# Exploring Different DAC Designs Exhibiting Logarithmic Behaviour for In Memory Computing

*A Project Report*

*submitted by*

**LAKSHYA NARAYAN PALLI**

*in partial fulfilment of the requirements*
*for the award of the degree of*

**BACHELOR OF TECHNOLOGY &**
**MASTER OF TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.**
**May 2019**

# THESIS CERTIFICATE

This is to certify that the thesis entitled **Exploring Different DAC Designs Exhibiting Logarithmic Behaviour for In Memory Computing**, submitted by **Lakshya Narayan Palli** (**EE14B092**), to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelors of Technology** and **Master of Technology**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Dr. Janakiraman Viraraghavan**
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 9th May 2019

# ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude to my project guide Dr.Janakiraman Viraraghavan for his valuable guidance and support throughout this project, for giving me an opportunity to work under him and providing his valuable time.

My colleagues for providing valuable insights, Lab mates and seniors for acting as a source of inspiration, special thanks to Praveen MV for guiding me in the right direction and helping with all the doubts and difficulties. I express my gratitude to my batchmates for their feedback and advise

Finally, I would like to thank my parents for their constant support right through my journey of becoming an engineer and My teacher for providing me with a plethora of knowledge.

I would like to thank everyone who helped me in one way or the other.

# ABSTRACT

KEYWORDS: Digital to Analog Converter with Logarithmic behaviour, MOS approch, String DAC,layered Resistor approach, TSMC 28nm tech

In Memory Computing, helps in mitigating one of the major bottlenecks in neural computing, reducing idle access time. In this project, we work on the implementation of MNIST data set on memory, (784 columns and layer dependent rows) For low power constraints we'll be working in the subthreshold region of the NMOS (Access transistor), and will try to eliminate the non-linear behavior of current, by using a logarithmic voltage at the gate, which will eventually cancel out the exponential current behaviour

The Project revolves around exploring different Digital to Analog Converter (DAC) designs with logarithmic output ( providing input to the gate of access transistor), since the DAC acts a localized component to each SRAM cell it has its own sizing constraint, limiting/restricting the size of DAC to an integral multiple of the SRAM pitch,

PMOS from TSMC.28nm tech is used as a base for the 1st DAC design (8 Bits), which is then simulated for potential failure/checks followed by mitigating to a more sturdy design, namely the String Resistor DAC, where the design is modified so as to minimize the number of components wrt the conventional approach, followed by discussion of some viable future implementations (Designs)

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| **DAC** | Digital to Analog Converter |
| **IITM** | Indian Institute of Technology, Madras |
| **I** | current |
| **IMC** | In Memory Computing |
| **MAC** | Multiply-Accumulate |
| **SRAM** | Static Random Access Memory |
| **WL** | Word Line |
| **eq.** | equation |
| **mux** | multiplexer |
| **wrt** | with respect to |
| **6T** | 6 Transistor |

# NOTATION

| | |
|---|---|
| $r$ | resistance, *ohm* |
| $R$ | resistor |
| $e$ | exponential |
| ln | log to natural base |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation

In Memory Computing play a very important role in countering the bottlenecks inherent to neural network computation: idle access time, optimizing just the hardware for faster computation will not solve the underlining issue of memory access also since the majority of operation in a neural network involves Multiplication and accumulation commonly known as the MAC operation, in In Memory Computation we search for a solution by providing/achieving some computation during the memory read cycle itself, creating a window for reduced unused access time

## 1.2 Methodology

During the Memory Read Operation (refer:*Barth* et al. *[2008] Khare* et al. *[2008]*) The bitwise multiplication of input and weight is aided by the access transistor ( weight being trained beforehand and stored in memory cell) The Multiplication result is indicated by Bit Line's discharge state, where the bit line voltage act as the measure of the result's magnitude , one important thing to note is that the process gets initiated only if we have a 0 stored at the memory cell as the bit line is pre-charged, and so the memory cell stores negation of weight data bits (!W)

Our Aim is to provide a current flow using the access transistor which will help in producing a linear discharge of Bit Line and assist in establishing a linear relationship between the result (multiplication) and bit line voltage.

$$I = C\frac{dV}{dT} \tag{1.1}$$

Due to energy constraints, the access transistor is operated in the subthreshold region, which results in the transistor following an exponential current relation wrt gate voltage. The Current equation for the NMOS, *eq.1.2* here the NMOS is switched on only when it has a voltage developed across it, ( $V_{ds} > 0$, drain and source),

$$I_D = I_s e^{\frac{V_{gs}}{nkt/q}}(1 - e^{\frac{-V_{ds}}{kt/q}})(1 + \lambda V_{ds}) \tag{1.2}$$

The linear behavior of the current is expected when we have a logarithmic value at the gate of the access transistor, i.e *to get a linear relation $V_{gs}$ has to be made logarithmic.* which in turn should be proportionate to the input, hence one of the key components in the design is a **DAC**, capable of delivering a logarithmic output,

Also, since we'll be switching all the WL (word lines) together in parallel we'll require the DAC units equivalent to our rows i.e one DAC localized for each row, this puts an area constraint on our design

In this Thesis, we consider the architecture of a Multi-bit DAC with logarithmic behaviour (part of an IMC architecture)

## 1.3 Design Constraint

As mentioned beforehand the DAC design is localized to every SRAM cell (row), meaning if our implementation of IMC has a column to row ratio of C X R, with 1 DAC associated with each row (one column activates per iteration), we'll be having **R** DACs for the SRAM array, this imposes some unique restriction on our DAC design:

- DAC's size should be an integral multiple of SRAM cell pitch
- minimizing the number of components
- minimizing the sizes for MOSFETs, if used (area)
- min voltage of operation (preventing heat: temp/power loss)
- feasible operation speed, in accordance with other components

We work with TSMC 28 nm technology,

In this technology the short channel effect becomes significant, presenting additional challenges for the design.

# CHAPTER 2

# Design of a 8bit logarithmic DAC

## 2.1 Binary Weighted Approach

The methodology used for designing this DAC takes inspiration from the binary bit weight approach, binary since each Input Bit powers a PMOS which subsequently have a doubling effect in terms of the current output,(achieved by increasing the width of the subsequent PMOS transistor by a factor of 2)

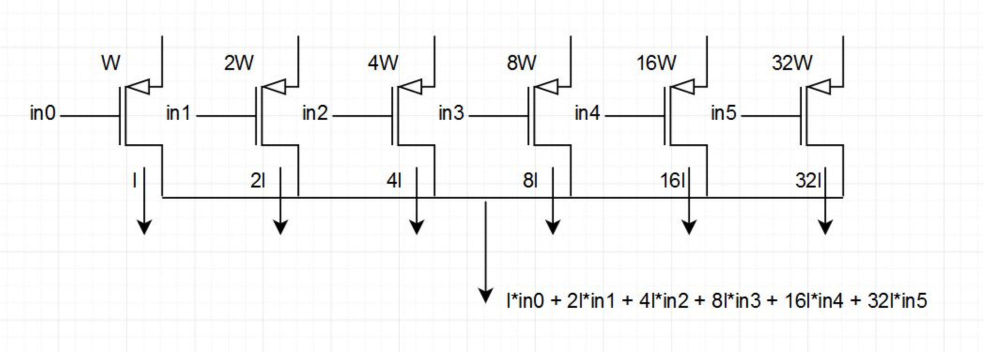


Figure 2.1: The binary model

### 2.1.1 Use of Parallel MOSFETs

If we follow the methodology and increment the width of every PMOS transistor in accordance with the bit weight, then the last MOSFET will have a width equivalent of 256 times the initial one, to counter this we use the technique of (placing) parallel MOSFETs and use finger technique (for layout consideration).

