register and sets the bit MSB-1 and presents this partial conversion to the DAC input.
v. The conversion continues for MSB-1, MSB-2, and finishes on the LSB bit.

The Predictive SAR-ADC proposal

Our proposal is a modification of the conventional SAR-ADC algorithm (Figure 2) to perform a conversion in fewer clock cycles. This is possible by a prediction of the consecutive 1's or 0's in the conversion result. Such prediction is based on the difference between V_{in} and V_{ref} signals. Our proposal is capable of predict 1 bit or 3 to 9 consecutive equal bits on a single clock cycle.

The conversion sequence of the proposed system is listed below:

i. The system samples the analog input V_{in} in a S&H circuit
ii. V_{in} and V_{ref} are converted to a time-domain signal via the Voltage to Time Converter (VTC). The outputs of the VTC are Pt_{in} and Pt_{ref}.
iii. The Time Amplifier (TA) extends the time difference between Pt_{in} and Pt_{ref}. The outputs of the TA are TAo1 and TAo2, these are send to the Arbiter and the Counter.
iv. The Arbiter determines in the time domain if Vin is lower or bigger than Vref to set or clear the corresponding bits of the current conversion step.

v. The SAR Logic uses the information from the Arbiter and the Counter to predict equal-consecutive bits and to adapt the conversion cycles

The propose predictive SAR improves the overall conversion time and therefore power dissipation per conversion cycle.

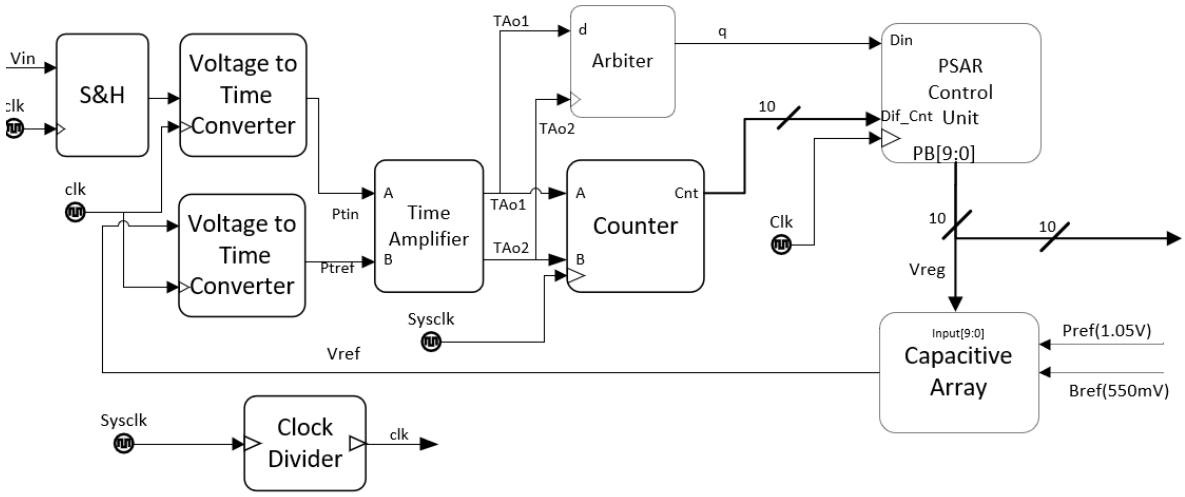

Figure 2. Block diagram of the predictive SAR proposal.

Design of SAR Control Unit for the Predictive SAR-ADC (PSAR-ADC)

The Predictive SAR Control Unit

The Predictive SAR Control Unit (PSAR) controls the analog-to-digital conversion process utilizing the proposed SAR-ADC algorithm. This module can predict from one to nine bits in only one clock cycle when these bits are equals and consecutive. The PSAR has the following inputs: din, start, Dif_CNT, rst, and clk. The outputs signals are eoc, PB and rst_DAC. The block diagram of the PSAR is shown in Figure 3.

