

## EE-413 VLSI TERM PROJECT

In this project, Custom Signal Manipulator is asked to design. The aim was according to inputs analog data (triangular wave) is intended to manipulate like in Project Manual. However, due to time limit, lack of enough network connection and delay of Cadence, I could only design some parts, serial to parallel converter, S/H circuit, comparator, DAC, operation register and controlling the start bit block. The whole circuit functions by taking triangular analog input and determining the operation instruction with digital serial input. All of the operations and functions of subblocks are decided according to that operation instruction. Moreover, the whole system has 4 inputs, which are analog triangular input, RESET, CLK and digital serial input. Also, according to these inputs, the system gives one analog output. Digital Serial Input can be considered as 3-bit encoded start signal and 3-bit instruction signal.

Figure 1 represents the whole system. This system consists of 8-Bit SAR ADC, serial to parallel converter, operation register, controlling bits subblock, ALU, 8X8 Bit Register and DAC. Also, 8- Bit SAR ADC includes S/H Circuit, comparator, SAR and DAC subblocks. In the system, I used 6 bit shift register as serial to parallel converter. Also, instead of control bits decoder I used one INV and one AND gate for controlling the start bits. Then, I connected output of this subblock to operation register as ‘Enable’ input of that with the select opcode. when start bits are correct, system started to function and operation instruction in the operation register is sent to ALU. Furthermore, I used some necessary circuit topologies in S/H Circuit, comparator and DAC subblocks. I used both p driver and n driver two stage differential amplifier for comparator. Moreover, I used R-2R ladder for DAC.

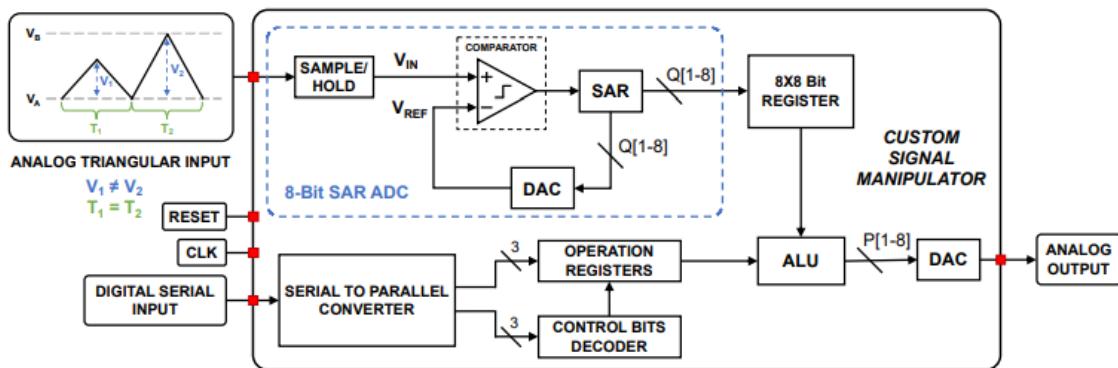


Figure 1 : Custom Signal Manipulator System Schematic

### Serial to Parallel Converter

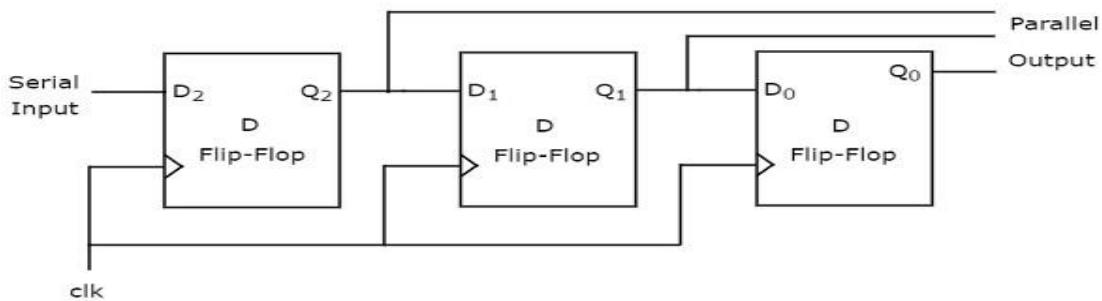


Figure 2 : Schematic of theory of converter

While designing this subblock, I used 6 DFFs as shown in Figure 2. When the digital input is entered to this subblock, digital input bits shifts to right with DFFs so that digital serial input is converted to parallel data by taking outputs from outputs of each DFFs. It should be noted that serial input should be in harmony with clock frequency. Otherwise data will be lost and operation might be wrong or even no operation will occur.

I adjusted the CLK frequency 50 MHz (20 ns period) as asked and did the simulations with different digital serial input bits.

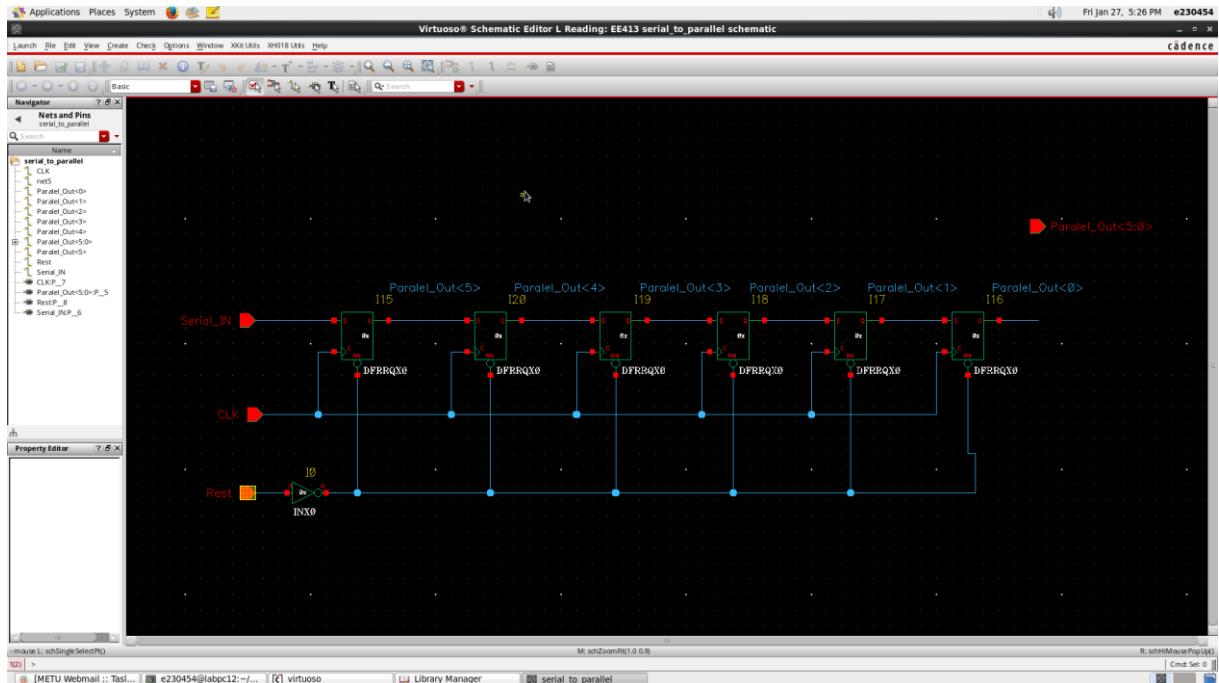


Figure 3 : Serial to Parallel Converter

This subblock converts the serial input to output in 6 clock cycles.



Figure 4 : Output of serial to parallel converter

## Control Decoder

This block checks whether operation is enabler or not. It is a straightforward block so I wont share the test setup. Since opcode is 101 one inverter with 3 input and will give us the begin command.

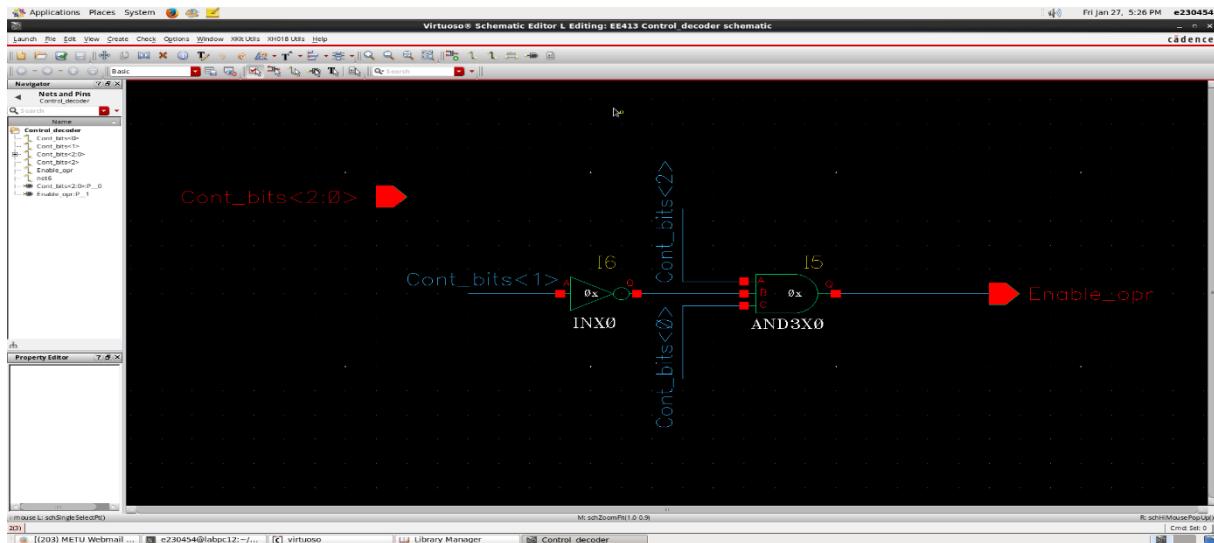


Figure 5 : Control decoder schematic

## ALU

This block will manipulate the signal coming from register. It will do 8 operations according to the operation code.

### All Zero and one Mode

This block designed as one and according to the select input operation will take into place as all zero and one. As we look in time domain the last operation code namely opcode<5> for my operations will determine. If opcode<5> is 1 and start code as 101 s given then output will be all ones vice versa all bits will be zero.

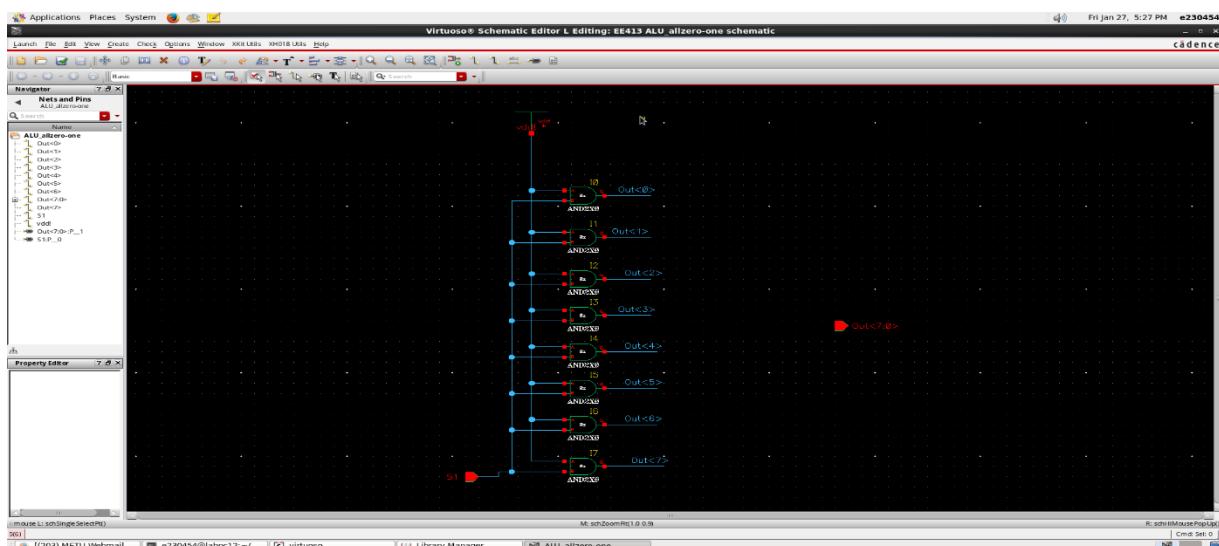


Figure 6 : All zero one mode ALU schematic

## Multiply and Divide Operation

I designed this block as one unit as well. As we know multiplying by two means logical left shift and dividing is logical shift right. Hence as a select input with 2X1 MUX circuits I designed a multiplier or divider circuitry.