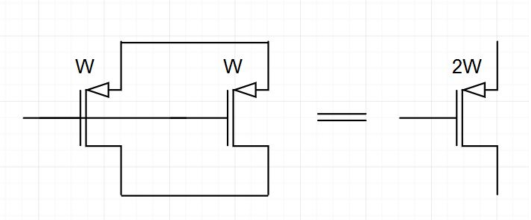


Figure 2.2: Parallel Cap approximate equivalence

### 2.1.2 Design Methodology

Here PMOS transistors are used, since we need to pump in the current at the node (access gate capacitor) and charge it, the binary approach will give us a current output in proportion to the input value, but we now have to use this current to develop a logarithmic voltage and so require a component with logarithmic behaviour between current passed and voltage produced

### 2.1.3 Using an NPN BJT, Logarithmic Voltage wrt Current

For producing logarithmic behaviour, we have two choices: MOSFETs (in subthreshold) or BJT, A MOSFET in subthreshold region would have to support the current using its $I_s$ current, which can only be done if the size of that MOSFET is very large (area constraints), another component which can perform the desired operation is an NPN BJT, connected in a diode configuration (Fig 2.3)

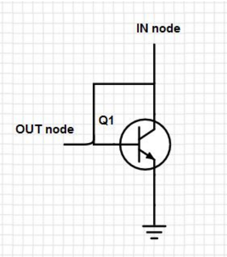


Figure 2.3: NPN transistor connect in diode configuration

### 2.1.4 Need for Current Mirror

After Simulations, the desired voltage graph is realised but with a displaced/shifted range of 700mV to 1.12V, which is 700mV above our desired range Fig. 2.10, hence we need to shift the output obtained by a value of 700mV (0-350mV, desired), the method adopted for producing the shift involves the idea of passing a minimal sufficient current (through the existing setup) which then coupled with a suitable resistor produces the desired alteration, Fig 2.4 (the concept revolves around using a constant current flow with a resistor for producing the voltage drop) a current mirror is selected for supplying the desired current (as the same current must be extracted from the output node, so as to avoid any deviations)

One important thing to note is that we can not directly select any current value as: if the current's magnitude is too small then the resistance value required to produce the voltage drop becomes very large and, if the current's magnitude is too high then the existing setup is disturbed due to a slight mismatch in the current mirrors capability to infuse and withdraw identical currents (input and output current mismatch)

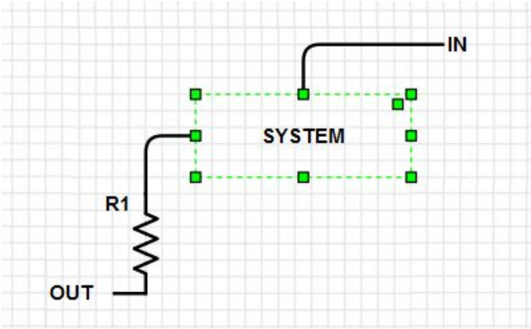


Figure 2.4: constant current pushed through IN node and extracted out of OUT node produces a voltage drop depending on the resistor value R

### 2.1.5 Designing the Current Mirror

One of the choices for current mirror design is the classical NMOS current mirror, but to push the current in the voltage node (IN), one cannot directly use the NMOS transistor, since the voltage is variable at the source, and will affect the current behaviour of the transistor, So we shift to a PMOS at the pushing side and NMOS at the receiving end with a resistance load in between to get the desired voltage shift. One of the problems faced here is the inaccurate matching of the NMOS and PMOS transistor, which leads to altered voltages *(due to the access current passing through the Base terminal of the BJT)*, at the output node.

So we shift to the second option of npn and pnp current mirror, for matching accuracy but they offer very low temp and voltage tolerance. Fig 2.11 and Fig.2.12

### 2.1.6 Simulations

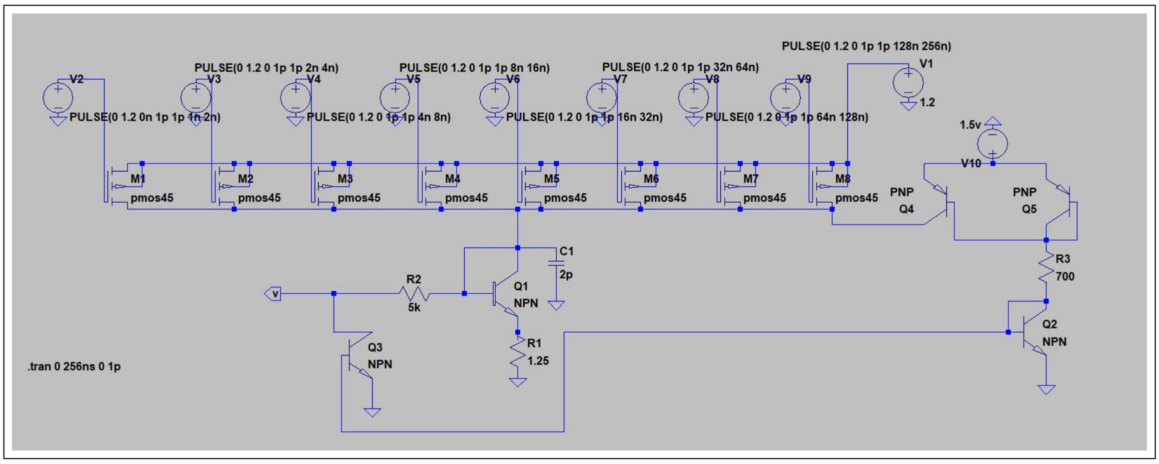

PULSE(0 1.2 0 1p 1p 128n 256n)
V1
1.2
PULSE(0 1.2 0 1p 1p 2n 4n)
PULSE(0 1.2 0 1p 1p 8n 16n)
PULSE(0 1.2 0 1p 1p 32n 64n)
V2
V3
V4
V5
V6
V7
V8
V9
PULSE(0 1.2 0n 1p 1p 1n 2n)
PULSE(0 1.2 0 1p 1p 4n 8n)
PULSE(0 1.2 0 1p 1p 16n 32n)
PULSE(0 1.2 0 1p 1p 64n 128n)
1.5v
V10
M1
pmos45
M2
pmos45
M3
pmos45
M4
pmos45
M5
pmos45
M6
pmos45
M7
pmos45
M8
pmos45
PNP
Q4
PNP
Q5
C1
2p
R3
700
R2
5k
v
Q1
NPN
R1
1.25
Q3
NPN
Q2
NPN
.tran 0 256ns 0 1p

