IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 3, Issue 3 (Sep. – Oct. 2013), PP 23-31
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Multi-bit High Speed Current Mode Analog to Digital Conversion Algorithm

[1] Veepsa Bhatia, [1] Ruchi, [2] Neeta Pandey, [2]Asok Bhattacharyya
([1] Department of ECE, India Gandhi Delhi Technical University, Delhi, India)
[2] Department of ECE, Delhi Technological University Delhi, India)

Abstract: *This paper proposes a generic algorithm for k-bit Analog to Digital Conversion. The proposed algorithm is very fast method of conversion of an analog signal into k-bit digital output in a multi-bit parallel fashion. The algorithm employs m stages, each of which is capable of producing bits such that k/m bits are converted at the same time. Thus, this algorithm can be considered a k = n X m conversion method. The algorithm proposed has been tested for a 4-bit current mode ADC implementation. The simulations have been performed in 0.18µm CMOS technology at the supply voltage of 1.8V using P-SPICE.*

I. Introduction

In the past ten years, the field of high-speed Analog to Digital conversion ([1]-[6]) has grown extensively. In general, an analog to digital converter takes input from an analog system such as voice, scanned images, etc., converts it into the corresponding digital format and then introduces the digital output to digital network. Therefore an ADC is considered a mixed signal block.

The medical and imaging technologies have presented a need for ADCs with sampling rates of the order of mega samples to tens of mega samples per second (Ms/s)([7]-[10]). Earlier the most popularly employed high speed data converters were Flash ADCs. However, these flash type ADCs have limited resolution, often restricted to 8 bits. As the ADCs market pushed towards higher resolution ADCs, two-step and pipelined architectures come into existence and promised higher resolution while having a smaller chip area and consuming lesser power. However, these new architectures had a limited operational speed in comparison to their flash counter parts. To overcome this limitation, various schemes and algorithms have been proposed which can offer a high speed and high resolution advantage in one scheme.

In this paper we have proposed one such algorithm that can provide a higher resolution while still maintaining high speed. We have implemented a 4-bit current-mode ADC, employing the proposed algorithm, which provides higher resolution while still maintaining high speed, making this ADC design highly flexible and thus finding application in wide variety of real time fields.

The paper has been divided into the following sections- Section II describes the proposed ADC algorithm which can be generalised in both current and voltage mode and also elaborates upon the advantages which this proposed algorithm offers. Section III proposes a balanced ADC architecture and its advantages. Section IV explains in detail the structure and working of various components used to implement a generic ADC in current-mode employing this proposed algorithm. The proposed ADC scheme has been simulated for exemplary (2X2) structure in section IV. Simulation results of this exemplary ADC have been discussed in section V. The final section summarizes the paper and gives some significant features and advantages of the proposed ADC scheme.

II. Analog To Digital Conversion Algorithm

In the proposed ADC scheme, to obtain a digital output of k bits, conversion of n bits is performed in m parallel stages thus producing k= m x n bits. Since multiple bits are being produced from multiple stages in parallel fashion, hence we call it as a *parallel multi-stage flash type converter* ([11]-[16]).

Fig. 1 depicts the general block diagram of the proposed algorithm comprising m stages. Of these m stages, the first stage generate most significant n bits from the analog input signal while in the mth stage, least significant n bits are generated. Each stage comprises of 3 basic elements- comparators to compare the input analog signal to a prefixed reference value, encoder to encode the compared value as obtained from the comparator into the required digital code and finally a digital to analog converter (DAC) to reconvert the digital code obtain from the preceding stage into analog form for conversion of next set of n bits required from succeeding stage. This DAC is called the reconstruction DAC.

