Journal of VLSI Design Tools and Technology
Volume 2, Issue (1-3), Compilation 2012

STM JOURNALS

Decimator Design for Sigma-Delta ADC

Tripti Sharma, Prashant Singh, Narendra Bahadur Singh
Central Electronics Engineering Research Institute(CSIR-CEERI), Pilani, Rajasthan-333031.

ABSTRACT

This paper describes the design of a decimator which is used in digital signal processing as well as 10-bit sigma-delta analog-to-digital converter to down sample the incoming signal for further applications. It is also used for verification of the sigma-delta modulator functionality to reconstruct the incoming signal. The design has been carried out using 180 nm TSMC, CMOS foundry parameters for spice level 49 model parameters in TANNER EDA (Tspice). The results for the presented decimator model are mentioned in the paper; the higher decimation factor can be taken similarly as per requirement.

Keywords: *DSP, decimator, counter, sigma-delta ADC, DAC, operational amplifier*

****Authors for Correspondence*** E-mail: tripti.mzn@gmail.com, nbs44254@gmail.com

1. INTRODUCTION

Multirate digital signal processing is required in digital systems where more than one sampling rate is required. Different sampling rates can be obtained using an up sampler or down sampler. The basic operations in multirate processing to achieve this are decimation and interpolation. Decimation is for reducing sampling rate and interpolation is for increasing sampling rate. In digital transmission systems like teletype, facsimile, and low bit rate speech, where data has to be handled in different rates, multirate signal processing is used. There are various areas in which multirate signal processing is used. Some of them are

- Communication Systems
- Speech and audio processing systems
- Antenna systems
- Radar systems

Various advantages of multirate signal processing are

- Computational requirement is less
- Storage for filter coefficient is less
- Finite arithmetic effects are less

1.1. Sampling

Let *x(t)* be a continuous time varying signal. The signal *x(t)* is sampled at regular interval of time with sampling period *T* as shown in Figures 1.a and 1.b. The sampled signal *x(nT)* is given here.

$$x(nT) = x(t)|_{t=nT} = x(nT), -\infty < n < \infty \quad (1)$$

A sampling process can also be interpreted as a modulation or multiplication process, as shown in Figures 2.a, 2.b, 2.c and 2.d.

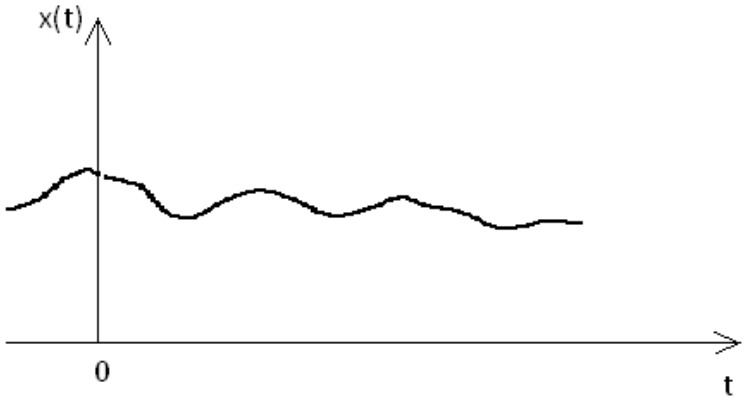

Fig. 1.a: *Input Signal.*

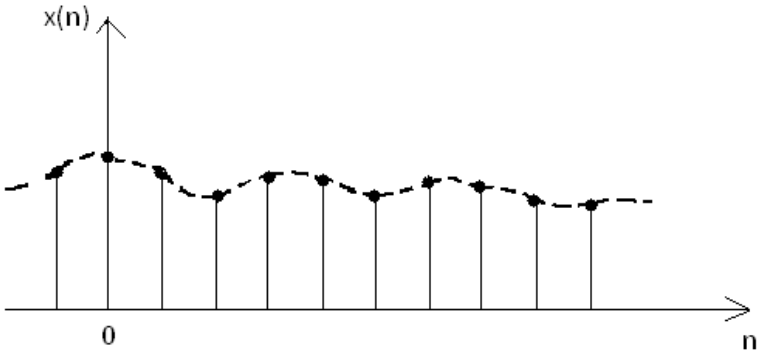

Fig. 1.b: *Sampled Sequences x(n), Sampling of Continuous Time Input Signal x(t).*