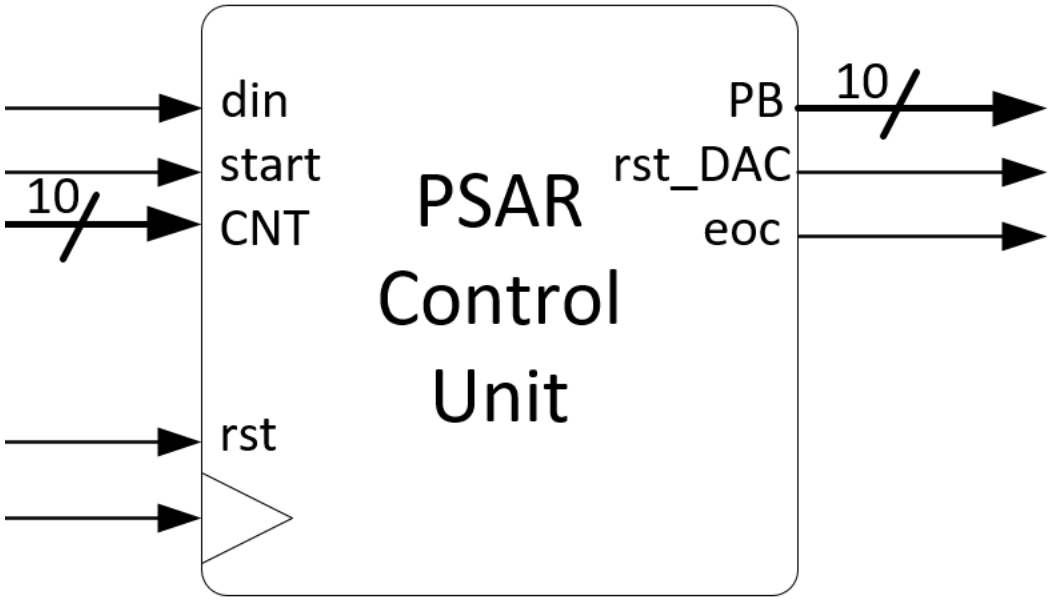

Figure 3. The black box of the predictive SAR control unit (PSAR).

The functionality of the PSAR is the following:

I. din: This input receives the q output of the Arbiter. This input indicates if TAo1 comes first than TAo2 or TAo2 comes first than TAo1. The din is used by the PSAR to predict a single bit or multiple-consecutive bits in high (1) or low (0).

II. start: External signal that enables the PSAR-ADC and it starts the conversion of V_{in} into a digital value.

III. CNT: This is a 4-bits input, which receives the Cnt output of the Counter as shown in Figure 2. The value of this input indicates to PSAR the number of differences between

Vin and Vref to predict. The minimum value of this bus is 1 and the maximum value is 512.

IV. rst: Reset of the PSAR-ADC system. This input is low activated and interrupts the conversion process and restart the PSAR-ADC.

V. clk: Clock signal of the PSAR ADC system. The clock is running at 200 kHz.

VI. PBR: The Predictive Bit Register (PBR) contains the temporal conversion and sends it to the DAC. The DAC converts the digital information into a discrete voltage value (Vref) to be compared with the analog input.

VII. eoc: This signal is activated at the end of a conversion cycle, this indicates to other device that the conversion is ready.

VIII. rst_DAC: This signal resets the DAC at the beginning of each conversion.

The Predictive SAR-Control Unit Architecture

The PSAR consist of a Finite State Machine (FSM), a 10-bits Prediction Bit Register (PBR), a 4-bit Comparator, a 4-bit subtract, a 4-bit Counter with Preload, and a Filler decoder (Filler). The PSAR architecture is shown in Figure 4.

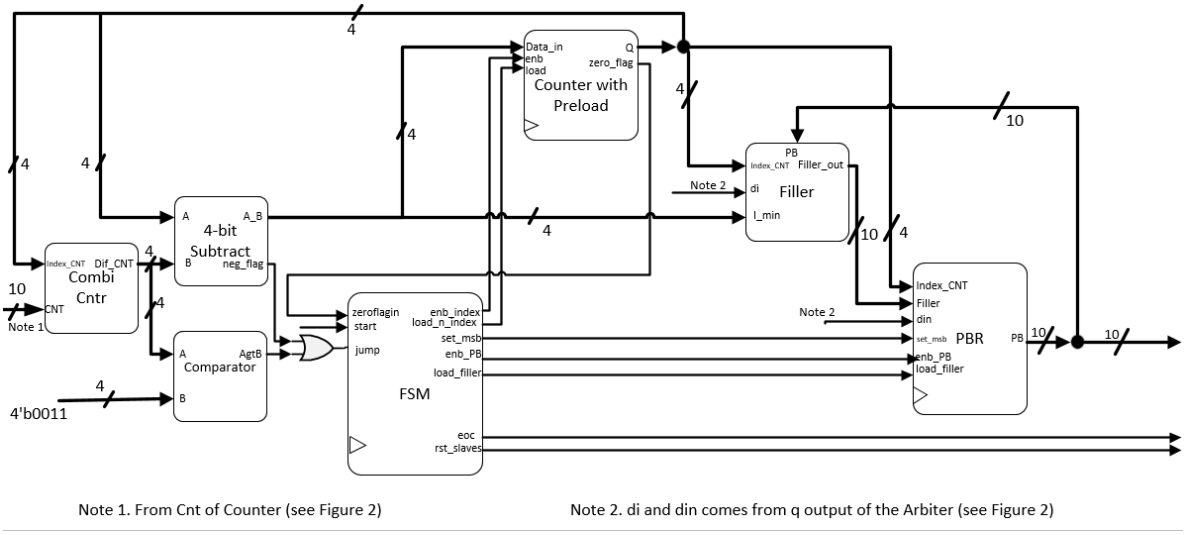

Figure 4. The block diagram of the PSAR architecture

The Combinational Counter Decoder

The PSAR receives the time differences between TAo1 and Tao2 and converts the difference in bits to predict in a PSAR cycle. The range of the time difference is quantized by the Counter module

To solve it, I proposed the following digital design to add in the PSAR: A Combinational Counter Decoder (Combi_Cntr). The Combi_Cntr is connected at the input of the PSAR PSAR control unit**.** The Combi_Cntr takes the current bit position to be converted in the conversion and the Cnt value from the Counter to calculates the number of the predictive bits to converts and send this calculus to the Dif_CNT bus of the PSAR control Unit. This module is shown in the Figure 5.