2.1 The Algorithm for Analog to Digital Conversion:

1. The value of 'k' is determined according to the application requirement.
2. The number of stages 'm', to be employed in the ADC is fixed.
3. The value of 'n' is computed as,

   $$n=k/m$$

   Thus, the number of bits to be produced at each stage is determined.
4. A step-size is calculated as,

   $$\text{Step size} = (M2-M1)/\,(2^{k}-1)$$

   Where [M1, M2] is the range of the input current.
5. Reference parameter value is employed at first stage to perform comparators function. It can be a voltage or current signal depending upon the mode of operation (Current or voltage mode) respectively. It can be calculated as:

   For comparator 1:

   $$P_{ref1}=step \times 2^{(n)}$$

   For comparator 2:

   $$P_{ref2}=step \times 2^{(n+1)}$$

   For comparator 3:

   $$P_{ref3}=step \times 2^{(n+3)}$$

   For last comparator (n+1) in the first stage:

   $$P_{ref(n+1)}= P_{ref1} + P_{ref2} + P_{ref3} +\ldots\ldots + P_{ref(n)}$$

6. The input parameter to the comparators can be calculated as-
   For the first stage, input to the comparators,

   $$P_{ci} = P_{in} - P_{refi}$$

   where i= 1, 2, 3, ……, (k/m +1).
   But for the rest of stages, input to comparators is:

   $$P_{ci} = P_{in} - (\, P_{const} + P_{dac}\,)$$

   where i= 1, 2, 3, …., (k/m +1)
   Here P_{ci} is the input to the comparator of the i^th^ stage and P_{dac} is the parameter generated from the DAC of $(i-1)^{th}$ stage.
7. The encoder is designed which encodes the output of the comparator in the desired digital format. The most common type of encoder used in ADCs is a thermometer to binary encoder. A [{(k/m) + 1} x n] encoder is used for this purpose.
8. For comparators 2 to m, a constant parameter P_{const} is to be added at each stage to the input of comparator as reference current and is calculated as below:

   $$P_{const1} = \text{ step} \times 2^{0}$$
   $$P_{const2} = \text{ step} \times 2^{1}$$
   $$P_{const(n+1)}= P_{fix1} + P_{fix2} +\ldots\ldots.+ \; P_{fix(n)}$$

   Similarly

   $$P_{const(n+2)}=step \times 2^{n}$$
   $$P_{const\,(n+3)}=step \times 2^{(n+1)}$$
   $$P_{const(2n+2)}= P_{const(n+2)} + P_{const\,(n+3)} +\ldots+ \; P_{const\,(2n+1)}$$

This parallel conversion of (k = mxn) bits at the same time in 'm' stages gives very high conversion speed to the ADC. Comparison of the proposed ADC scheme with the different versions of flash type ADC reveals an interesting advantage it offers over them [1]. In flash type ADC, the number of comparators needed for a digital output of k bits is $(2^{k}-1)$ whereas in the proposed ADC scheme the number of comparators needed for generating a k-bit digital output is only (k+2). As the demand for high resolution increases, the hardware required grows exponentially; thus making the converter highly convenient and less expensive. Also the use of reconstruction DACs provide more accuracy over the flash type ADCs.

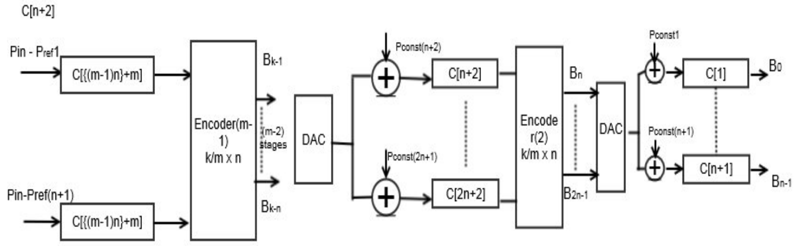
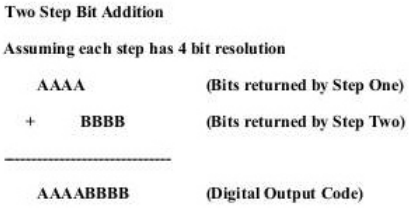


Fig 1. Block diagram of Generic ADC

III. The Balanced ADC

The balanced ADC is a generalisation of this multi-stage parallel ADC. Each individual stage in the structure is a Flash ADC resolving the finest bits in the system. As shown in fig. 2, the bits from each stage are added together to form a digital output code.


Two Step Bit Addition

Assuming each step has 4 bit resolution

AAAA	**(Bits returned by Step One)**
+ BBBB	**(Bits returned by Step Two)**

AAAABBBB	**(Digital Output Code)**

Fig. 2 Addition of Bits at each Stage.

