

**MANIPAL SCHOOL OF INFORMATION SCIENCES**

**(A Constituent unit of MAHE, Manipal)**

**Implementation of SAR AND FLASH ADC using Verilog**

|  |  |  |
| --- | --- | --- |
| **Reg. Number** | **Name** | **Branch** |
| **201038012** | **Shree Jnanesh** | **VLSI-A** |
| **201038037** | **Tummala Vamshi** | **VLSI-B** |
| **201038055** | **Avinash R** | **VLSI-B** |

**Under the guidance of**

**Dr. Prashanth Kumar Shetty**

Associate Professor -Senior Scale,

Manipal School of Information Sciences,

MAHE, MANIPAL

**20/01/2021**

****

**DECLARATION**

We declare that this mini project, submitted for the evaluation of course work of Mini Project to Manipal School of Information Sciences, is our original work using an existing concept available at IEEE named “**A less-Area and performance-efficient SAR ADC design”**, conducted under the supervision of my guide Dr. Prashanth Kumar Shetty and panel members, Dr. Prashanth Kumar Shetty, Dr. Chaitanya C.V.S References, help and material obtained from other sources have been duly acknowledged.

**ABSTRACT**

The process of converting an analog voltage into an equivalent digital signal is known as Analog to Digital Conversion abbreviated as ADC. An ADC is an electronic circuit which converts its analog input to corresponding binary value. The output depends up on the coding scheme followed in the ADC circuit. For example, Analog value may convert to Gray code, excess 3 code and so on. Resolution, Speed and Power Consumption are the three parameters of ADC.

**CONTENTS**

[**1.** **INTRODUCTION** 1](#_Toc62144667)

[**1.1** **ADC WORKING** 1](#_Toc62144668)

[**1.2** **SPECIFICATIONS** 2](#_Toc62144669)

[**2** **ARCHITECTURE OF ADC TO BE IMPLEMENTED** 4](#_Toc62144670)

[**2.1** **FLASH ADC ARCHITECTURE** 4](#_Toc62144671)

[**2.2** **SUCCESSIVE APPROXIMATION ADC ARCHITECTURE** 5](#_Toc62144672)

[**3** **IMPLEMENTATION** 6](#_Toc62144673)

[**3.1** **IMPLEMENTATION OF SAR ADC** 6](#_Toc62144674)

[**3.2** **IMPLEMENTATION OF FLASH ADC** 8](#_Toc62144675)

[**4** **RESULTS** 10](#_Toc62144676)

[**4.1** **RESULTS OBTAINED FOR SAR ADC** 10](#_Toc62144677)

[**4.1.1** **SYNTHESIS OF SAR ADC** 12](#_Toc62144678)

[**4.1.2** **SIMULATED OUTPUT RESULT LOG FILE OF SAR ADC** 14](#_Toc62144679)

[**4.1.3** **OBSERVATIONS RECORDED POST SYNTHESIS OF SAR ADC** 14](#_Toc62144680)

[**4.2** **RESULTS OBTAINED FOR FLASH ADC** 15](#_Toc62144681)

[**4.2.1** **SIMULATED OUTPUT RESULT LOG FILE OF FLASH ADC** 16](#_Toc62144682)

[**5** **REFERENCES** 17](#_Toc62144683)

[**6** **APPENDIX A** 18](#_Toc62144684)

[**6.1** **SIMULATION OUTPUT OF SAR ADC** 18](#_Toc62144685)

[**6.2** **REPORTS GENERATED BY THE GENUS TOOLS FOR SYNTHESIS OF SAR ADC** 20](#_Toc62144686)

[**6.2.1** **POWER** 20](#_Toc62144687)

[**6.2.2** **AREA** 20](#_Toc62144688)

[**6.3** **SIMULATION OUTPUT OF FLASH ADC** 21](#_Toc62144689)

**LIST OF FIGURES**

**Figure 1.1: Analog to Digital Converter.**

**Figure 1.2: Low Sampling Rate and High Sampling Rate.**

**Figure 2.1: Flash ADC block diagram.**

**Figure 2.2: Successive Approximation ADC.**

**Figure 3.1: State diagram of 8 bit SAR type ADC**

**Figure 3.2: Block diagram of 8-bit SAR ADC**

**Figure 3.3: Flowchart of working of 8-bit FLASH ADC**

**Figure 4.1a: Waveform of SAR ADC**

**Figure 4.1b: Output of SAR at Time interval = 207ns**

**Figure 4.1c: Output of SAR at Time interval = 315ns**

**Figure 4.1.1a: Synthesis for SAR – Verilog Elaborated**

**Figure 4.1.1b: Synthesis for SAR – Synthesized**

**Figure 4.1.1c: Output of SAR using Cadence post Synthesis.**

**Figure 4.1.1d: Output of SAR at Time interval = 205.1ps**

**Figure 4.1.1e: Output of SAR at Time interval = 315.43ns**

**Figure 4.2a: Output of FLASH at Time interval = 30ns**

**Figure 4.2b: Output of FLASH at Time interval = 1010ns**

**Figure 4.2c: Output of FLASH at Time interval – 3120ns**

**LIST OF TABLES**

**Table 3.2: Specification of Designed Flash ADC**

**Table 3.2: Specification of Designed Flash ADC**

**Table 4.1.2 observations recorded for SAR ADC post synthesis.**

**ABBREVIATIONS**

|  |  |
| --- | --- |
| ADC | Analog to Digital Converter |
| DAC | Digital to Analog Converter |
| PGAs | Programmable Gain Amplifiers |
| LSB | Least Significant Bit |
| MSB | Most Significant Bit |
| SAR | Successive Approximation Register |

**NOMENCLATURE**

|  |  |
| --- | --- |
| *Q* | Resolution of the ADC |
| *Vin* | Input Voltage |
| *Vref* | Refence voltage |
| *Dout* | Digital output |
| *N* | Number of Bits |
| *S/H* | Sample and hold |
|  |  |

# **INTRODUCTION**

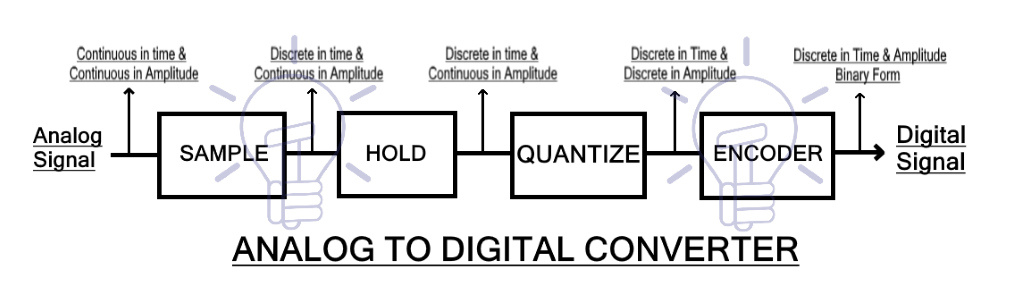
Signals in the real world are analog: light, sound, you name it. So, real-world signals must be converted into digital, using a circuit called ADC (Analog-to-Digital Converter), before they can be manipulated by digital equipment. But, why digital? There are some basic reasons to use digital signals instead of analog, noise being the number one.

Since analog signals can assume any value, noise is interpreted as being part of the original signal. Digital information isn’t only restricted to computers. When you talk on the phone, for example, your voice is converted into digital (at the central office switch, if you use an analog line, or at you home, if you use a digital line like ISDN or DSL), since your voice is analog and the communication between the phone switches is done digitally. But, why digital? There are some basic reasons to use digital signals instead of analog, noise being the number one.

Since analog signals can assume any value, noise is interpreted as being part of the original signal. For example, when you listen to a LP record, you hear noise because the needle is analog and thus do not know the difference from the music originally recorded from the noise inserted by dust or cracks.

## **ADC WORKING**

The basic ADC function is shown in Figure. This could also be referred to as a quantizer. Most ADC chips also include some of the support circuitry, such as clock oscillator for the sampling clock, reference (REF), the sample and hold function, and output data latches. In addition to these basic functions, some ADCs have additional circuitry built in. These functions could include multiplexers, sequencers, auto-calibration circuits, programmable gain amplifiers (PGAs), etc.



**Fig 1.1 :- Analog to Digital Converter**

Similar to DACs, some ADCs use external references and have a reference input terminal, while others have an output from an internal reference. In some instances, the ADC may have an internal reference that is pinned out through a resistor. This connection allows the reference to be filtered (using the internal R and an external C) or by allowing the internal reference to be overdriven by an external reference. If an ADC has an internal reference, its overall accuracy is specified when using that reference. If such an ADC is used with a perfectly accurate external reference, its absolute accuracy may actually be worse than when it is operated with its own internal reference. This is because it is trimmed for absolute accuracy when working with its own actual reference voltage, not with the nominal value.

## **SPECIFICATIONS**

1. **RESOLUTION**

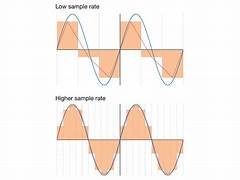
Resolution can also be defined electrically, and expressed in volts. The minimum change in voltage required to guarantee a change in the output code level is called the least significant bit (LSB) voltage. The resolution Q of the ADC is equal to the LSB voltage.

Q= (Vmax - Vmin)/N

where N = 2n

1. **SAMPLING RATE**

The analog signal is continuous in time and it is necessary to convert this to a flow of digital values. It is therefore required to define the rate at which new digital values are sampled from the analog signal. The rate of new values is called the sampling rate or sampling frequency of the converter. Implementation and Comparison Of Different ADC.



**Fig 1.2 :- Low Sampling Rate and High Sampling Rate**

1. **ACCURACY**

An ADC has several sources of errors. Quantization error and (assuming the ADC is intended to be linear) non-linearity are intrinsic to any analog-to-digital conversion. These errors are measured in a unit called the least significant bit (LSB).

* 1. **NON-LINEARITY**

All ADCs suffer from non-linearity errors caused by their physical imperfections, causing their output to deviate from a linear function (or some other function, in the case of a deliberately non-linear ADC) of their input. These errors can sometimes be mitigated by calibration, or prevented by testing.