Fig. 2.a: *Sampling System Block Diagram.*

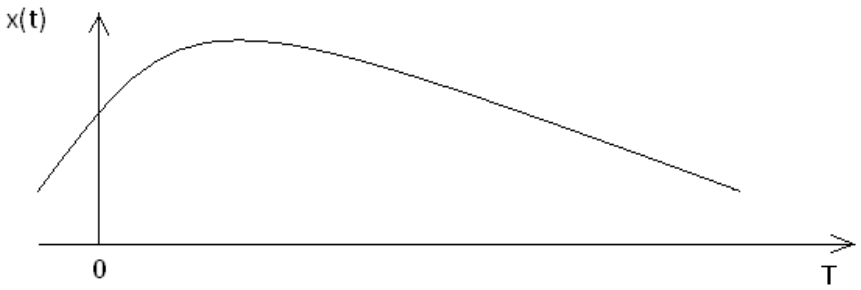

Fig. 2.b: *Input Signal Waveform to be Sampled.*

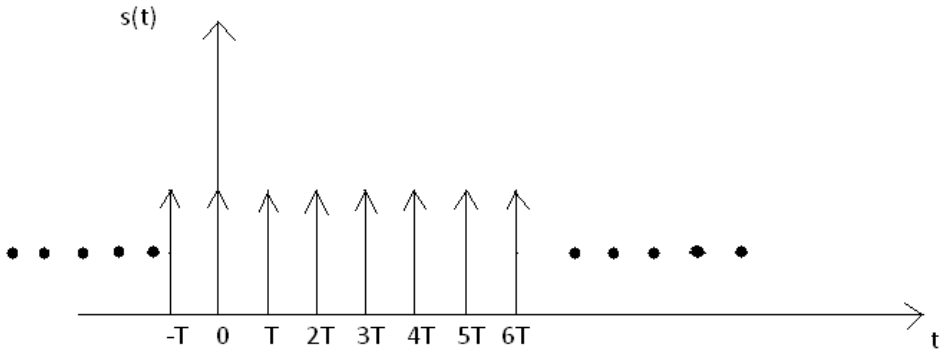

Fig. 2.c: *Sampling Signal Waveform.*

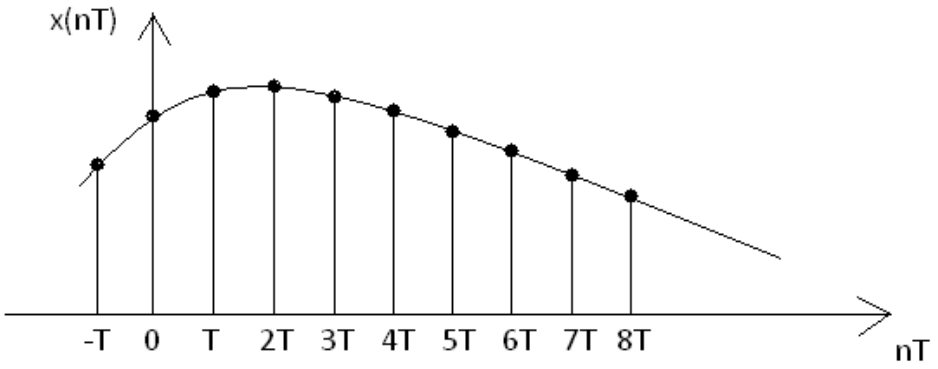

Fig. 2.d: *Periodic Sampling of x(t), Sampled Signal Waveform.*

The continuous time signal *x(t)* is multiplied by the sampling function *s(t)* which is a series of impulses (periodic impulse train); the resultant signal is a discrete time signal *x(n)* [1–3].

$$x(n) = x(t)\, s(t)|_{t=nT}, -\infty < n < \infty \quad (2)$$

1.2. Sampling Theorem

According to the sampling theorem, a band-limited signal *x(t)* having finite energy, which has no frequency components higher than f_h Hz, can be completely reconstructed from its samples taken at the rate of $2f_h$ samples per second, i.e., $f_s \geq 2f_h$, where f_s is sampling frequency and f_h is the highest signal frequency.

The sampling rate of $2f_h$ samples per second is the Nyquist rate and its reciprocal $1/2f_h$ is Nyquist period [5].

1.3, Sampling Rate Conversion

Sampling rate conversion is the process of converting the sequence *x(n)* which is taken from sampling the continuous time signal *x(t)* with period *T*, to another sequence *y(k)* obtained from sampling *x(t)* with period *T*'.

The new sequence *y(k)* can be obtained by first reconstructing the original signal *x(t)* from the sequence *x(n)* and then sampling the reconstructed signal with a period *T′*. Figure 3 shows the reconstruction of original signal with a D/A converter, low-pass filter and resampler with sampling period *T′*.

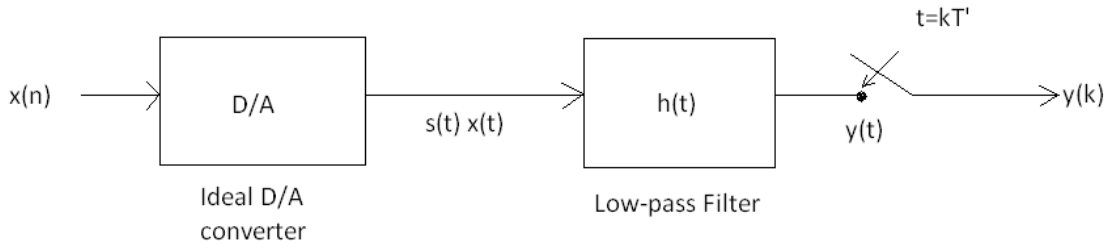

Fig. 3: *Conversion of a Sequence x(n) to Another Sequence y(k).*

1.4. Decimation

The process of reducing sampling rate of a signal is called decimation (sampling rate compression). An example of decimated signal is shown in Figures 4.a and 4.b. The input sequence is given by *x(n)* and output sequence, i.e., decimated signal is given by *y(n)*. In this example, let us consider the decimation factor is "3" [3–5].