The converter has been called 'Balanced' because its architecture allows expansion in both horizontal as well as in vertical direction.

The balanced parameter for the ADC can be calculated as:

Balanced Ratio = m/n

Where m=number of stages; n=number of bits at each stage.

- If the balanced ratio>1 then the number of comparators required in each stage for parallel conversion will be less and the number of stages will be more. This increased number of stages will make use of proportionately increased number of reconstruction DACs which in turn offers greater accuracy with minimum number of comparators.
- If the balanced ratio<1 then the number of comparators used for parallel conversion will be increased but this will provide higher speed.
- If the balanced ratio = 1 then the convertor so obtained is called as 'Square type ADC'.

The digital output bits k= n x m, and n = m, therefore, k = n^2 = m^2. The 'Square type ADC' so obtained will provide both optimum accuracy and speed with minimum number of comparators.

So, by changing the value of balanced ratio a large number of different architectures of parallel ADC's can be designed.

IV. Current-Mode Adc Components

Components used in the proposed ADC are a popular current comparator, a thermometer to binary encoder, a Digital to Analog converter, current adder, and current subtractor. They are discussed in detail in the following sub-sections([22]-[23]).

4.1 CURRENT COMPARATOR

Current comparators are crucial part of most current mode ADCs. They compare the analog input current with a reference value and generate a logic level '1' if input is greater than the reference, and logic level '0' otherwise.

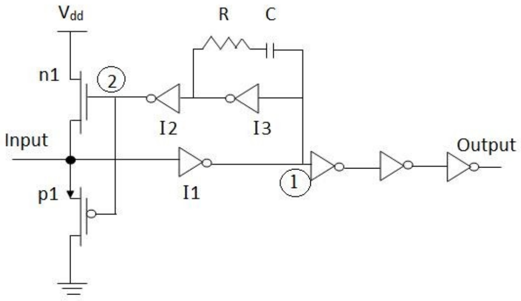


Fig. 3 Tang's Current comparator

Different types of current comparators have been proposed in literature over the past few years ([17]-[20]). They are designed using different design principles. Most popular among them is the Traff's current comparator [17]. Various architectures have since been proposed to improvise upon Traff's structure. One such improvisation is the structure proposed by Tang [19]. Fig. 3 shows the Tang's Comparator architecture.

Operationally, when the current comparison begins, transistors n1 and p1 are in deep sub-threshold state i.e. in OFF mode. Now if the current applied at the input is of low value, then the input of inverter I1 charges and discharges slowly. As a result of which voltage at node 1 also changes slowly. In this situation, n1 and p1 take a long time to turn ON and then to activate the feedback. To overcome the long response time two inverters I2 and I3 are introduced, together acting as a non-inverting amplifier in the feedback loop. For a low input current, the voltage changes at the input node and node 1 are small and for a short time, as that in Traff's. But because of the amplification by inverters I2 and I3, voltage at node 2 changes substantially, simultaneously turning on n1 or p1 and activating the feedback loop. Hence the response time is sharply reduced. The added inverters will increase power consumption slightly, compared to Traff's circuit. But the speed is improved significantly for low input currents, which is more important in many applications [19].

Because of this distinguishing feature of Tang's current comparator, we have used it in the proposed ADC scheme.

We have simulated the structure in 0.18µm technology with a supply voltage of 1.8V. Fig. 4 shows the simulations result for input current in the range [-3, 3] µA. For the sake of simplicity, the reference current is taken to be 0 µA. Input to the comparator is difference between input current and the reference current. If this difference is positive, then the output voltage is logic 1 otherwise logic 0.The value of capacitor C is chosen to be 0.1pF and resistance R is taken as 10k Ω. The delay observed in the simulation output is 0.154ns.