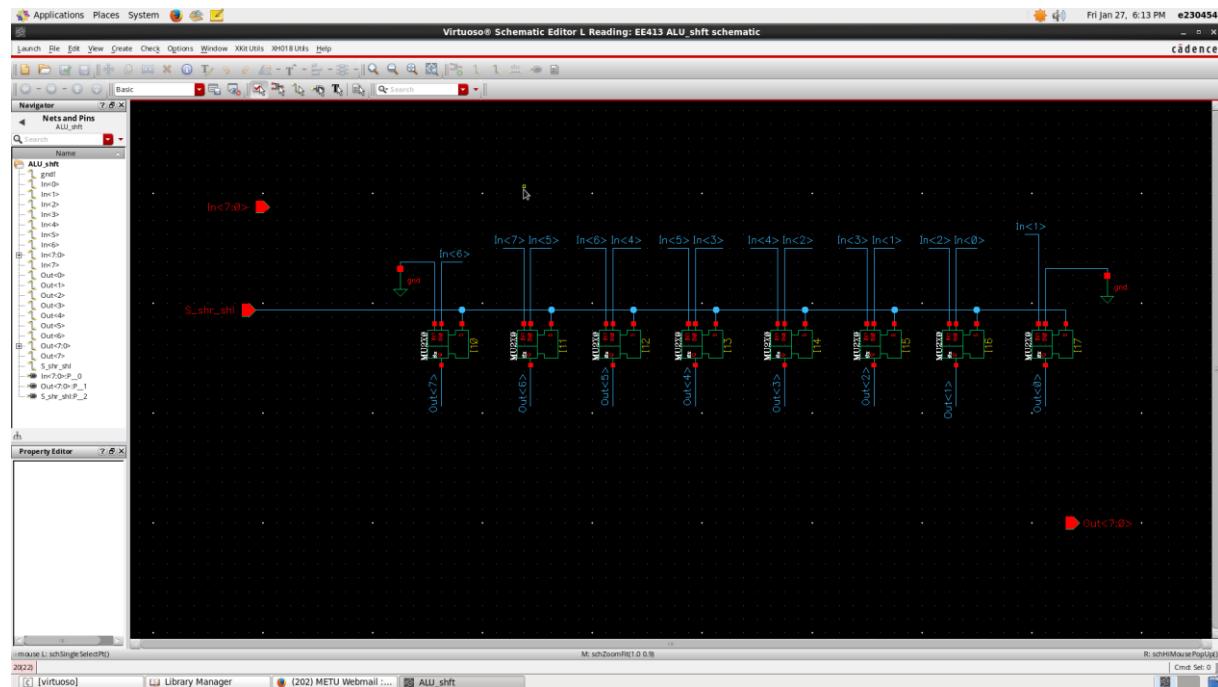


Figure 7 : Shift circuitry



Figure 8 : TB results of divide and multiply

As you see when the select is 0 B input is shifted left while when select is 1 B is shifted right.

## Buffer

Buffer operation is also a straightforward action when this command comes signal will directly given to the DAC at the end. Only MUXEs at the end will select this function and no subblock is utilized for this purpose.

## Offset Addition and Removal

Also this one is pretty straight forward. At the phase one I thought I should use a adder subtractor circuitry for this purpose , I understand that we use 8 bit not signed demonstration of a number. Hence subtracting will be impossible and even if I use signed bit or 2's complement this will result in complication and also lower resolution. And I know that offset is at least 500 mV also input is always less than VDD/2 so by just manipulating the MSB of the input I can add 900 mV offset. And also can remove that by pulling MSB down to ground.

### SSSS



Figure 9 : Schematic of Offset Addition and Removal



Figure 10 : Output of Offset Addition and Removal

As we see we can manipulate MSB by a select input. It should be noted that opcode<5> is the only select signal in my design I did this on purpose to reduce complexity.

### Input Flipping

This one is also pretty straightforward. Signal can be flipped by negating each bit parallelly. That was what I did.



Figure 11 : Flip Schematic



Figure 12 : Flipped Output

We can see that we accomplished our goal.

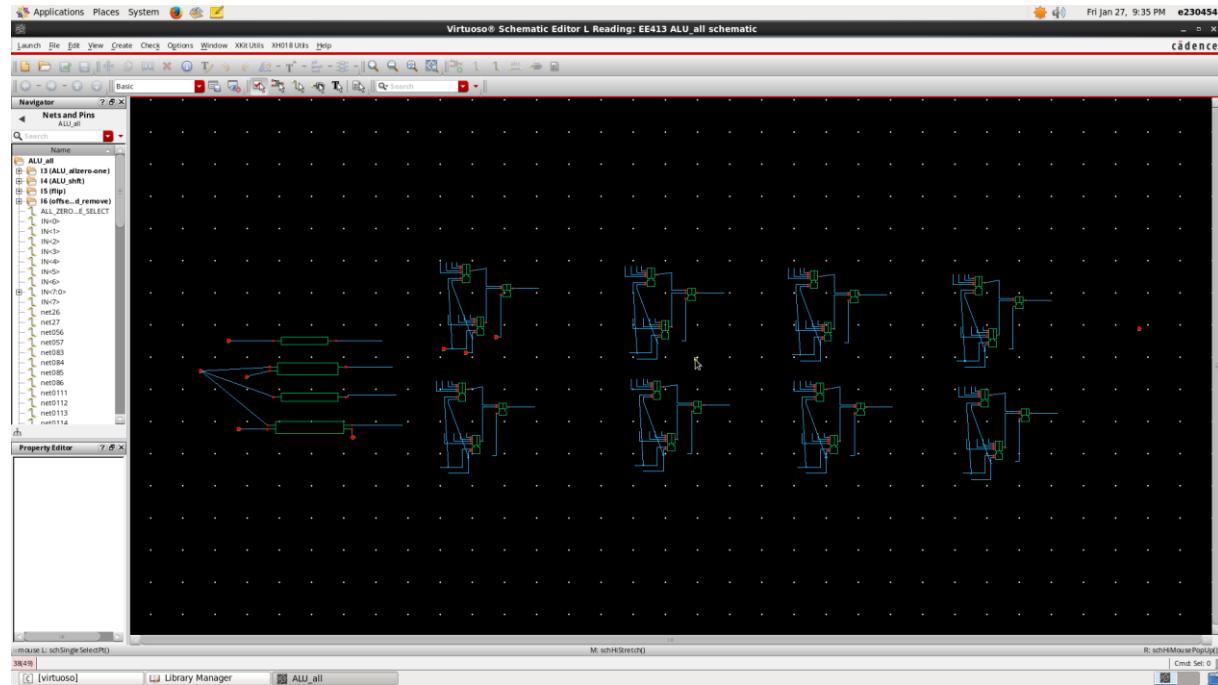


Figure 13 : Overall ALU BLOCK

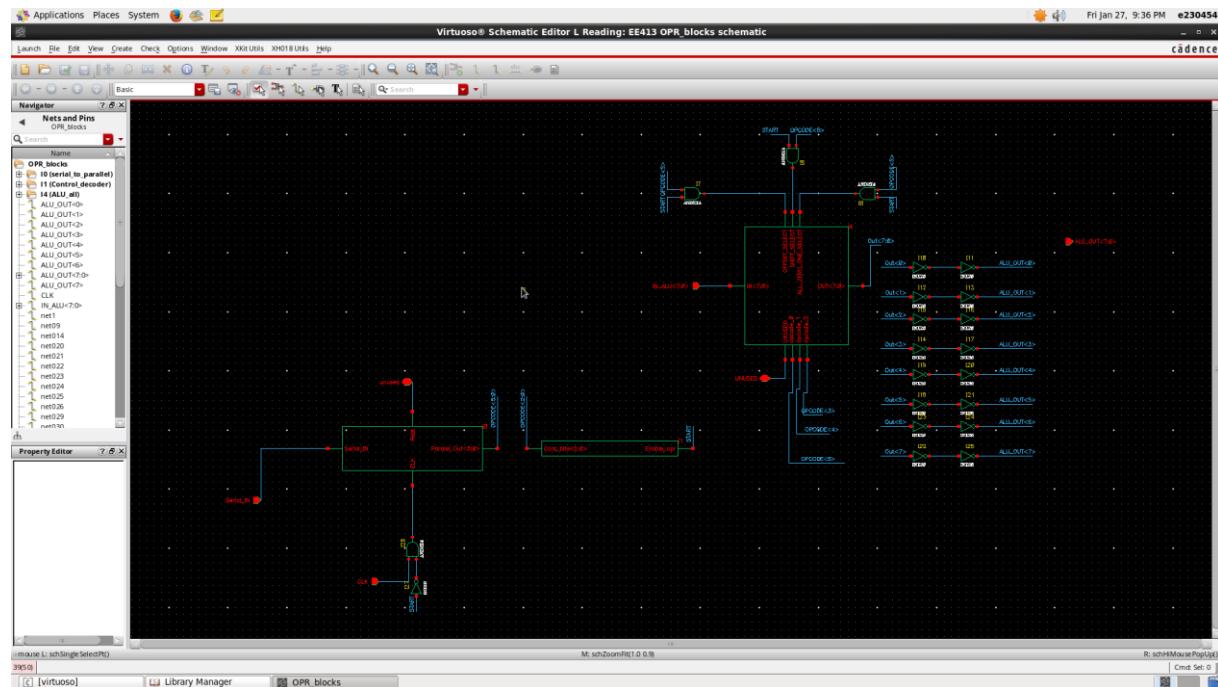


Figure 14 : ALU with Enable Block

## ADC

### Sample Hold Unit

I created a transmission gate by using CLK and CLK'. By controlling the CLK of this unit I sample and hold it for 8 cycle.