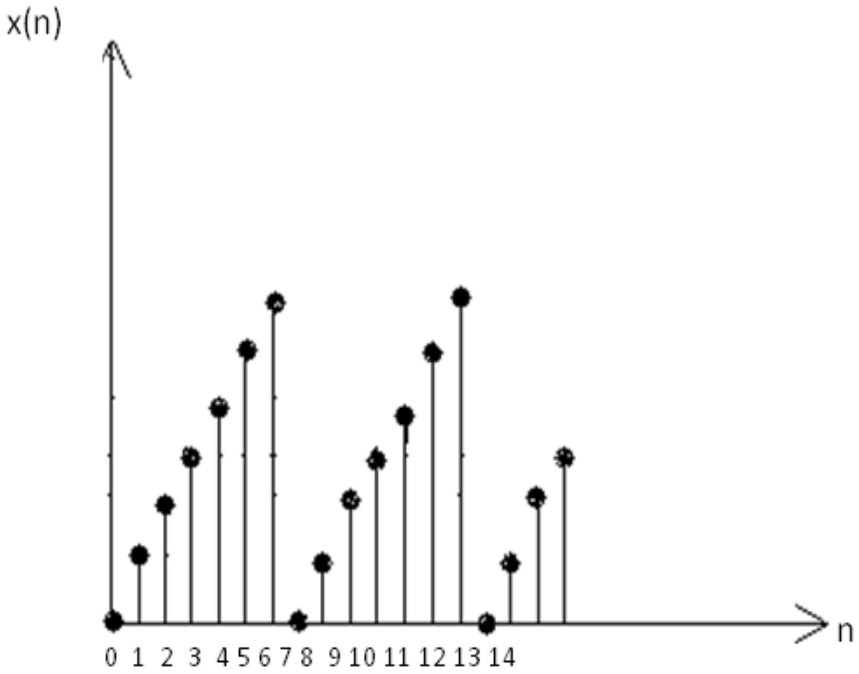

Fig. 4.a: *Input Sequences x(n).*

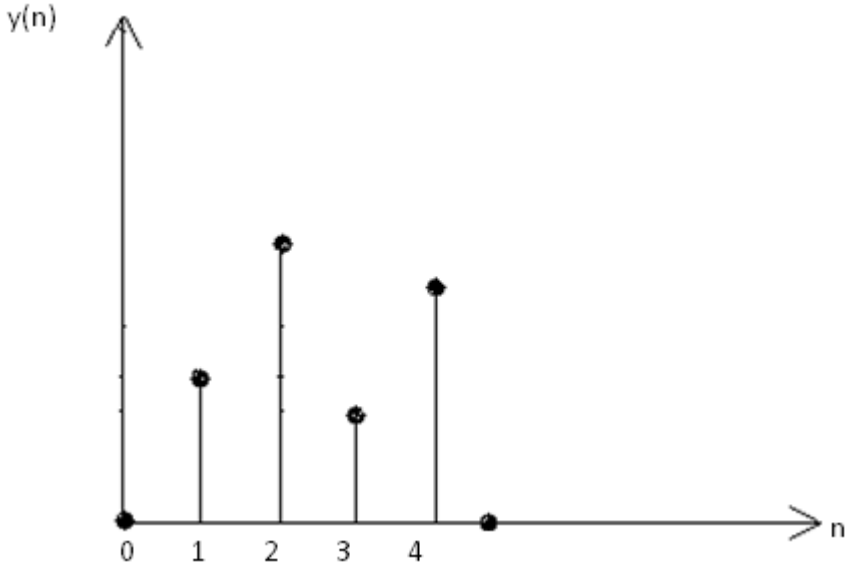

Fig. 4.b: *Output Sequences with Decimation Factor "3".*

2. GENERAL PURPOSE DECIMATOR DESIGN

The decimator can be designed as shown in Figure 5 and it is clear from the given schematic diagram that the decimator circuit consists of various other circuits presented here like counter, shift register and logic gates [6–10]. The spice files were exported for the given schematic, then their circuit simulations were carried out using twin-well epitaxial mixed mode technology (TSMC) 180 nm foundry parameters.

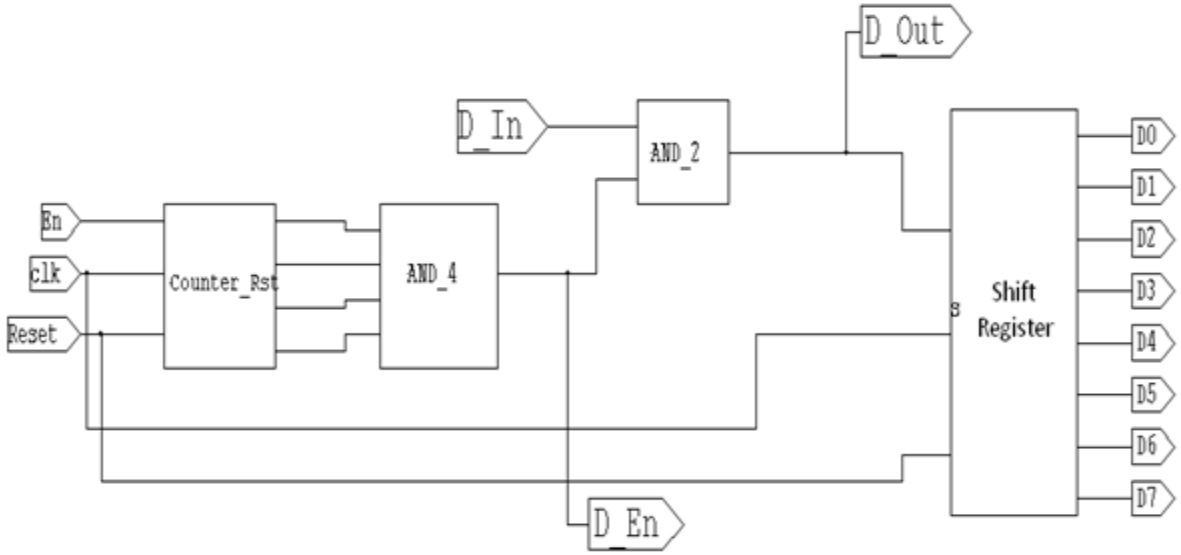

Fig. 5: *General Purpose Decimator Circuit Schematic.*

2.1. Counter Design

Here we use a synchronous counter, whose output bits change state simultaneously, with no ripple. Such type of counter can be designed from J-K flip flops, connecting all the clock inputs together so that each and every flip flop receives the same clock pulse at the same time. The regular pattern can be seen from 4-bit counter in Figures 6 and 7.

