Workshop on Band gap reference design

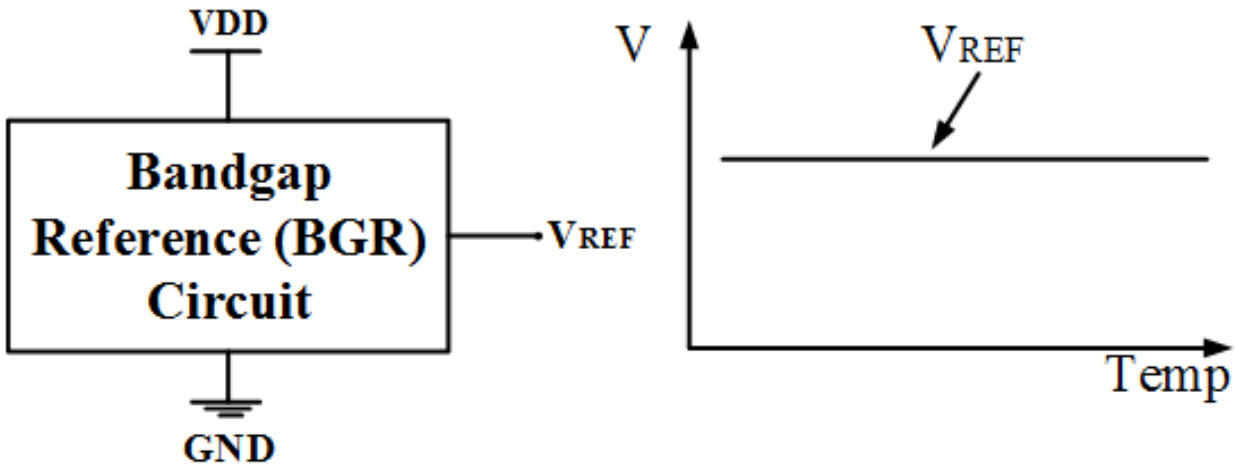
Summary:

2-day hands on lab-based workshop on Analog Bandgap IP design using SKY 130, an opensource 130nm PDK made available by collaboration between Google and Sky Water Technology Foundry. Workshop covers BGR are design from scratch to post layout simulation. On Day1 the course provides profound background on the theory of BGR, BGR's description, application, implementation and need. Day 2 provide hands on experience in BGR design using ngspice, magic and netgen. After the course one will be able to understand concept of BGR, how CTAT and PTAT blocks are designed to get remove temperature dependency and implement solution to generate constant voltage as per the required specs.

INDEX

PART1: Introduction to BGR

BGR is a temperature independent voltage reference widely used in integrated circuits. It produces a constant voltage regardless of power supply and variation in temperature.



Why BGR

There is a need for a device to provide a constant voltage independent of variation in power supply and temperature. A battery drops voltage over time. A typical power supply is noisy or has ripples and IC using Zener cannot be used due to high thermal noise and unavailability of lower voltage Zener's.

Solution

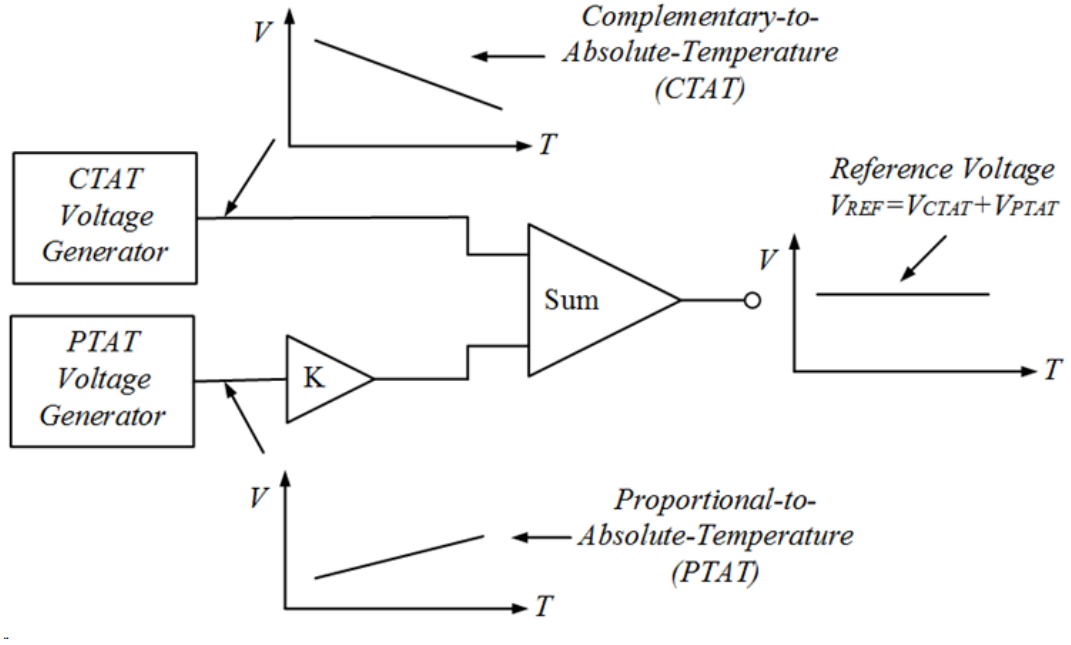
- A Bandgap reference which can be integrated in bulk CMOS, Bi-CMOS or Bipolar technologies without the use of external components.

Applications of BGR

- Low dropout regulators (LDO)
- DC-to-DC buck converters
- Analog-to-Digital Converter (ADC)
- Digital-to-Analog Converter (DAC)

BGR Principle

All devices inherently either have positive temperature coefficient or negative temperature coefficient. The operation principle of BGR circuits is to sum a voltage with negative temperature coefficient with another one with positive temperature coefficient. The two-temperature coefficient are designed so as to cancel each other out so that overall resultant is temperature independent. Generally, constant current supplied into semiconductor diode behave as CTAT i.e., complement to absolute temp. So, we need to design a PTAT i.e., proportional to absolute temp (VT=KT/Q) which can cancel out the CTAT nature i.e., with rise in temp.



Types of BGR

Based on architecture there are two times

1. Self-biased current mirror: Stable and simple design but has low PSRR and needs start-up circuit.
2. Using operational amplifier

Based on application BGR is characterised into 4 types

1. Low-voltage BGR
2. Low-power BGR
3. High-PSRR and low noise BGR
4. Curvature compensated BGR

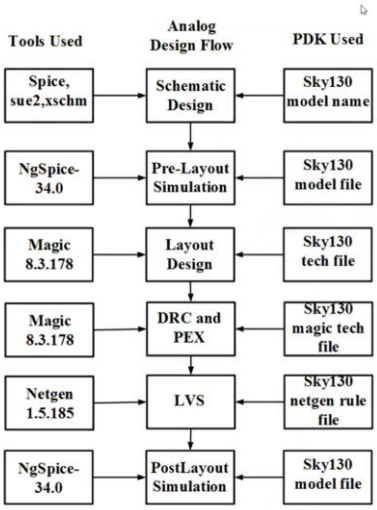
PART 2: Tools and PDK setup

- Tools used:

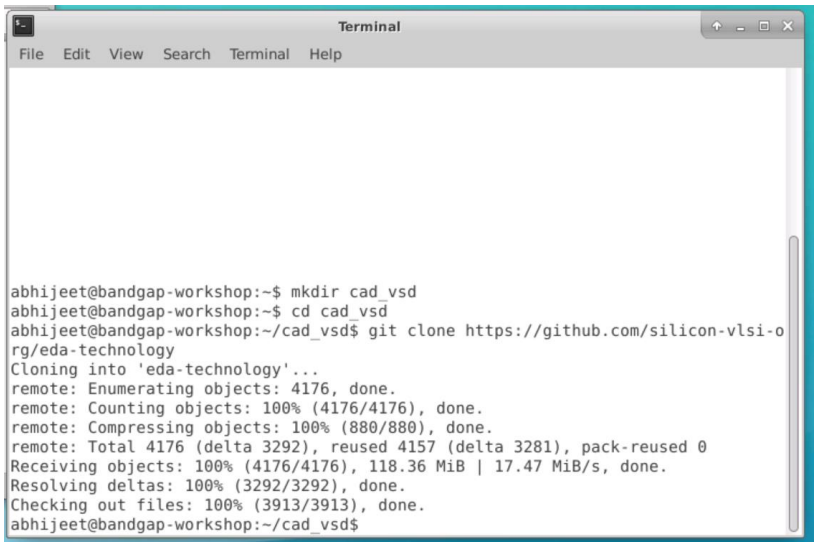
    - Ngspice: It is used for the transistor level circuit simulation and design.
    - Magic: It is used for layout design and parasitic extraction

  Both Ngspice and Magic are open-source tools which need to be downloaded and installed to the work environment.