Important parameters for linearity are integral non-linearity (INL) and differential nonlinearity (DNL). These non-linearities reduce the dynamic range of the signals that can be digitized by the ADC, also reducing the effective resolution of the ADC.

# **ARCHITECTURE OF ADC TO BE IMPLEMENTED**

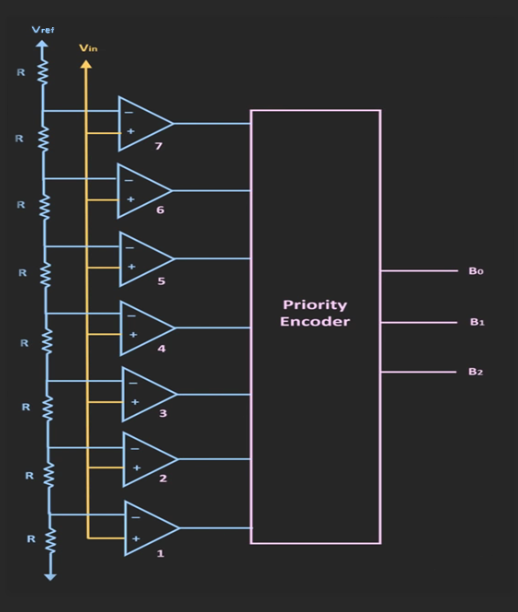
## **FLASH ADC ARCHITECTURE**

Flash analog-to-digital converters, also known as parallel ADCS, are the fastest way to convert an analog signal to a digital signal. They are suitable for applications requiring very large bandwidths. However, flash converters consume a lot of power, have relatively low resolution, and can be quite expensive. This limits them to high frequency applications that typically cannot be addressed any other way. Examples include data acquisition, satellite communication, radar processing, sampling oscilloscopes, and high-

Density disk drives.

Flash or parallel ADC s, are the fastest way to convert an analog signal to a digital signal. They are suitable for applications requiring very large bandwidths. However, flash converters consume a lot of power, have relatively low resolution, and can be quite expensive. This limits them to high frequency applications such as data acquisition, satellite communication, radar processing, sampling oscilloscopes, and high-density disk drives. To save the power consumption, quite an amount of power reduction techniques have been published for flash ADC, such as folding and interpolating, pipelined look-ahead architecture, distortion correction and so on .

Conceptually, in the flash architecture a set of 2n-1 comparators is used to directly measure an analog signal to a resolution of n bits. For a 4-bit flash ADC, the analog input is fed into 15 comparators, each of which is biased to compare the input to a discrete transition value. These values are spaced one least-significant bit (LSB=FS/2n) apart. The comparator outputs simultaneously present 2n-1 discrete digital output states. For an N-bit converter, the circuit has 2 N -1 comparators. A resistive divider network with 2N resistors provides the reference voltage. Each comparator produces a 1 when its analog input voltage is higher than the reference voltage applied to it. Otherwise, the comparator output is "0". A regenerative latch at each comparator output stores the result.

****

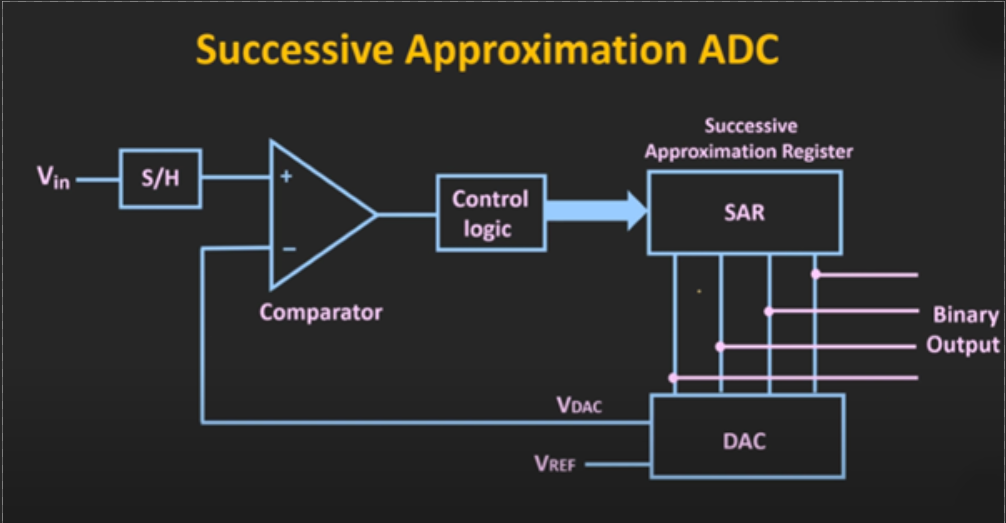
**Fig 2.1 :- Flash ADC block diagram**

## **SUCCESSIVE APPROXIMATION ADC ARCHITECTURE**

The successive-approximation [analog-to-digital converter](https://en.wikipedia.org/wiki/Analog-to-digital_converter) circuit typically consists of four chief subcircuits:

1. A [sample-and-hold](https://en.wikipedia.org/wiki/Sample-and-hold) circuit to acquire the input [voltage](https://en.wikipedia.org/wiki/Voltage) *V*in.
2. An analog voltage comparator that compares *V*in to the output of the internal [DAC](https://en.wikipedia.org/wiki/Digital-to-analog_converter) and outputs the result of the comparison to the successive-approximation [register](https://en.wikipedia.org/wiki/Processor_register) (SAR).
3. A successive-approximation register subcircuit designed to supply an approximate digital code of *V*in to the internal DAC.
4. An internal reference DAC that, for comparison with *V*ref, supplies the [comparator](https://en.wikipedia.org/wiki/Comparator) with an analog voltage equal to the digital code output of the SAR.

The successive approximation register is initialized so that the [most significant bit](https://en.wikipedia.org/wiki/Most_significant_bit) (MSB) is equal to a [digital](https://en.wikipedia.org/wiki/Digital_data) 1. This code is fed into the DAC, which then supplies the analog equivalent of this digital code (*V*ref/2) into the comparator circuit for comparison with the sampled input voltage. If this analog voltage exceeds *V*in, then the comparator causes the SAR to reset this bit; otherwise, the bit is left as 1. Then the next bit is set to 1 and the same test is done, continuing this [binary search](https://en.wikipedia.org/wiki/Binary_search_algorithm) until every bit in the SAR has been tested. The resulting code is the digital approximation of the sampled input voltage and is finally output by the SAR at the end of the conversion .



**Fig 2.2:- Successive Approximation ADC**

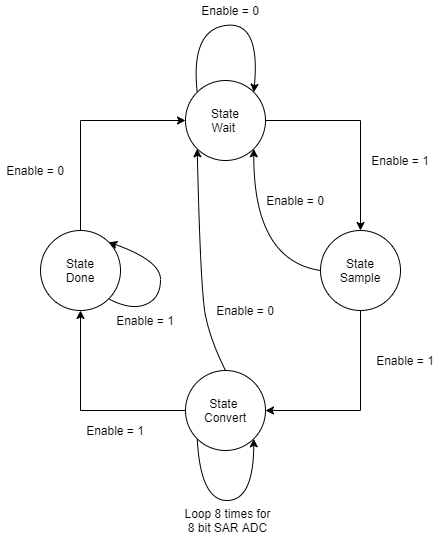
# **IMPLEMENTATION**

## **IMPLEMENTATION OF SAR ADC**

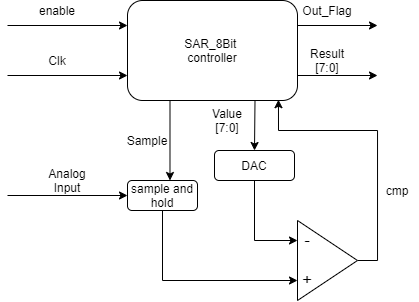
The SAR logic for the ADC is a state machine which takes the Clock signal as input. The clock signal is divided by 8 to generate the clock required for Sample & Hold circuit. 8 clock cycles are required for one complete conversion. Two additional clock cycles are for Start of Conversion and End of Conversion (Fig 3.1).

1. The Start of Conversion (State Wait) – 1st Clock cycle is used to reset all the registers in the logic block.
2. The 2nd clock cycle (State Sample and hold) – At this clock cycle the sample is taken from the input signal and held for further conversion.
3. The 3rd Clock cycle (State Convert) – Comparator performs comparison between Input data and reference voltage and generates a 1 or 0. The data is stored in the registers and applied to the DAC. The DAC sets the reference voltage for the next conversion cycle.
4. The same process is repeated for 3rd to 10th clock cycle to complete conversion process.
5. The End of Conversion (State Done) – 11th Clock cycle latches the output data to data bus and the out flag is set to 1.

**Fig 3.1: State diagram of 8-bit SAR type ADC**



The SAR logic operated over an 11-clock cycle period, with 2 cycles allotted for SOC & EOC and 8 clock cycles for digital data. An 8-bit conversion scheme (fig 3.2) is used to divide the entire voltage. For each clock cycle to the SAR controller, if the enable is 0, the controller will be inactive. When enable is one at State sample the controller sends a signal to sample and hold circuit to hold an analog value. Then a reference value is generated from the controller and given as input to DAC. The voltage from output of DAC and the held potential from sample and hold is compared. If sample value is greater than the generated reference voltage, bit will be set of not it is reset. Similarly, this operation is performed for all 8 bits of data and then the result is obtained. When the conversion is complete the out flag is set to 1.