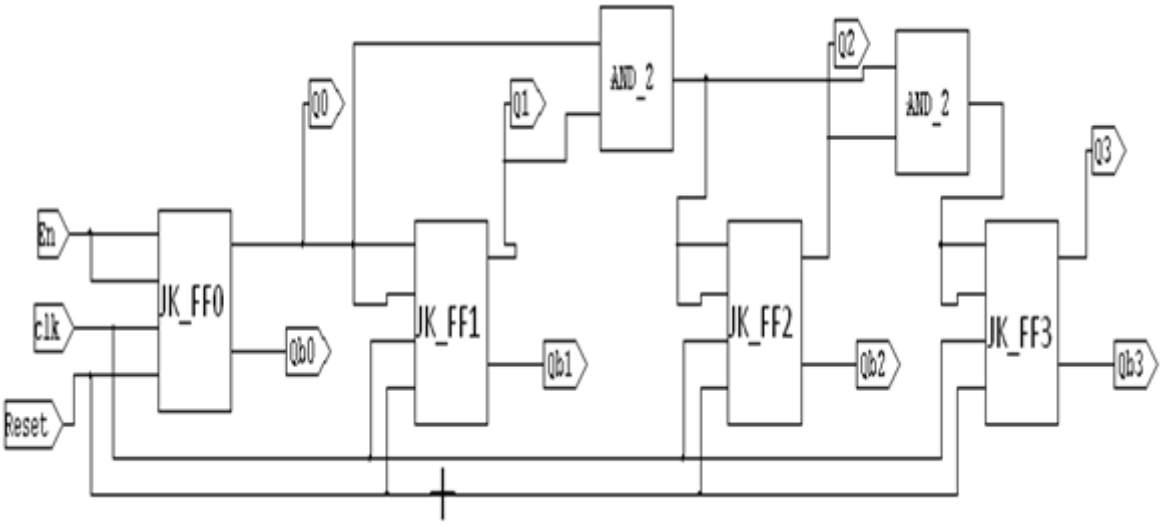

Fig. 6: *4-Bit Synchronous Up Counter.*

Each of higher order flip flops is made ready to toggle (both J and K inputs are HIGH) if the Q outputs of all previous flip flops are HIGH. Otherwise the J and K inputs for that flip flop will both be low, placing it into the latch mode where it will maintain its present output state at the next clock pulse. Since the first LSB flip-flop needs to toggle at every clock pulse, its J and K inputs are connected to VDD, where they will be HIGH all the time. The next flip-flop need only recognize that the first flip-flop's Q output is HIGH to be made ready to toggle, so no AND gate is needed. However, the remaining flip-flops should be made ready to toggle only when all lower order output bits are HIGH, thus the need of AND gates. The reset input clears the output of counter to zero [11].

The binary count sequence of a counter is shown in Table I. The operation of synchronous counter is, the flip flop in the least significant position is complemented with every pulse. A flip flop in any other position is complemented when all the bits in the lower significant positions are equal to 1. For example, if the present state of a 4-bit counter is Q3 Q2 Q1 Q0 = 0111, the next count is 1000. Q0 is always complemented. Q1 is complemented because the present state of Q0 = 1. Q2 is complemented because the present state of Q1 Q0 = 11. Q3 is also complemented because the present state of Q2 Q1 Q0 = 111, which gives all 1's condition [6].

Let us take another example. If present state is Q3 Q2 Q1 Q0 = 011, the next count is 1100. Q0 is always complemented. Q1 is complemented because present state of Q0 = 1. Q2 is complemented because the present state of Q1 Q0 = 11. Q3 is not complemented because the present state of Q2 Q1 Q0 = 011, which does not give an all 1's condition. In this way the counter counts in a regular pattern from 0000 to 1111.

Table I: *Binary Count Sequence.*

Q3	Q2	Q1	Q0	Count
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

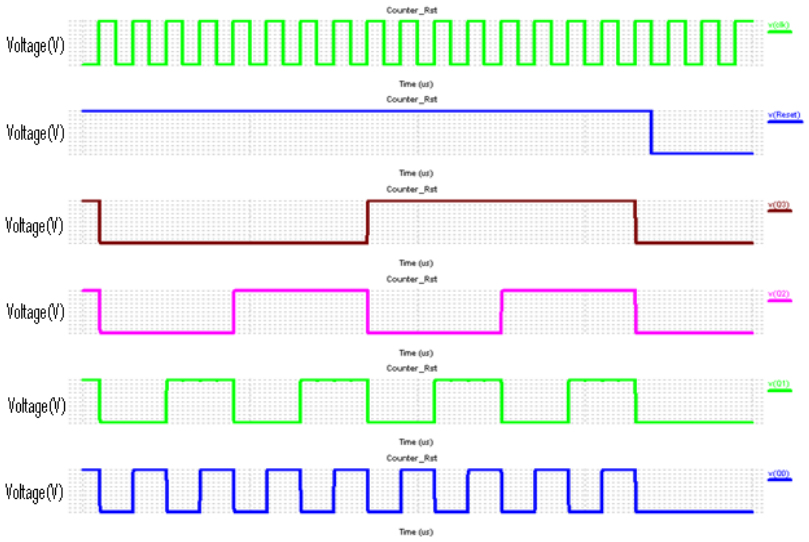

Fig. 7: *Output Waveform of Up Counter.*

2.2. Shift Registers

A register capable of shifting its binary information in one or both directions is called a shift register. The logical configuration of a shift register consists of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of next flip-flop. All flip-flops receive common clock pulses, which activate the shift from one stage to the next.