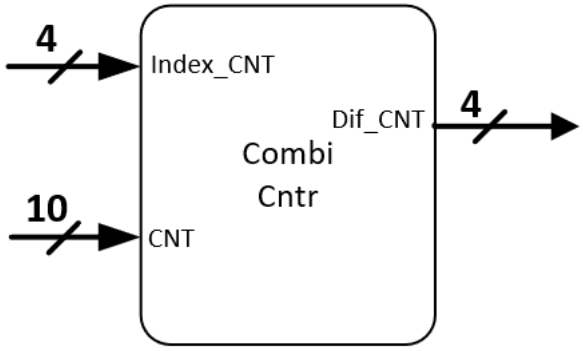

Figure 5. The Combi Cntr black box

The 4-bit Comparator

The 4-bit Comparator is used to detect if an analog to digital conversion has consecutive equal bits to predict. If a CNT is equal or greater than 3 counts, the comparator indicates a predictive conversion to the FSM. The 4-bits Comparator has two input pins and one output pin called A, B, and AgtB respectively as shown in Figure 6.

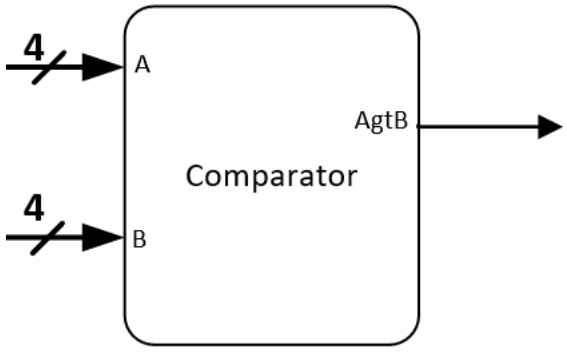

Figure 6. The black box of 4-bits Comparator

The function of Comparator pins is:

I. A: A 4-bit input. This input pin receives the calculated Dif_CNT and compares it with the pin B.
II. B: The value of this 4-bit input is set to decimal 3. The value of this input is compared with the value of the input pin A.
III. AgtB: This output pin is set to high if the input A is equal or greater than the input B and indicates a predictive conversion to the FSM.

The 4-bits Subtract module

The PSAR needs to calculate a new index to set in the PBR register. The 4-bits Subtract module calculates the new index by subtracting the input signals Q and Dif_CNT. The result of the operation is the signal A_B and it cannot be less than 1. If the result is negative the neg_flag output is high, and the FSM does not perform the predictive conversion. The block diagram of the 4-bits Subtract is shown in Figure 7.

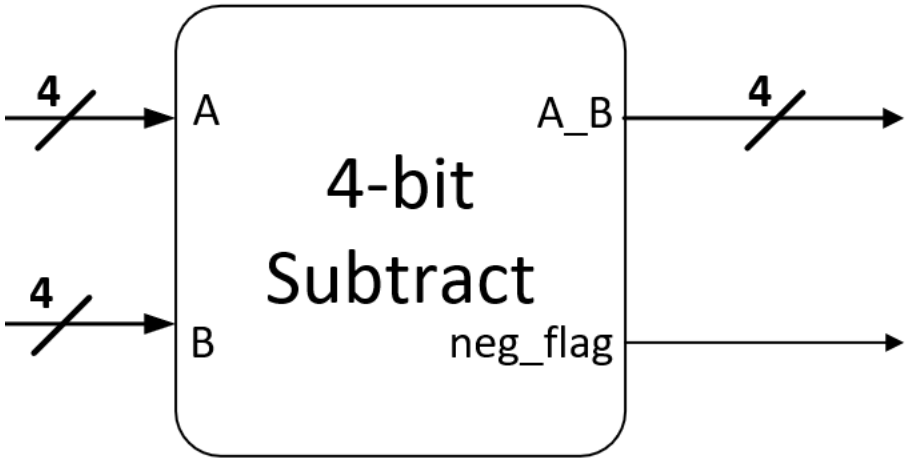

Figure 7. The black box of 4-bits Subtract module

Counter with preload

The Counter with preload is very fundamental in the architecture because the count of this counter is the conversion-bit position in the PBR. It performs a descendent count from nine to zero.

The Counter with preload module has five inputs: clk, prst, a 4-bit Data_in, enable, and load. The outputs are zero_flag and the 4-bit output Q. The block diagram of the Counter with preload is shown in the Figure 8.