- Development Flow and tool used:



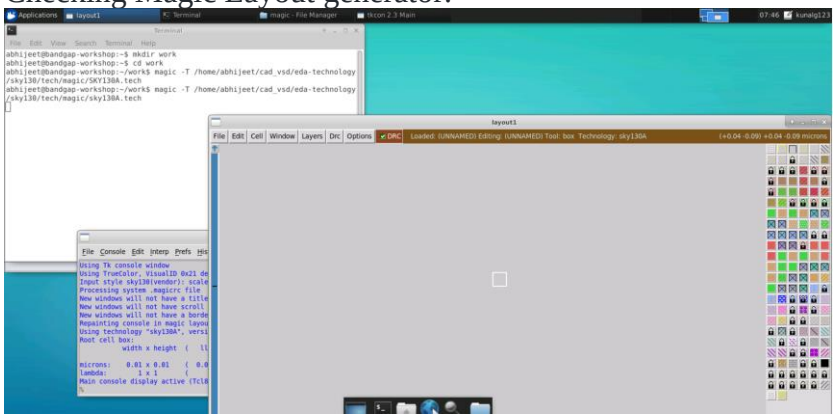
- Next, we need we have to clone the google skywater pdk. On the terminal type git clone https://github.com/google/skywater-pdk-libs-sky130_fd_pr.git or Git clone https://github.com/silicon-vlsi-org/eda-technology



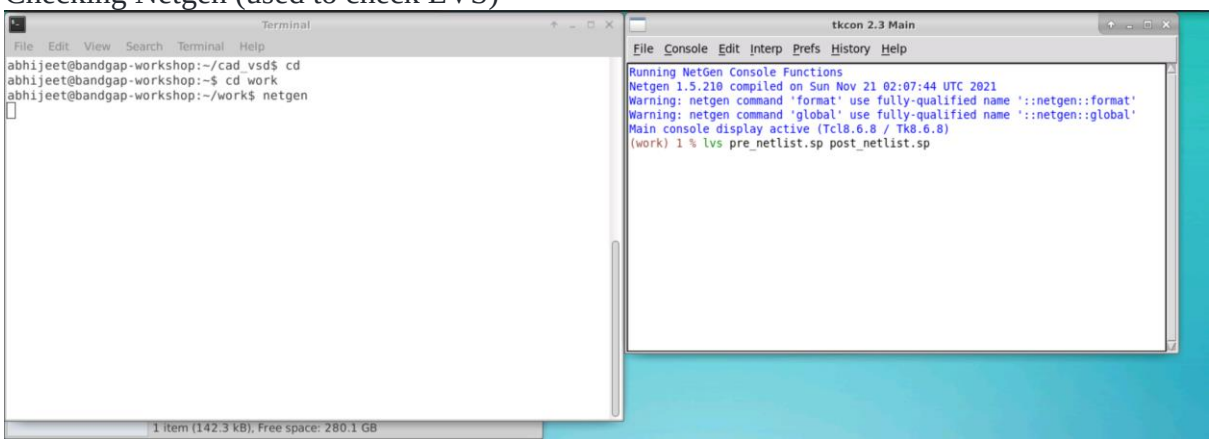
- Once all tools and pdk are set up ensure all tools are working properly.
  1. Checking all model file.

```
Terminal
File  Edit  View  Search  Terminal  Help
abhijeet@bandgap-workshop:~$ cd cad_vsd
abhijeet@bandgap-workshop:~/cad_vsd$ ls
eda-technology
abhijeet@bandgap-workshop:~/cad_vsd$ cd eda-technology
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology$ cd sky130/
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130$ ls
libraries  models  tech
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130$ cd models
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models$ ls
spice
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models$ cd spice
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models/spice$ ls
cells  models  tech
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models/spice$ cd models
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models/spice/models$ ls
all.spice            sky130.lib.spice
capacitors           sky130_fd_pr__model__diode_pd2nw_11v0.model.spice
corners              sky130_fd_pr__model__diode_pw2nd_11v0.model.spice
correl1.spice        sky130_fd_pr__model__inductors.model.spice
correl2.spice        sky130_fd_pr__model__linear.model.spice
correl3.spice        sky130_fd_pr__model__pnp.model.spice
correl4.spice        sky130_fd_pr__model__r+c.model.spice
custom.spice         sonos
head.spice           sonos_e
parameters           sonos_p
parasitics           sonos_see
r+c                  sonos_see_e
r+c.mrplmonte.spice  sonos_see_p
abhijeet@bandgap-workshop:~/cad_vsd/eda-technology/sky130/models/spice/models$
```

  2. Checking Magic Layout generator.



  3. Checking Netgen (used to check LVS)



Part 3: Design spec and data analysis

First thing before getting into design is to define the specifications. The device needs to be operated with in the specifications. For BGR below mentioned are the general specifications.

- Supply voltage = 1.8V
- Temperature: -40 to 125 Deg Cent.
- Power Consumption < 60uW
- Off current < 2uA
- Start-up time < 2us
- Tempco. Of Vref < 50 ppm

It can be observed that for BGR the power consumption, temperature coefficient and start up time need to be low.

Well defined specification helps us in choosing the Mosfets, resistors and transistors need to design BGR. For the current design following devises will be used.

MOSFET

	NFET (MOSFET)	PFET (MOSFET)
Type	LVT	LVT
Voltage	1.8V	1.8V
Thres. Voltage	~ 0.4V	~-0.6V
Model Name	Sky130_fd_pr__nfet_01v8_lvt	Sky130_fd_pr__pfet_01v8_lvt

BJT

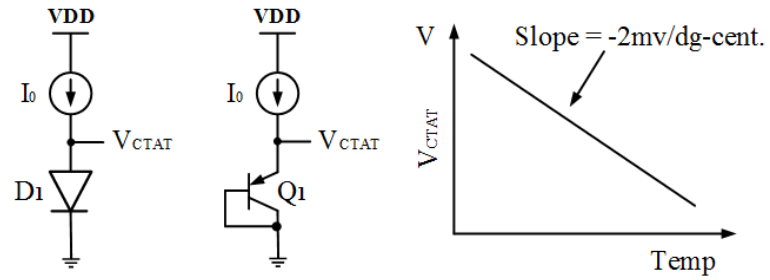
	PNP (BJT)
Current Rating	1uA-10uA/um2
Beta	~12
Emitter Area	11.56 um2
Model Name	Sky130_fd_pr__pnp_05v5_W3p40l3p40

RESISTOR

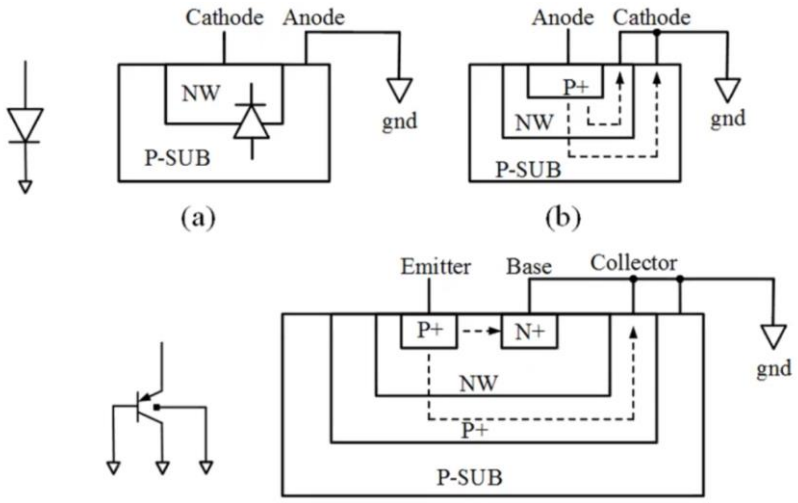
	RPOLYH (RESISTOR)
Sheet Resistance	~350 Ohm
Tempoco.	2.5 Ohm/Deg Cent.
Bin width	0.35u, 0.69u, 1.41u, 2.85u and 5.73u
Model Name	Sky130_fd_pr__res_high_po

Part 4: CTAT voltage generation circuit

If we consider constant current is flowing through a forward biased diode, then with increase in temp. we can observe that the voltage across the diode is decreasing. Generally, it is found that the slope of the V~Temp is -2mV/deg Centigarde.



While manufacturing a diode it forms a parasitic PNP transistor which leaks large current into substrate if there is a small base current. This may affect other devices present in the same substrate. In order to avoid it a diode connected transistor is used to generate CTAT. P+ diffusion is done in P-SUB and PN junction is formed in the P+ region. Now when there is base current in N+ region the large current flows from emitter to collector (P+) and doesn't leak into P-sub.