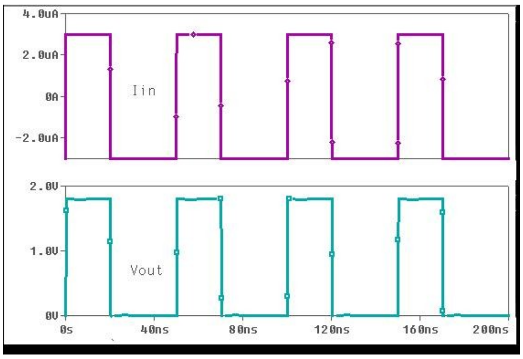


Fig.4 Simulation results of Tang's Current Comparator

In order to perform the subtraction between input current and reference current, a current subtractor [21] is used. It is discussed next.

4.2. CURRENT SUBTRACTOR

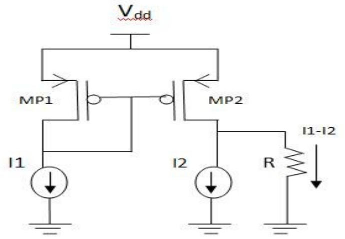


Fig 5. Current Subtractor

This ADC requires a current subtractor to compute the difference between the input current and a reference current. We have implemented a current mirror based current subtractor for our implementation, shown in Fig.5. It subtracts two input currents I1 and I2 and generates a difference output I1-I2. Fig. 6 shows the simulation result for our subtractor. The delay observed in our subtractor design is 0.350 ns.

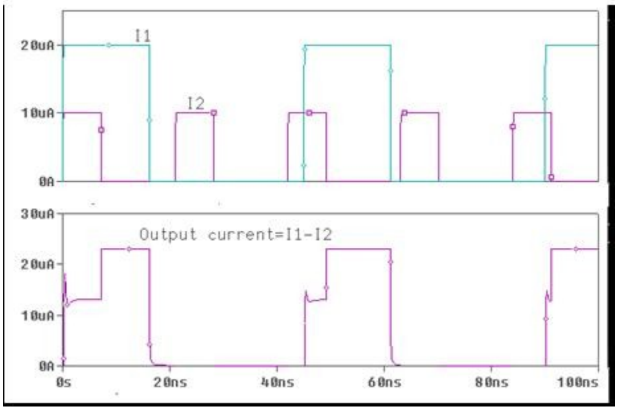


Fig.6 Simulation results of Current Subtractor

4.3 ENCODER

In accordance with step 3 of section II, outputs of current comparators are fed into a [{(k/m) + 1} x n)] size encoder implemented using CMOS technology. The implementing functions for the encoder are derived by framing a truth table, shown in Table I. For the sake of simplicity, we assume k=4, m=n=2 ADC.

Table I Truth Table of Encoder

C(3)	C(2)	C(1)	B3	B2
0	0	0	0	0
0	0	1	0	1
0	1	1	1	0
1	1	1	1	1

B3= C(1) AND C(2)
B2= C(1) AND (C(2) XNOR C(3))

It is observed that if the output of any comparator C(i) is ‘1’, the output of comparator C(j) would be also ‘1’ because $I_j < I_i$ as long as j<i. I_j and I_i stand for inputs in comparator j and i respectively. Simulation results of AND and OR gate are shown in Fig.7 and 8 respectively.

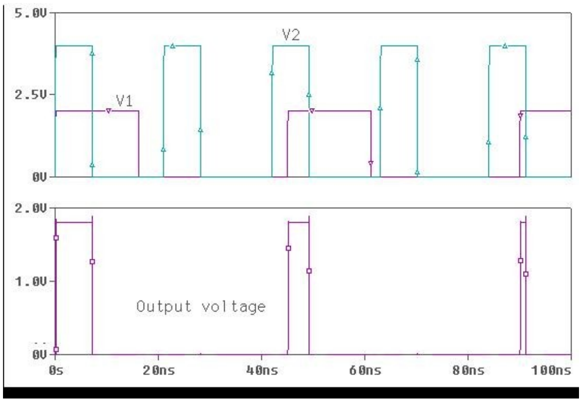


Fig.7 Simulation results of AND gate.