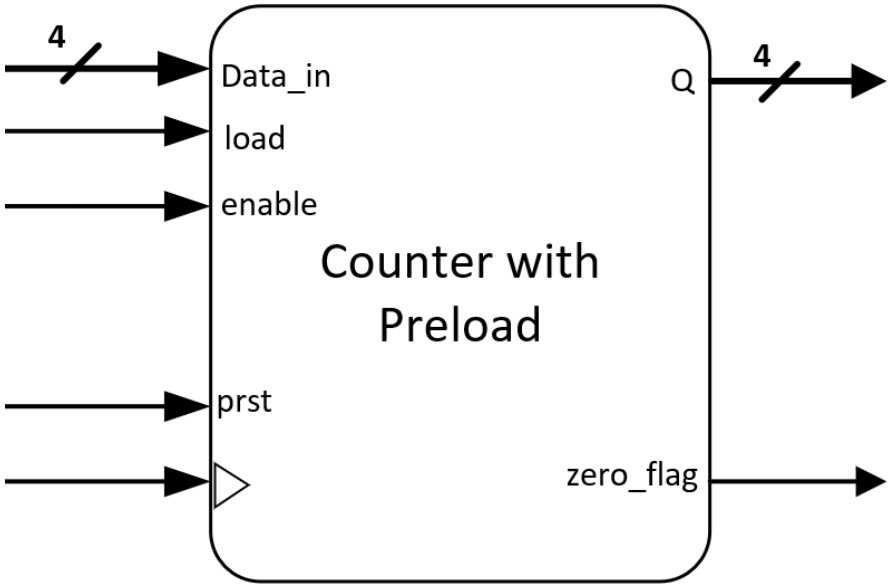

Figure 8. The black box of the Counter with preload.

The function of the Counter with preload pins is the following:

I. Data_in: This 4-bit bus input receives the new index calculated in the 4-bit Subtract A_B output and this is loaded in Q when the load signal of the FSM is high.

II. load: The input receives a signal in high state from the FSM to load the calculated predictive index in Q.

III. enable: This input signal is high when a conversion is in progress and enables the Counter. The signal comes from the FSM.

IV. prst: This signal allows set the value of the count Q to nine at the start of the analog to digital conversion.

V. Q: This 4-bits output, registers the current predictive index. The maximum value of this output is nine and the minimum value is zero.
The value of Q is automatically reset to nine at the end of each data conversion.

VI. zero flag: This signal is connected to the FSM and it is high when the value of Q is zero to finish the analog to digital conversion.

At the beginning of each V_{in} conversion, the count is set to nine because it is the MSB bit in the PBR. When the conversion is running the count is decreased until zero for a standard conversion. If the PSAR-ADC is performing a predictive conversion, the Counter with preload loads the predictive index calculated in the 4-bit subtract. Also, the value of Q is automatically reset to nine at the end of each Vin conversion.

The Filler decoder

The Filler decoder has four inputs (Figure 9): di, Index_CNT, PB and I_min, and a single output called Filler_out.

The predictive conversion is based in a decoder that converts the count Dif_CNT in a binary word with the predicted bits ready to be loaded in the PBR. The output register is only loaded into the PBR if a predictive conversion is running.

The function of Index_CNT is to receive the Q signal of the Counter with preload and the I_min is received from A_B and represents the new predictive index to the PBR. Also, the current PBR output is received and is filled with Index_CNT bits with the din value and loaded in the filler_out.

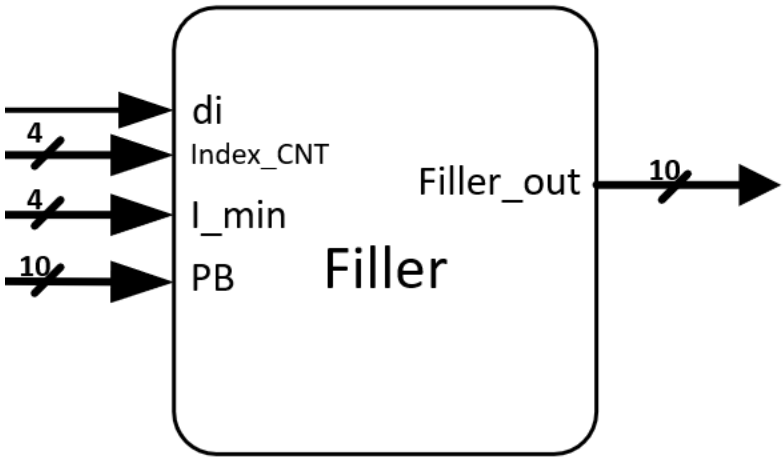

Figure 9. Black box of the Filler Decoder.

The Predictive Bit Register (PBR)

The PSAR needs to predict and register each intermediate conversion to compare the Vref with the analog input Vin either in a standard or predictive conversion. This function is performed by the PBR. The block diagram of the PBR is shown in the Figure 10.