Figure 2.5: Schematic of the Design, used for initial testing on LTSpice

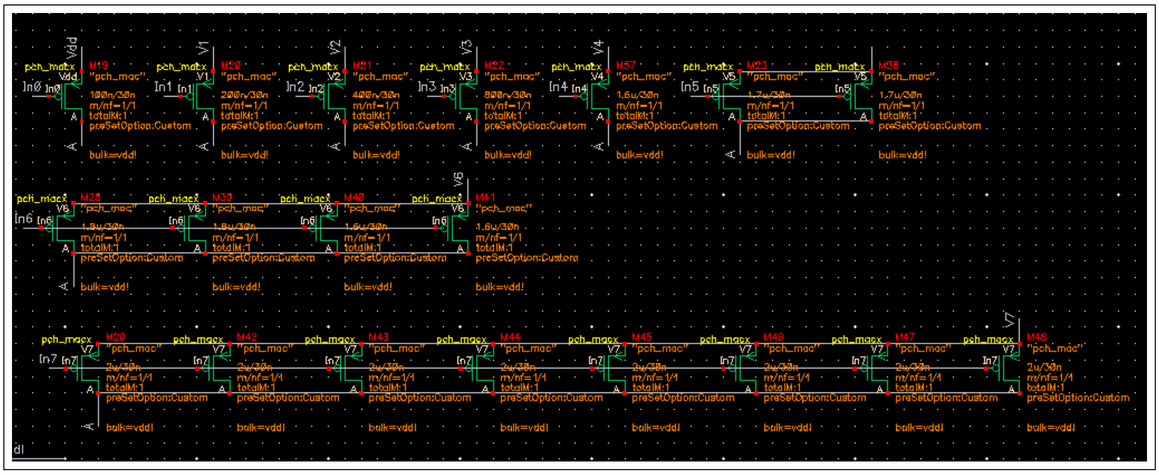


Figure 2.6: PMOS structure for Binary Array

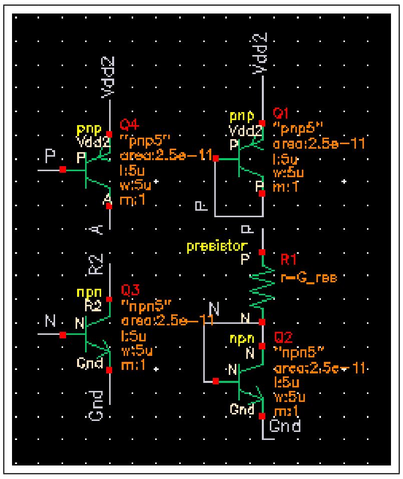


Figure 2.7: BJT current mirror structure

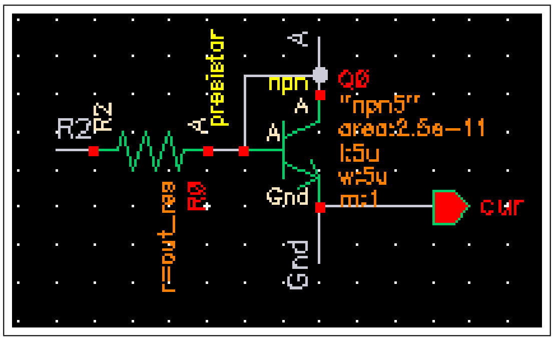


Figure 2.8: BJT used for generating the exponential voltage

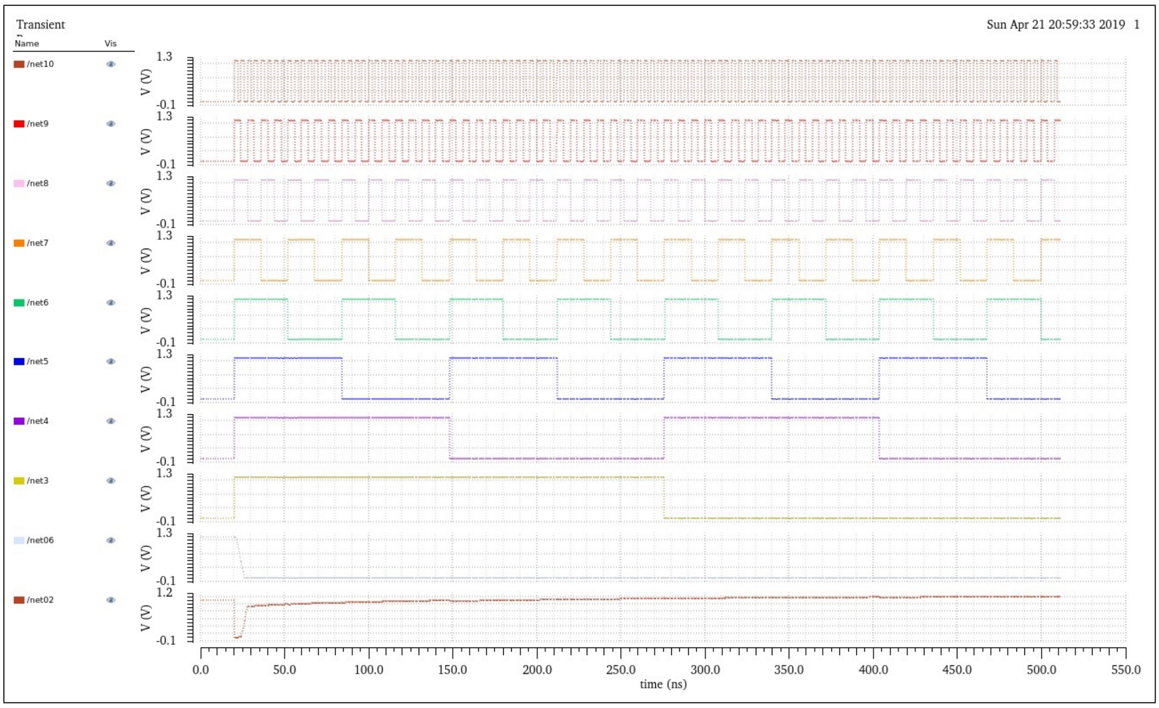


Figure 2.9: input waveform

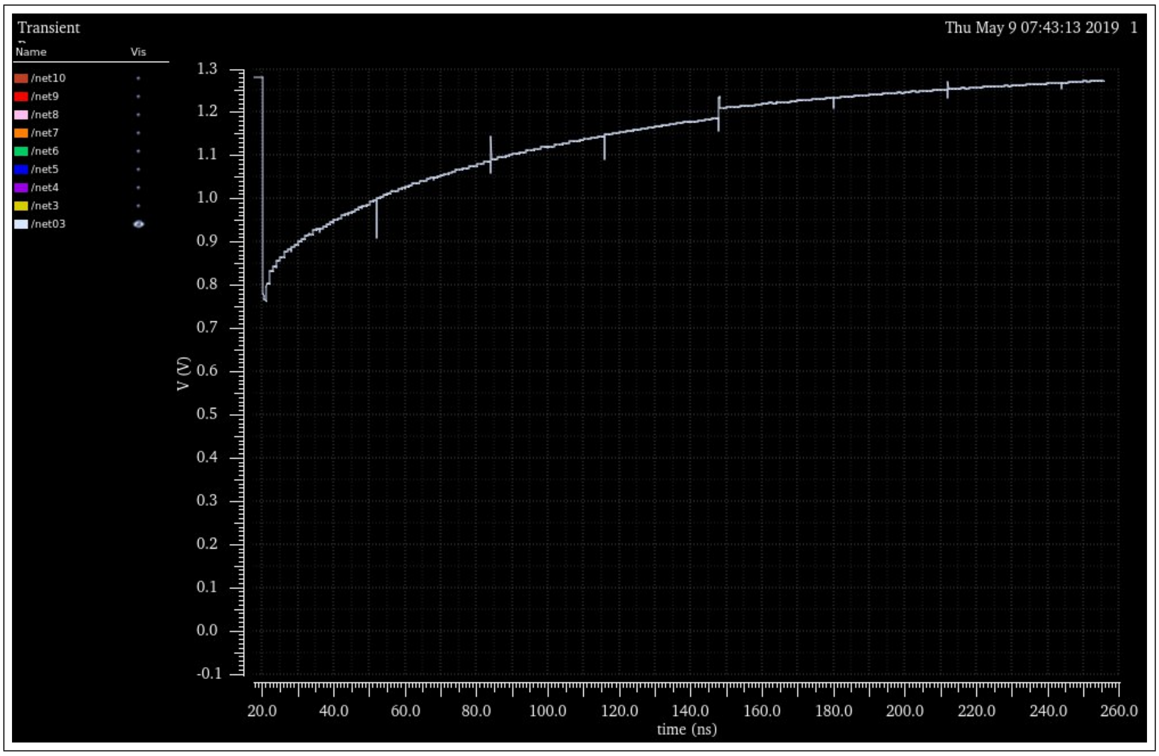


Figure 2.10: output voltage wrt Time

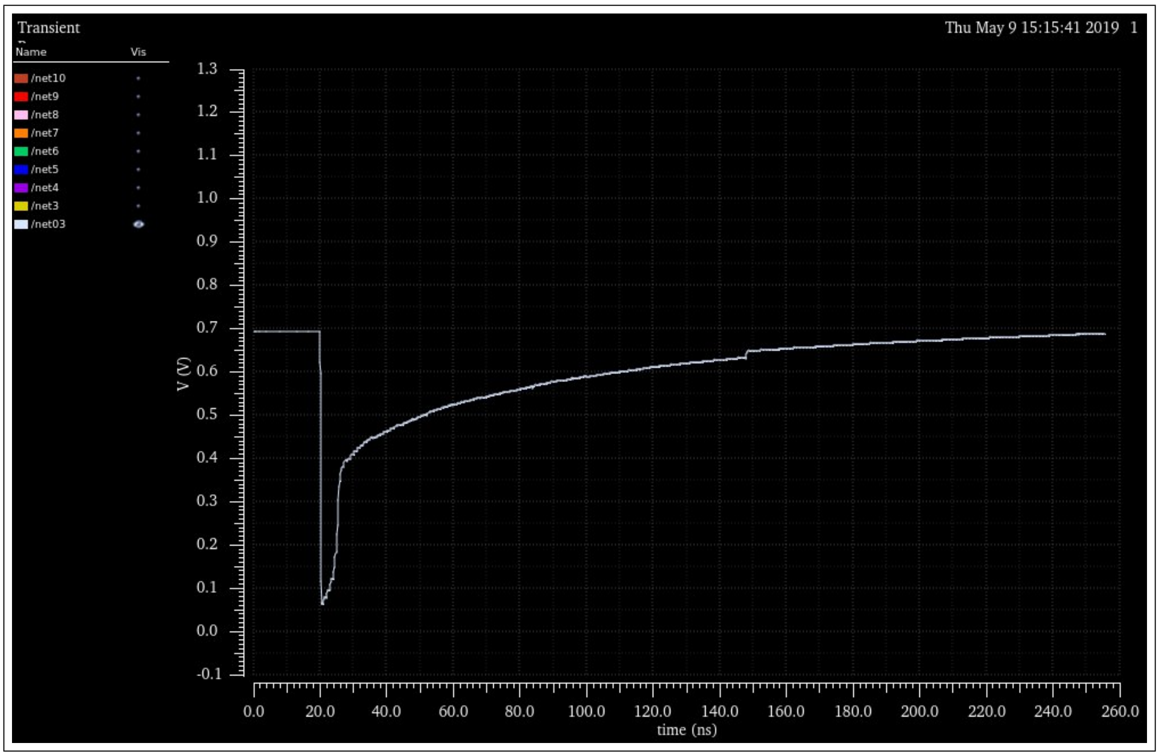


Figure 2.11: Shift produced when 1.9 V is used for Current mirror, output range 65mV to 700mV

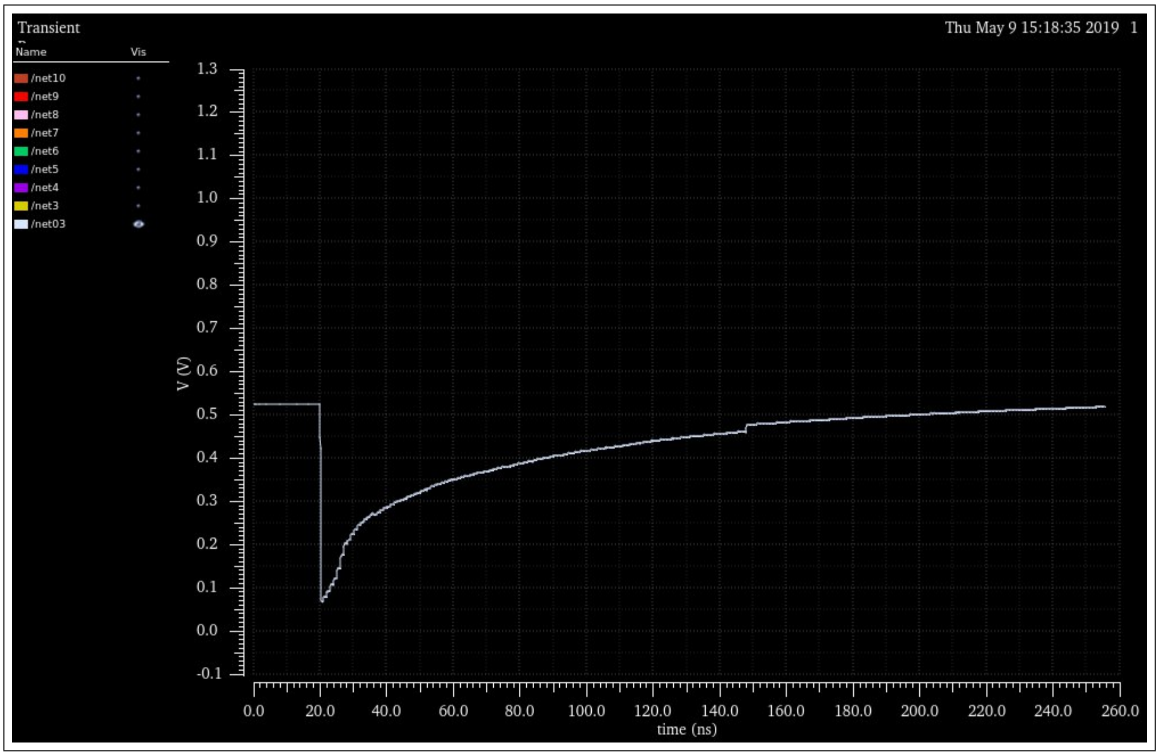


Figure 2.12: Shift produced when 2 V is used for Current mirror, output range 65mV to 518mV

### 2.1.7 Limitation and Failures

The DAC designed by using the BJT as a current mirror is Highly sensitive to voltage variations, (as depicted in Fig 2.11 and 2.12), with a slight change of .1V the voltage mapping is altered, also after being incorporated on-chip with other components of IMC architecture the DAC is expected to encounter temperature variations which can modify the current and cause deviations, therefore even though one can achieve the desired voltage behaviour, one cannot directly shift it to the desired range, (with the existing current mirror design), hence some other more robust approach is required, or a better mechanism for shifting the node voltage is desired.

# CHAPTER 3

# Design of 5bit Resistor String DAC

## 3.1 String Approach

To improve upon the tolerance and robustness of the earlier model (binary weighted), we move on to the Resistor-String DAC approach *(refer:A.1)* As required the nodal voltages are calculated beforehand, since they are the normalized log equivalent of the input *(refer:A.4)* log series (range: 0-$V_t$ of the mos, $V_t$ is 0.35V)