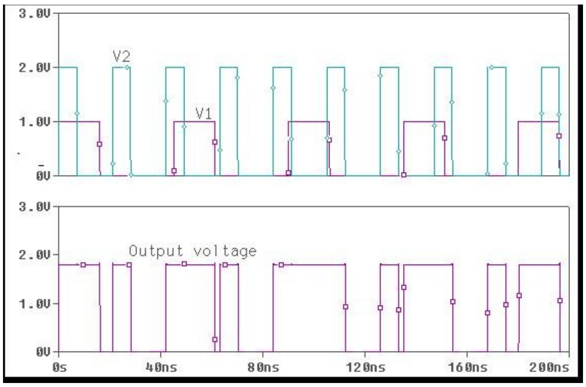


Fig.8 Simulation results of OR gate

4.4. DIGITAL TO ANALOG CONVERTOR (DAC)

A DAC converts digital value to corresponding analog value. The purpose of DAC in the proposed circuit is to generate the currents corresponding to the n MSBs obtained after every stage so that these n currents, in combination, can then be used for obtaining reference currents for next stage after adding constant currents by a current adder. Fig.9 shows the DAC as employed in our implementation. This DAC is simply a Common Source MOS [21] with binary input applied at gate of NMOS to either switch it ON or OFF. The simulation results as shown in Fig.10 exhibit that this DAC design introduces a delay of 0.085ns.

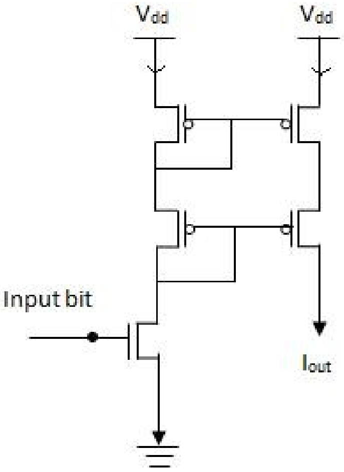


Fig.9 1-Bit Digital to Analog Converter.

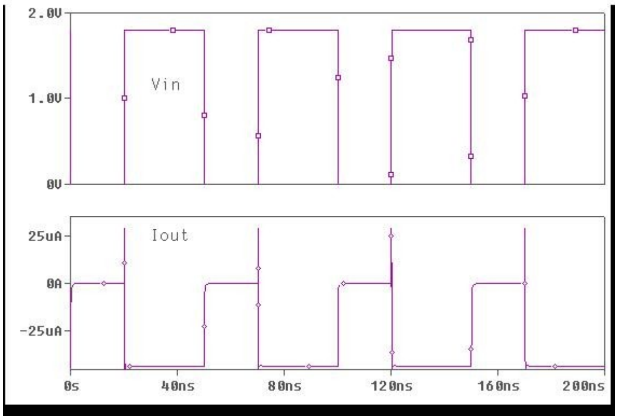


Fig. 10 Simulation results of 1-bit DAC

4.5. CURRENT ADDER

To add currents obtained from the DAC and that obtained from step 8 of the algorithm (I_{dac} and I_{const}), a current adder [21] is used. The working of current adder is similar to the working of current subtractor, only the direction of current source in parallel to the resistor R is reversed. The value of constant current I_{const} for each stage is calculated as-

$$I_{const1}=\text{step x } 2^{0}$$
$$I_{const2}=\text{step x } 2^{1}$$
$$I_{const(k/m+1)} = I_{const1} + I_{const2} +\ldots\ldots.+ I_{const(k/m)}$$

Similarly,

$$I_{const(k/m+2)}=\text{step x } 2^{n}$$
$$I_{const(k/m+3)}=\text{step x } 2^{(n+1)}$$
$$I_{const(2k/m+2)}= I_{const(k/m+2)} + I_{const(k/m+3)} +\ldots\ldots.. + I_{const(2k/m+1)}$$