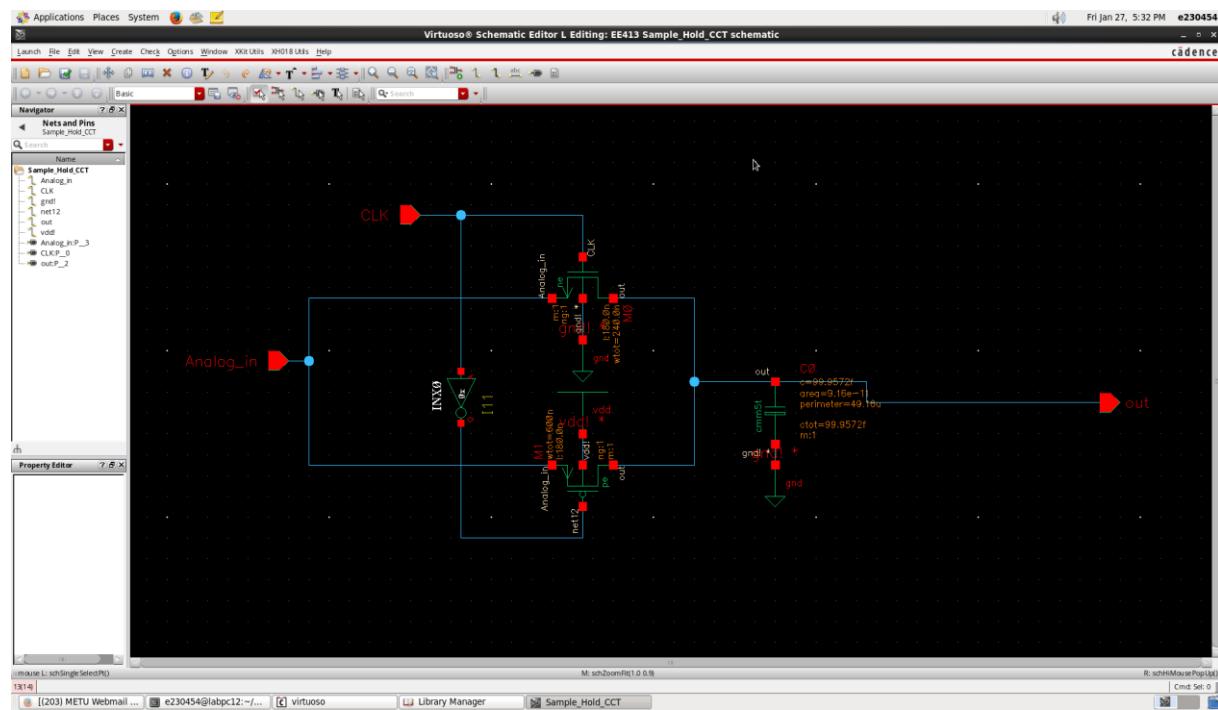


Figure 15 : Sample Hold Unit

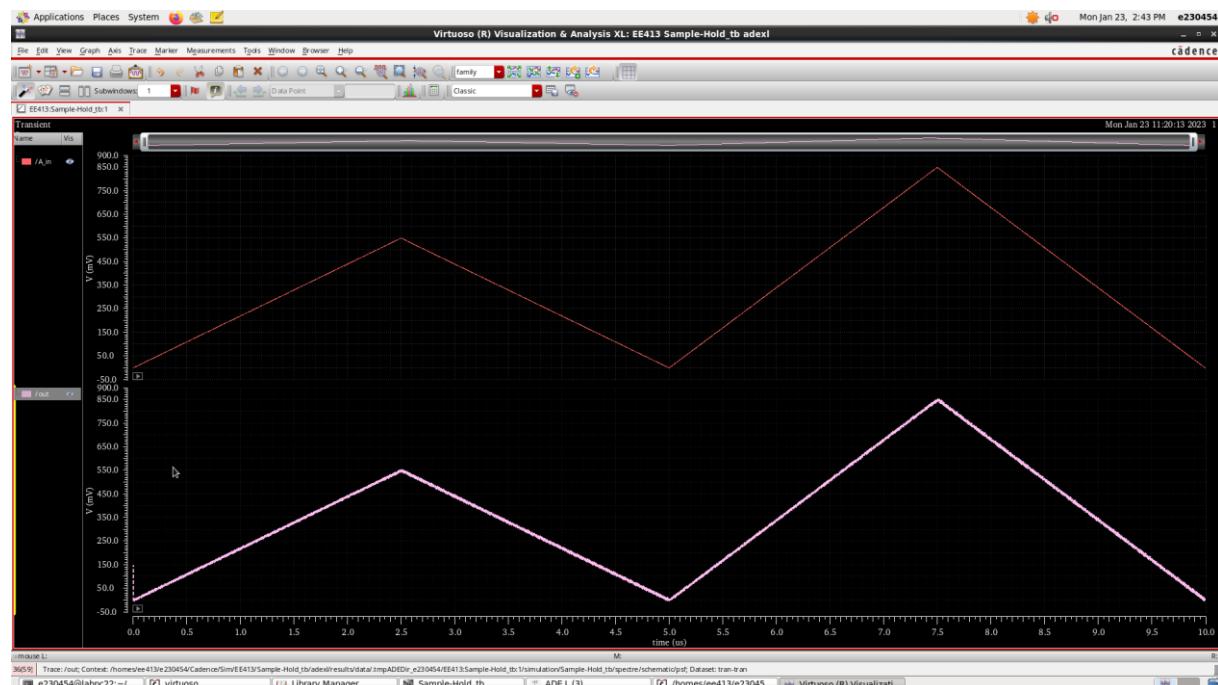


Figure 16 : Sample Hold Unit output

## Comparator

I designed two comparator. One of them is p type MOSFET driver and the other one is n type MOSFET driver comparator. To increase the slew rate output MOSFET's had a W/L ratio of approximately 2 and L is minimum. Also by using current mirror with 10 gain I used high power in an exchange of lower delays.

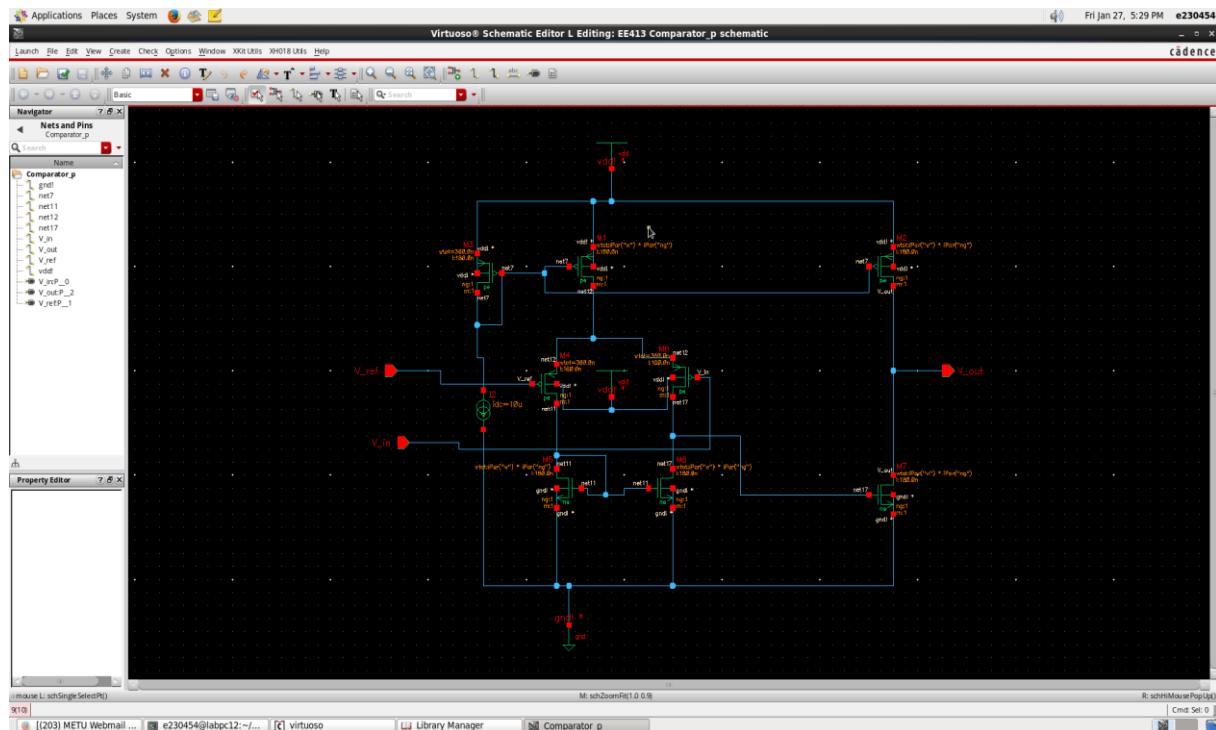


Figure 17 : P type driver Comparator

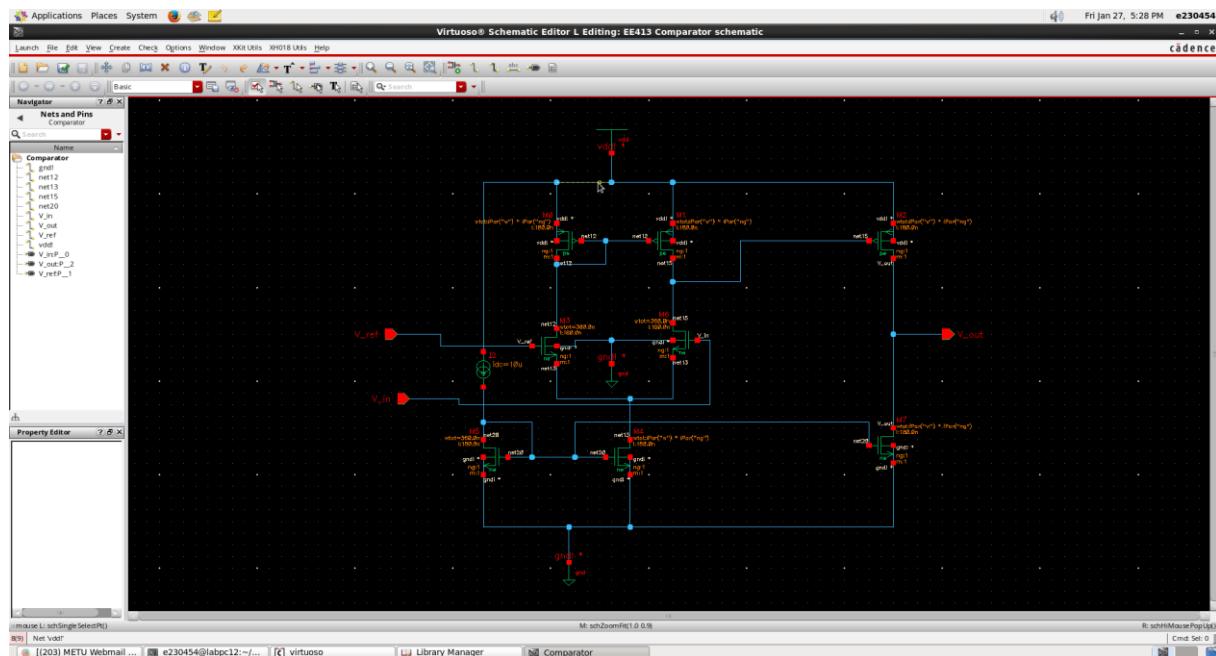


Figure 18 : N type driver Comparator

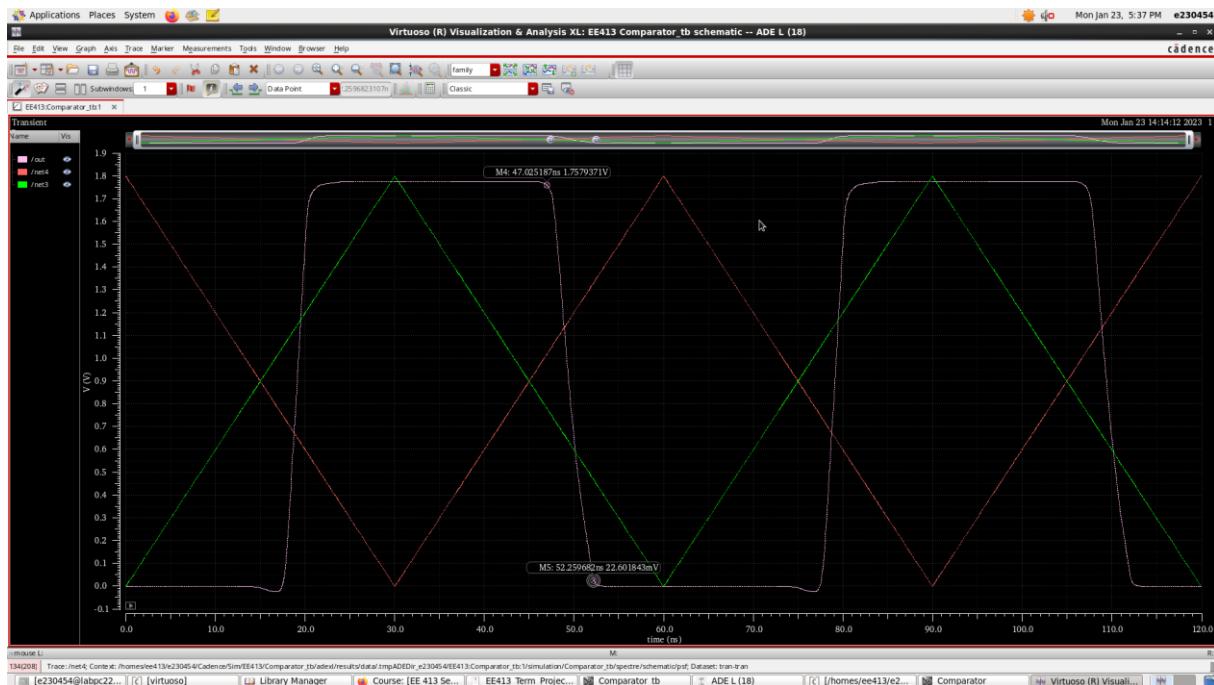


Figure 19 : Comparator result

By using a OR gate at the output we get the overall comparator output.