****

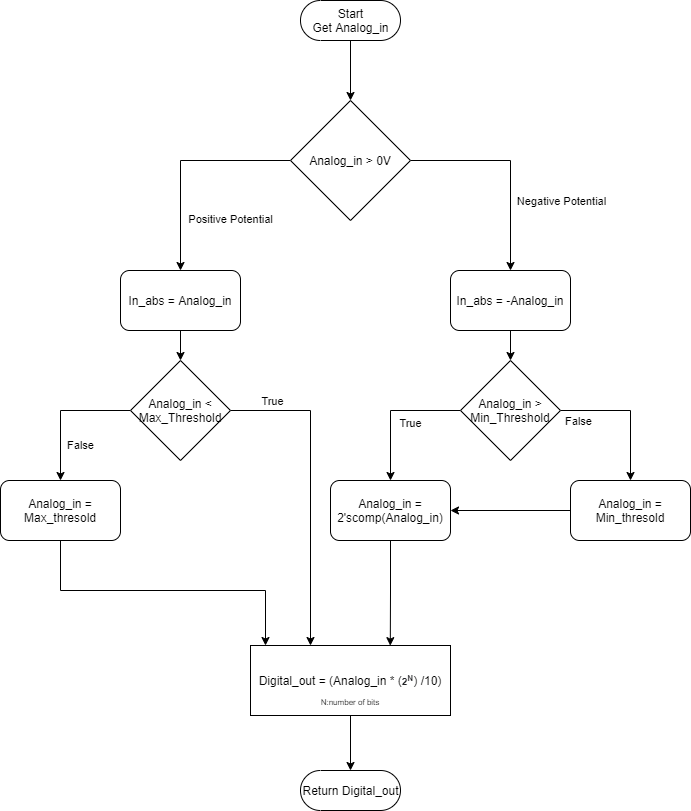
**Fig. 3.2: Block diagram of 8-bit SAR ADC**

|  |  |
| --- | --- |
| Parameter | Specification |
| Max Input Voltage | 256V |
| Min Input Voltage | 0V |
| Sampling | 100 samples / signal |
| Max Digital output | 88 h |
| Min Digital Output | 0 h |
| Clk Period for SAR | 90ps |

**Table 3.2 Specification of Designed Flash ADC**

## **IMPLEMENTATION OF FLASH ADC**

The Specification of designed Flash ADC is as shown in Table 3.2 The flash ADC architecture takes analog input and process analog input and gives digital output at an instance of the time. The analog input is validated if the voltage is positive or negative, if the voltage is negative then the absolute value is taken for further calculation. The input voltage is verified if the voltage parameter is in between Max Input voltage and Min Input voltage then the value is converted into respective digital value using the below equation. The complete flow chart of the flash ADC implementation is fig 3.3.



**Fig. 3.3: Flowchart of working of 8-bit FLASH ADC**

Digital\_out = (Analog\_in)\*2N /10

Where,

Analog\_in: input voltage applied to the ADC

N : is the Number of bits -1

Digital\_out : Digital encoded output

N is reduced by 1 since the most significant bit is signed bit.

|  |  |
| --- | --- |
| Parameter | Specification |
| Max Input Volatage | 10V |
| Min Input Voltage | -10V |
| Sampling | 1000 samples / signal |
| Max Digital output | 7F h |
| Min Digital Output | 81 h |

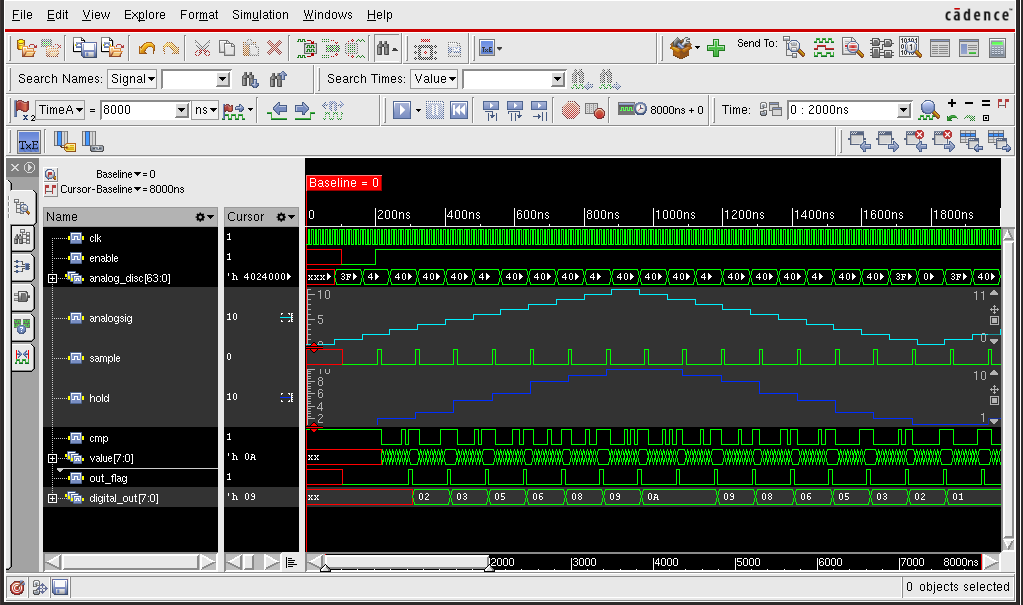
**Table 3.2 Specification of Designed Flash ADC**

# **RESULTS**

As per the specification defined in section 3 Verilog code is developed for respective analog to digital converter. The obtained results for different ADC are as shown in the below section.

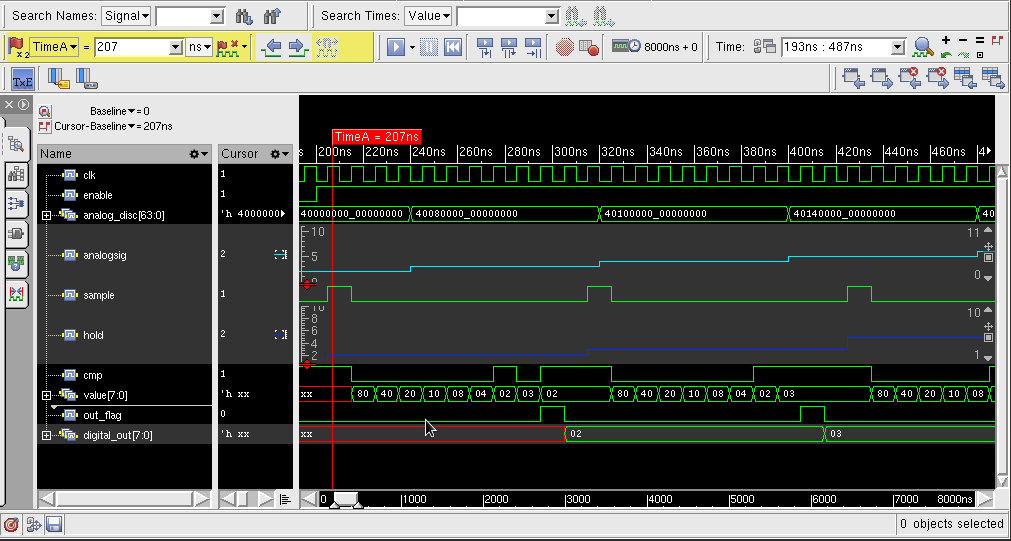
## **RESULTS OBTAINED FOR SAR ADC**

Below is the screenshot of simulation waveform of SAR ADC. Fig 4.1 show that the input voltage is varied from 0 to 10 V. The sample and hold circuit are triggered by the SAR logic, when the SAR sends the logic high to the Sample and hold block, it samples the input signal and holds the value of sampled input signal in the block. SAR generates the digital value, and that value is converted into Analog and compared with the input signal. If the input signal is equal to SAR generated value, out flag will become high and the process repeats for all samples of inputs.

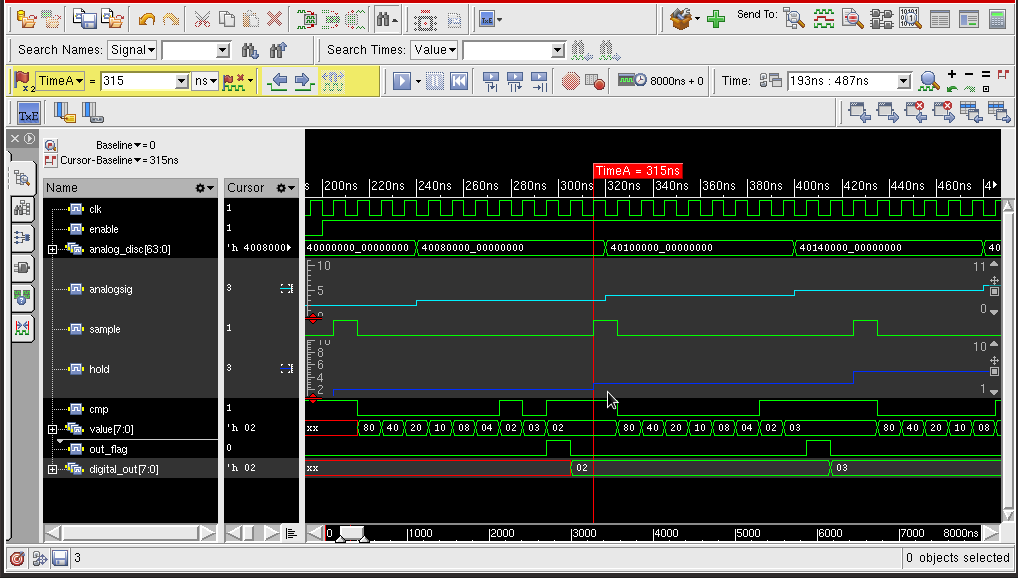


**Fig. 4.1a Waveform of SAR ADC**

As described above in fig 4.1b and fig 4.1c shows the conversion of analog input into digital signal. when enable becomes high SAR module sends a signal to sample and hold module to sample the data. Then the SAR module starts to generate the digital values which are compared in 8 clock cycles. Each bit is compared, then the corresponding binary values are obtained. As soon as the process is complete the out flag will become high and then the output will be sent to the output ports.