The Diode current is defined as

$$I_D = I_S e^{\frac{V_D}{V_t}} \quad \text{.....(1)}$$

Which can be written as

$$V_D = V_t \ln\left(\frac{I_0}{I_S}\right) \quad \text{.....(2)}$$

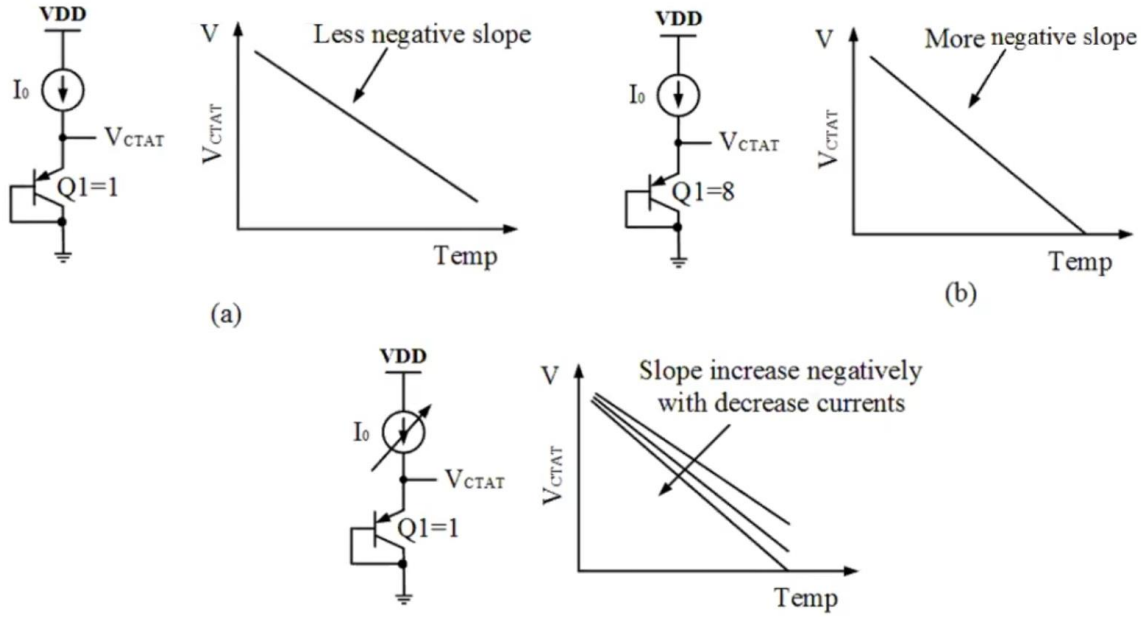
Variation of Vd wrt to temperature can be obtained by differentiating the equation 2 and it can be shown that voltage Vctat can be expressed as below

$$\frac{dV}{dT} = \frac{V_D - (4+m)V_t - \frac{E_g}{q}}{T} \quad \text{……..(3)}$$

By replacing the standard values for the above variable, it can be seen that temperature coefficient of diode connected mosfet is closed to 1.88mv/degK

$$\frac{dV}{dT} = \frac{0.7 - (4-1.5) * 0.026 - 1.2}{300} = -1.88mv/\deg k$$

The negative slope of CTAT circuit can be controlled by varying the current supplied to the transistor or by having multiple transistors connected in parallel.



Current mirror is highly dependent on VDD hence self-biased current mirror is used.

Implantation of CTAT:

CTAT is implemented by connecting a constant current source to a diode connected mosfet.

Ensure to include the sky130.lib.spice TT (TT for typical- typical process) and

Sky130_fd_pr__model__pnp.model.spice

DC simulation is run from -40 deg C to 125 deg C.

```
*** ctat voltage generation ***
.lib "/home/abhijeet/cad_vsd/eda-technology/sky130/models/spice/models/sky130.lib.spice tt"
.include "/home/abhijeet/cad_vsd/eda-technology/sky130/models/spice/models/sky130_fd_pr__model__pnp.model.spice"

.global vdd gnd
.temp=27

*** bjt definition ***
xqp1 gnd gnd qp1 gnd sky130_fd_pr__pnp_05v5_W3p40L3p40 m=1

*** supply voltage and current
vsup vdd gnd dc 2
isup vdd qp1 dc 10u
.dc   temp -40 125 5

.control
run
plot v(qp1)
.endc
.end
```

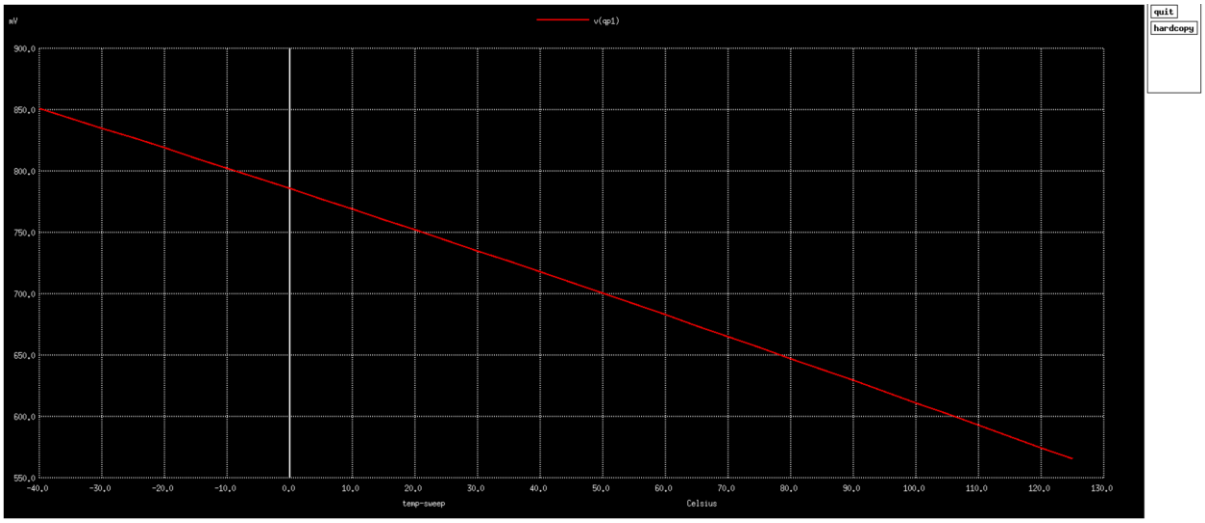
```
abhijeet@bandgap-workshop:~$
abhijeet@bandgap-workshop:~$
abhijeet@bandgap-workshop:~$ cd work
abhijeet@bandgap-workshop:~/work$  vim ptat_curcuit.sp
abhijeet@bandgap-workshop:~/work$  vim ptat_curcuit.sp
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$ vim ctat_voltage_gen.sp.
abhijeet@bandgap-workshop:~/work$ ngspice ctat_voltage_gen.sp
```

```
abhijeet@bandgap-workshop:~$
abhijeet@bandgap-workshop:~$
abhijeet@bandgap-workshop:~$ cd work
abhijeet@bandgap-workshop:~/work$  vim ptat_curcuit.sp
abhijeet@bandgap-workshop:~/work$  vim ptat_curcuit.sp
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$
abhijeet@bandgap-workshop:~/work$ vim ctat_voltage_gen.sp.
abhijeet@bandgap-workshop:~/work$ ngspice ctat_voltage_gen.sp
******
** ngspice-27 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Tue Dec 26 17:10:20 UTC 2017
******
ctat_voltage_gen.sp: No such file or directory
ngspice 1 ->
```

*Note: Lesson learnt- Be careful for additional. and typos while creating and naming files. Typos can be hard to debug in large files.(Error fix “.sp.” to “.sp”)