### DAC

Initially due to power consumption I decided to use C-2C logic however in design procedure I faced with errors that I couldn't solve so I changed my topology to R-2R design.

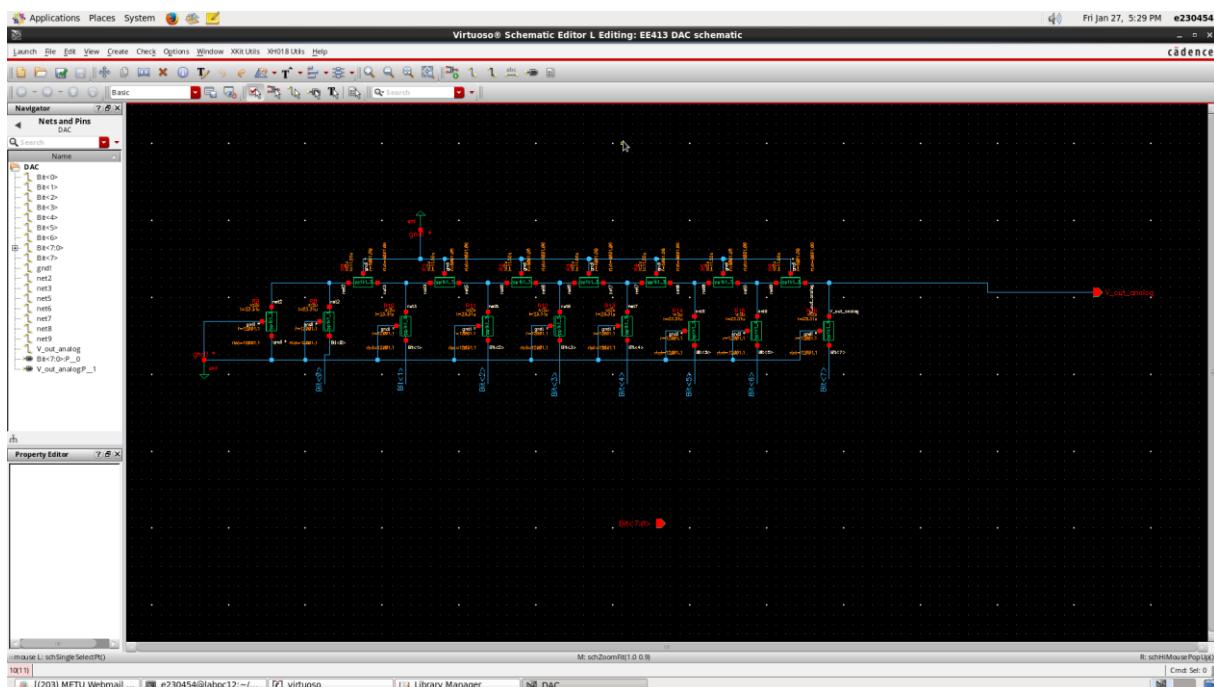


Figure 20 : R-2R DAC schematic

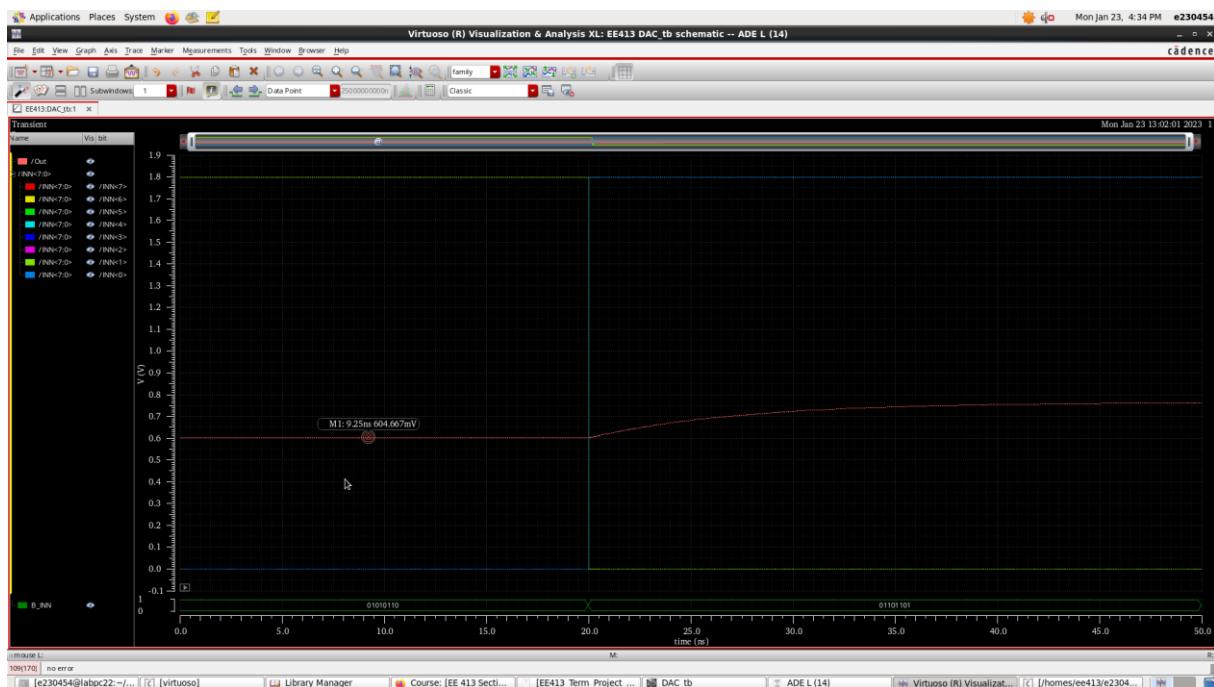


Figure 21 : Output of the DAC

I gave an input as 01010110 and get the output as 604 mV. It is expected due to the rule states

$$\text{Analog Out} = 1.8 * (2^{-1} * \text{MSB} + 2^{-2} * \text{BIT}(N-1) + \dots + 2^{-N-1} * \text{BIT}(0))$$

That equation is verified.

#### SAR

This block is responsible for the determining the right conversion of the analog signal to the digital signal. This is done by looking at the result of the comparison in the previous stage. At the first cycle MSB is set to 1 so the overall register is '10000000' and compared with the sampler output. If the result of the comparison is less than 0 then MSB set to 0 and vice versa. Hence this goes on and on for 8 bit of 9 cycles to generate an output to be written on 8x1 bit register. Since this process goes on for 9 cycles sampling process should be 9 times slower than this.

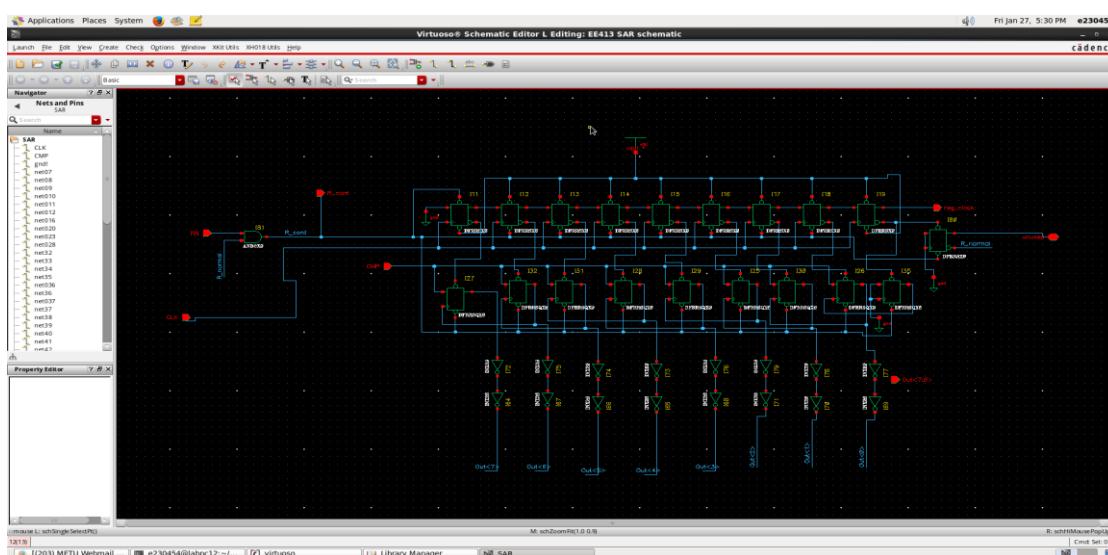


Figure 22 : SAR Block Schematic

SAR block begins with reset this initializes the MASTER part and according to the comparator result Slave part holds the result or pulls it down. At the end of D-FF's there is a control output that suggest all conversion is done so now reset it and begin another cycle. Also I added a buffer at the output of the digital point this one is done because of the SAR pull up is not enough. A bit is one when the center bus is 0,9 V but the analog output becomes 1,2 V which is not desired so I pulled it high by adding 2 inverter.

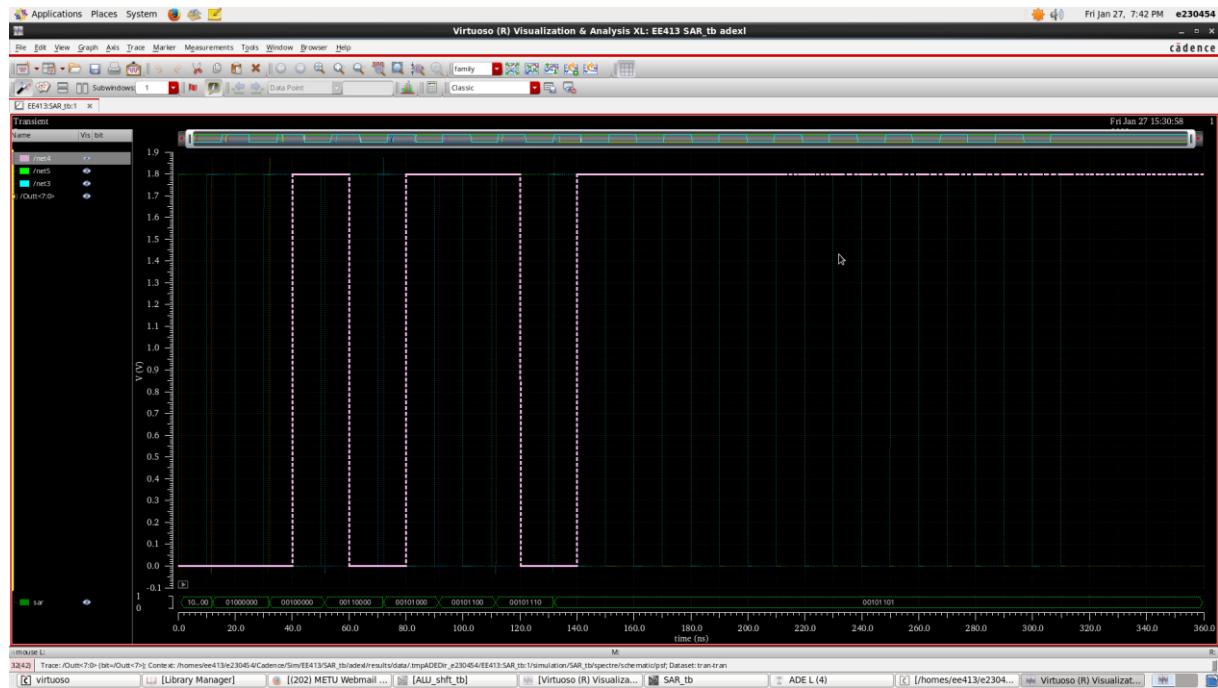


Figure 23 : SAR result

For just SAR without reset input you can see that for input 00101101 it gives the exact correct result. When we implement in overall system it will be resetted so that system starts to work again.