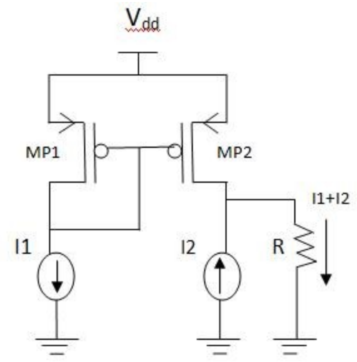


Fig.11 Current Adder

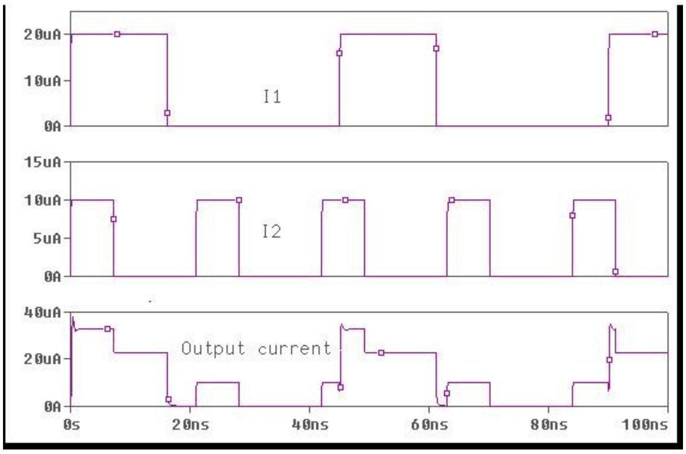


Fig.12 Simulation results of current adder

Simulation results of the adder in Fig. 12 show that the delay observed in our current adder design is 0.227ns.

V. EXEMPLARY k=4, m=n=2 current mode ADC

To better explain the concept, we have implemented an exemplary k=4-bit ADC [21]. It has two parallel stages, each step producing a 2-bit output. Therefore m=n=2, hence making it a square ADC.
The block diagram this ADC is depicted in Fig13.

$$\text{Step Size} = (M2 – M1)/(2^{k} - 1) = 0.5\mu A$$
$$I_{ref1} = \text{Step x } 2^{(k/m)} = 0.5 \text{ x } 2^{(4/2)} = 0.5 \text{ x } 4 = 2.0\mu A$$
$$I_{ref2} = \text{Step x } 2^{(k/m)} = 0.5 \text{ x } 2^{(4/2+1)} = 0.5 \text{ x } 8 = 4.0\mu A$$
$$I_{ref3} = I_{ref1} + I_{ref2} = 6.0\mu A$$

The total number of comparators used are 3 per stage thus, 3x2=6 total comparators are employed. The implemented ADC operates on the input current range of 0 to 7.5μA. So, M1=0μA, M2=7.5μA and k=4, we calculate the step size as-

- For input current, I_{in} = 3μA, input to comparator is (I_{in} – I_{ref1}) = 3-2=1.0μA. Similarly input to comparator5 and comparator6 is -1.0μA and -3.0μA.
- Also the values of I_{const} are calculated as per the formulae. The results obtained are-

$$I_{const1}=0.5\mu A$$
$$I_{const2}=1.0\mu A$$

I_{const3}=1.5µA

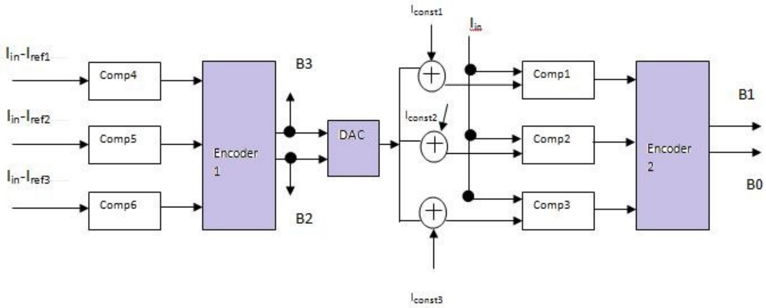


Fig13. Block diagram for 2 x 2 ADC

VI. Simulation Results

A 2x2-bit current mode square ADC is designed using the proposed algorithm and simulation results are discussed here. Total delay encountered in k=4, m=n=2 proposed ADC is calculated as follows-

- Number of series stages of comparator = 2
  Delay due to 1 Tang's comparator=0.154ns
  Therefore, delay due to two comparators in series= 2 x 0.154= 0.308ns
- Number of series stages of encoder =2 Number of AND gates in series = 2 Number of OR gates in series = 2 Delay offered by 1 AND gate= 1.072ns Delay offered by 1 OR gate= 0.118ns
  Therefore, delay due to 2 AND and 2 OR gates in series= 2x1.072 + 2x0.118 = 2.380ns
- Total number of DAC in series=1
  Delay due to 1 DAC=0.085ns
- Total number of series stages of current adders used=1
  Delay due to 1 current adder=0.227ns
- Total number of series stages of current subtractor = 1 (current subtractor has not been used in I_{in}-I_{ref1} , I_{in}-$I_{ref2,}$ I_{in}-I_{ref3} process i.e. input to the first stage of current comparators is given as such.)