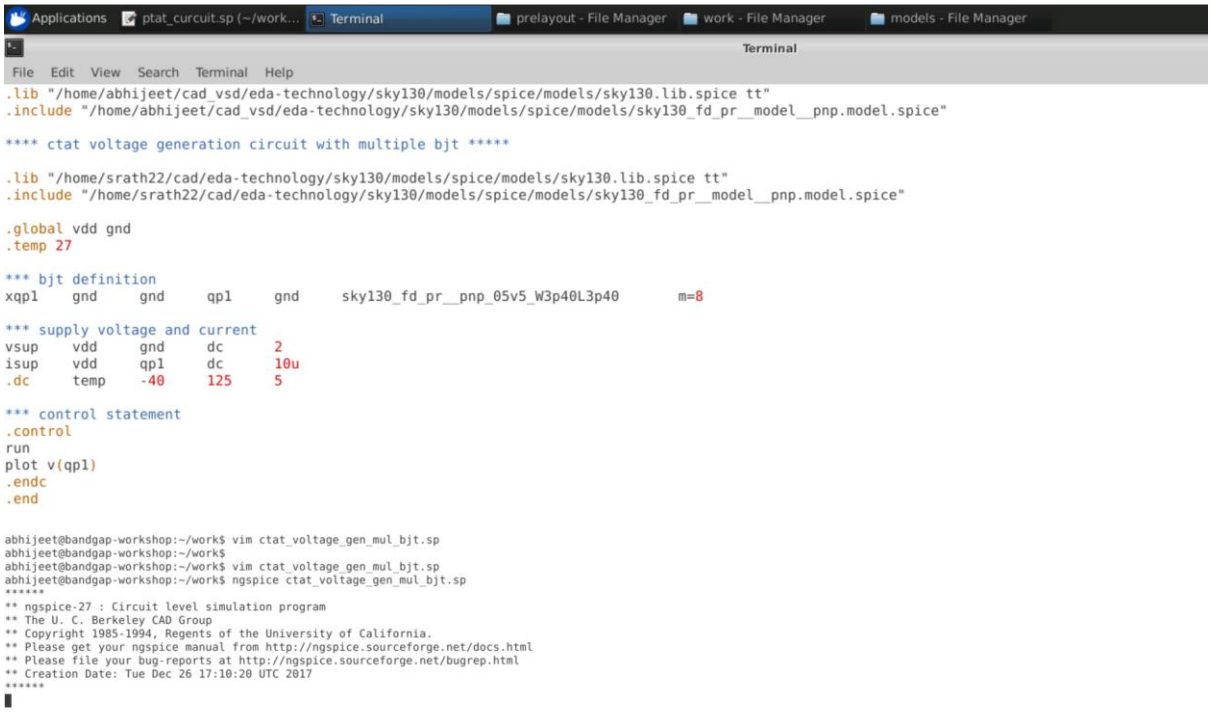
```
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Tue Dec 26 17:10:20 UTC 2017
******
ctat_voltage_gen.sp: No such file or directory
ngspice 1 ->
ngspice 1 ->
ngspice 1 -> exit
abhijeet@bandgap-workshop:~/work$ ngspice ctat_voltage_gen.sp
******
** ngspice-27 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Tue Dec 26 17:10:20 UTC 2017
******


Circuit: *** ctat voltage generation ***

Scale set
Doing analysis at TEMP = 0.000000 and TNOM = 27.000000

Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
Warning: include: has no value, DC 0 assumed
```

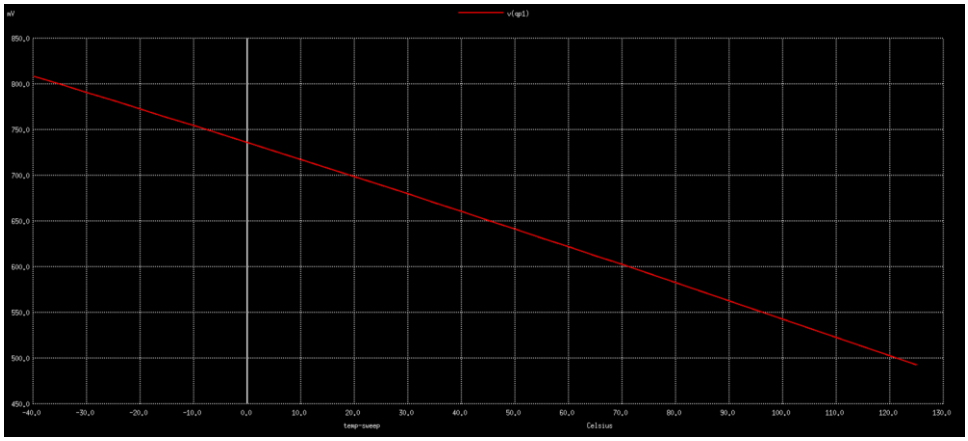
- Test case1: Id=10uA, M=1. Resultant slope=-0.0017074




```
No. of Data Rows : 34
ngspice 1 ->
x0 = -19.8864, y0 = 0.818023     x1 = 78.5227, y1 = 0.65
dx = 98.4091, dy = -0.168023
dy/dx = -0.0017074     dx/dy = -585.687
```

- Test case2: Id=10uA, M=8. Resultant slope=-0.0019138

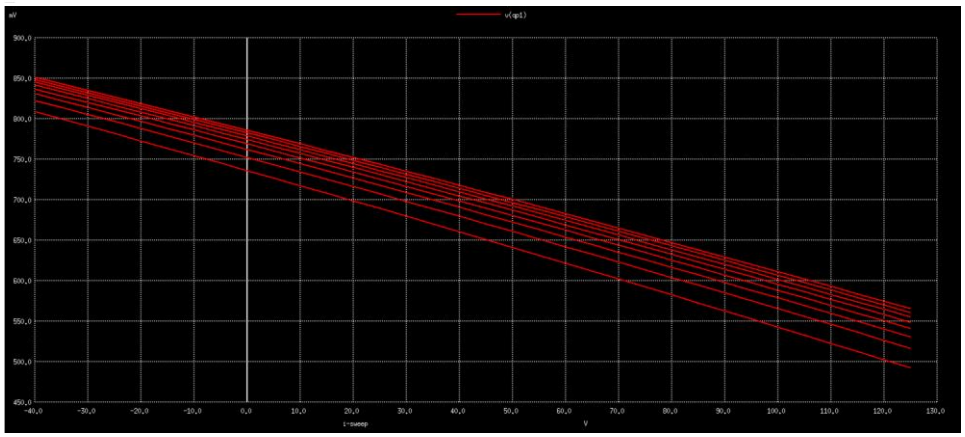





```
No. of Data Rows : 34
ngspice 1 ->
x0 = 0.113636, y0 = 0.733333    x1 = 95.9091, y1 = 0.55
dx = 95.7955, dy = -0.183333
dy/dx = -0.0019138    dx/dy = -522.521
```

- Test case3: Id=1.25u to10uA, M=1. Resultant slope=-0.0017166 to -0.00187869

```
abhijeet@bandgap-workshop:~/work$ vim ctat_voltage_gen_var_current.sp
abhijeet@bandgap-workshop:~/work$ ngspice ctat_voltage_gen_var_current.sp
******
** ngspice-27 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Please get your ngspice manual from http://ngspice.sourceforge.net/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Tue Dec 26 17:10:20 UTC 2017
******
```




```
No. of Data Rows : 272
ngspice 1 ->
x0 = 126.477, y0 = 0.776119

x0 = -19.8864, y0 = 0.770896    x1 = 70.6818, y1 = 0.600746
dx = 90.5682, dy = -0.170149
dy/dx = -0.00187869    dx/dy = -532.287

x0 = -29.8864, y0 = 0.833582    x1 = 109.659, y1 = 0.59403
dx = 139.545, dy = -0.239552
dy/dx = -0.00171666    dx/dy = -582.526
```

PART5: PTAT Voltage Generation

From Diode current equation (2) we can find that it has two parts, i.e.

- Vt (Thermal Voltage) which is directly proportional to the temp. (KT/Q)
- Is (Reverse saturation current) which is directly proportional to the temp. ($AukTni^2$), as this Is term is in denominator so with increase in temp. the ln (Io/Is) decreases which is responsible for CTAT nature of the diode.

$$I_D = I_S e^{\frac{V_D}{V_t}}$$

$$V_D = V_t \ln\left(\frac{I_0}{I_S}\right)$$

$$V_t = \frac{kT}{q}$$

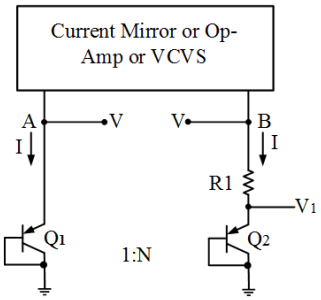
$$I_S = A\mu kT n_i^2$$

$$\mu \propto \mu_0 T^m, m = -3/2$$

$$n_i^2 \propto T^3 e^{\left[\frac{-E_g}{kT}\right]}$$

$$I_S = AT^{(4+m)} e^{\left[\frac{-E_g}{kT}\right]}$$

So, to get a PTAT Voltage generation circuit Vt needs to be separated from Is.