The base case for designing a 5 bit DAC (32 distinct value): Consist of N+1 (33) resistor, and involves selecting suitable VDD, GND and resistor values for achieving desired nodal voltages.

To connect the respective node (respective voltage) to the output (gate of access transistor), we require a 32:1 multiplexer, the multiplexer is designed using pass gates, this approach is used since it minimizes the transistors usage, but the mux itself leads to an additional overhead of 62 transistors *(refer:A.2)*

## 3.2 Minimizing Design Components

Taking inspiration from Resistor-String DAC approach (Appendix A.1) to reduce the resistor count we create 2 different layers of resistor arrays: layer1, layer2 (layer$2_0$, layer$2_1$), (the role and requirement of 2 different arrays at layer 2 will be discussed in the next section)

Only 2 layers are created so as to terminate/mitigate the influence of following(secondary) layer loading the initial(primary) layer, when the two layers are connected using the multiplexer, the second layer acts as a parallel load for the primary one and can lead to disturbed nodal voltages, to solve this problem we make the secondary layer's magnitude relatively large when compared to the primary.

$$R_{el} = \frac{1}{R_{primary}} + \frac{1}{R_{secondary}} \tag{3.1}$$

if $R_{secondary} >> R_{primary}$,then

$$R_{el} = \frac{1}{R_{primary}} \tag{3.2}$$

conventional method involves introducing a buffer in between layers but due to the area constraints, we will avoid buffers.