Therefore total delay encountered in k=4, m=n=2 proposed ADC= 0.308ns + 2.380ns + 0.085ns + 0.227ns + 0.350ns= 3.35ns.

Simulation results for k=4, m=n=2, and I_{in}=3µA are shown in Fig. 14. In Fig. 15, time axis of Fig. 14 has been expanded and shown. It is known that the digital output for input current= 3µA should be 0011. It can be clearly seen from Fig. 15 that the digital combination B3B2B1B0 between blue and yellow lines is 0011. And the combination keeps on repeating at fixed intervals.

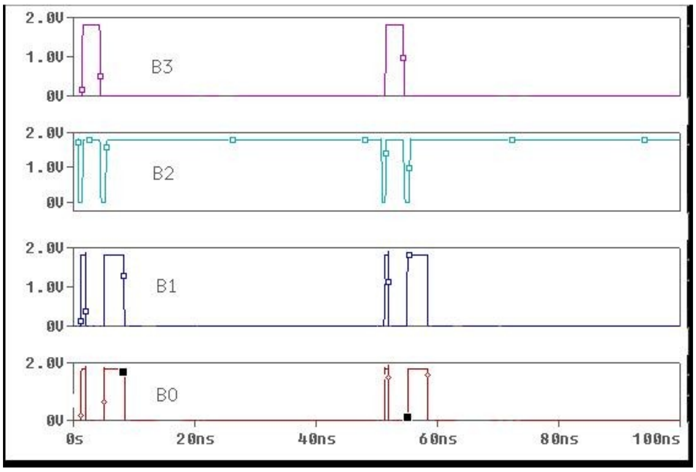


Fig.14 Simulation results of 4- Bit ADC

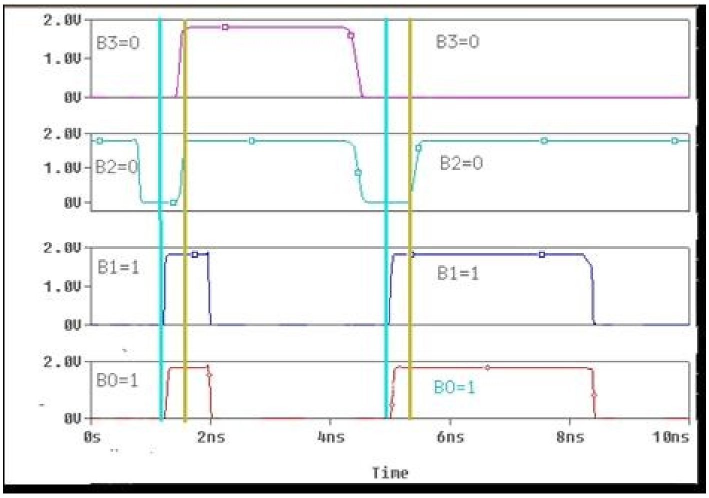


Fig.15 Time Axis of Fig. 14 expanded.

VII. Conclusion

In this paper, a fast method for k-bit (n x m) analog to digital conversion is proposed. It has a parallel configuration and provides conversion of n bits at the same time in m stages which gives a very high conversion speed. In addition to this, the comparator used is highly efficient and gives negligible response time. The rise and fall time of the pulses is also very small.

References

[1] Veepsa Bhatia, Neeta Pandey, Asok Bhattacharyya, Application Based comparison of different analog to digital Converter architectures", International Journal of Engineering Science and Technology 2(8), 2010,3396-3404.

[2] Nairn, D.G., Salama, C.A.T., Current- Mode Algorithmic Analog to Digital Converters, IEEE Journal of Solid-State Circuits 25, 1990,997-1004.