In the above circuit same amount of current I is flowing in both the branches. So, the node voltage A and B are going to be same V. Now in the B branch if we subtract V1 from V, we get Vt independent of Is.

$$V = V_t \ln\frac{I}{I_s} \quad and \quad V_1 = V_t \ln\frac{I/N}{I_s}$$

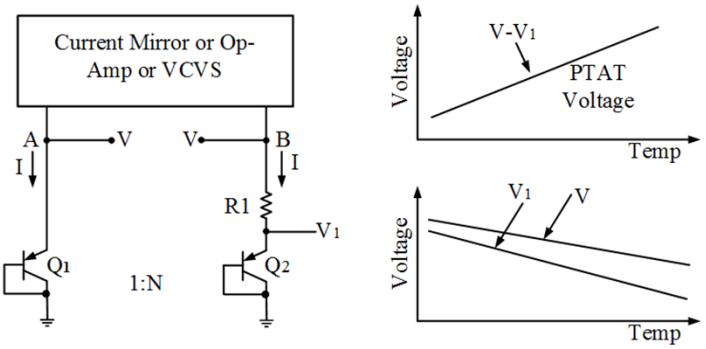
$$V - V_1 = V_t \ln(N)$$

$$V_t = PTAT \; and \; \ln(N) = constant$$

$$V_t = \frac{kT}{q}, \qquad \frac{d(V_t)}{dT} = \frac{k}{q} = 85uV/DegCent.$$

From above we can see that the voltage V-V1 is PTAT in nature, but its slope is very less as compared to the CTAT, so we have to increase the slope.

In order to increase the slope, we can use multiple BJTs as diode, so that current per individual diode will be less and it the slope of V-V1 will increase.

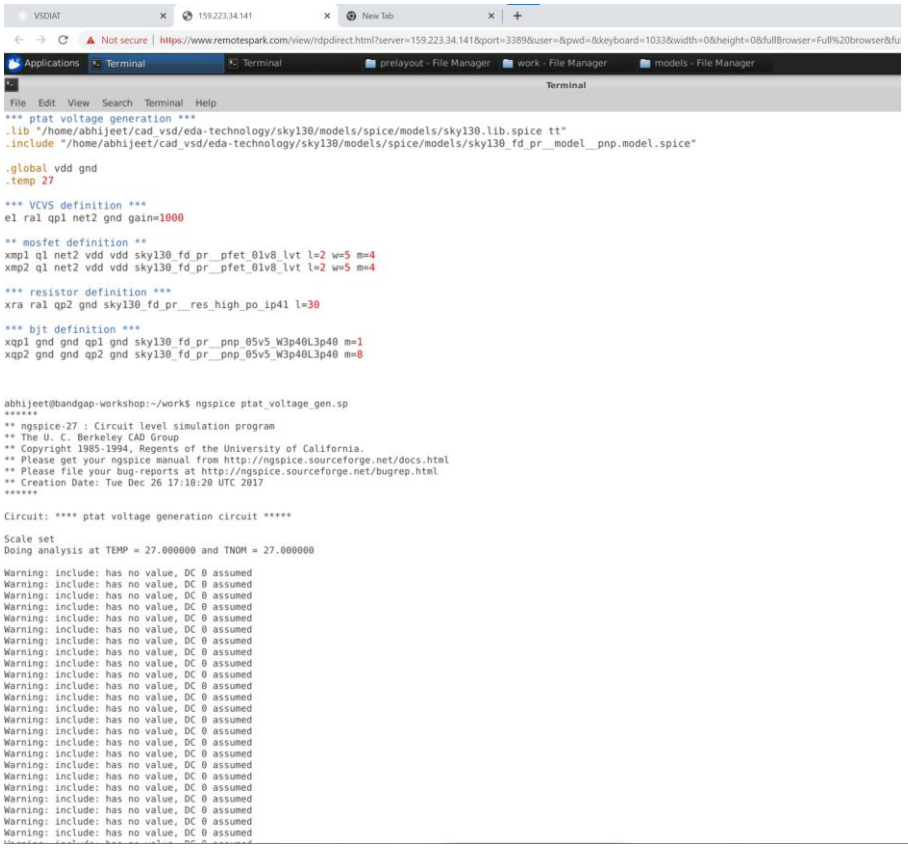


Implantation of PTAT:

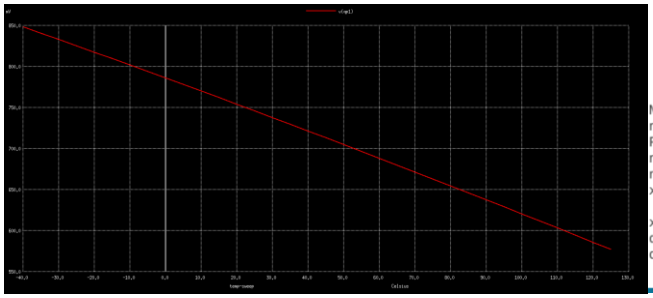
PTAT is implemented by connecting a constant current source to a diode connected mosfets in node A and B with a resistor R1 connected to node B.

Ensure to include the sky130.lib.spice TT (TT for typical- typical process) and

Sky130_fd_pr__model__pnp.model.spice

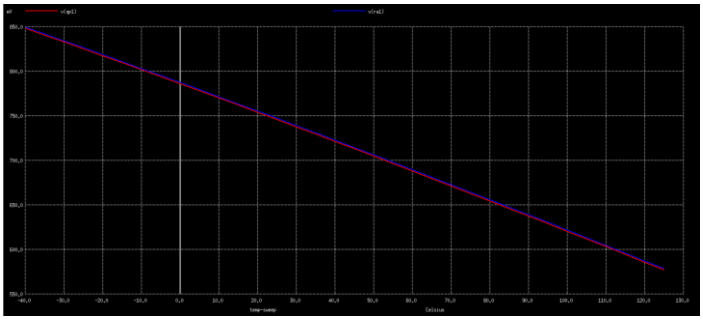


qp1 result is CTAT in nature

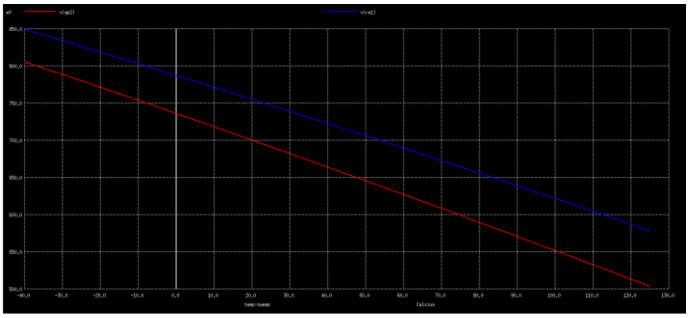


```
No. of Data Rows : 34
ngspice 1 -> plot (v(qp1)
PPerror: syntax error
ngspice 1 -> plot v(qp1)
ngspice 1 ->
x0 = 130.455, y0 = 0.64703

x0 = -8.86364, y0 = 0.8    x1 = 82.6136, y1 = 0.649505
dx = 91.4773, dy = -0.150495
dy/dx = -0.00164516    dx/dy = -607.842
```

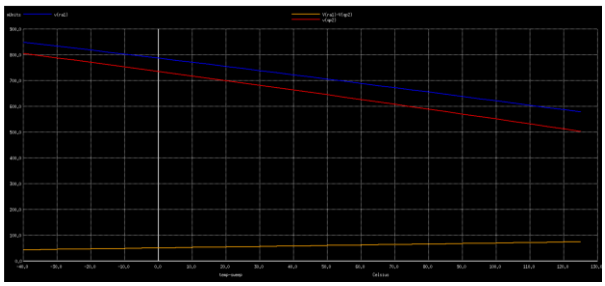
V(qp1)≈V(Ra1)

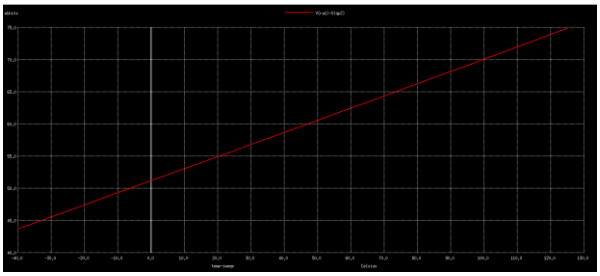


In below wave form it can be observed that V(qp2) has different slope than V(ra1)