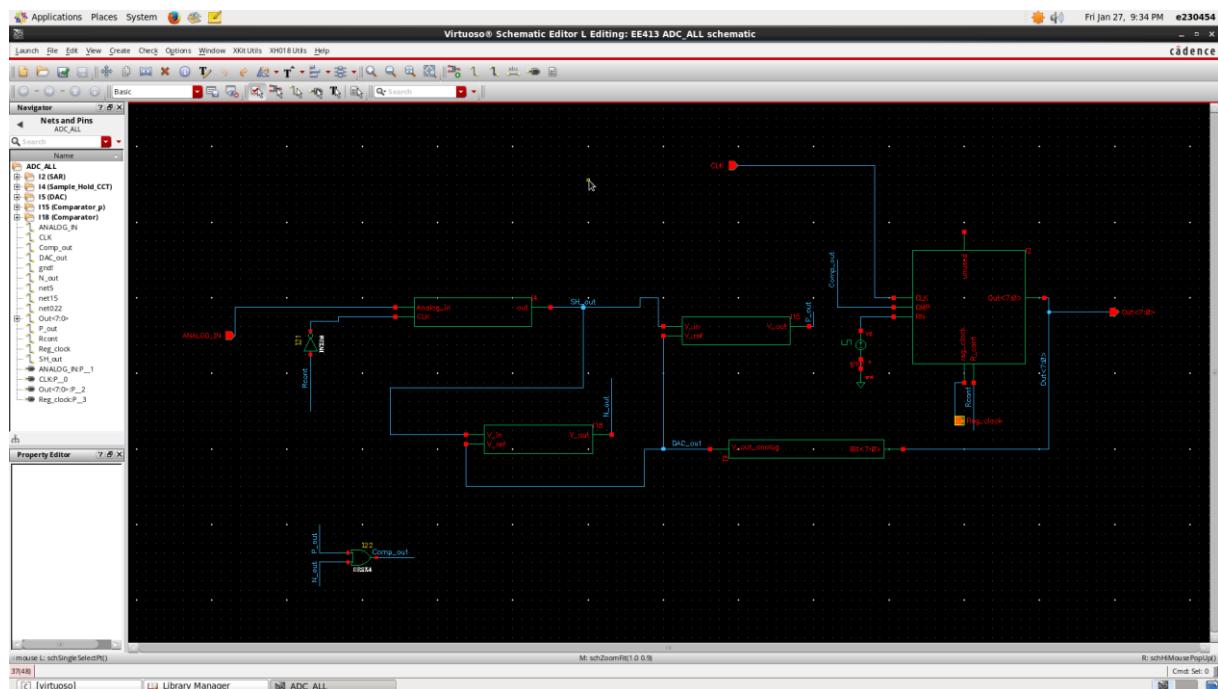


Figure 24 : Overall ADC structure

## 8 bit Register

Normally I have planned to design a 8x8 bit register. However as I more think about it I believe that 8x1 bit register is enough for us. However this is done with a price of resolution decrease. This is something we can accept because if we would use 8\*8 bit it will consume more power and also to increase the resolution we should use each 8 bit part with a phase difference. For example if our first 8 bit starts with 0 phase shift clock second one should have 45 degree third one is 90 degree and goes on like that until reset. That also increase the complexity.

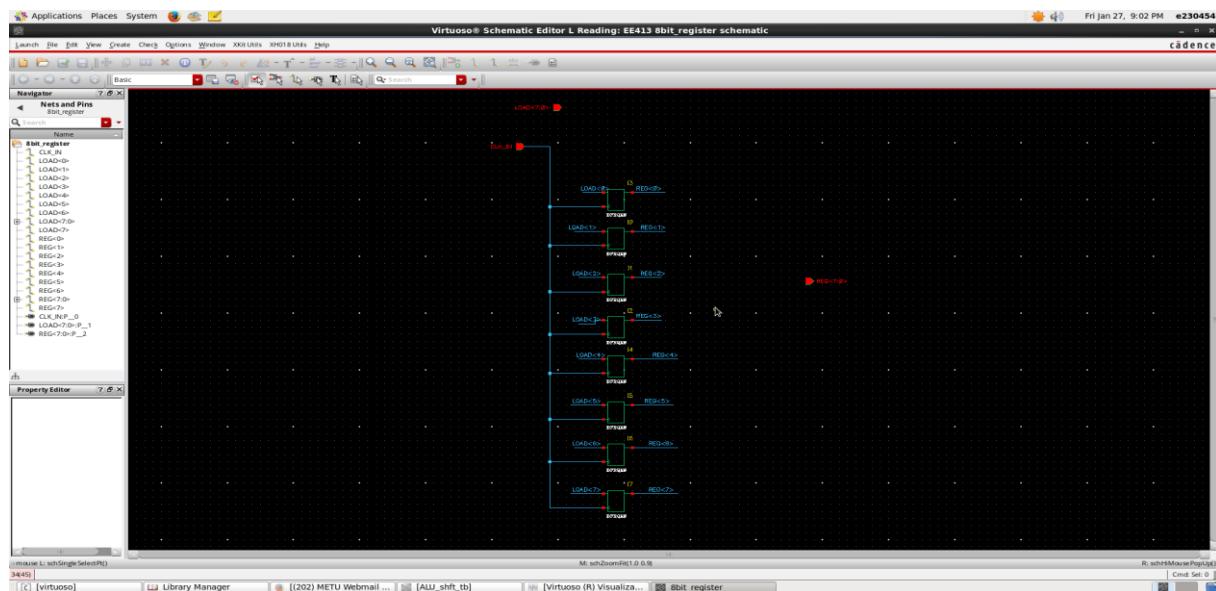


Figure 25 : 8 bit register

It has a trivial output as we expect because it is a parallel load 8 bit register. The clock of D flip flops are controlled with SAR DFF last bit in time domain. By this way when ever SAR has finished its job register passes the new value to the ALU.

## Overall System Outputs

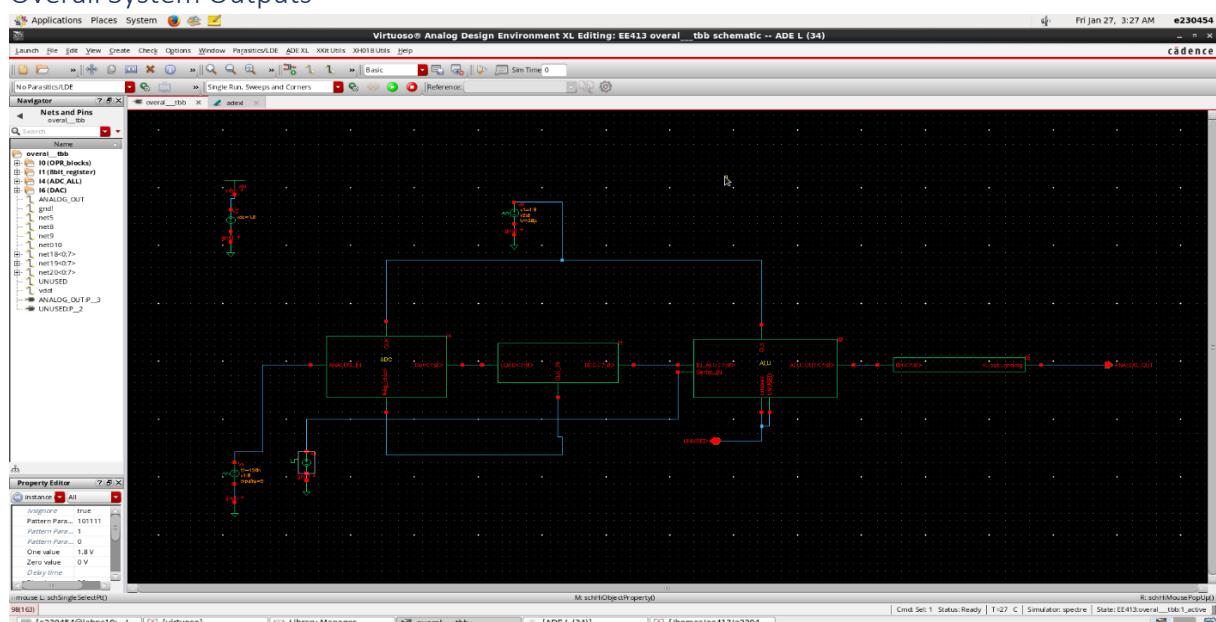


Figure 26 : Overall System Setup

In the above figure we can see over all test setup. As we mentioned earlier SAR will control the Register CLK while SAR reset will control the sample hold circuitry and by this way we will sample once and hold for 8 cycles.

In the below figures you will see every operation result with respect to their opcode ;