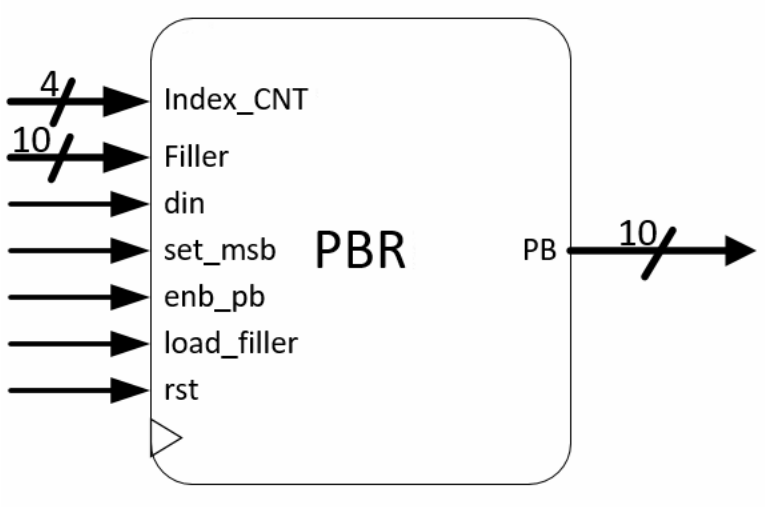

Figure 10. The black box of the PBR

The PBR has the following inputs:

I. Index_CNT**:** This 4-bits input is the predictive index calculated in the 4-bit Subtract. In the conversion process bits are set or clear according to this index position.

II. Filler: This 10-bits input is the new predictive PBR calculated in the Filler and loaded into the PBR register.

III. din**:** This input receives the q value from the Arbiter and is used by the PBR to set a current bit conversion with the value of din**.**

IV. set_msb**:** This input is received from the FSM and sets high the MSB in the PBR if it is high

V. enb_pb: This signal is received from the FSM and enables the PBR functions.

VI. load_filler: This control signal comes from FSM**.** The PBR loads the value of the Filler Decoder if the load_filler is high.

VII. rst: Reset of the module and is connected to the SAR-ADC's system reset.

VIII. clk: This input is the clock signal, of the SAR ADC's system and it is running at 200 kHz.

Finally, the module has an only 10-bits output: PB output register. The PB register saves the conversion process of every bit in the PSAR-ADC conversion. The PB is sent to the DAC and update the Vref voltage.

The Finite State Machine (FSM)

The PSAR FSM is the fundamental module in this architecture: controls the conversion in each clock cycle. Also, the FSM performs the predictive conversion.

The FSM is a mealy FSM and has five inputs (Figure 11): start, jump, zeroflagin, rst, clk**.** The outputs of the PSAR FSM are correct_index, enable_Index, load_n_Index, en_pb, load_filler, setmsb, rst_DAC, eoc.

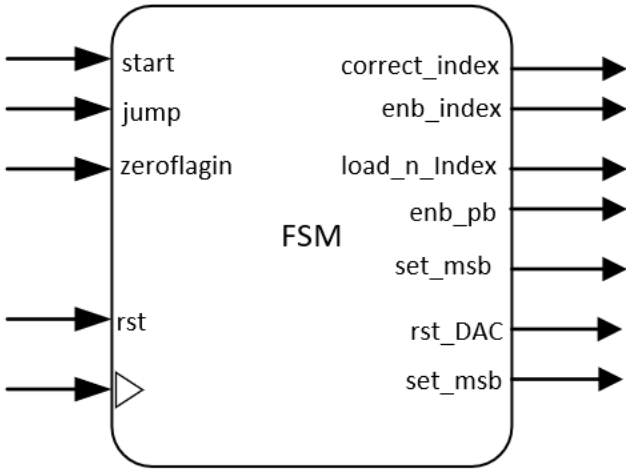

Figure 11. The PSAR FSM black box diagram

The functionality of the PSAR FSM inputs is the following:

I. start**:** External signal to start the PSAR-ADC system to perform the V_{in} conversion.

II. jump**:** This input comes from the OR operation between zero_flag and AgtB**.** The FSM performs the predictive conversion if this input is high. Otherwise, the FSM performs the standard conversion.

III. zeroflagin**:** This input comes from Counter with preload and indicates to the FSM the end of the V_{in} conversion.

IV. clk**:** This input is the clock signal of the PSAR-ADC. In every clock cycle the FSM performs a predictive conversion of one or more bits.

V. rst**:** Reset of the SAR-ADC system.

The outputs have the following functions:

I. correct_index: This signal is only used to convert the last bit in the PB output of the PBR.

II. enable_Index**:** This signal is connected to Counter with preload and enable it to calculate the new predictive index and perform the V_{in} conversion.

III. load_n_Index**:** The signal is connected to the Counter with preload to indicate the load of the predictive index in a conversion of three or more bits

IV. enb_pb**:** The value of this signal is set high only to activate the conversion in the PBR

V. load_filler**:** The signal makes the PBR capture the new predicted conversion built in the Combinational Filler output

VI. setmsb**:** This signal is connected to PBR register to set the MSB in the PBR. It sets the MSB in the PBR when its value is high.