On subtracting V(ra1) V(qp2) PTAT result is obtained, Resultant slope=0.00190636





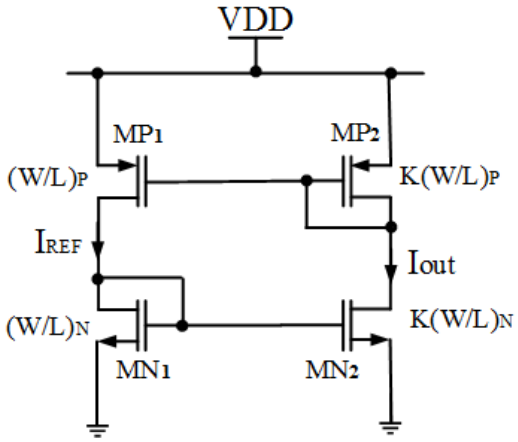
```
ngspice 1 -> plot v(qp2) v(ra1) V(ra1)-V(qp2)
ngspice 1 -> plot  V(ra1)-V(qp2)
ngspice 1 ->
x0 = -37.0787, y0 = 0.055407

x0 = -6.17978, y0 = 0.0499419    x1 = 89.8876, y1 = 0.0682558
dx = 96.0674, dy = 0.018314
dy/dx = 0.000190636    dx/dy = 5245.59
```

Part 6: Biased current mirror circuit

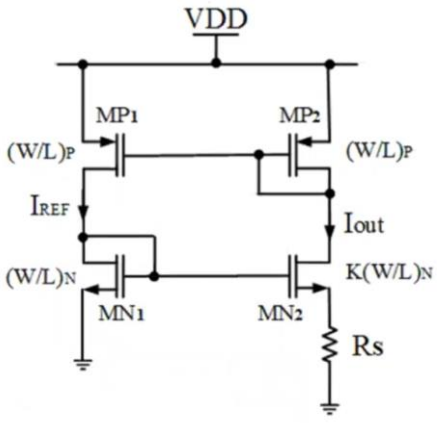
For both CTAT and PTAT design it was observed previously that a constant current source is required in order to obtain the desired slope of voltage wrt to temperature.

In a current mirror Iref sets Iout but Iout is tied closely with supply voltage. any variation in supply affects Iref and there by affecting the voltage reference. In order to improve PSRR of the system the current mirror should be able to bias itself i.e, Iref is derived from Iout.

The Self-biased current mirror is a type of current mirror which requires no external biasing. This current mirrors biases itself to the desired current value without any external current source reference.



In the above circuit MP2 and MP1 copy Iout and define Iref (Iref is bootstrapped to Iout). The device is governed by Iref=K*Iout, hence it can support any current.



Rs is used to uniquely define current, loop gain is always less than 1. i.e, the circuit is generally stable and an external circuit is need to drive the device to correct bias point.

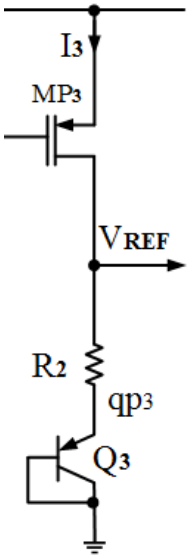
$$I_{out} = \frac{2}{\mu_n C_{ox}(W/L)_N} \cdot \frac{1}{R_S^2}\left(1 - \frac{1}{\sqrt{K}}\right)^2$$

Iout is defined by the above equation.

Part 7: Reference voltage branch circuit

CTAT, PTAT and their current source is designed. Next step is to add CTAT and PTAT to generate the required reference voltage.

The reference circuit branch performs the addition of CTAT and PTAT voltages and gives the final reference voltage. MP3 is used as a mirror transistor to keep the current same as current mirror branches, R2 is used to scale PTAT to the same value as CTAT so that they cancel each other out and a BJT as diode in the reference branch.



R2= α*R1

$$\frac{dV_{R2}}{dT} + \frac{dV_{Q3}}{dT} = 0$$

$$\frac{d\alpha * V_{R1}}{dT} + \frac{dV_{Q3}}{dT} = 0$$

$$\frac{d\,(\alpha * V_t \ln N)}{dT} + \frac{dV_{Q3}}{dT} = 0$$

$$(\alpha * \ln N)\frac{dV_t}{dT} + \frac{dV_{Q3}}{dT} = 0$$

$$\frac{dV_{Q3}}{dT} = -1.6\,mV/\deg cent.$$

$$\frac{dV_t}{dT} = 85\,uV/\deg cent.$$

Part 8: Start-up circuit

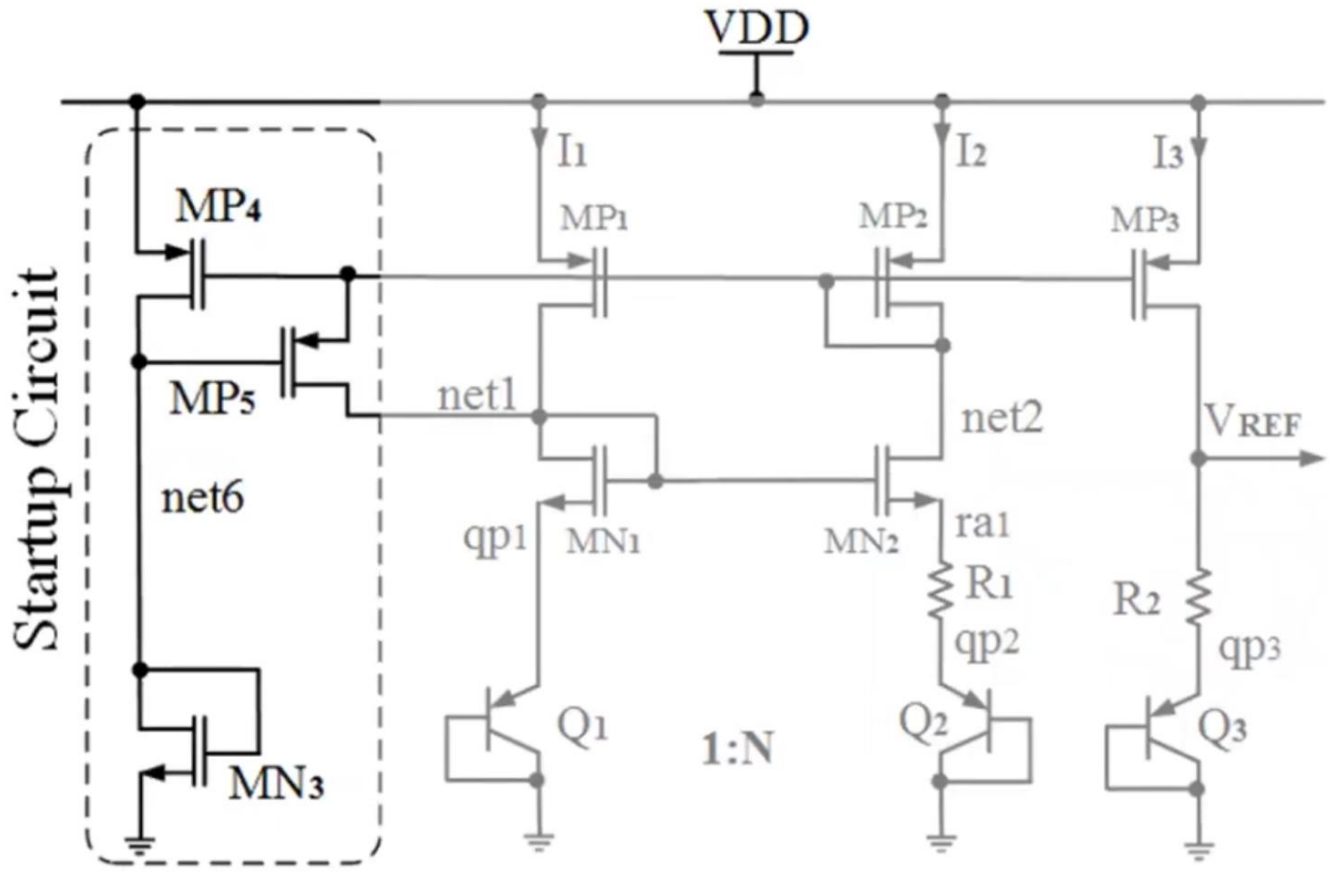
Why Start-up?

In self-biased current mirror circuit if channel length modulation is ignored there is very little dependency on the supply voltage. This leads to existence of a degenerate bias point i.e device is stable at Iin=Iout =0A. Even if the Supply is turned on there is no force to drive the currents and therefore voltage to the required operating point.

A circuit is added to drive the device into the desired operating point and once device operation reaches steady state the circuit should no longer interfere.