### OPCODE: 000

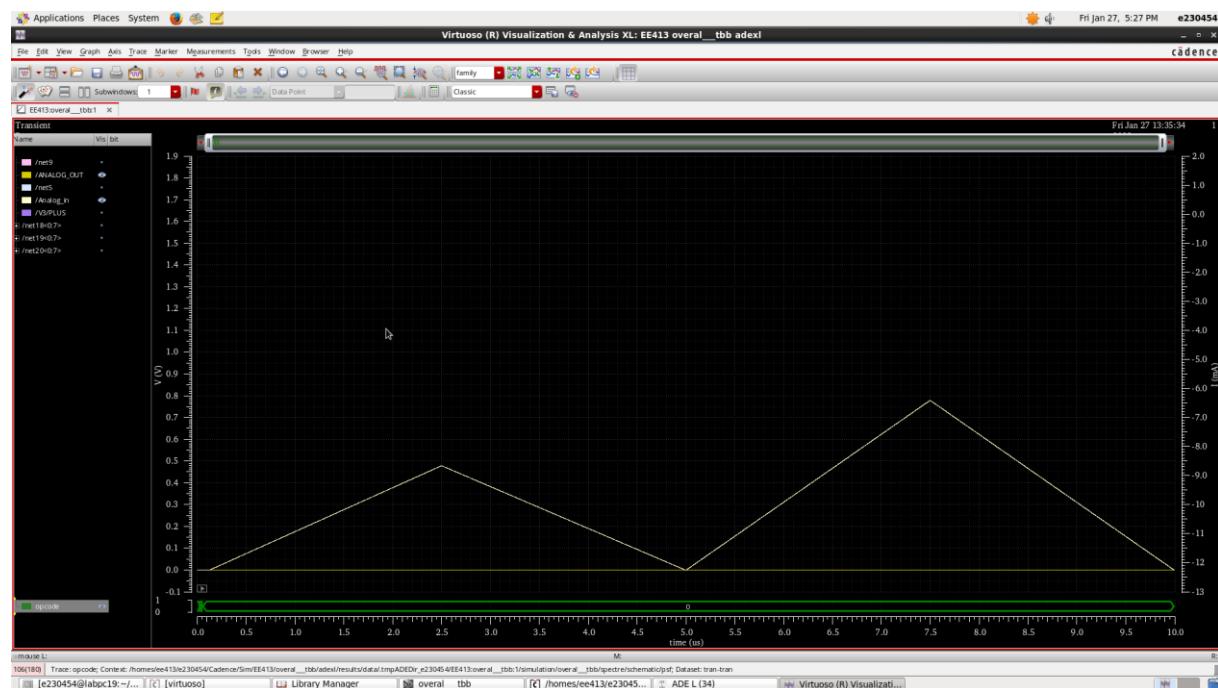


Figure 27 : All Zero Output

### OPCODE : 001

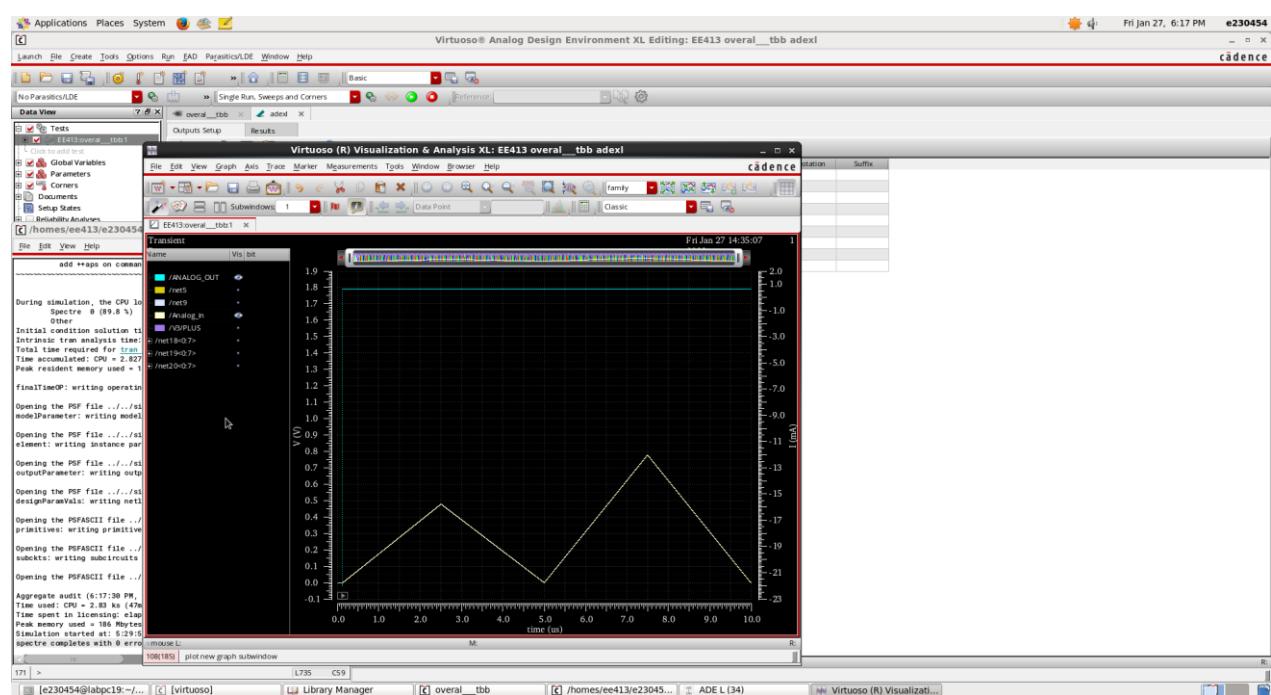


Figure 28 : ALL One output

## OPCODE 010

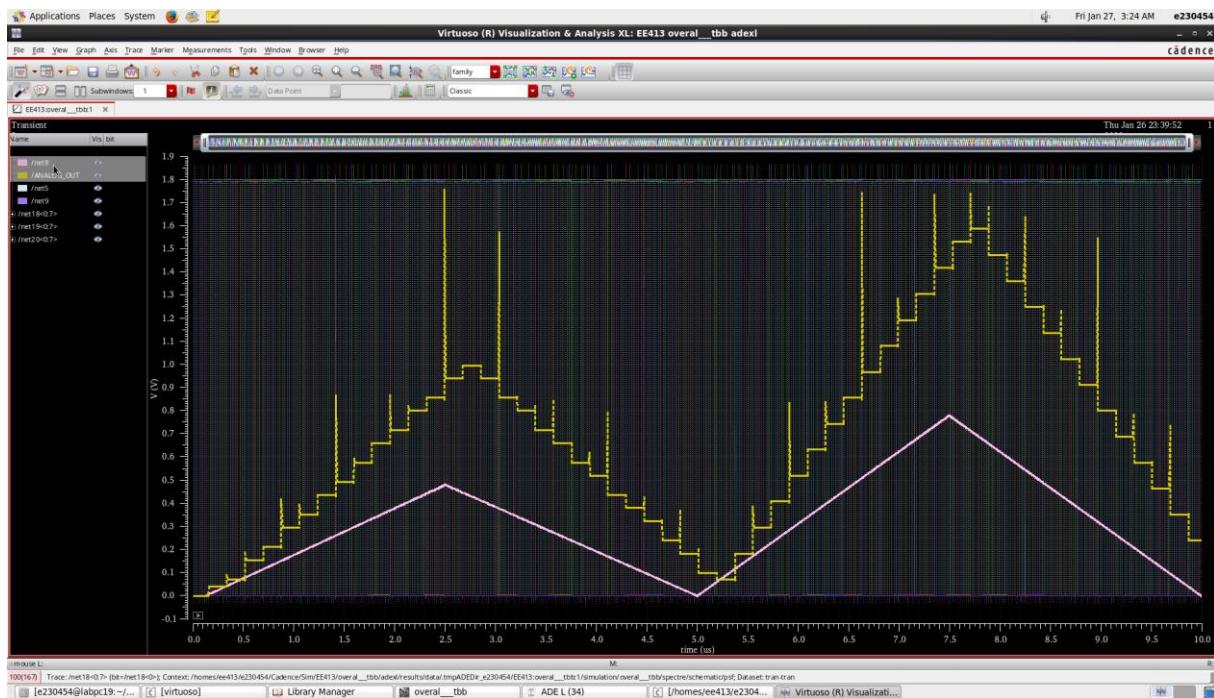


Figure 29 : Multiply by Two Output

## OPCODE : 011

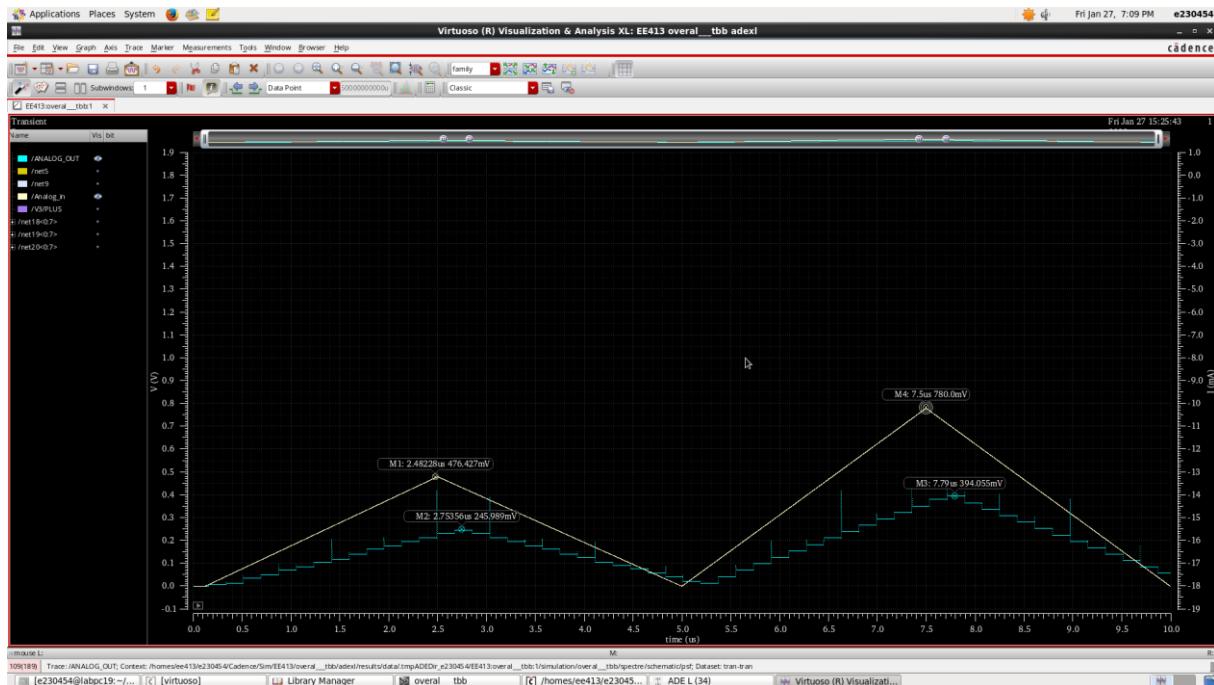


Figure 30 : Divide by Two Output

### OPCODE : 100

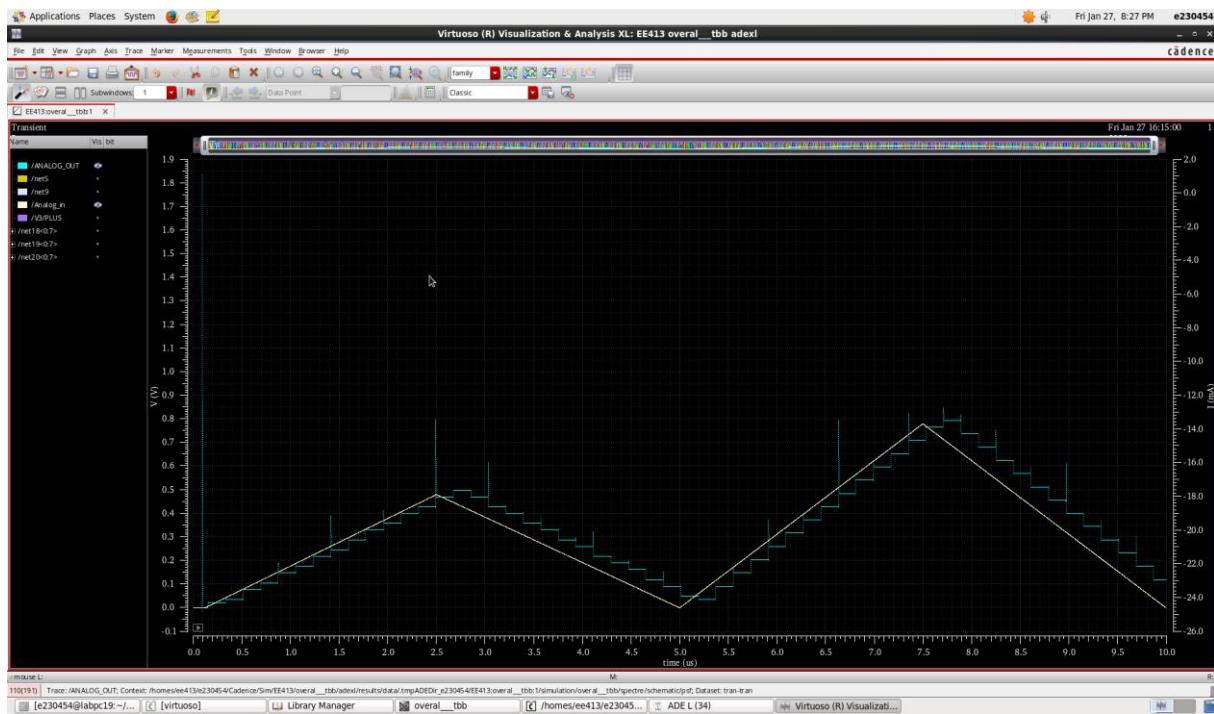


Figure 31 : Buffer Operation Output