[3] C. P. Hurrell, C. Lyden, D. Laing, D. Hummerston, and M. Vickery, "An 18 b 12.5 MS/s ADC with 93 dB SNR" ", IEEE Journal of Solid-State Circuits 45, 2010, 2647-2654.

[4] B. A. Wooley, K. Vleugels, Pipelined A/D Conversion, Handout #22, 2002, EE315.

[5] Ken Poulton, R. Ne®, A. Muto, W. Liu, A. Burstein, and M. Heshami, A 4GSample/s 8bADC in 0.35-^{1}m CMOS", IEEE International Solid-State Circuits Conference, Digest of Technical Papers, 2002,166-167.

[6] Robert H. Walden, Analog-to-digital converter survey and Analysis, IEEE Journal on Selected Areas in Communication, 17(4), 1999,539-550

[7] INC, IMS and IDC,A Novel 1GSPS Low Offset Comparator for High Speed AD,NCM '09 Fifth International Joint Conference, 2009, 1251 – 1254

[8] Matsuzawa, ASIC, Trends in high Speed ADC design, ASICON'07, 7th International Conference on Digital Object Identifier, 2007, 245-248.

[9] R. Khoini-Poorfard, L.B. Lim and D.A. Johns, Time-interleaved oversampling A/D converters, IEEE Transaction on Circuits and Systems - II Analog and Digital Signal Processing 44, 1997, 634-645.

[10] M. Choi and A. A. Abidi, A 6-bit 1.3-Gsample/s A/D converter in 0.35 um CMOS, IEEE Journal of Solid-State Circuits 36, 2001,1847-1858.

[11] I. Ahmed and D.A. Johns, An 11-bit 45MS/s pipelined ADC with rapid calibration of DAC errors in a multi-bit pipeline stage, IEEE Journal of Solid-State Circuits 41, 2008,1578-1588.

[12] I. Ahmed and D.A. Johns, A Low-Power Capacitive Charge Pump Based Pipelined ADC, IEEE Journal of Solid-State Circuits 45, 2010,5

[13] P. Prasad Rao, K. Lal Kishore, Optimizing the Number of Bits/Stage in 10-Bit,50Ms/Sec Pipelined A/D Converter Considering Area, Speed, Power and Linearity, World Academy of Science, Engineering and Technology, 2012,62.

[14] W. Liu and Y. Chiu , Time-interleaved analog-to-digital conversion with online adaptive equalization, IEEE Transactions on Circuits and Systems 59, 2012, 1384-1395.

[15] A. Gharbiya, T.C. Caldwell and D.A. Johns , High-speed oversampling analog-to-digital converters, International Journal of High Speed Electronics and Systems 15(2), 2005,297-317

[16] Boris Murmann ,Bernhard E. Boser ,A 12-bit 75-MS/s Pipelined ADC Using Open-Loop Residue Amplification, IEEE Journal of Solid-State Circuits 38, 2003, 12.

[17] Traff. H,Novel approach to high speed CMOS current comparators, Electron. Letters 28, 1992,310–312.

[18] Ravezzi L, Stoppa D, Dallabetta G.F, "Simple high-speed CMOS current comparator, Electronics Letters 33, 1997, 1829–1830

[19] X.Tang and K.P. Pun., High-performance CMOS current comparator, Electronics Letters 45, 2009,20

[20] Tolga Kaya Ali Zeki., A High-Speed Cmos Current Comparator Suitable For Algorithmic Analog-To-Digital Converters, IU-JEEE 8, 2008,2.

[21] Veepsa Bhatia, Neeta Pandey, Asok Bhattacharyya , A 4-bit expandable current mode ADC based on different current-mode architectures, International Journal of Engineering Science And Technology 2(12), 2010,7380-7384

[22] Golnar Khodabnedehlo, Mitra Mirhassani And Majid Ahmadh, An area-speed efficient method for current mode analog to Digital converters", Circuit Theory and Design, ECCTD. European Conference, 2009,201-204

[23] P.R Grey and R.G Meyer, Analysis and Design of Analog integrated circuits (Springer US, Wiley 1984).