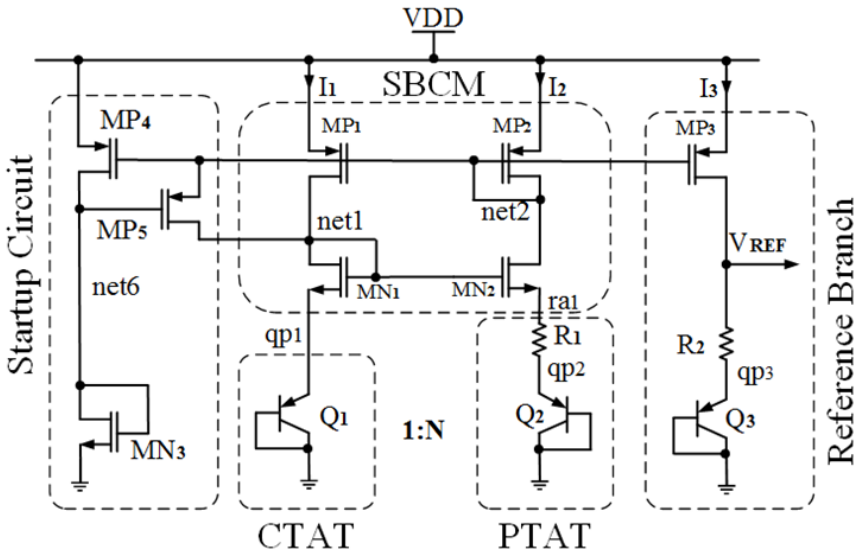
Implementation of Start-up:

In the below circuit net2 follows VDD. As net2 voltage rises greater than the threshold voltage with reference to net6 MP5 turns on there by allowing current through itself to net1. This regulates the net2 voltage and once device turns on net 6 remains at high voltage and there by isolating itself after turn on.



Part 9: Complete BGR circuit

After implementing all the blocks. The blocks are connected to get the complete BGR circuit.



BGR Circuit Design

1. Current Calculation

- Max. power Consumption < 60uW
- Max Total Current = 60 uW/1.8V=33.33uA (1.8V VDD)
- So, we have chosen 10uA/branch, (3*10=30uA)
- Start-up current 1-2 uA

2. Choosing Number of BJT in parallel in Branch2

- Less number of BJT: require less resistance value but matching hampers
- More number of BJT: requires higher resistance value but gives good matching
- So a moderate number have chosen (8 BJT) for better layout matching and moderate resistance value.

3. Calculation of R1

- R1= Vt* ln (8)/I =26 mv *ln(8)/10.7uA=5 KOhm
- R1 size: W=1.41um, L=7.8um, Unit res value: 2k Ohm
- Number of resistance needed: 2 in series and 2 in parallel (2+2+(2||2))

4. Calculation of R2

- Current through ref branch:I3=I2=Vt*ln(8)/R1
- Voltage across R2: R2*I3=R2/R1(V*tln(8))
- Slope of VR2= R2/R1 (ln(8)*115uv)/Deg Cent.

- Slope of VQ3=-1.6mV/Deg cent
- Adding both and equating to zero, R2 will be around 33k Ohm
- Number of resistance needed: 16 in series and 2 in parallel (2+2...+2+ (2||2))

5. SBCM Design (Self-biased Current Mirror)

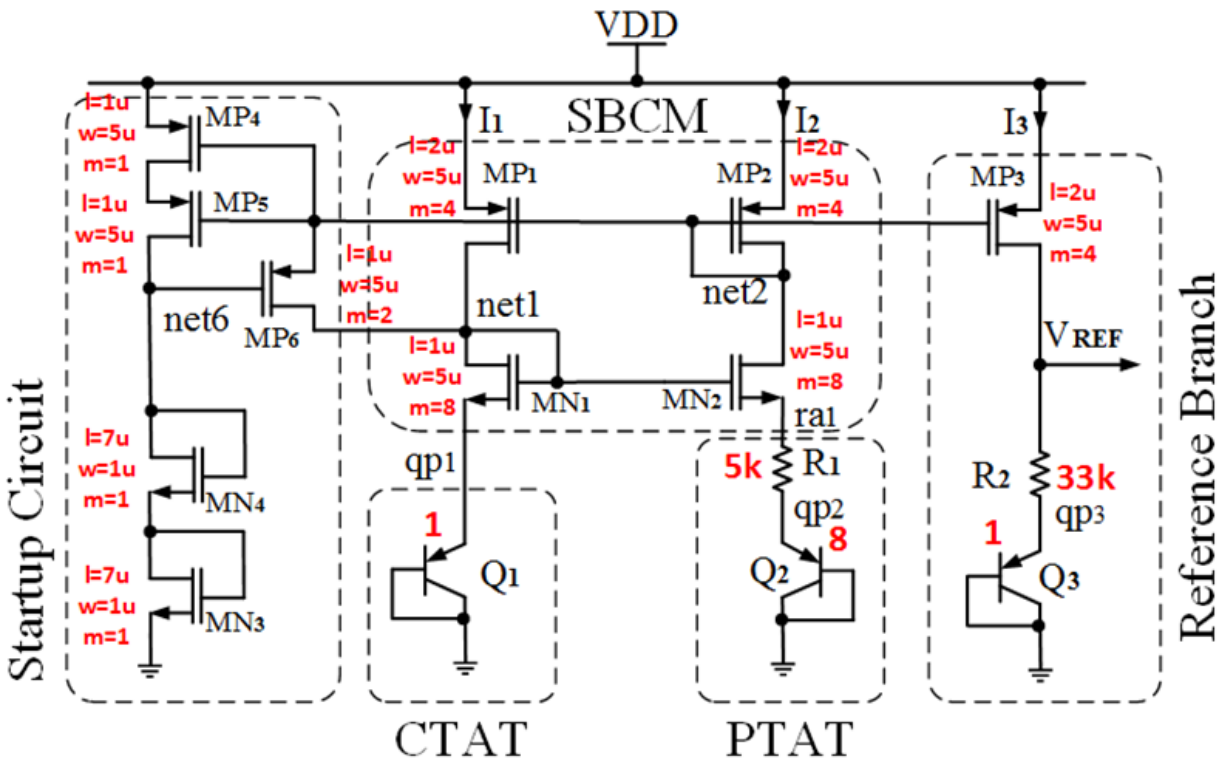
A. PMOS Design in SBCM

- Make both the MP1 and MP2 well in Saturation
- To reduce channel length modulation used L=2um
- Finally the size is **L=2u, W=5u and M=4**

B. NMOS Design in SBCM

- Make both the MN1 and MN2 either in Saturation or in deep sub-threshold
- We have made it in deep sub-threshold
- To reduce channel length modulation used L=1um
- Finally the size is **L=1u, W=5u and M=8**

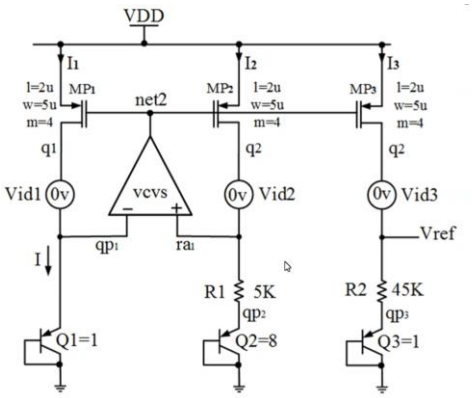
Final Circuit



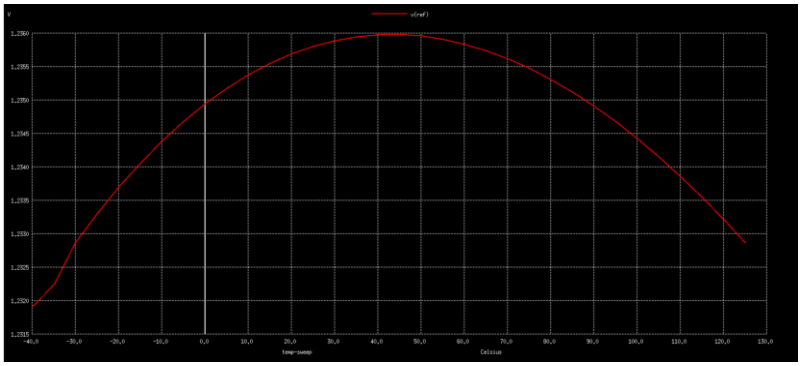
Implementation:

BGR using Ideal OpAmp

After all blocks are done. BGR with VCVS or opamp is tested.