By using the layered structure, we have reduced the number of required resistors from 33 to 21 (2 layers of 5 and (8 + 8) resistors) Fig 3.1

The input defined is 5 bits in length, which gives us 32 different possible combinations, now we have the option of distributing the values into 2 layers with each layer having nodal distribution in power of 2. (it simplifies the mux select line logic), of the available choices **4X8 and 8X4** are good, (where 4X8 signifies 4 resistors in first layer and 8 in the second), we select the later one as it gives an advantage in the number of MOSFETs required to construct the multiplexers, Fig 3.2

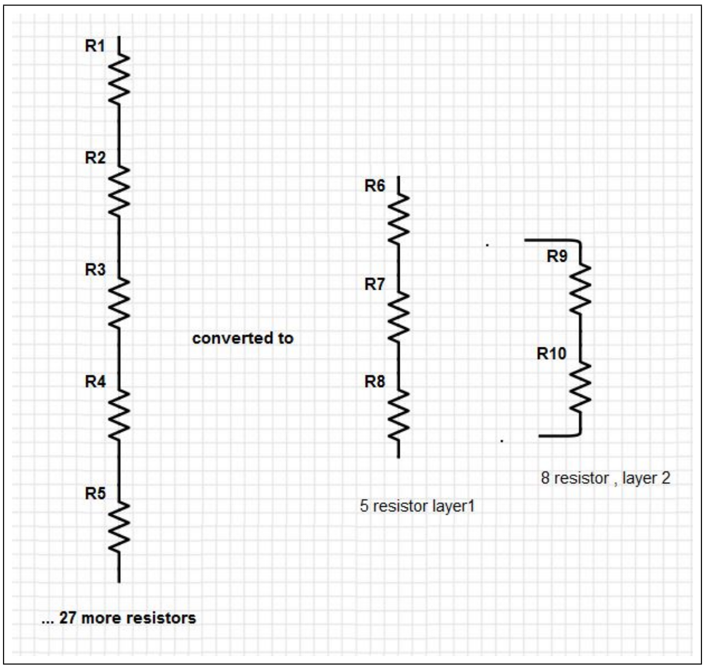
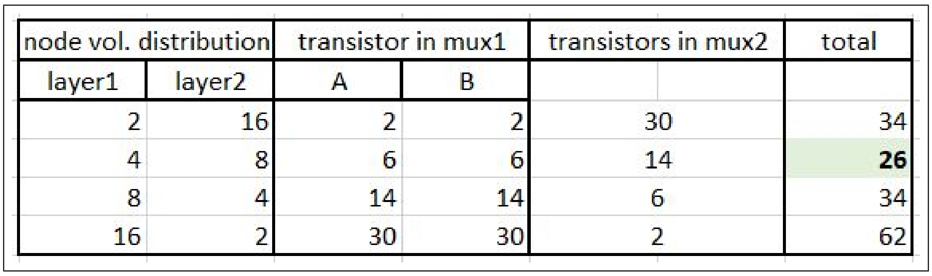


Figure 3.1: The base case and layered resistor comparison



| node vol. distribution | | transistor in mux1 | | transistors in mux2 | total |
|---|---|---|---|---|---|
| layer1 | layer2 | A | B | | |
| 2 | 16 | 2 | 2 | 30 | 34 |
| 4 | 8 | 6 | 6 | 14 | **26** |
| 8 | 4 | 14 | 14 | 6 | 34 |
| 16 | 2 | 30 | 30 | 2 | 62 |

Figure 3.2: Transistor Table for selecting nodal distribution into 2 layers

### 3.2.1 Working and Advantage

The role of layer1 is to provide node values by division-8, meaning the node values corresponding to 0, 8, 16, 24 and 32 (input), configuration of layer1: consist of 5 resistors Fig 3.3, given a Vdd of 500mV, since the voltage range is 0 - 350mV, (one can decrease the voltage up to 350mV if required, but it'll modify the resistor value too),Fig 3.4 for determining the resistance ratio for achieving the respective nodal voltages, another constraint to keep in mind is the high power dissipated in this array structure, (resistances in series connection and supply voltage of 500 mV).

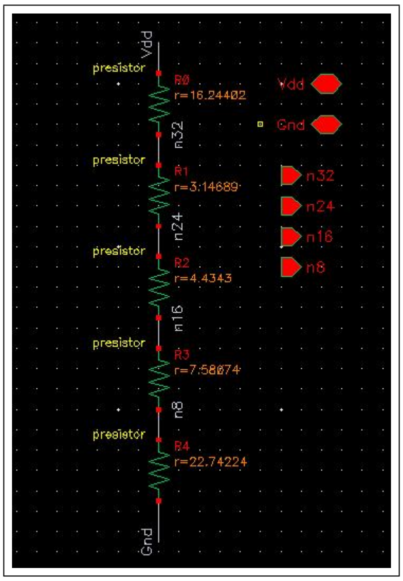
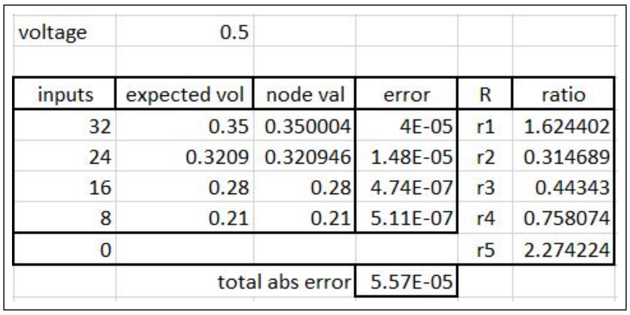


Figure 3.3: schematic of layer 1, depicting the resistor values



| voltage | 0.5 | | | | |
|---|---|---|---|---|---|
| inputs | expected vol | node val | error | R | ratio |
| 32 | 0.35 | 0.350004 | 4E-05 | r1 | 1.624402 |
| 24 | 0.3209 | 0.320946 | 1.48E-05 | r2 | 0.314689 |
| 16 | 0.28 | 0.28 | 4.74E-07 | r3 | 0.44343 |
| 8 | 0.21 | 0.21 | 5.11E-07 | r4 | 0.758074 |
| 0 | | | | r5 | 2.274224 |
| | | total abs error | 5.57E-05 | | |

Figure 3.4: excel simulation for selecting the resistor ratio of layer 1

The advantage we obtain by creating layer$2_0$ *(refer:A.3)* is the reduction in the number resistor, from a base case of 32: (32:24)8 + (24:16)8 + (16:8)8 + (8:0)8 we reduce it to (layer1)5 + (layer$2_0$)8 + (layer$2_1$)8 = 21,i.e with some small error hit one can map the resistor set for 3 intervals , 32:24, 24:16, 16:8 to a single set, referred to as layer$2_0$, *(refer:A.3)*

The overlapping is possible due to relative linear affinity (relation) between the logarithmic value of 3 intervals 32:8, which can be modelled by a single resistor array, but inserting the 7:0 interval leads to an increase in the error of the simulated(prediction) data, if we decide to match the 7:0 values to the resistor array we get a higher error for the logarithmic values of the later intervals (32:8), even in the 32:8 interval there is an internal trade-off between error percentage of the 32:16 interval and 16:8 interval (here we prioritize 32:16 interval)

The best solution here is to have two different resistor array. one for the 32:8 values, and another one for 7:0 values, the trade-off we face for improving the accuracy is increased in the number of transistors, we now require one more multiplexer structure for the 8:0 ($layer2_1$), required to connect to the output.