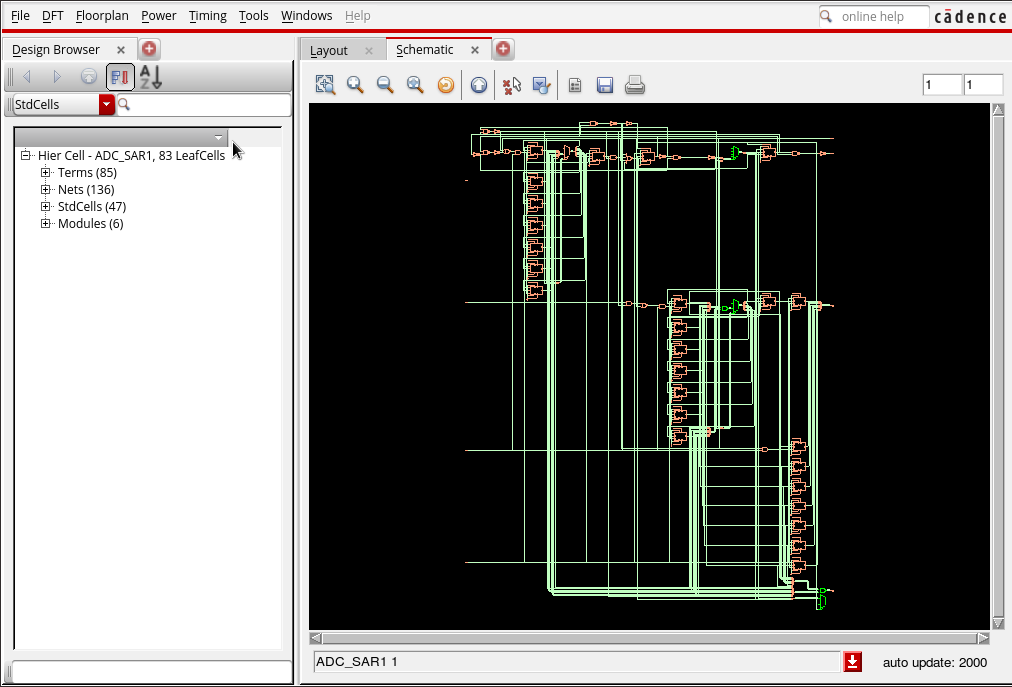
**Fig. 4.1b: Output of SAR at Time interval = 207ns**



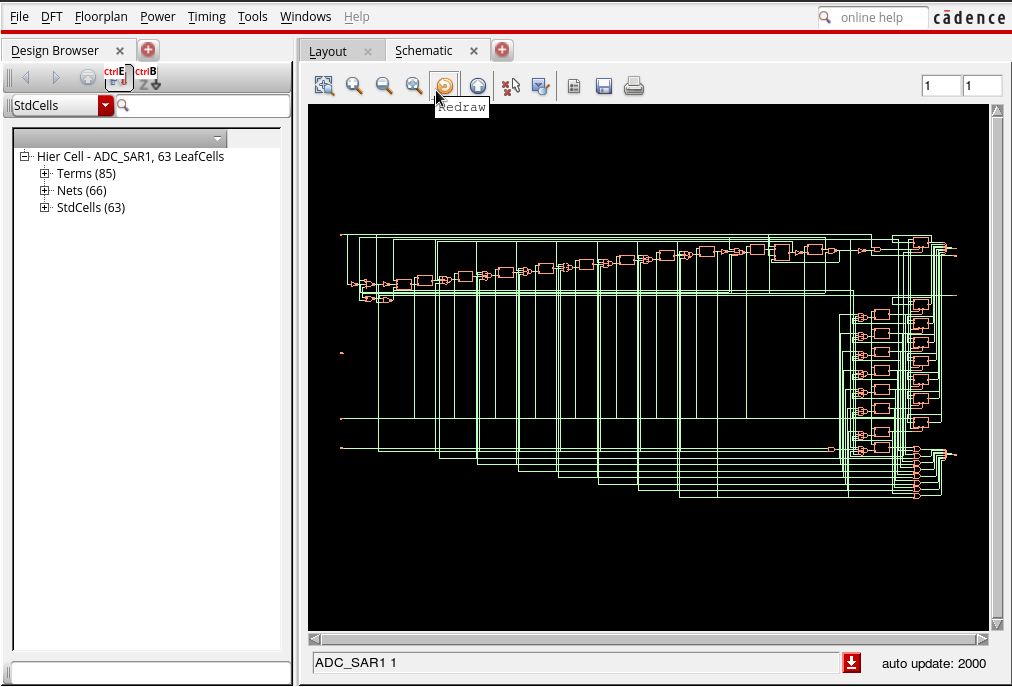
**Fig. 4.1c: Output of SAR at Time interval = 315ns**

### **SYNTHESIS OF SAR ADC**

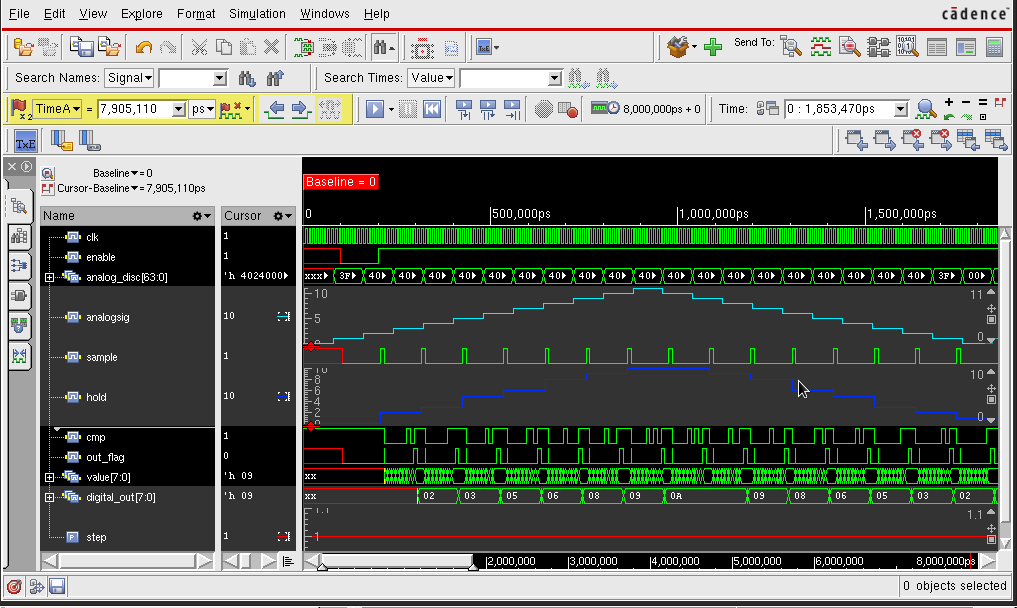
Netlist for the designed Verilog code is Generated using Genus tool by cadence. These are the results obtained.



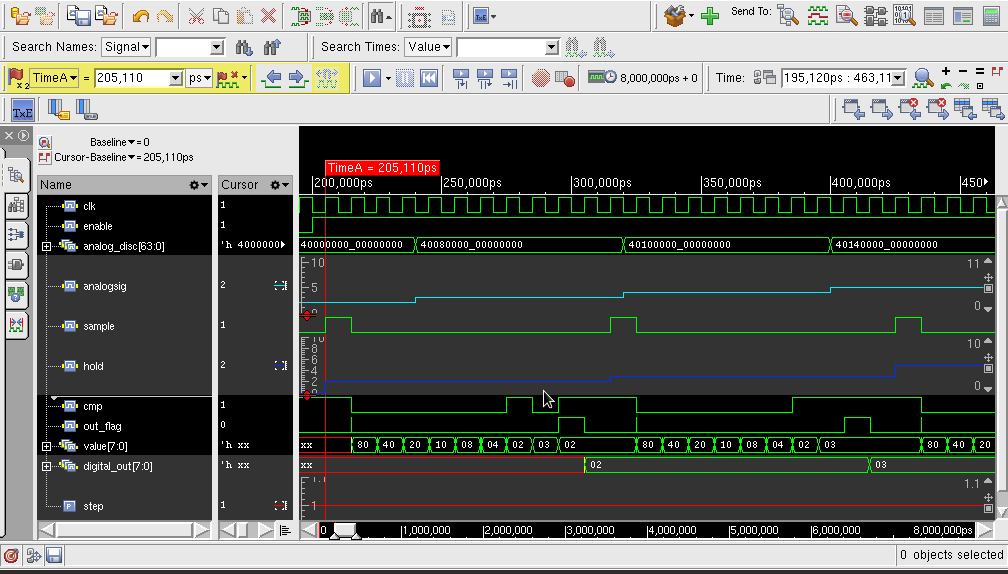
**Fig. 4.1.1a: Synthesis for SAR – Verilog Elaborated**



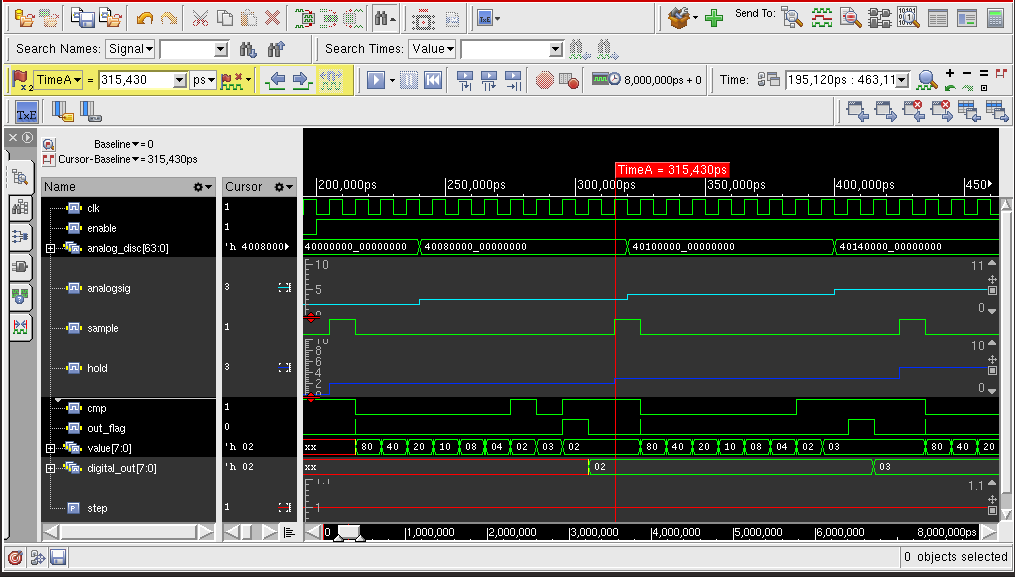
**Fig. 4.1.1b: Synthesis for SAR – Synthesized**