VII. rst_DAC**:** This output signal resets the DAC in every V_{in} conversion sending a low signal.

VIII. eoc**:** This output is set to high if the V_{in} conversion is ready.

The FSM states

The FSM controls the process to convert a V_{in} sample into a digital word. This is performed using six states. Each state is executed in one clock cycle: Idle, ModeSelect, NormalMode, PredicMode and EOC. The state diagram is shows in the Figure 12.

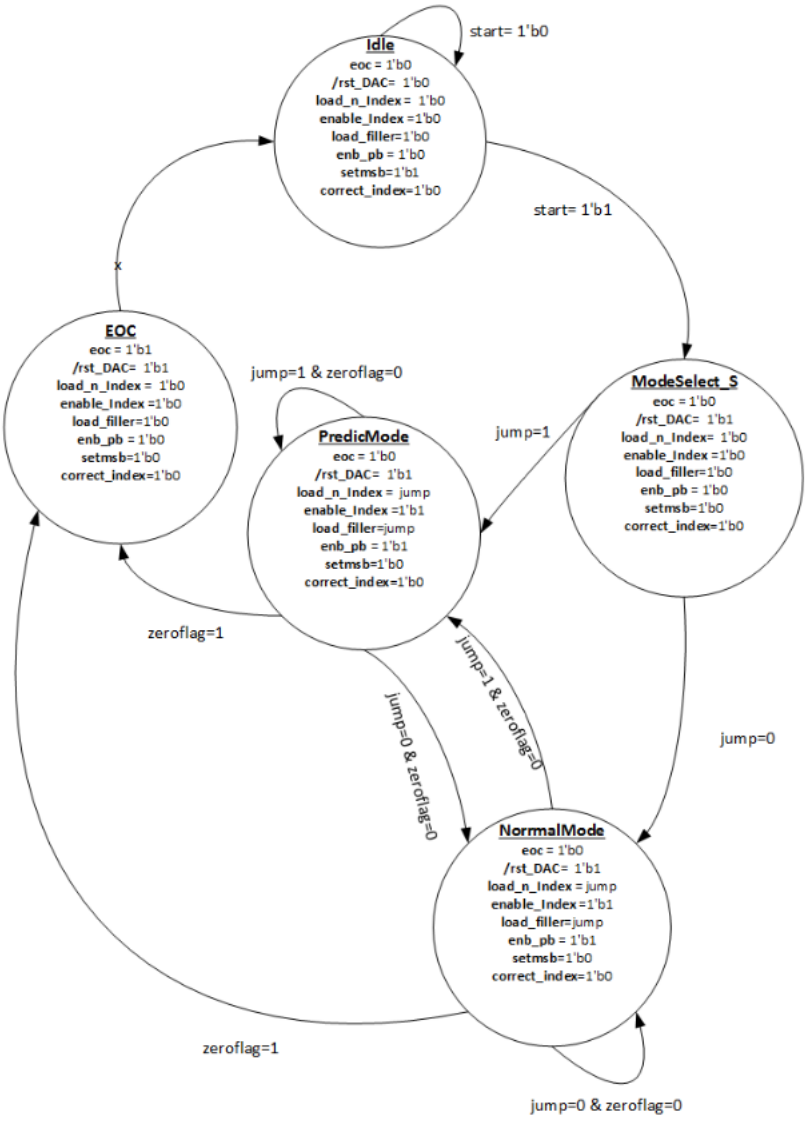

Figure 12. The FSM states diagram

The Idle state is always waiting for the start signal to change to the ModeSelect state. Also, in the Idle state the FSM resets the DAC, set the MSB in the PBR. In the ModeSelect, the FSM is waiting the first comparison of the PSAR ADC and receives the jump signal status. If the jump signal is high, the next state is PredicMode; else, the next state is NormalMode.

In the NormalMode the PSAR performs bit by bit conversion. At this state, the output signals enb_pb and enable_Index are high. The values of the outputs load_n_Index and load_filler has the value of jump signal. The NormalMode can change to PredicMode if jump signal is

set to high and the zeroflagin is set to low. Finally, the NormalMode pass to EOC if the zeroflagin is set to high.

In the PredicMode state, the PSAR performs the predictive conversion. At the PredicMode state, the output signals enb_pb and enable_Index are high. The values of the outputs load_n_Index and load_filler have the value of jump signal. The PredicMode can change to NormalMode if the jump signal is low and the zeroflagin is low. Finally, the PredicMode pass to EOC if zeroflagin is set.

Finally, the EOC state only sets the end of conversion signal (eoc) to notifies that data converted is ready in the output of the PBR register. This state only works in a one clock cycle and automatic pass to IDLE state.

SAR Control Unit RTL

The PSAR is described using 2001 Verilog standard [4]. The Verilog model is synthesized using Quartus Prime Software and the EP4CE115F29C7 FPGA.