Here we use serial-in, parallel-out shift register. It is the simplest shift register that uses only flip-flops as shown in figure 9. It shifts data into internal storage elements and shifts data out at the serial out, data out pin. It makes all the internal stages available as outputs. Therefore, serial-in, parallel-out shift register converts data from serial format to parallel format.. A basic 8-bit shift register can be constructed by using only 8D flip-flops. The register is first reset, forcing all 8 outputs to zero. The input data is then applied sequentially to the D-input of the first flip-flop on the left. During each clock pulse, one bit is transmitted from left to right. The least significant bit of data has to be shifted through register from the leftmost flip-flop to the rightmost flip-flop (D_FF0 to D_FF7). If eight data bits are shifted in by eight pulses via a single wire at data-in, the data becomes available simultaneously on the eight outputs D0 to D7 after the eighth clock pulse [11].

3. FUNCTION OF A DECIMATOR

The decimator circuit is shown in Figure 5. First of all a 4-bit synchronous counter is taken, it is up counter having a count sequence from 0000 to 1111 where En is counter enable and clk is the clock input. All the four outputs of counter are given at the inputs of a four input AND gate, as we know that output of AND gate is “1” if and only if all of its inputs are “1” otherwise it is “0” so the output of AND gate is “0” for the count sequence from 0000 to 1110, it will give “1” only when the count is 1111. Here in this circuit the output of four input AND gate is “D_clk” and it acts as a clock for the decimation. Now a two input AND gate is taken whose one input is D_In, the bit stream that has to be decimated and another input is D_clk. The output of AND gate is “D_Out” that gives the output only when A is “1”, i.e., output is obtained after every 15 bits. This output is given to the shift register, which stores the data.