### OPCODE : 101

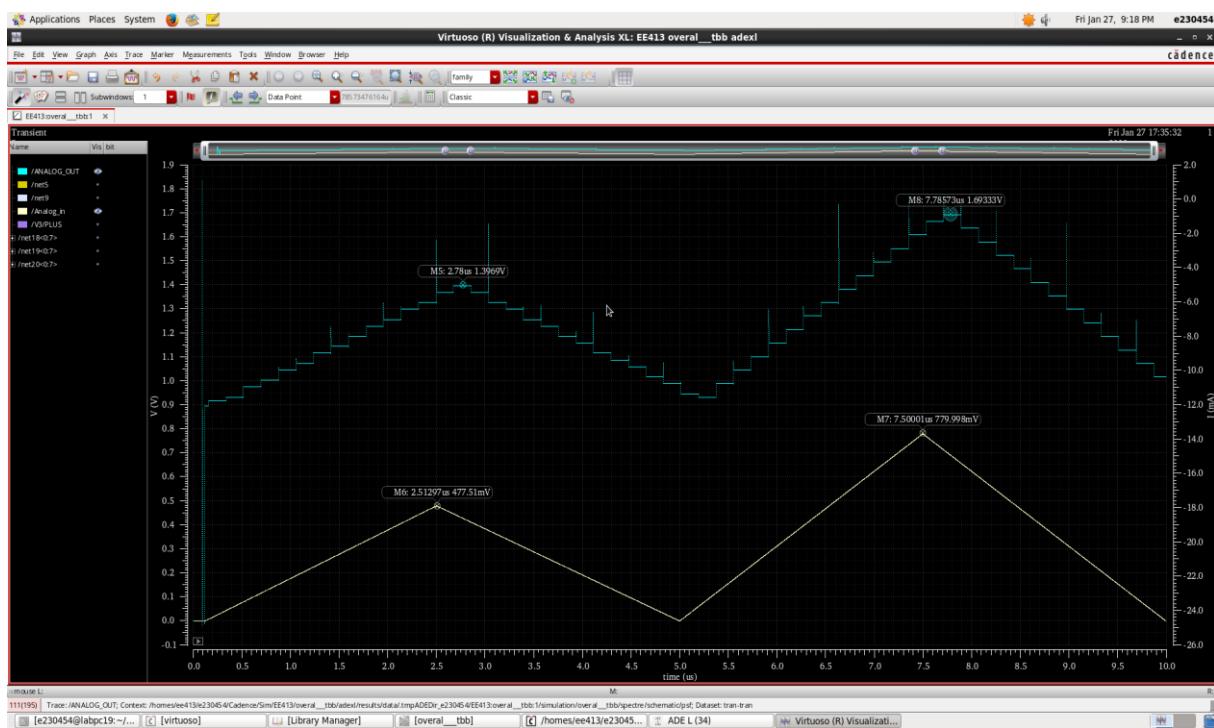


Figure 32 : OFFSET addition of 900 mV

### OPCODE : 110

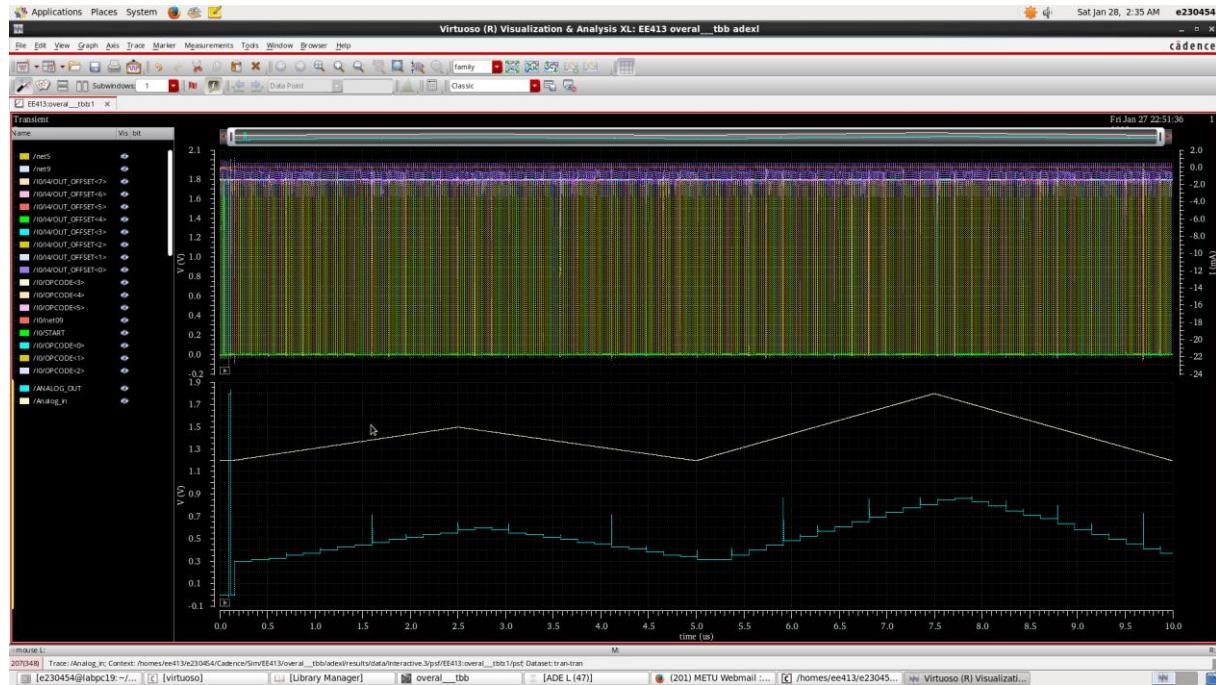


Figure 33 : Offset Remove Operation

### OPCODE : 111

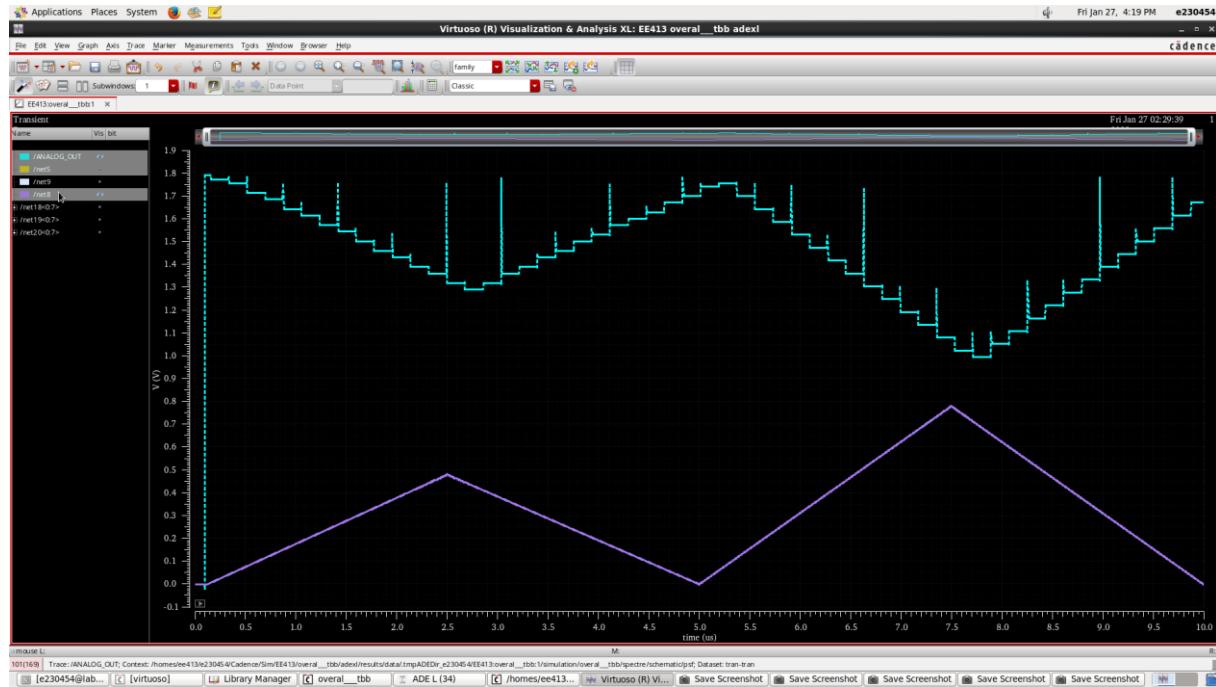


Figure 34 : Flip Operation Output

### Power Consumption

I didn't investigate each block power consumption however I examined average power consumption of the system for each operation.