The total synthesized logic elements are 157, and the total employed registers are 19. The RTL schematic diagram is shown in the Figure 13.

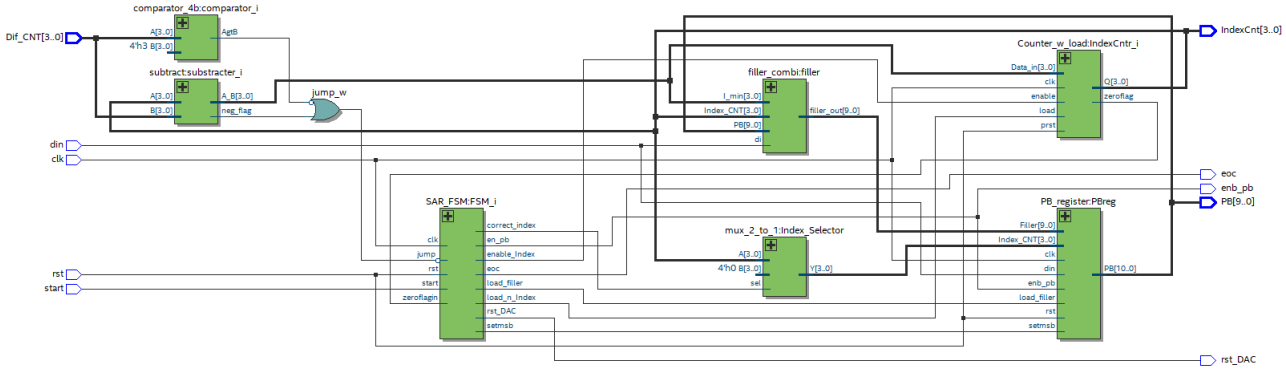

Figure 13. The RTL schematic diagram of the PSAR.

The Modified PSAR Control Unit simulation

The simulation of the PSAR is performed in the Modelsim Software. For this simulation, a test-bench file is created. In this file the PSAR is verified with the all possible inputs in a digital format from 0 to 1023. The digital values are used because the the VTC and TA are analog full-custom modules and their Verilog models are not available. Also, the digital input value is used as reference and allows compare this value with the PSAR conversion. The Figure 14. shows the simulation of a no predictive conversion for a Vin of 0.850585938 V. The binary value for this V_{in} is 00110011100. The test bench loads this input in the Dig_Vin test register (white signal). The value of the Dig_Vin is one-left-bit shifted every clock cycle to simulate the comparator behavior in a bit to bit SAR conversion.

The PSAR FSM states in the Figure 14 are shows in blue color. These states start in Idle state, I, continuous through ModeSelect state, M, Normal conversion state, N, and finishes in the EOC state, E. The entire conversion process is showed in the PB (Cyan signal), that starts with the MSB set to high and performs the standard PSAR conversion in the next ten clock cycles. Finally, the simulation shows the conversion and the final data converted in PB is equals to the Dig_Vin value. The Vin was converted in thirteen clock cycles.

Figure 14. A standard PSAR conversion simulation.

In the Figure 15, a Dig_Vin of 511 (0.874755859V) is converted by the PSAR. The simulation shows that conversions starts in the Idle state. The conversion continues with the ModeSelect state and changes to Normal conversion state. The PSAR performs the standard mode of the MSB bit. After, the PSAR performs 9-bit predictive conversion; the current state in the conversion is the Predictive Mode state. Finally, the PSAR enters the EOC state and finish the conversion. The Vin was converted in six clock cycles using the predictive conversion.

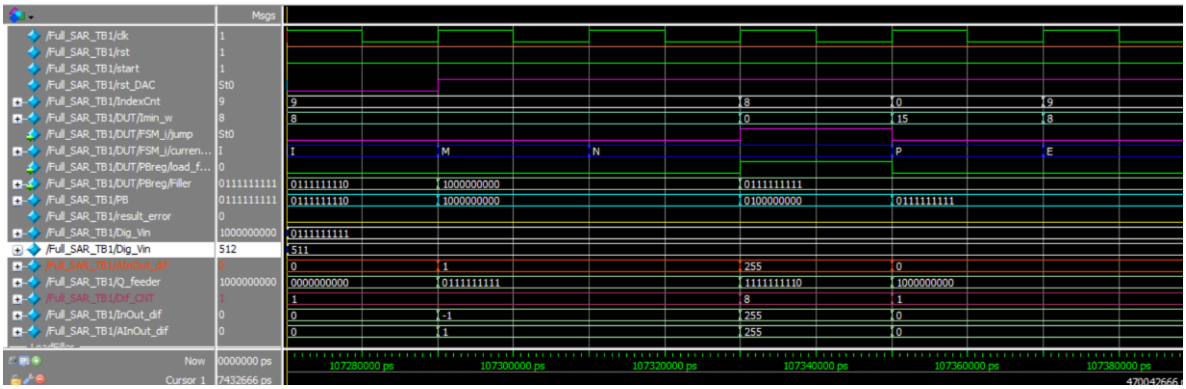

Figure 15. Simulation of a predictive PSAR conversion

The PSAR time efficiency

Comparing the total conversion time for all possible inputs in a standard SAR with the conversion time in the PSAR, the predictive conversion time is lower than the standard SAR conversion.