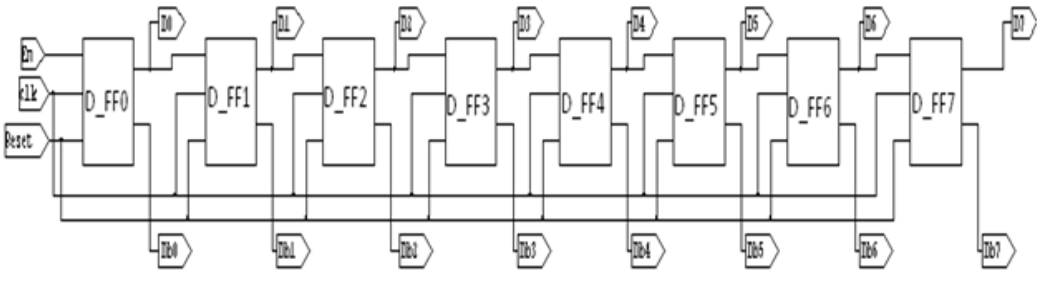

Fig. 8: *8-Bit Shift Register*

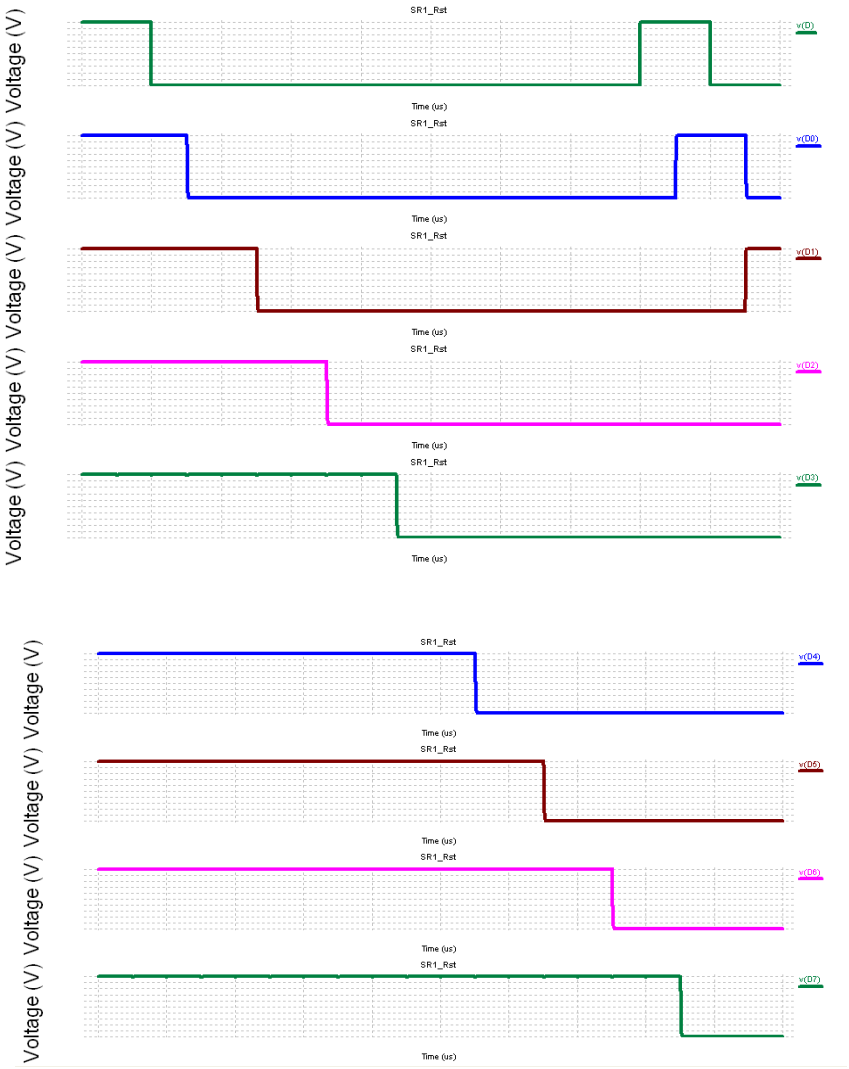

Fig. 9: *Output of Shift Register.*

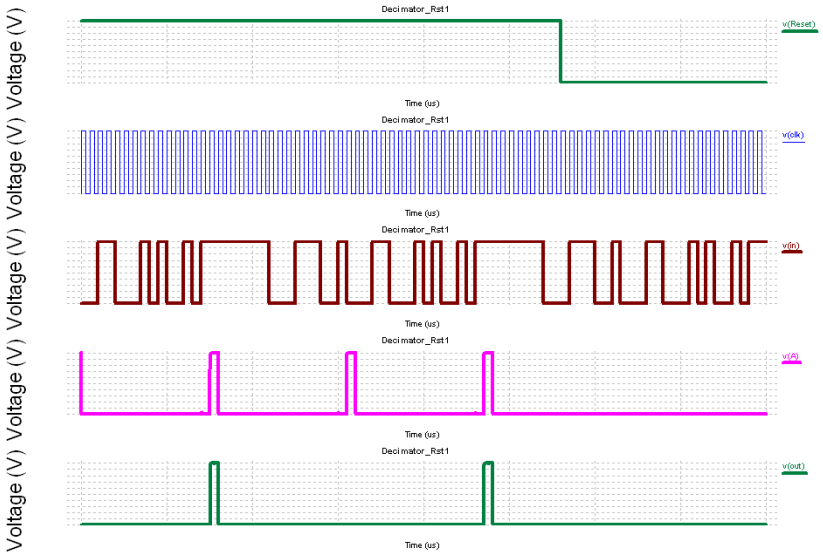

Fig. 10: *Decimator Simulation Waveform, Decimation by a Factor of 16.*

The simulation result of decimator is as shown in Figure 10. A is obtained after every 15 bits of input bit stream, the decimated output "out" is "1" in first count as the input is "1" on 16th bit then the out is "0" in next count as the input is "0" on next 16th bit, similarly the output is obtained according to the input on every successive 16th bit. This output is given to the shift register where it is stored. The output of shift register is shown in Figure 9, it also converts the serial data into parallel data.

4. DECIMATOR FOR SIGMA-DELTA ADC

The decimator circuit presented in Figure 11 is designed for the verification of the modulator functionality to be designed for the Sigma-Delta(∑-Δ) ADC [12–14]. It follows the following steps to reconstruct the one-bit digital serial stream of the Sigma-Delta ADC modulator into an analog waveform. Steps involved to implement the functions are

- Count the number of ones/zeros in the serial stream within a fixed time period using binary ripple counter.
- At the end of the time period, shift the value into a set of registers and reset the counter to store the values from the counter.
- Use a summer (D-A converter) to reconstruct the digitized signa