### 3.2.2 The requirement of layer$2_1$

As mentioned the variation in the value corresponding to input 8 down to 0 cannot be approximated with the same resistor set used for 32:8, since the variation in the slope is quite steep,(Fig 3.7),one can note the difference in the slope value (slope graduation) is quite significant for the earlier part as compared to the later values, the crux of the solution lies in dividing the logarithmic graph into different segments and then modelling them into linear models (resistors in our case).

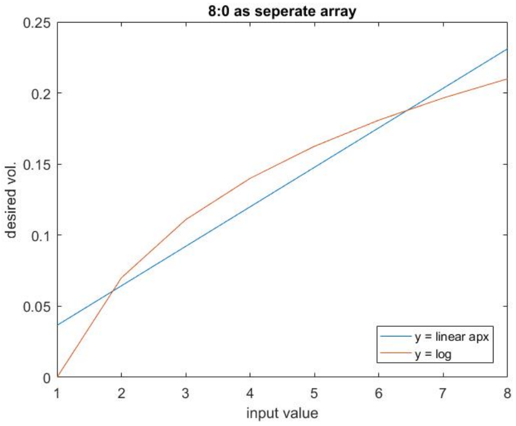


Figure 3.5: when 8:0 is treated as a special case

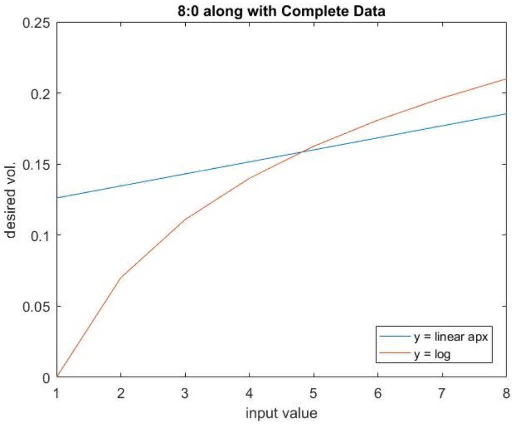


Figure 3.6: when 8:0 is combined with other interval

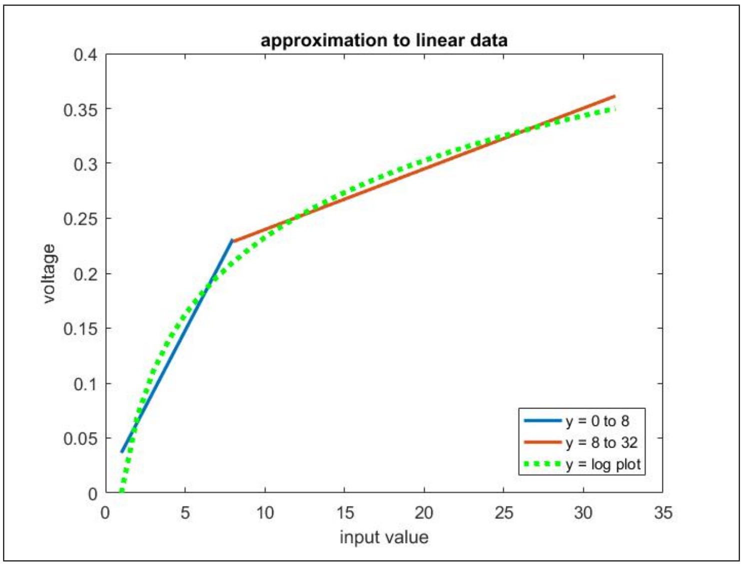


Figure 3.7: linearization of logarithmic values, by considering 2 different interval sets. 32:8 and 8:0

### 3.2.3 Transistor sizing

The transistor sizes are decided and set in accordance with their functioning.

For example in the multiplexer used between layer 1 and layer 2 the transistors used for connecting the source node to the upper node of the layer2 are sized relatively smaller when compared to the transistors used for connecting the Lower node of layer2, this is done in order to keep the voltage drop across the connecting transistors(NMOS) to the minimum. (current of the order of 160nA)

Table 3.1: the Respective MOS sizing

| Role | From | To | Number | Size | Reason |
|---|---|---|---|---|---|
| mux1 | layer1 | (A)layer2 | 5 | 500n | providing sufficient current |
| mux1 | layer1 | (B)layer2 | 5 | 1u | reducing voltage drop |
| $mux2_0$ | $layer2_0$ | output | 14 | 500n | sufficient current for charging |
| $mux2_1$ | $layer2_1$ | output | 13 | 500n | sufficient current, rise time |
| $mux3_0$ | $layer2_0$ | out | 2 | 500n/250n | sufficient current, rise time |
| $mux3_1$ | $layer2_1$ | out | 2 | 500/1000n | sufficient current, rise time |

## 3.3 Simulations and Results

The Simulation were conducted with output capacitance as 1p F (load), (which was later modified to 30f F),

Table 3.2: The Simulation's Output, *Calculated : Theoretical voltage, Expected: the voltage expected after introducing the linear Models (Resistors).*

| Input | Calculated (mV) | Expected (mV) | Output (mV) | Error (mV) |
|---|---|---|---|---|
| 32 | 350.000 | | | |
| 31 | 346.794 | 346.957 | 346.580 | 0.214 |
| 30 | 343.482 | 343.771 | 343.490 | 0.008 |
| 29 | 340.059 | 340.436 | 340.160 | 0.101 |
| 28 | 336.515 | 336.939 | 336.580 | 0.065 |
| 27 | 332.842 | 333.262 | 333.028 | 0.186 |
| 26 | 329.031 | 329.386 | 329.400 | 0.369 |
| 25 | 325.070 | 325.289 | 325.440 | 0.370 |
| 24 | 320.947 | 321.170 | 321.346 | 0.399 |
| 23 | 316.649 | 316.649 | 316.170 | 0.479 |
| 22 | 312.160 | 312.160 | 311.800 | 0.360 |
| 21 | 307.462 | 307.462 | 307.200 | 0.262 |
| 20 | 302.535 | 302.535 | 302.310 | 0.225 |
| 19 | 297.355 | 297.355 | 297.300 | 0.055 |
| 18 | 291.895 | 291.895 | 292.020 | 0.125 |
| 17 | 286.122 | 286.122 | 286.370 | 0.248 |
| 16 | 280.000 | 280.320 | 280.600 | 0.600 |
| 15 | 273.482 | 272.627 | 271.770 | 1.712 |
| 14 | 266.515 | 264.955 | 264.300 | 2.215 |
| 13 | 259.031 | 256.925 | 256.400 | 2.631 |
| 12 | 250.947 | 248.504 | 248.170 | 2.777 |
| 11 | 242.160 | 239.651 | 239.500 | 2.660 |
| 10 | 232.535 | 230.319 | 230.430 | 2.105 |
| 9 | 221.895 | 220.453 | 220.700 | 1.195 |
| 8 | 210.000 | 210.536 | 210.970 | 0.970 |
| 7 | 196.515 | 196.515 | 196.600 | 0.085 |
| 6 | 180.947 | 180.945 | 181.073 | 0.126 |
| 5 | 162.535 | 162.535 | 162.744 | 0.209 |
| 4 | 140.000 | 140.005 | 140.350 | 0.350 |
| 3 | 110.947 | 110.945 | 110.949 | 0.002 |
| 2 | 70.000 | 70.000 | 71.000 | 1.000 |
| 1 | 0.000 | 0.000 | 11.000 | 11.000 |
| 0 | 0.000 | 0.000 | 17.000 | 17.000 |

The output Waveform for Input value 5 is shown as it experience the slowest rise time due to capacitor sizing.