**Fig. 4.1.1c: Output of SAR using Cadence post Synthesis**



**Fig. 4.1.1d: Output of SAR at Time interval = 205.1ps**



**Fig. 4.1.1e: Output of SAR at Time interval = 315.43ns**

### **SIMULATED OUTPUT RESULT LOG FILE OF SAR ADC**

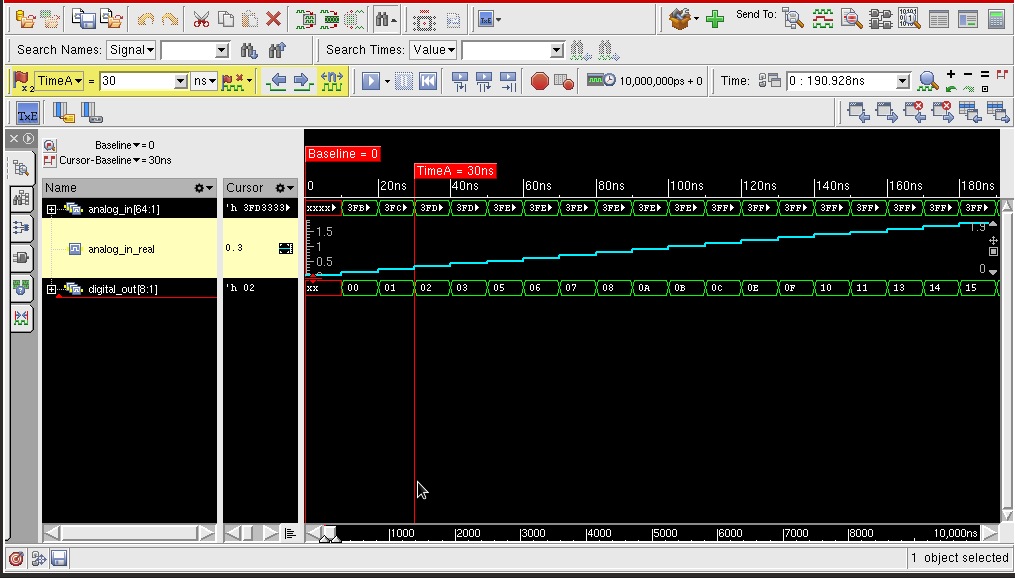
Please refer Appendix A

|  |  |
| --- | --- |
| Parameter | Results |
|  | After Synthesis Timing as Unconstrained |
| Area | 770.524 |
| Leakage Power | 9134.821 nW |
| Dynamic power | 109825.907 nW |
| Total Power | 18960.628 nW |
| Number of Leaf cells | 63 |

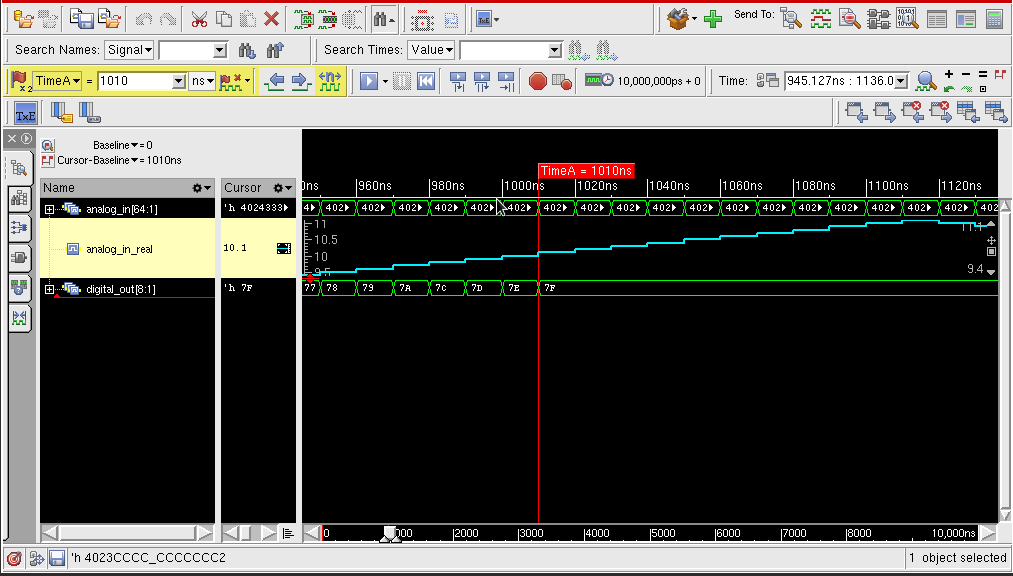
### **OBSERVATIONS RECORDED POST SYNTHESIS OF SAR ADC**

**Table 4.1.2 observations recorded for SAR ADC post synthesis**

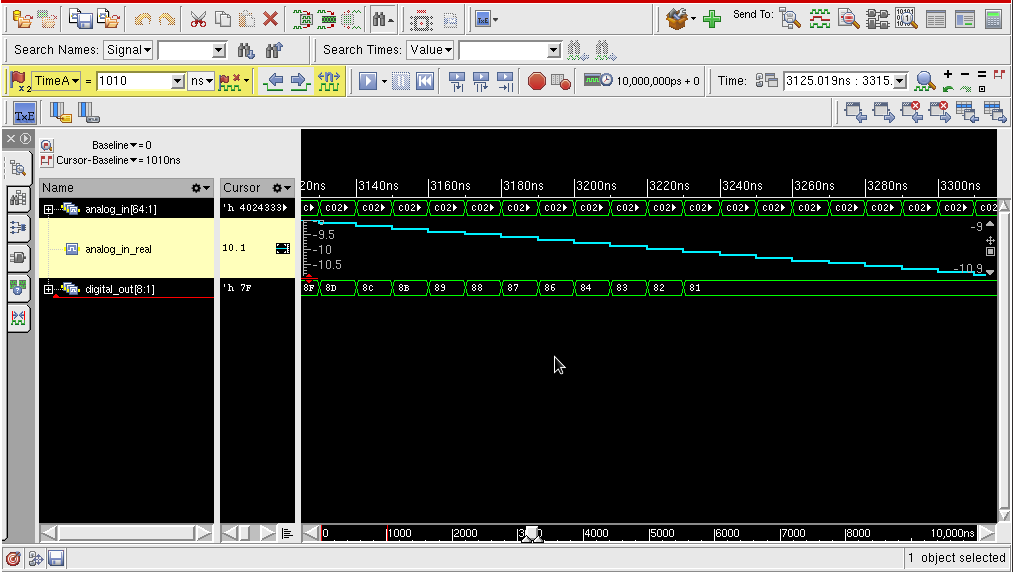
## **RESULTS OBTAINED FOR FLASH ADC**

Below is the screenshot of simulation waveform of Flash ADC. Fig 4.2a shows the output waveform of the initial voltage inputs and their corresponding converted digital signal. Each sampled input signal is converted using the equation as described in section 3.2

**Fig. 4.2a: Output of FLASH at Time interval = 30ns**



**Fig. 4.2b: Output of FLASH at Time interval = 1010ns**



**Fig. 4.2c: Output of FLASH at Time interval – 3120ns**

### **SIMULATED OUTPUT RESULT LOG FILE OF FLASH ADC**

Please refer to appendix A section 2

# **REFERENCES**

1. Mirzaie, Nahid, Ahmed Alzahmi, Chung-Ching Lin, Insoo Kim, and Gyung-Su Byun. "A low-power and performance-efficient SAR ADC design." In 2017 International SoC Design Conference (ISOCC), pp. 3-4. IEEE, 2017.
2. Huailiang Li and Jing Hu,” The Research on SAR ADC Integrated Circuit”,Journal of Physics: Conference Series, Volume 1314, 3rd International Conference on Electrical, 9–11 August 2019.
3. https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1023.html
4. https://components101.com/articles/analog-to-digital-adc-converters
5. https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1870.html
6. https://www.allaboutcircuits.com/textbook/digital/chpt-13/delta-sigma-adc/

# **APPENDIX A**

## **Simulation output of SAR ADC**

irun(64): 15.20-s069: (c) Copyright 1995-2019 Cadence Design Systems, Inc.

TOOL: irun(64) 15.20-s069: Started on Dec 12, 2020 at 16:00:46 IST

irun

ADC\_SAR.v

ADC\_SAR\_tb.v

-access rw

+gui

-------------------------------------

Relinquished control to SimVision...

ncsim>

ncsim> source /cad/INCISIVE152/tools/inca/files/ncsimrc

ncsim> database -open waves -into waves.shm -default

Created default SHM database waves

ncsim> probe -create -shm testbench.clk testbench.cmp testbench.enable testbench.hold testbench.holdanalog testbench.out\_flag testbench.result testbench.sample testbench.value

Created probe 1

ncsim> run

0enable=x out\_flag=x result=xxxxxxxx sample=x value=xxxxxxxx cmp=x state=xx mask=xxxxxxxx

100enable=0 out\_flag=x result=xxxxxxxx sample=x value=xxxxxxxx cmp=x state=xx mask=xxxxxxxx

110enable=0 out\_flag=0 result=xxxxxxxx sample=0 value=xxxxxxxx cmp=x state=00 mask=xxxxxxxx

200enable=1 out\_flag=0 result=xxxxxxxx sample=0 value=xxxxxxxx cmp=x state=00 mask=xxxxxxxx

210enable=1 out\_flag=0 result=xxxxxxxx sample=1 value=xxxxxxxx cmp=x state=01 mask=xxxxxxxx

230enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=0 state=10 mask=10000000

250enable=1 out\_flag=0 result=00000000 sample=0 value=01000000 cmp=1 state=10 mask=01000000

270enable=1 out\_flag=0 result=01000000 sample=0 value=01100000 cmp=0 state=10 mask=00100000

290enable=1 out\_flag=0 result=01000000 sample=0 value=01010000 cmp=0 state=10 mask=00010000

310enable=1 out\_flag=0 result=01000000 sample=0 value=01001000 cmp=0 state=10 mask=00001000

330enable=1 out\_flag=0 result=01000000 sample=0 value=01000100 cmp=1 state=10 mask=00000100

350enable=1 out\_flag=0 result=01000100 sample=0 value=01000110 cmp=1 state=10 mask=00000010

370enable=1 out\_flag=0 result=01000110 sample=0 value=01000111 cmp=0 state=10 mask=00000001