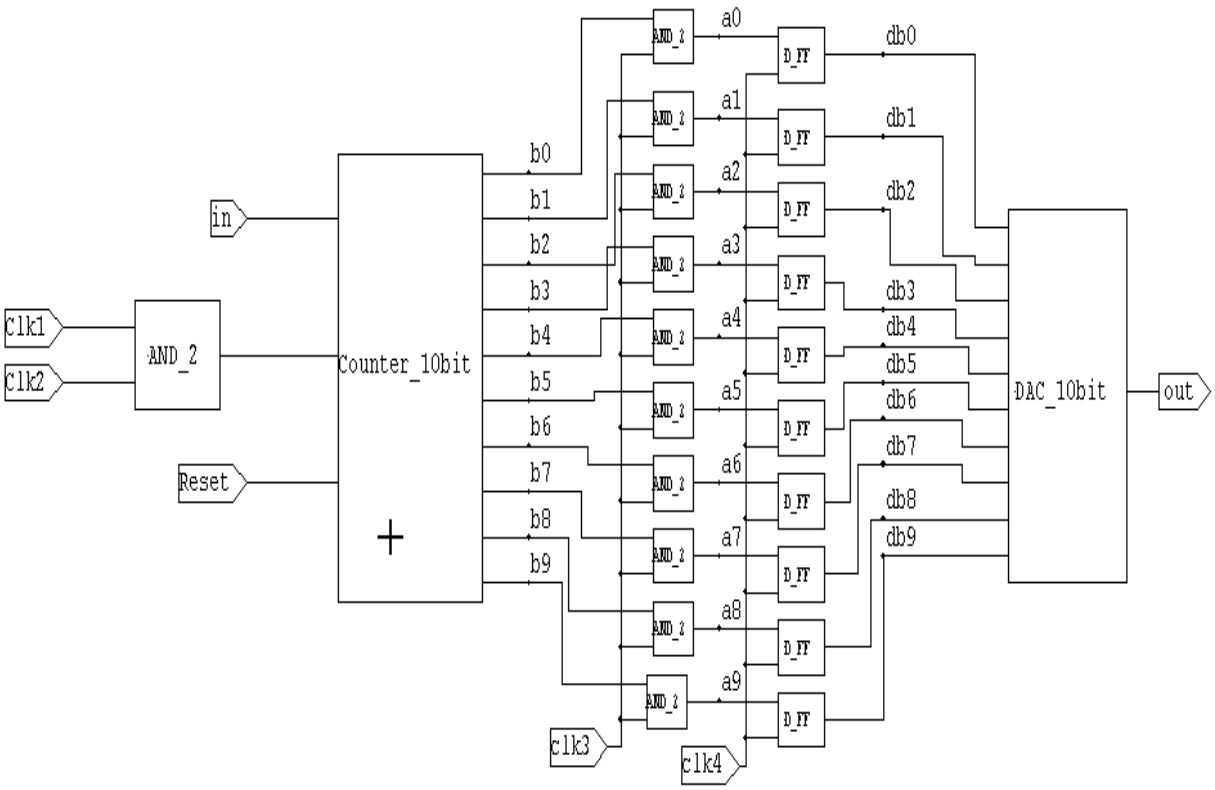

Fig. 11: *Schematic of the Decimator for 10-Bit Sigma-Delta ADC.*

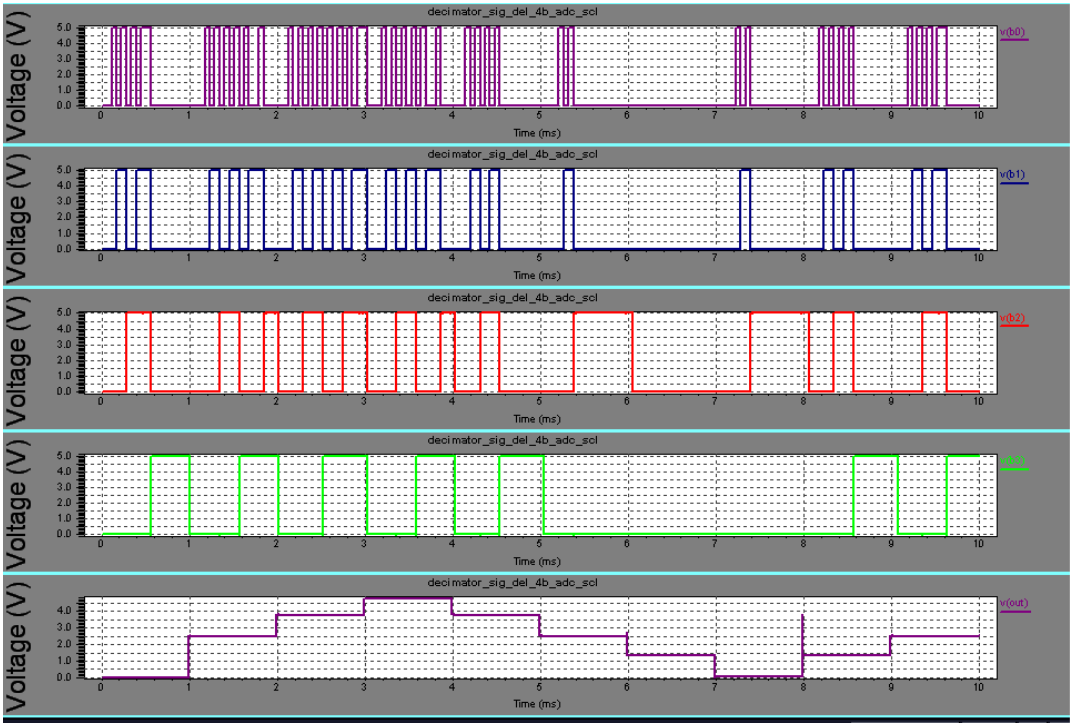

Fig. 12: *Simulation Result for the Circuit in Fig. 11.*

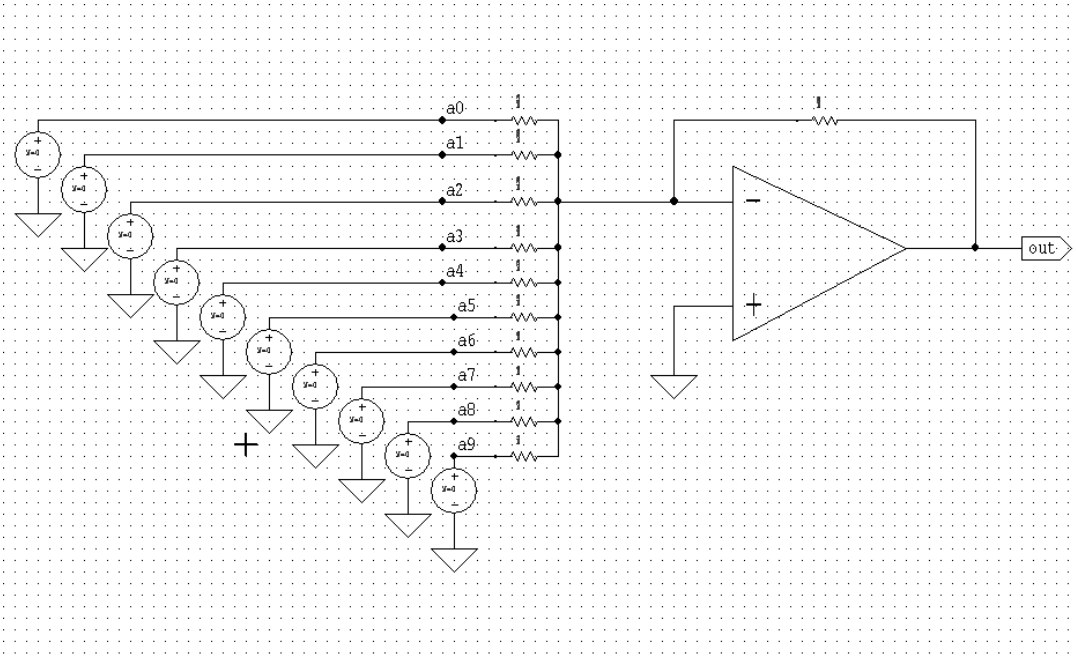

Fig. 13:*10-Bit Digital to Analog Converter Schematic Used in Fig. 11.*

Parameters of the operational amplifier used in the design are Gain = 74 dB, Phase Margin = 71 ° for Load < = 100 pF.

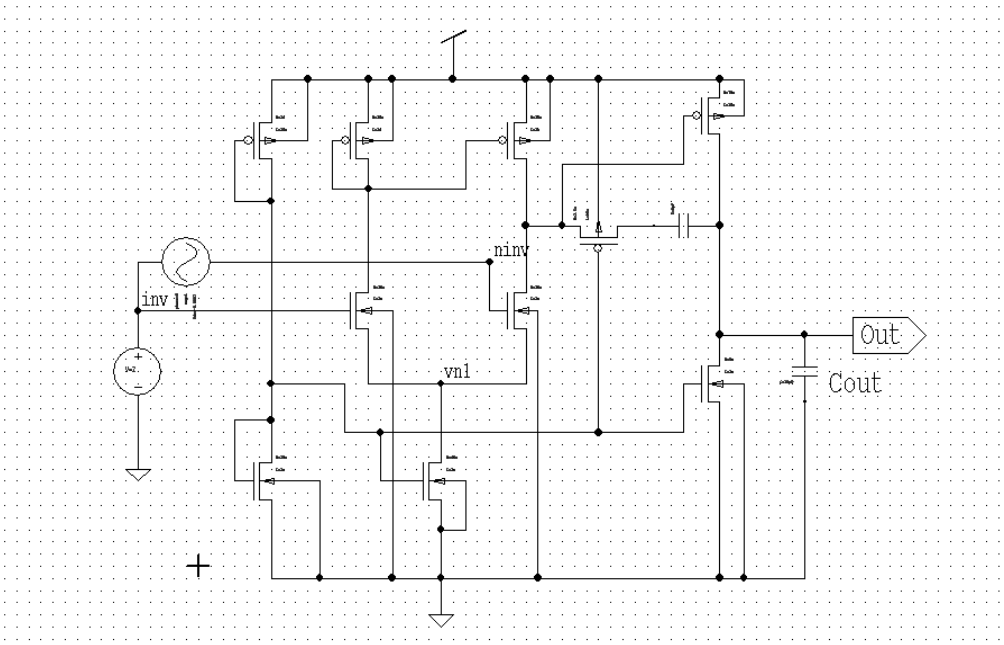

Fig. 14: *Operational Amplifier Circuit Used in Fig. 11.*

5. CONCLUSIONS

Complete design of a decimator applicable to 10-bit Sigma-Delta ADC from its schematic to the MOS transistor circuit level has been presented in this paper for the targeted semiconductor foundry. The simulation results are also presented for the decimator as well as other components. It is a part of the sigma-delta analog-to-digital converter where decimator is used for the verification of sigma-delta ADC modulator to reconstruct the signal. It has other numerous applications in digital signal processing.

REFERENCES

1. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, New Jersey. 1989. 236,and 441p.
2. L. R. Rabiner and B. Gold. *The Theory and Application of Digital Signal Processing.* Prentice-Hall, Englewood Cliffs New Jersey. 1975. 206, 273, and 288p.