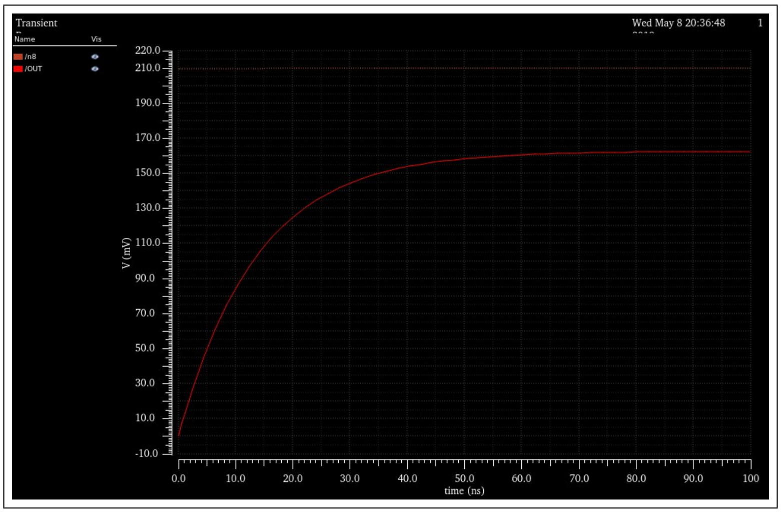


Figure 3.8: Rise Time for Input Value 5, with Output Load as 1p F, approximately 95ns

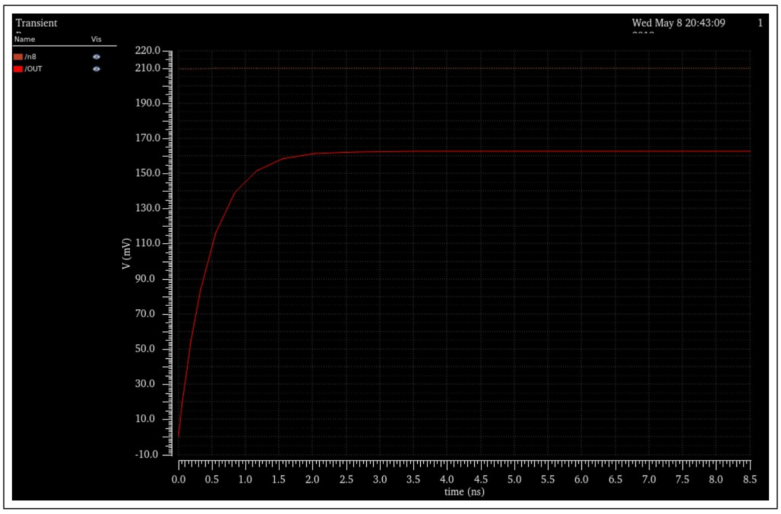


Figure 3.9: Rise Time for Input Value 5, with Output Load as 30f F, approximately 3.5ns

### 3.3.1 Limitation

*Mentioned in the Next Section*

# CHAPTER 4

# Conclusion and Future Work

## 4.1 Conclusion

### 4.1.1 Binary Weighted DAC

The DAC is designed for 8 bits input, implying the distribution of 350mV of value into 256 different division (in logarithmic fashion), (Appendix A.4) which requires very good accuracy for differentiating between (later) values, if we reduce the bit size to 5, (the division decreases to 32 distinct value), and the overall performance and accuracy can be improved.

But the issue that we face is of providing a constant voltage shift with good tolerance, as the accuracy (logarithmic behaviour) of the output can be improved by changing certain circuit parameters for example: improving the current linearity by modifying the PMOS width, which can then be used to produce a logarithmic relation, But producing the voltage shift is tricky,

The designed model can be used in circuits where there's a requirement for logarithmic output and the desired range is 700 mV to 1.2V or even a fraction of this range (achievable through resistor divider), in-fact any range above threshold region can be mapped, again if the area constraints are relaxed enough to allow the usage of an Opamp then one can design a more robust system and produce a constant voltage shift.

### 4.1.2 String Resistor Model

Although this model provides a good tolerance with respect to the temperature and voltage variations, the power consumed by the model is on the higher side when compared to the Binary model (resistors vs transistors), the design is better when compared to the Base model, but the area occupied by the model is reasonably large when compared to our desired size. *area estimate of 20um X 19um (using cadence)*

## 4.2 Future Work

### 4.2.1 Binary DAC Approach

One has to carefully verify the linear current variations if we plan to move on to the 8-bit model, the gate width adjusted suitably, as mentioned before, After the initial verification phase, one can experiment with a better current mirror for providing the voltage shift with good tolerance, one can also look for other suitable ways of shifting the voltage, one proposed method incorporates the idea of setting the voltage levels high enough so that one can attain the desired voltage range simply by using a resistor divider, but irrespective of the approach used for getting the shift the additional current introduced if any should be small enough to not cause any disturbance to the existing current flow (leading to modification of exponential voltage)

### 4.2.2 String DAC

The future work involves searching for components (exploring different designs) which can be substituted in place of the resistor, following similar working principle, and consuming much lesser power, One approach involves the idea of using transistors for producing the desired nodal voltages, (power saving) but here one has to be careful while working with the idea of using the transistors in the subthreshold region (desired output voltage range) and should ensure its proper functioning, biasing and robustness. Other work involves finding a better mechanism for coupling the 2 resistor layers together and even coming up with a new technique for muxing the $layer2_0$ and $layer2_1$ in the pre-final stage (minimizing the number of transistors).

# APPENDIX A

# APPENDIX

## A.1 Resistor String DAC Method

Resistor-String DAC approach is a novel idea for generating expected voltages in a series resistor connection which can be adjusted by changing the resistor ratios. let's take an example for better understanding:

Say we require 0 to 100 volts with step size of 1V i.e 100 distinct values, one way of generating the voltage is by taking 100 equivalent Resistances connecting them in series and assigning *(the top node voltage)* Vdd = 100V, then each node in series connection will represent our steps, from which the desired voltage can be tapped as an when required (say with help of a multiplexer)

Another smarter way is to divide the set of resistor into 2 arrays A1 and A2, A1 consisting of 10 equivalent resistors (R1), and A2 consisting of another 10 equivalent resistors (R2), *where R1 may or may not be equal to R2*, Vdd=100 is connect to the top node of Array1 now we can simply connect the immediate series nodes of Array1 to the end terminal of the A2, to get a step size of 1

say we extract the values from node1 (100v) and node2 (90v) and transfer it to the terminal nodes of A2, we'll be able to get the approximate value of all intermediate steps from A2 with just 20 Resistors in place.

*Note :* We need to put buffers between A1 and A2 layer network, else the initial (A1) node voltages will get disturbed due to the parallel loading by A2 resistors.