In this architecture, the standard SAR conversion takes 13 clock cycles (t_{CLK}): One cycle in the Idle state, one cycle in the Mode select state, ten clock cycles in the standard SAR conversion and the last clock cycle in the End of conversion state. To calculate the total time of the SAR T_{conv} at 200kHz it is necessary to use the following equation:

$$T_{conv} = 13 * t_{CLK} \quad (1)$$

Were T_{conv} is the conversion time of SAR and t_{CLK} is the period of a clock cycle at 200 kHz (5 ns). The total time conversion of a standard SAR on this architecture takes 66.56us for 1024 possible values

The total time conversion for a predictive PSAR conversion in the simulation for 1024 values is 52.58 us. It implies a time reduction of 21% in comparison with the SAR conversion. Finally, the equation of the average time conversion for the PSAR is:

$$PT_{conv} = 13 * t_{CLK} * .79 \quad (2)$$

comparing this architecture in the standard conversion mode with the AD7298-1[7], the PSAR employs 13 t_{CLK} and the AD7298-1 employs 16 t_{CLK}. The big difference is the communication protocol: the AD7298-1 employs the protocol SPI, has comparator, control logic, DAC and the PSAR is only the control logic. Also, this architecture is compared with the Texas Instruments' ADS7039-Q1[8] the PSAR employs one more cycle than TI design (12 t_{CLK}). Finally, comparing the frequency of these three designs the ADS7039-Q1frequency is 28 MHz, the AD7298 frequency is 20 MHz and the PSAR is designed with a theoretical frequency of 200 kHz It is important to specify that the PSAR is a theorical control unit and needs the Comparator module, the DAC, S&H, Counter to determine if the 200 kHz of speed is possible.

Conclusions

This document presented the implementation of a predictive successive approximation register (PSAR) for a 10-bits analog-to-digital converter (ADC) that works in the time domain. The comparison between the analog input voltage (Vin) and the voltage reference (Vref) provided by the DAC is done by converting these voltages into pulse widths. The time difference between the pulse widths is proportional to the number of consecutive bits to be predicted. The proposed PSAR is able to predict from 3 to 9 consecutive bits in one conversion cycle; this characteristic reduces the average time conversion up to 21% when a ramp input with the 1024 possible values is applied

The PSAR utilize 13 clock cycles to perform a typical bit-to-bit conversion when more than 2 consecutive bits prediction is not possible. The above is a point to improve, considering that the Texas Instruments 10-bit successive approximation ADC (ADC ADS7039-Q1) with SPI serial output only requires 12 clock cycles to perform a conversion.

The proposed PSAR is a part of a predictive ADC project, so it needs to be integrated with other modules such as sample and hold (S&H), counter of differences, voltage-to-time converters, time amplifier and an arbiter to perform the system functional test. These modules have been developed by colleagues of the ITESO EDSEC.

References

[1] V. P. Singh, G. K. Sharma and A. Shukla, "Power efficient SAR ADC designed in 90 nm CMOS technology," *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, Noida, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343542.

[2] Rashid, M. (2000). *"*Analog-to-Digital- Converters*"*. En M. Rashid, Microelectronics Circuits. Analysis and Design pp. 879-881. Florida: Thompson.

[3] Baker, R. J. (2010). "The Successive Approximation ADC". En R. J. Baker, CMOS. Circuit design, Layout and Design pp. 1003-1007. Wiley.

[4] Kalmeshwar N. Hosur, Dr. Girish V. Attimarad and Dr. Harish M. Kittur "Design and Implementation of 10bit,2MS/s Split SAR ADC using 0.18um CMOS" *International Journal of VLSI design & Communication Systems (VLSICS)* Vol.6, No.3, India, June 2015 pp 16-18

[5] IEEE Computer Society. (2001). IEEE Standard Verilog® Hardware. USA: The Institute of Electrical and Electronics Engineers, Inc.

[6] Jaime Hernández, Esteban Martinez Guerrero and Cuauhtémoc Rafael Aguilera Galicia, "Implementing Time Amplifier for a Low Power SAR-ADC with Adapative Conversión Cycle for High Quality Audio Applications in 0.18um TSMC CMOS Technology" Internal Report, 2020.

[7] Analog Devices. "8-Channel, 1 MSPS, 10-Bit SAR ADC". URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD7298-1.pdf

[8] Texas Instruments (2017). "ADS7039-Q1 Small-Size, Low-Power, 10-Bit, 2-MSPS, SAR ADC".

URL: https://www.ti.com/lit/ds/sbas788a/sbas788a.pdf?ts=1600819402556&ref_url=https%253A%252F%252Fwww.google.com%252F2