390enable=1 out\_flag=1 result=01000110 sample=0 value=01000110 cmp=1 state=11 mask=00000000

400enable=0 out\_flag=1 result=01000110 sample=0 value=01000110 cmp=1 state=11 mask=00000000

410enable=0 out\_flag=0 result=01000110 sample=0 value=01000110 cmp=1 state=00 mask=00000000

420enable=1 out\_flag=0 result=01000110 sample=0 value=01000110 cmp=1 state=00 mask=00000000

430enable=1 out\_flag=0 result=01000110 sample=1 value=01000110 cmp=1 state=01 mask=00000000

450enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=0 state=10 mask=10000000

470enable=1 out\_flag=0 result=00000000 sample=0 value=01000000 cmp=1 state=10 mask=01000000

490enable=1 out\_flag=0 result=01000000 sample=0 value=01100000 cmp=0 state=10 mask=00100000

510enable=1 out\_flag=0 result=01000000 sample=0 value=01010000 cmp=0 state=10 mask=00010000

530enable=1 out\_flag=0 result=01000000 sample=0 value=01001000 cmp=0 state=10 mask=00001000

550enable=1 out\_flag=0 result=01000000 sample=0 value=01000100 cmp=1 state=10 mask=00000100

570enable=1 out\_flag=0 result=01000100 sample=0 value=01000110 cmp=1 state=10 mask=00000010

590enable=1 out\_flag=0 result=01000110 sample=0 value=01000111 cmp=0 state=10 mask=00000001

610enable=1 out\_flag=1 result=01000110 sample=0 value=01000110 cmp=1 state=11 mask=00000000

620enable=0 out\_flag=1 result=01000110 sample=0 value=01000110 cmp=1 state=11 mask=00000000

630enable=0 out\_flag=0 result=01000110 sample=0 value=01000110 cmp=1 state=00 mask=00000000

640enable=1 out\_flag=0 result=01000110 sample=0 value=01000110 cmp=1 state=00 mask=00000000

650enable=1 out\_flag=0 result=01000110 sample=1 value=01000110 cmp=1 state=01 mask=00000000

670enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=0 state=10 mask=10000000

690enable=1 out\_flag=0 result=00000000 sample=0 value=01000000 cmp=1 state=10 mask=01000000

710enable=1 out\_flag=0 result=01000000 sample=0 value=01100000 cmp=1 state=10 mask=00100000

730enable=1 out\_flag=0 result=01100000 sample=0 value=01110000 cmp=0 state=10 mask=00010000

750enable=1 out\_flag=0 result=01100000 sample=0 value=01101000 cmp=1 state=10 mask=00001000

770enable=1 out\_flag=0 result=01101000 sample=0 value=01101100 cmp=1 state=10 mask=00000100

790enable=1 out\_flag=0 result=01101100 sample=0 value=01101110 cmp=1 state=10 mask=00000010

810enable=1 out\_flag=0 result=01101110 sample=0 value=01101111 cmp=0 state=10 mask=00000001

830enable=1 out\_flag=1 result=01101110 sample=0 value=01101110 cmp=1 state=11 mask=00000000

840enable=0 out\_flag=1 result=01101110 sample=0 value=01101110 cmp=1 state=11 mask=00000000

850enable=0 out\_flag=0 result=01101110 sample=0 value=01101110 cmp=1 state=00 mask=00000000

860enable=1 out\_flag=0 result=01101110 sample=0 value=01101110 cmp=1 state=00 mask=00000000

870enable=1 out\_flag=0 result=01101110 sample=1 value=01101110 cmp=1 state=01 mask=00000000

890enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=0 state=10 mask=10000000

910enable=1 out\_flag=0 result=00000000 sample=0 value=01000000 cmp=1 state=10 mask=01000000

930enable=1 out\_flag=0 result=01000000 sample=0 value=01100000 cmp=1 state=10 mask=00100000

950enable=1 out\_flag=0 result=01100000 sample=0 value=01110000 cmp=0 state=10 mask=00010000

970enable=1 out\_flag=0 result=01100000 sample=0 value=01101000 cmp=1 state=10 mask=00001000

990enable=1 out\_flag=0 result=01101000 sample=0 value=01101100 cmp=1 state=10 mask=00000100

1010enable=1 out\_flag=0 result=01101100 sample=0 value=01101110 cmp=1 state=10 mask=00000010

1030enable=1 out\_flag=0 result=01101110 sample=0 value=01101111 cmp=0 state=10 mask=00000001

1050enable=1 out\_flag=1 result=01101110 sample=0 value=01101110 cmp=1 state=11 mask=00000000

1060enable=0 out\_flag=1 result=01101110 sample=0 value=01101110 cmp=1 state=11 mask=00000000

1070enable=0 out\_flag=0 result=01101110 sample=0 value=01101110 cmp=1 state=00 mask=00000000

1080enable=1 out\_flag=0 result=01101110 sample=0 value=01101110 cmp=1 state=00 mask=00000000

1090enable=1 out\_flag=0 result=01101110 sample=1 value=01101110 cmp=1 state=01 mask=00000000

1110enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=1 state=10 mask=10000000

1130enable=1 out\_flag=0 result=10000000 sample=0 value=11000000 cmp=1 state=10 mask=01000000

1150enable=1 out\_flag=0 result=11000000 sample=0 value=11100000 cmp=0 state=10 mask=00100000

1170enable=1 out\_flag=0 result=11000000 sample=0 value=11010000 cmp=0 state=10 mask=00010000

1190enable=1 out\_flag=0 result=11000000 sample=0 value=11001000 cmp=1 state=10 mask=00001000

1210enable=1 out\_flag=0 result=11001000 sample=0 value=11001100 cmp=1 state=10 mask=00000100

1230enable=1 out\_flag=0 result=11001100 sample=0 value=11001110 cmp=0 state=10 mask=00000010

1250enable=1 out\_flag=0 result=11001100 sample=0 value=11001101 cmp=0 state=10 mask=00000001

1270enable=1 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1280enable=0 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1290enable=0 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1300enable=1 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1310enable=1 out\_flag=0 result=11001100 sample=1 value=11001100 cmp=1 state=01 mask=00000000

1330enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=1 state=10 mask=10000000

1350enable=1 out\_flag=0 result=10000000 sample=0 value=11000000 cmp=1 state=10 mask=01000000

1370enable=1 out\_flag=0 result=11000000 sample=0 value=11100000 cmp=0 state=10 mask=00100000

1390enable=1 out\_flag=0 result=11000000 sample=0 value=11010000 cmp=0 state=10 mask=00010000

1410enable=1 out\_flag=0 result=11000000 sample=0 value=11001000 cmp=1 state=10 mask=00001000

1430enable=1 out\_flag=0 result=11001000 sample=0 value=11001100 cmp=1 state=10 mask=00000100

1450enable=1 out\_flag=0 result=11001100 sample=0 value=11001110 cmp=0 state=10 mask=00000010

1470enable=1 out\_flag=0 result=11001100 sample=0 value=11001101 cmp=0 state=10 mask=00000001

1490enable=1 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1500enable=0 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1510enable=0 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1520enable=1 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1530enable=1 out\_flag=0 result=11001100 sample=1 value=11001100 cmp=1 state=01 mask=00000000

1550enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=1 state=10 mask=10000000

1570enable=1 out\_flag=0 result=10000000 sample=0 value=11000000 cmp=1 state=10 mask=01000000

1590enable=1 out\_flag=0 result=11000000 sample=0 value=11100000 cmp=0 state=10 mask=00100000

1610enable=1 out\_flag=0 result=11000000 sample=0 value=11010000 cmp=0 state=10 mask=00010000

1630enable=1 out\_flag=0 result=11000000 sample=0 value=11001000 cmp=1 state=10 mask=00001000

1650enable=1 out\_flag=0 result=11001000 sample=0 value=11001100 cmp=1 state=10 mask=00000100

1670enable=1 out\_flag=0 result=11001100 sample=0 value=11001110 cmp=0 state=10 mask=00000010

1690enable=1 out\_flag=0 result=11001100 sample=0 value=11001101 cmp=0 state=10 mask=00000001

1710enable=1 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1720enable=0 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1730enable=0 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1740enable=1 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1750enable=1 out\_flag=0 result=11001100 sample=1 value=11001100 cmp=1 state=01 mask=00000000

1770enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=1 state=10 mask=10000000

1790enable=1 out\_flag=0 result=10000000 sample=0 value=11000000 cmp=1 state=10 mask=01000000

1810enable=1 out\_flag=0 result=11000000 sample=0 value=11100000 cmp=0 state=10 mask=00100000

1830enable=1 out\_flag=0 result=11000000 sample=0 value=11010000 cmp=0 state=10 mask=00010000

1850enable=1 out\_flag=0 result=11000000 sample=0 value=11001000 cmp=1 state=10 mask=00001000

1870enable=1 out\_flag=0 result=11001000 sample=0 value=11001100 cmp=1 state=10 mask=00000100

1890enable=1 out\_flag=0 result=11001100 sample=0 value=11001110 cmp=0 state=10 mask=00000010

1910enable=1 out\_flag=0 result=11001100 sample=0 value=11001101 cmp=0 state=10 mask=00000001

1930enable=1 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1940enable=0 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

1950enable=0 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1960enable=1 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

1970enable=1 out\_flag=0 result=11001100 sample=1 value=11001100 cmp=1 state=01 mask=00000000

1990enable=1 out\_flag=0 result=00000000 sample=0 value=10000000 cmp=1 state=10 mask=10000000

2010enable=1 out\_flag=0 result=10000000 sample=0 value=11000000 cmp=1 state=10 mask=01000000

2030enable=1 out\_flag=0 result=11000000 sample=0 value=11100000 cmp=0 state=10 mask=00100000

2050enable=1 out\_flag=0 result=11000000 sample=0 value=11010000 cmp=0 state=10 mask=00010000

2070enable=1 out\_flag=0 result=11000000 sample=0 value=11001000 cmp=1 state=10 mask=00001000

2090enable=1 out\_flag=0 result=11001000 sample=0 value=11001100 cmp=1 state=10 mask=00000100