## A.2 MUX Design

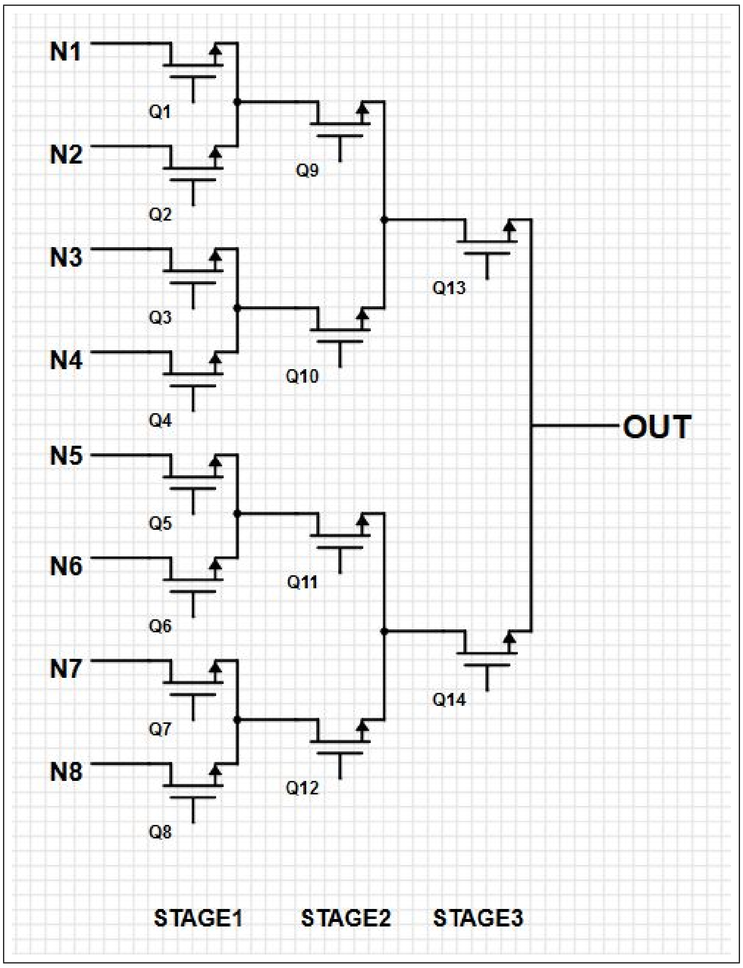


Figure A.1: example of 8 to 1 multiplexer, consisting of 3 stages

In each stage we see $2^{3-stage.num}$ NMOS, in total for 8 to 1 multiplexer we require 14 MOSFETS.

$$X = 2^1 + 2^2 + .....2^N. \tag{A.1}$$

$$summation\ of\ a\ GP \tag{A.2}$$

$$S = \frac{a(r^n - 1)}{r - 1} \tag{A.3}$$

$$here\ r = 2,\ a = 2,\ n = 3 \tag{A.4}$$

$$S = 14 \tag{A.5}$$

## A.3 Resistance Layer Values

Table A.1: The Resistor values in $layer2_1$ and in $layer2_0$, all value here are in Kohm

| Node | Layer$2_1$ | Layer$2_0$ |
|---|---|---|
| Resistor1 | 3.820 | 2.330 |
| Resistor2 | 4.471 | 2.716 |
| Resistor3 | 4.679 | 3.212 |
| Resistor4 | 4.907 | 3.931 |
| Resistor5 | 5.159 | 5.070 |
| Resistor6 | 5.438 | 7.143 |
| Resistor7 | 5.749 | 12.213 |
| Resistor8 | 5.779 | |

## A.4 Normalization

The following procedure is used to normalize the input data values to respective logarithmic (ln) voltages

$$X = [1, 2, ....N]; \tag{A.6}$$

$$Y = log(X); \tag{A.7}$$

$$Normalizing\ Y[N] = 1 \tag{A.8}$$

$$by\ using :: Y = Y/Y(N); \tag{A.9}$$

$$Y = Y * 0.35; \tag{A.10}$$

Table A.2: the normalization of 5 bit input value to the desired voltage range 0:0.35V, here Voltage is given in mV

| Input Value | Log | Normalizing | Expected Voltage |
|---|---|---|---|
| 1 | 0.000 | 0 | 0 |
| 2 | 0.693 | 0.200 | 0.070 |
| 3 | 1.099 | 0.317 | 0.111 |
| 4 | 1.386 | 0.400 | 0.140 |
| 5 | 1.609 | 0.464 | 0.163 |
| 6 | 1.792 | 0.517 | 0.181 |
| 7 | 1.946 | 0.561 | 0.197 |
| 8 | 2.079 | 0.600 | 0.210 |
| 9 | 2.197 | 0.634 | 0.222 |
| 10 | 2.303 | 0.664 | 0.233 |
| 11 | 2.398 | 0.692 | 0.242 |
| 12 | 2.485 | 0.717 | 0.251 |
| 13 | 2.565 | 0.740 | 0.259 |
| 14 | 2.639 | 0.761 | 0.267 |
| 15 | 2.708 | 0.781 | 0.273 |
| 16 | 2.773 | 0.800 | 0.280 |
| 17 | 2.833 | 0.817 | 0.286 |
| 18 | 2.890 | 0.834 | 0.292 |
| 19 | 2.944 | 0.850 | 0.297 |
| 20 | 2.996 | 0.864 | 0.303 |
| 21 | 3.045 | 0.878 | 0.307 |
| 22 | 3.091 | 0.892 | 0.312 |
| 23 | 3.135 | 0.905 | 0.317 |
| 24 | 3.178 | 0.917 | 0.321 |
| 25 | 3.219 | 0.929 | 0.325 |
| 26 | 3.258 | 0.940 | 0.329 |
| 27 | 3.296 | 0.951 | 0.333 |
| 28 | 3.332 | 0.961 | 0.337 |
| 29 | 3.367 | 0.972 | 0.340 |
| 30 | 3.401 | 0.981 | 0.343 |
| 31 | 3.434 | 0.991 | 0.347 |
| 32 | 3.466 | 1.000 | 0.350 |

# REFERENCES

**Barth, J.**, **W. R. Reohr**, **P. Parries**, **G. Fredeman**, **J. Golz**, **S. E. Schuster**, **R. E. Matick**, **H. Hunter**, **C. C. Tanner**, **J. Harig**, **H. Kim**, **B. A. Khan**, **J. Griesemer**, **R. P. Havreluk**, **K. Yanagisawa**, **T. Kirihata**, and **S. S. Iyer** (2008). A 500 mhz random cycle, 1.5 ns latency, soi embedded dram macro featuring a three-transistor micro sense amplifier. *IEEE Journal of Solid-State Circuits*, **43**(1), 86–95. ISSN 0018-9200.

**Cheng, M. .** and **T. . Huang** (1991). Switched-capacitor pipelined logarithmic a/d and d/a convertors. *IEE Proceedings G - Circuits, Devices and Systems*, **138**(6), 714–716. ISSN 0956-3768.

**Elmallah, A.**, **M. Gamal Ahmed**, **A. El-Kholy**, **W.-S. Choi**, and **P. Kumar Hanumolu**, A 1.6ps peak-inl 5.3ns range two-step digital-to-time converter in 65nm cmos. 2018.

**Ghanaatian-Jahromi, A.**, **A. Abrishamifar**, and **A. Medi** (2011). A novel voltage-to-voltage logarithmic converter with high accuracy. *Przeglad Elektrotechniczny*, **87**.

**Guilherme, J.** and **J. E. Franca**, A logarithmic digital-analog converter for digital cmos technology. *In Proceedings of APCCAS'94 - 1994 Asia Pacific Conference on Circuits and Systems*. 1994.

**Guo, J.** and **S. Sonkusale** (2009). An area-efficient and low-power logarithmic a/d converter for current-mode sensor array. *IEEE Sensors Journal*, **9**(12), 2042–2043. ISSN 1530-437X.

**Jan M. Rabaey, B. N.** and **A. Chandrakasan**, *Digital Integrated Circuits*. Pearson Education Inc., 2014.

**Khare, K.**, **N. Khare**, **V. Kumar Kulhade**, and **P. Deshpande**, Vlsi design and analysis of low power 6t sram cell using cadence tool. *In 2008 IEEE International Conference on Semiconductor Electronics*. 2008.

**Lu, C.**, **P. Yin**, **C. Hsiao**, **M. F. Chang**, and **Y. Lin** (2012). A 10-bit resistor-floating-resistor-string dac (rfr-dac) for high color-depth lcd driver ics. *IEEE Journal of Solid-State Circuits*, **47**(10), 2454–2466. ISSN 0018-9200.

**Motamed, A.**, **C. Hwang**, and **M. Ismail** (1997). Cmos exponential current-to-voltage converter. *Electronics Letters*, **33**(12), 998–1000. ISSN 0013-5194.