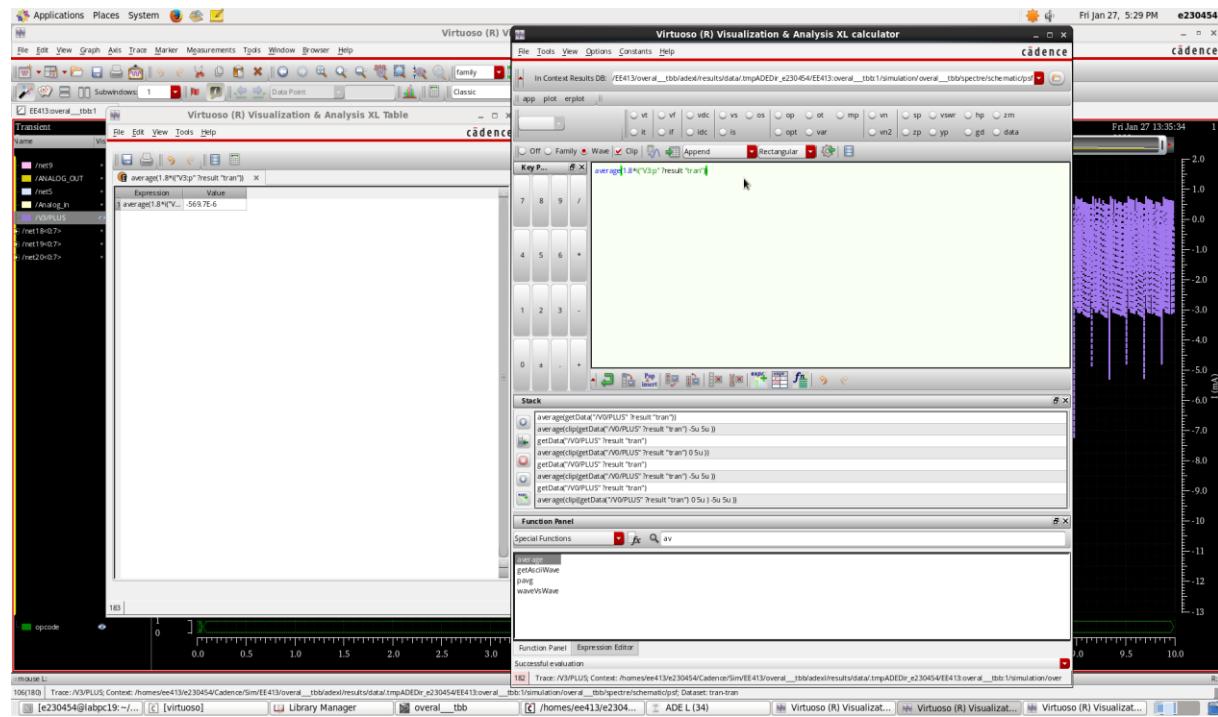


Figure 35 : All zero output power consumption

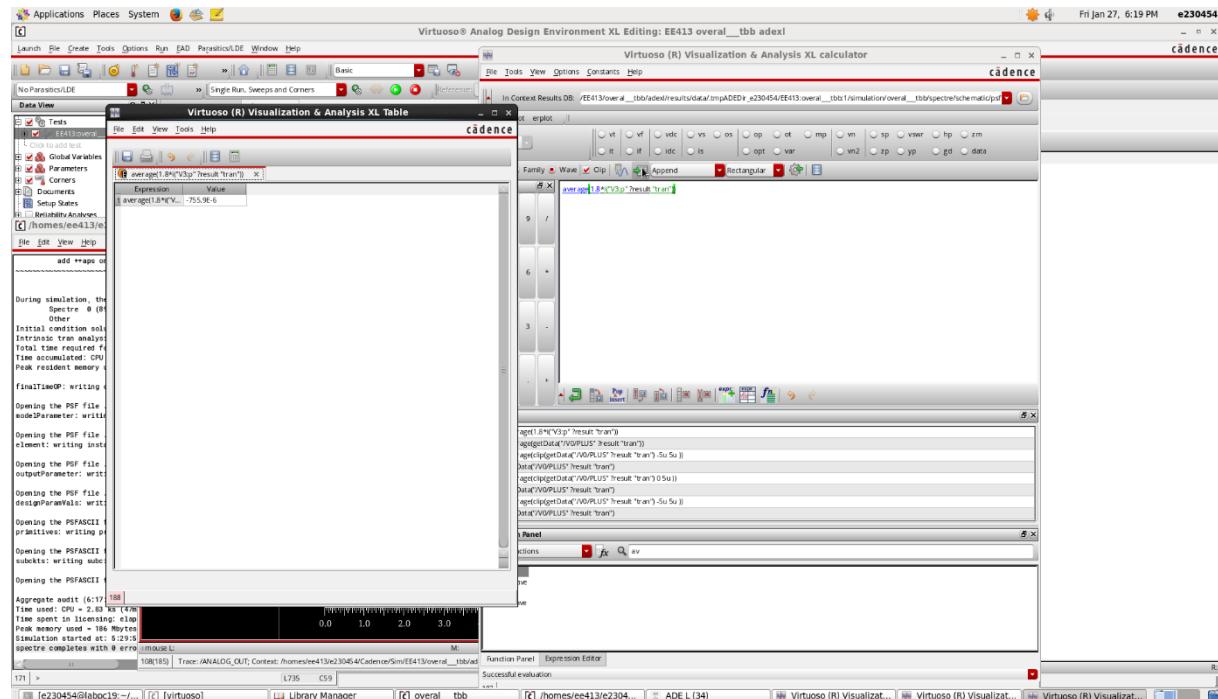


Figure 36 : All one output average power

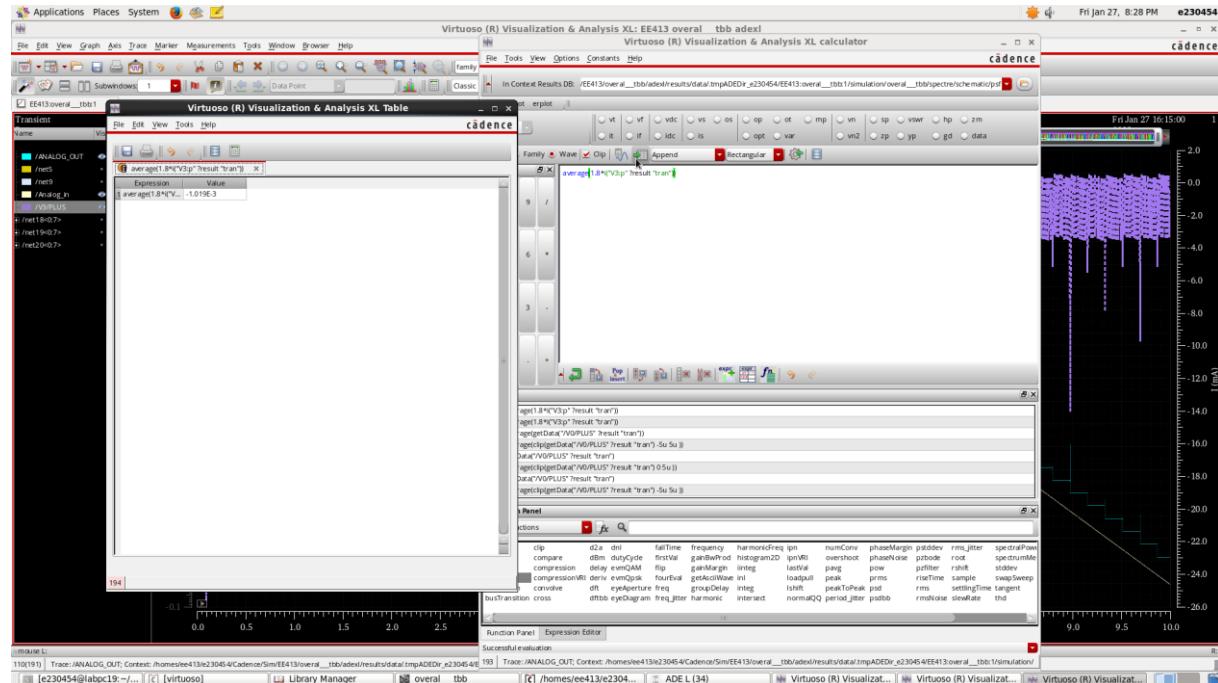


Figure 37 : Multiply/Divide Power Output

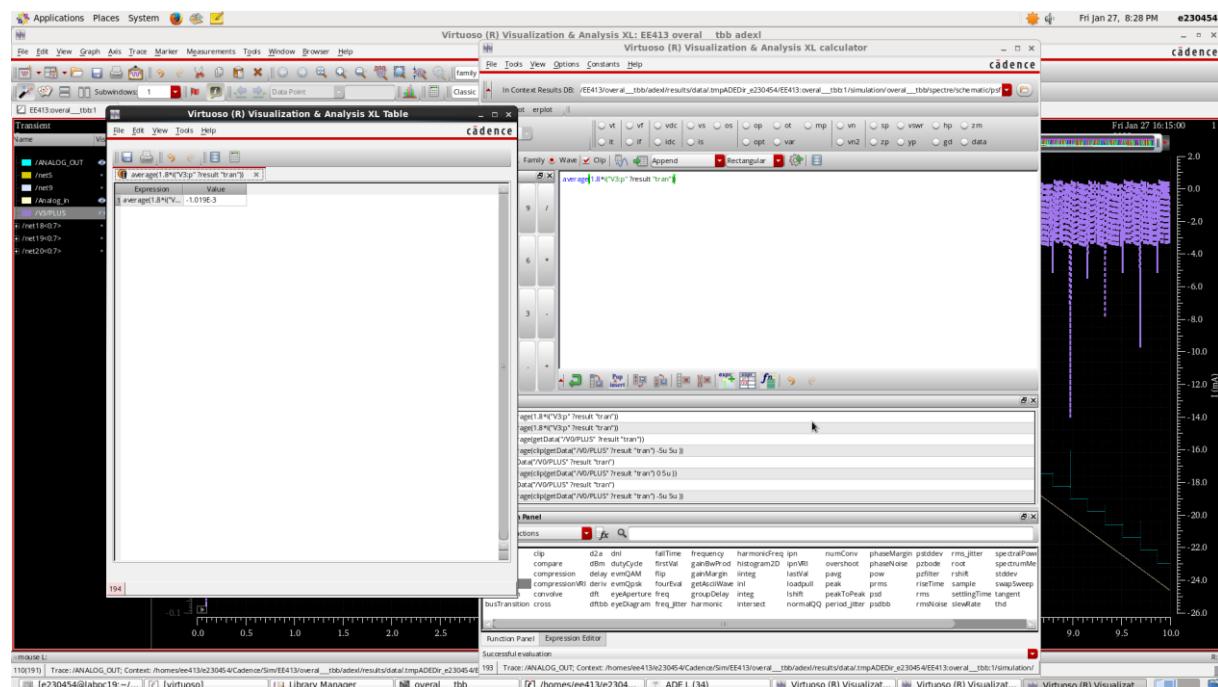


Figure 38 : Buffer Power Output

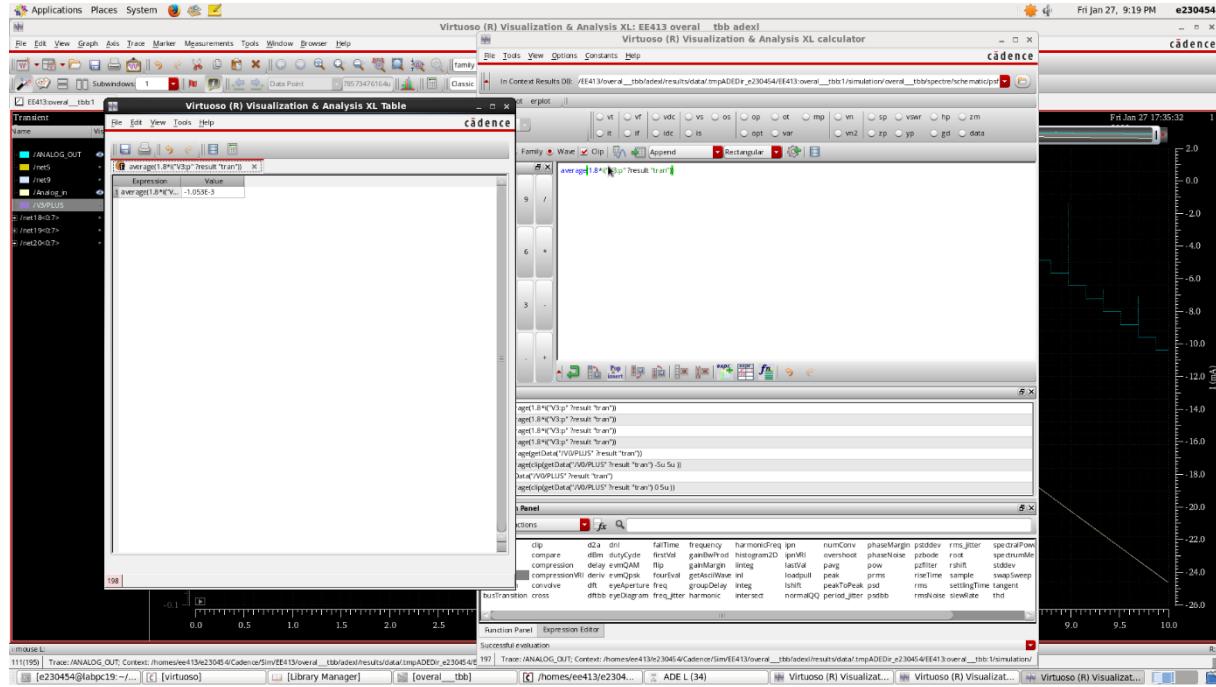


Figure 39 : Offset Addition Output

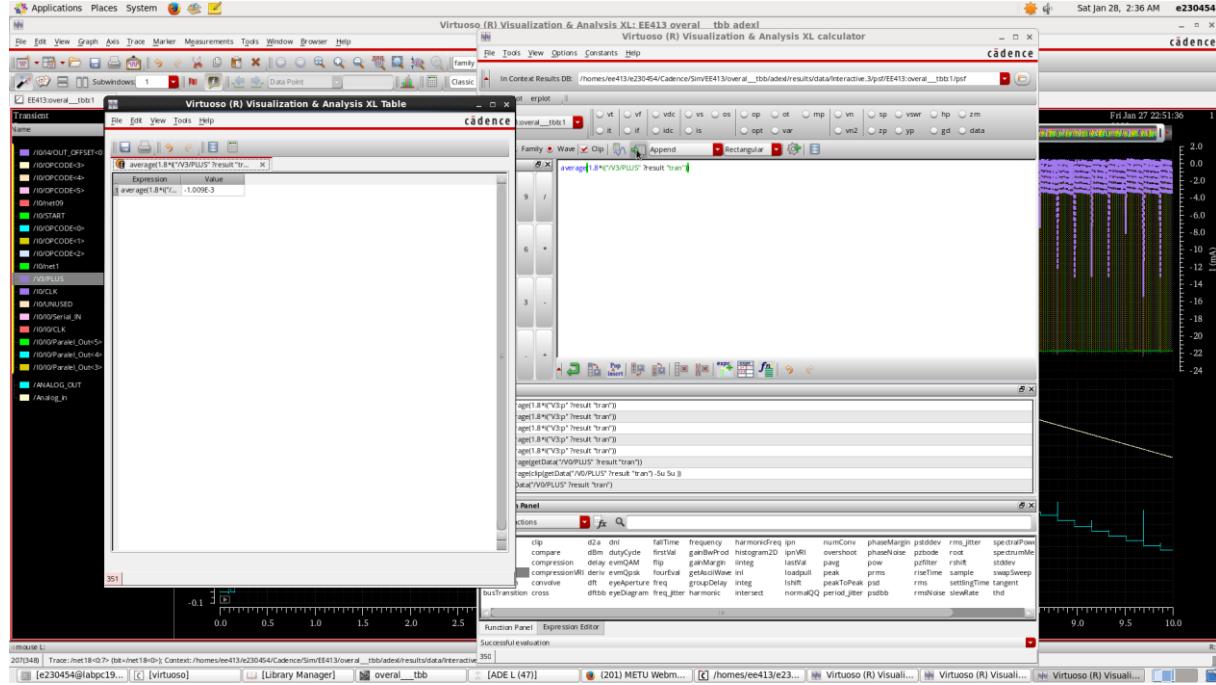


Figure 40 : Offset Remove operation Power

As we can see from the above figures highest occurs in offset addition. This was a little surprising for me since in offset addition we only manipulate one bit. And other bits had not so much effect

however in this operation I used largest possible gate. This increases the current drawn which results in high power consumption.

### Conclusion

The goal of the project is manipulating the analog signal, which is triangular wave, based on the digital serial input, which consists of start bits (101) and instruction data indicated in the manual. In the project, I learned how to deal with design of a circuit and how to approach to when error occurs as a candidate engineer. By researching and analyzing the information in the Internet and my background knowledge, I could design S/H circuit, comparator, SAR circuit, DAC, serial to parallel converter, controlling start bits and operation register. I got correct results in their waveforms. I demonstrated them for two different input examples.

### References :

- [1] <https://www.irjet.net/archives/V4/i7/IRJET-V4I7657.pdf>
- [2] <https://www.electronics-tutorials.ws/combination/r-2r-dac.html>
- [3] <https://www.allaboutcircuits.com/textbook/digital/chpt-12/parallel-in-serial-out-shift-register/>
- [4] [http://www.ijirset.com/upload/2018/may/152\\_An.pdf](http://www.ijirset.com/upload/2018/may/152_An.pdf)
- [5] [https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8314&context=etd\\_theses](https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8314&context=etd_theses)

Yiğit Utku Er 2304541

Yiğit Utku Er 2304541