2110enable=1 out\_flag=0 result=11001100 sample=0 value=11001110 cmp=0 state=10 mask=00000010

2130enable=1 out\_flag=0 result=11001100 sample=0 value=11001101 cmp=0 state=10 mask=00000001

2150enable=1 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

2160enable=0 out\_flag=1 result=11001100 sample=0 value=11001100 cmp=1 state=11 mask=00000000

2170enable=0 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

2180enable=1 out\_flag=0 result=11001100 sample=0 value=11001100 cmp=1 state=00 mask=00000000

2190enable=1 out\_flag=0 result=11001100 sample=1 value=11001100 cmp=1 state=01 mask=00000000

Simulation stopped via $stop(1) at time 2200 NS + 0

ncsim> exit

...Regained control from SimVision

-------------------------------------

TOOL: irun(64) 15.20-s069: Exiting on Dec 12, 2020 at 16:02:47 IST (total: 00:02:01)

## **Reports generated by the Genus Tools for Synthesis of SAR ADC**

### **Power**

============================================================

Generated by: Genus(TM) Synthesis Solution 18.10-p003\_1

Generated on: Jan 20 2021 03:00:22 pm

Module: ADC\_SAR1

Operating conditions: fast (balanced\_tree)

Wireload mode: enclosed

Area mode: timing library

============================================================

Leakage Dynamic Total

Instance Cells Power(nW) Power(nW) Power(nW)

------------------------------------------------

ADC\_SAR1 63 9134.821 109825.807 118960.628

### **Area**

============================================================

Generated by: Genus(TM) Synthesis Solution 18.10-p003\_1

Generated on: Jan 20 2021 03:00:22 pm

Module: ADC\_SAR1

Operating conditions: fast (balanced\_tree)

Wireload mode: enclosed

Area mode: timing library

============================================================

Instance Module Cell Count Cell Area Net Area Total Area Wireload

----------------------------------------------------------------------------

ADC\_SAR1 63 770.524 0.000 770.524 <none> (D)

(D) = wireload is default in technology library

## **Simulation Output of Flash ADC**

irun(64): 15.20-s069: (c) Copyright 1995-2019 Cadence Design Systems, Inc.

TOOL: irun(64) 15.20-s069: Started on Jan 13, 2021 at 14:41:33 IST

irun

FlashDesignModule.v

FlashTestbench.v

-access rw

+gui

Caching library 'worklib' ....... Done

Elaborating the design hierarchy:

Top level design units:

ADC\_8bit\_tf

Building instance overlay tables: .................... Done

Building instance specific data structures.

Loading native compiled code: .................... Done

Design hierarchy summary:

Instances Unique

Modules: 2 2

Registers: 12 12

Vectored wires: 2 -

Always blocks: 1 1

Initial blocks: 2 2

Cont. assignments: 0 1

Pseudo assignments: 1 1

Simulation timescale: 1ps

Writing initial simulation snapshot: worklib.ADC\_8bit\_tf:v

-------------------------------------

Relinquished control to SimVision...

ncsim>

ncsim> source /cad/INCISIVE152/tools/inca/files/ncsimrc

ncsim> database -open waves -into waves.shm -default

Created default SHM database waves

ncsim> probe -create -shm ADC\_8bit\_tf.analog\_in ADC\_8bit\_tf.analog\_in\_real ADC\_8bit\_tf.digital\_out