3. S Salivahanan, A Vallavraj and G Gnanapriya. *Digital Signal Processing*. McGraw Hill Education Private Limited, New Delhi. 2010
4. Fred J. Harris. *Multirate Signal Processing for Communication Systems*. Prentice Hall. 2004.
5. R. J. Marks II. *Introduction to Shannon Sampling and Interpolation Theory*. Springer-Verlag. 1991.
6. M. Morris. Mano and C. R. Kime. *Logic and Computer Design*

Fundamentals. 2nd edn. Prentice Hall, Upper Saddle River, NJ. 2000.

7. J. P. Hayes. *Introduction to Digital Logic Design.* Reading, MA. Addison-Wesley. 1993.
8. J. F. Wakerly. *Digital Design: Principles And Practices*. 3rd edn. Prentice Hall. Upper Saddle River, NJ. 2000.
9. Jan M. Rabaey, Anantha Chandrakasan and Borivoje Nikolic. *Digital Integrated Circuits*. Prentice Hall.2007
10. Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design*. Pearson Education. 2004.
11. www.allaboutcircuits.com
12. S. R. Norsworthy, R. Schreier, and G. C. Temes. *Theory, Design and Simulation*. Piscataway, NJ. IEEE Press. 1997.
13. Paul Jespers. *D to A and A to D Architectures, Analysis and Simulation.* Oxford university Press. Jun 2001.
14. P. E. Allen and D. R. Holberg. *CMOS Analog Circuit Design.* 2nd edn. Oxford University Press. 2002.