Created probe 1

ncsim> run

0 0 x

10 0.1 x

20 0.2 x

25 0.2 2

30 0.3 2

40 0.4 2

50 0.5 5

60 0.6 5

70 0.7 5

75 0.7 8

80 0.7999999999999999 8

90 0.8999999999999999 8

100 0.9999999999999999 11

110 1.1 11

120 1.2 11

125 1.2 15

130 1.3 15

140 1.4 15

150 1.5 17

160 1.6 17

170 1.7 17

175 1.7 21

180 1.8 21

190 1.900000000000001 21

200 2 24

210 2.100000000000001 24

220 2.200000000000001 24

225 2.200000000000001 28

230 2.300000000000001 28

240 2.400000000000001 28

250 2.500000000000001 30

260 2.600000000000001 30

270 2.700000000000001 30

275 2.700000000000001 34

280 2.800000000000001 34

290 2.900000000000001 34

300 3.000000000000001 37

310 3.100000000000001 37

320 3.200000000000002 37

325 3.200000000000002 40

330 3.300000000000002 40

340 3.400000000000002 40

350 3.500000000000002 43

360 3.600000000000002 43

370 3.700000000000002 43

375 3.700000000000002 47

380 3.800000000000002 47

390 3.900000000000002 47

400 4.000000000000002 49

410 4.100000000000001 49

420 4.200000000000001 49

425 4.200000000000001 53

430 4.300000000000001 53

440 4.4 53

450 4.5 56

460 4.6 56

470 4.699999999999999 56

475 4.699999999999999 60

480 4.799999999999999 60

490 4.899999999999999 60

500 4.999999999999998 62

510 5.099999999999998 62

520 5.199999999999998 62

525 5.199999999999998 66

530 5.299999999999997 66

540 5.399999999999997 66

550 5.499999999999996 69

560 5.599999999999996 69

570 5.699999999999996 69

575 5.699999999999996 72

580 5.799999999999995 72

590 5.899999999999995 72

600 5.999999999999995 75

610 6.099999999999994 75

620 6.199999999999994 75

625 6.199999999999994 79

630 6.299999999999994 79

640 6.399999999999993 79

650 6.499999999999993 81

660 6.599999999999993 81

670 6.699999999999992 81

675 6.699999999999992 85

680 6.799999999999992 85

690 6.899999999999991 85

700 6.999999999999991 88

710 7.099999999999991 88

720 7.19999999999999 88

725 7.19999999999999 92

730 7.29999999999999 92

740 7.39999999999999 92

750 7.499999999999989 94

760 7.599999999999989 94

770 7.699999999999989 94

775 7.699999999999989 98

780 7.799999999999988 98

790 7.899999999999988 98

800 7.999999999999988 101

810 8.099999999999987 101

820 8.199999999999987 101

825 8.199999999999987 104

830 8.299999999999986 104

840 8.399999999999986 104

850 8.499999999999986 107

860 8.599999999999985 107

870 8.699999999999985 107

875 8.699999999999985 111

880 8.799999999999985 111

890 8.899999999999984 111

900 8.999999999999984 113

910 9.099999999999984 113

920 9.199999999999983 113

925 9.199999999999983 117

930 9.299999999999983 117

940 9.399999999999983 117

950 9.499999999999982 120

960 9.599999999999982 120

970 9.699999999999982 120

975 9.699999999999982 124

980 9.799999999999981 124

990 9.899999999999981 124

1000 9.99999999999998 126

1010 10.09999999999998 126

1020 10.19999999999998 126

1025 10.19999999999998 127

1030 10.29999999999998 127

1040 10.39999999999998 127

1050 10.49999999999998 127

1060 10.59999999999998 127

1070 10.69999999999998 127

1080 10.79999999999998 127

1090 10.89999999999998 127

1100 10.99999999999998 127

1110 11.09999999999998 127

1120 10.99999999999998 127

1130 10.89999999999998 127

1140 10.79999999999998 127

1150 10.69999999999998 127

1160 10.59999999999998 127

1170 10.49999999999998 127

1180 10.39999999999998 127

1190 10.29999999999998 127

1200 10.19999999999998 127

1210 10.09999999999998 127

1220 9.99999999999998 127

1230 9.899999999999981 127

1240 9.799999999999981 127

1250 9.699999999999982 125

1260 9.599999999999982 125

1270 9.499999999999982 125

1275 9.499999999999982 121

1280 9.399999999999983 121

1290 9.299999999999983 121

1300 9.199999999999983 119

1310 9.099999999999984 119

1320 8.999999999999984 119

1325 8.999999999999984 115

1330 8.899999999999984 115

1340 8.799999999999985 115

1350 8.699999999999985 112

1360 8.599999999999985 112

1370 8.499999999999986 112

1375 8.499999999999986 108

1380 8.399999999999986 108

1390 8.299999999999986 108

1400 8.199999999999987 106

1410 8.099999999999987 106

1420 7.999999999999988 106

1425 7.999999999999988 102

1430 7.899999999999988 102

1440 7.799999999999988 102

1450 7.699999999999989 99

1460 7.599999999999989 99

1470 7.499999999999989 99

1475 7.499999999999989 95

1480 7.39999999999999 95

1490 7.29999999999999 95

1500 7.19999999999999 93

1510 7.099999999999991 93

1520 6.999999999999991 93

1525 6.999999999999991 89

1530 6.899999999999991 89

1540 6.799999999999992 89

1550 6.699999999999992 87

1560 6.599999999999993 87

1570 6.499999999999993 87

1575 6.499999999999993 83

1580 6.399999999999993 83

1590 6.299999999999994 83

1600 6.199999999999994 80

1610 6.099999999999994 80

1620 5.999999999999995 80

1625 5.999999999999995 76

1630 5.899999999999995 76

1640 5.799999999999995 76

1650 5.699999999999996 74

1660 5.599999999999996 74

1670 5.499999999999996 74

1675 5.499999999999996 70

1680 5.399999999999997 70

1690 5.299999999999997 70

1700 5.199999999999998 67

1710 5.099999999999998 67

1720 4.999999999999998 67

1725 4.999999999999998 63

1730 4.899999999999999 63

1740 4.799999999999999 63

1750 4.699999999999999 61

1760 4.6 61

1770 4.5 61

1775 4.5 57

1780 4.4 57

1790 4.300000000000001 57

1800 4.200000000000001 55

1810 4.100000000000001 55

1820 4.000000000000002 55

1825 4.000000000000002 51

1830 3.900000000000002 51

1840 3.800000000000002 51

1850 3.700000000000002 48

1860 3.600000000000001 48

1870 3.500000000000001 48

1875 3.500000000000001 44

1880 3.400000000000001 44

1890 3.300000000000001 44

1900 3.200000000000001 42

1910 3.100000000000001 42

1920 3.000000000000001 42

1925 3.000000000000001 38

1930 2.900000000000001 38

1940 2.800000000000001 38

1950 2.700000000000001 35

1960 2.600000000000001 35

1970 2.5 35

1975 2.5 32

1980 2.4 32

1990 2.3 32

2000 2.2 29

2010 2.1 29

2020 2 29

2025 2 25

2030 1.9 25

2040 1.8 25

2050 1.7 23

2060 1.6 23

2070 1.5 23

2075 1.5 19

2080 1.399999999999999 19

2090 1.299999999999999 19

2100 1.199999999999999 16

2110 1.099999999999999 16

2120 0.9999999999999992 16

2125 0.9999999999999992 12

2130 0.8999999999999992 12

2140 0.7999999999999993 12

2150 0.6999999999999993 10

2160 0.5999999999999993 10

2170 0.4999999999999993 10

2175 0.4999999999999993 6

2180 0.3999999999999994 6

2190 0.2999999999999994 6

2200 0.1999999999999994 3

2210 0.09999999999999937 3

2220 -6.38378239159465e-16 3

2225 -6.38378239159465e-16 0

2230 -0.1000000000000006 0

2240 -0.2000000000000006 0

2250 -0.3000000000000007 254

2260 -0.4000000000000007 254

2270 -0.5000000000000007 254

2275 -0.5000000000000007 250

2280 -0.6000000000000006 250

2290 -0.7000000000000006 250

2300 -0.8000000000000006 248

2310 -0.9000000000000006 248

2320 -1.000000000000001 248

2325 -1.000000000000001 244

2330 -1.100000000000001 244

2340 -1.200000000000001 244

2350 -1.300000000000001 241

2360 -1.400000000000001 241

2370 -1.500000000000001 241

2375 -1.500000000000001 237

2380 -1.600000000000001 237

2390 -1.700000000000001 237

2400 -1.800000000000001 235

2410 -1.900000000000001 235

2420 -2.000000000000001 235

2425 -2.000000000000001 231

2430 -2.100000000000001 231

2440 -2.200000000000002 231

2450 -2.300000000000002 228

2460 -2.400000000000002 228

2470 -2.500000000000002 228

2475 -2.500000000000002 224

2480 -2.600000000000002 224

2490 -2.700000000000002 224

2500 -2.800000000000002 222

2510 -2.900000000000002 222

2520 -3.000000000000002 222

2525 -3.000000000000002 218

2530 -3.100000000000002 218

2540 -3.200000000000002 218

2550 -3.300000000000002 216

2560 -3.400000000000003 216

2570 -3.500000000000003 216

2575 -3.500000000000003 212

2580 -3.600000000000003 212

2590 -3.700000000000003 212

2600 -3.800000000000003 209

2610 -3.900000000000003 209

2620 -4.000000000000003 209

2625 -4.000000000000003 205

2630 -4.100000000000002 205

2640 -4.200000000000002 205

2650 -4.300000000000002 203

2660 -4.400000000000001 203

2670 -4.500000000000001 203

2675 -4.500000000000001 199

2680 -4.600000000000001 199

2690 -4.7 199

2700 -4.8 196

2710 -4.899999999999999 196

2720 -4.999999999999999 196

2725 -4.999999999999999 193

2730 -5.099999999999999 193

2740 -5.199999999999998 193

2750 -5.299999999999998 190

2760 -5.399999999999998 190

2770 -5.499999999999997 190

2775 -5.499999999999997 186

2780 -5.599999999999997 186

2790 -5.699999999999997 186

2800 -5.799999999999996 184

2810 -5.899999999999996 184

2820 -5.999999999999996 184

2825 -5.999999999999996 180

2830 -6.099999999999995 180

2840 -6.199999999999995 180

2850 -6.299999999999994 177

2860 -6.399999999999994 177

2870 -6.499999999999994 177

2875 -6.499999999999994 173

2880 -6.599999999999993 173

2890 -6.699999999999993 173

2900 -6.799999999999993 171

2910 -6.899999999999992 171

2920 -6.999999999999992 171

2925 -6.999999999999992 167

2930 -7.099999999999992 167

2940 -7.199999999999991 167

2950 -7.299999999999991 164

2960 -7.399999999999991 164

2970 -7.49999999999999 164

2975 -7.49999999999999 161

2980 -7.59999999999999 161

2990 -7.69999999999999 161

3000 -7.799999999999989 158

3010 -7.899999999999989 158

3020 -7.999999999999988 158

3025 -7.999999999999988 154

3030 -8.099999999999989 154

3040 -8.199999999999989 154

3050 -8.299999999999988 152

3060 -8.399999999999988 152

3070 -8.499999999999988 152

3075 -8.499999999999988 148

3080 -8.599999999999987 148

3090 -8.699999999999987 148

3100 -8.799999999999986 145

3110 -8.899999999999986 145

3120 -8.999999999999986 145

3125 -8.999999999999986 141

3130 -9.099999999999985 141

3140 -9.199999999999985 141

3150 -9.299999999999985 139

3160 -9.399999999999984 139

3170 -9.499999999999984 139

3175 -9.499999999999984 135

3180 -9.599999999999984 135

3190 -9.699999999999983 135

3200 -9.799999999999983 132

3210 -9.899999999999983 132

3220 -9.999999999999982 132

3225 -9.999999999999982 129

3230 -10.09999999999998 129

3240 -10.19999999999998 129

3250 -10.29999999999998 129

3260 -10.39999999999998 129

3270 -10.49999999999998 129

3280 -10.59999999999998 129

3290 -10.69999999999998 129

3300 -10.79999999999998 129

3310 -10.89999999999998 129

3320 -10.99999999999998 129

3330 -11.09999999999998 129

3340 -10.99999999999998 129

3350 -10.89999999999998 129

3360 -10.79999999999998 129

3370 -10.69999999999998 129

3380 -10.59999999999998 129

3390 -10.49999999999998 129

3400 -10.39999999999998 129

3410 -10.29999999999998 129

3420 -10.19999999999998 129

3430 -10.09999999999998 129

3440 -9.999999999999982 129

3450 -9.899999999999983 129

3460 -9.799999999999983 129

3470 -9.699999999999983 129

3475 -9.699999999999983 132

3480 -9.599999999999984 132

3490 -9.499999999999984 132

3500 -9.399999999999984 135

3510 -9.299999999999985 135

3520 -9.199999999999985 135

3525 -9.199999999999985 139

3530 -9.099999999999985 139

3540 -8.999999999999986 139

3550 -8.899999999999986 141

3560 -8.799999999999986 141

3570 -8.699999999999987 141

3575 -8.699999999999987 145

3580 -8.599999999999987 145

3590 -8.499999999999988 145

3600 -8.399999999999988 148

3610 -8.299999999999988 148

3620 -8.199999999999989 148

3625 -8.199999999999989 152

3630 -8.099999999999989 152

3640 -7.999999999999989 152

3650 -7.89999999999999 154

3660 -7.79999999999999 154

3670 -7.69999999999999 154

3675 -7.69999999999999 158

3680 -7.599999999999991 158

3690 -7.499999999999991 158

3700 -7.399999999999991 161

3710 -7.299999999999992 161

3720 -7.199999999999992 161

3725 -7.199999999999992 164

3730 -7.099999999999993 164

3740 -6.999999999999993 164

3750 -6.899999999999993 167

3760 -6.799999999999994 167

3770 -6.699999999999994 167

3775 -6.699999999999994 171

3780 -6.599999999999994 171

3790 -6.499999999999995 171

3800 -6.399999999999995 173

3810 -6.299999999999995 173

3820 -6.199999999999996 173

3825 -6.199999999999996 177

3830 -6.099999999999996 177

3840 -5.999999999999996 177

3850 -5.899999999999997 180

3860 -5.799999999999997 180

3870 -5.699999999999998 180

3875 -5.699999999999998 184

3880 -5.599999999999998 184

3890 -5.499999999999998 184

3900 -5.399999999999999 186

3910 -5.299999999999999 186

3920 -5.199999999999999 186

3925 -5.199999999999999 190

3930 -5.1 190

3940 -5 190

3950 -4.9 192

3960 -4.800000000000001 192

3970 -4.700000000000001 192

3975 -4.700000000000001 196

3980 -4.600000000000001 196

3990 -4.500000000000002 196

4000 -4.400000000000002 199

4010 -4.300000000000002 199

4020 -4.200000000000003 199

4025 -4.200000000000003 203

4030 -4.100000000000003 203

4040 -4.000000000000004 203

4050 -3.900000000000003 205

4060 -3.800000000000003 205

4070 -3.700000000000003 205

4075 -3.700000000000003 209

4080 -3.600000000000003 209

4090 -3.500000000000003 209

4100 -3.400000000000003 212

4110 -3.300000000000003 212

4120 -3.200000000000003 212

4125 -3.200000000000003 216

4130 -3.100000000000003 216

4140 -3.000000000000003 216

4150 -2.900000000000003 218

4160 -2.800000000000002 218

4170 -2.700000000000002 218

4175 -2.700000000000002 222

4180 -2.600000000000002 222

4190 -2.500000000000002 222

4200 -2.400000000000002 224

4210 -2.300000000000002 224

4220 -2.200000000000002 224

4225 -2.200000000000002 228

4230 -2.100000000000002 228

4240 -2.000000000000002 228

4250 -1.900000000000002 231

4260 -1.800000000000002 231

4270 -1.700000000000002 231

4275 -1.700000000000002 235

4280 -1.600000000000001 235

4290 -1.500000000000001 235

4300 -1.400000000000001 237

4310 -1.300000000000001 237

4320 -1.200000000000001 237

4325 -1.200000000000001 241

4330 -1.100000000000001 241

4340 -1.000000000000001 241

4350 -0.9000000000000009 244

4360 -0.8000000000000009 244

4370 -0.700000000000001 244

4375 -0.700000000000001 248

4380 -0.600000000000001 248

4390 -0.500000000000001 248

4400 -0.400000000000001 250

4410 -0.300000000000001 250

4420 -0.200000000000001 250

4425 -0.200000000000001 254

4430 -0.100000000000001 254

4440 -1.02695629777827e-15 254

Simulation stopped via $stop(1) at time 10 US + 0

ncsim> exit

...Regained control from SimVision

-------------------------------------

TOOL: irun(64) 15.20-s069: Exiting on Jan 13, 2021 at 14:45:03 IST (total: 00:03:30)