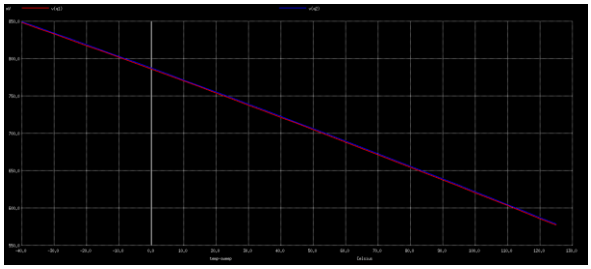


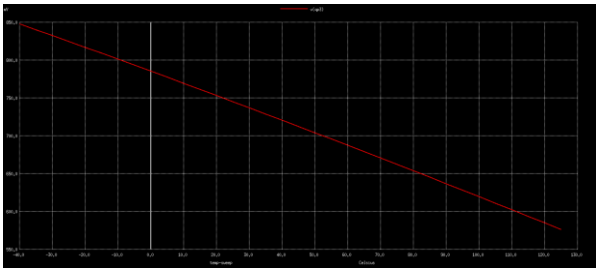

```
*** vcvs definition
e1 net2 gnd ra1 qp1 gain=1000

xmp1    qp1     net2    vdd     vdd     sky130_fd_pr__pfet_01v8_lvt     l=2     w=5     m=4
xmp2    ra1     net2    vdd     vdd     sky130_fd_pr__pfet_01v8_lvt     l=2     w=5     m=4
xmp3    ref     net2    vdd     vdd     sky130_fd_pr__pfet_01v8_lvt     l=2     w=5     m=4

*** bjt definition
xqp1    gnd     gnd     qp1     gnd     sky130_fd_pr__pnp_05v5_W3p40L3p40        m=1
xqp2    gnd     gnd     qp2     gnd     sky130_fd_pr__pnp_05v5_W3p40L3p40        m=8
xqp3    gnd     gnd     qp3     gnd     sky130_fd_pr__pnp_05v5_W3p40L3p40        m=1

*** high-poly resistance definition
xra1    ra1     na1     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xra2    na1     na2     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xra3    na2     qp2     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xra4    na2     qp2     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8

xrb1    ref     nb1     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb2    nb1     nb2     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb3    nb2     nb3     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb4    nb3     nb4     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb5    nb4     nb5     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb6    nb5     nb6     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb7    nb6     nb7     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb8    nb7     nb8     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb9    nb8     nb9     vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb10   nb9     nb10    vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb11   nb10    nb11    vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
xrb12   nb11    nb12    vdd     sky130_fd_pr__res_high_po_1p41      w=1.41      l=7.8
```



In this simulation we should get the reference voltage as an umbrella shaped curve.

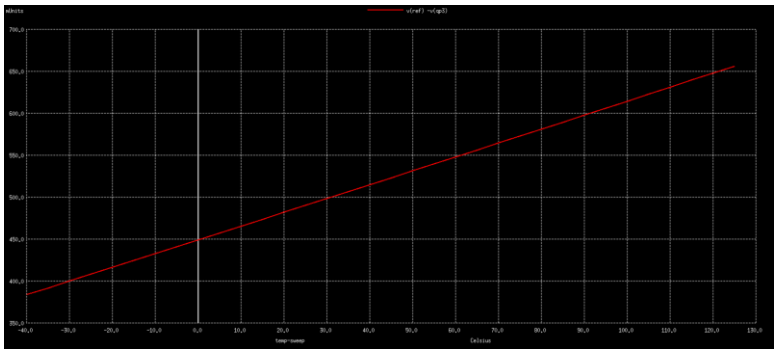
CTAT:





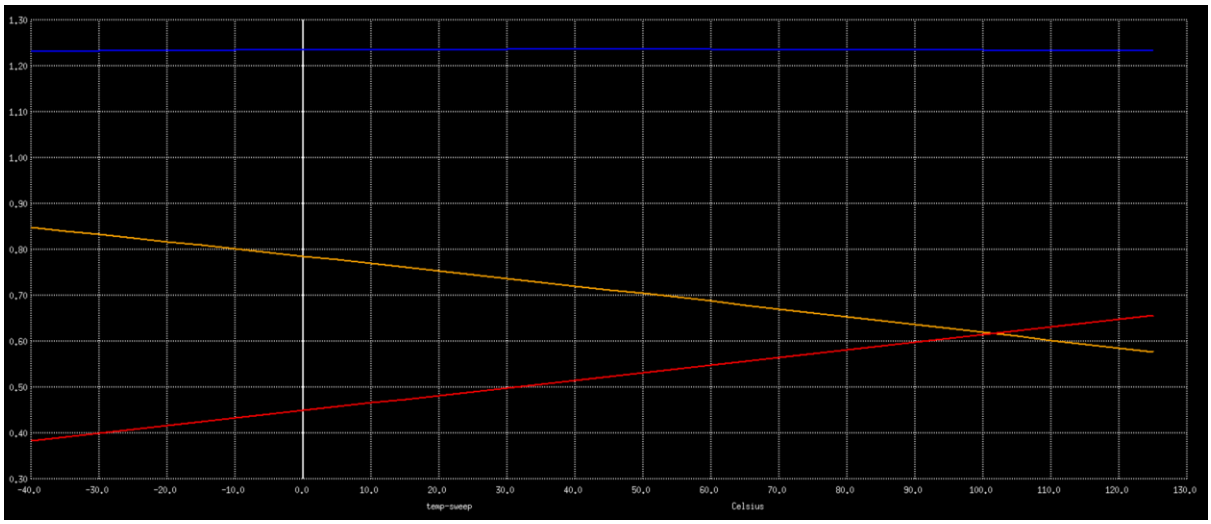
```
ngspice 1 -> plot vid1#branch vid2#branch
ngspice 1 -> plot v(qp3)
ngspice 1 ->
x0 = -35.4545, y0 = 0.74505

x0 = -19.8864, y0 = 0.815842    x1 = 82.2727, y1 = 0.649505
dx = 102.159, dy = -0.166337
dy/dx = -0.00162821    dx/dy = -614.171
```

PTAT:



```
ngspice 1 -> plot v(ref) -v(qp3)
ngspice 1 ->
x0 = -32.6136, y0 = 0.523256

x0 = -0.113636, y0 = 0.448256    x1 = 90.5682, y1 = 0.6
dx = 90.6818, dy = 0.151744
dy/dx = 0.00167337    dx/dy = 597.597
```

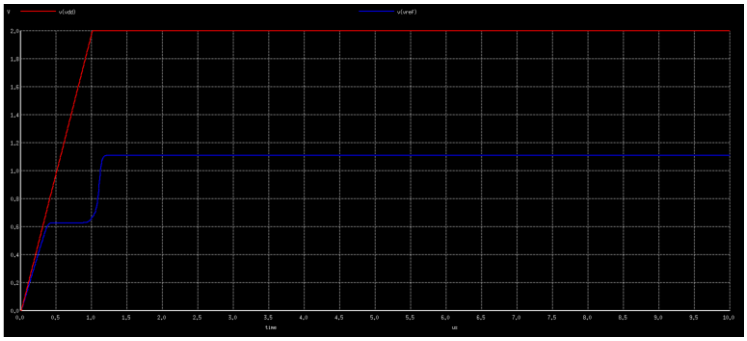
CTAT + PTAT = VREF



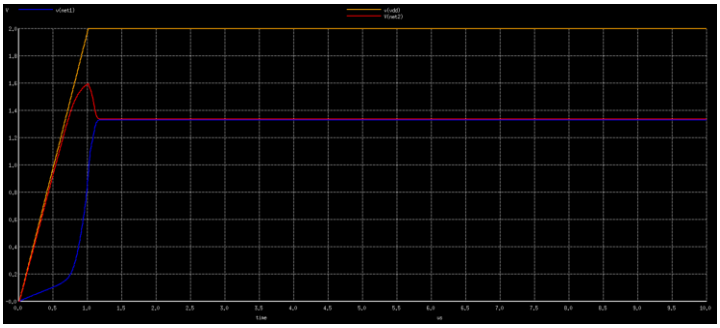
BGR with Start-up (SBCM)

With SBCM it is necessary to implement start-up circuit. Transient analysis is run to observe start-up behaviour of the circuit.

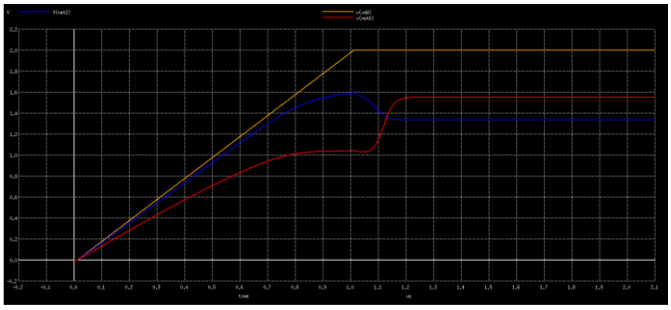
Vref raises as VDD increases and it can be observed start-up time is ≈1.25us which is < 2us specification.



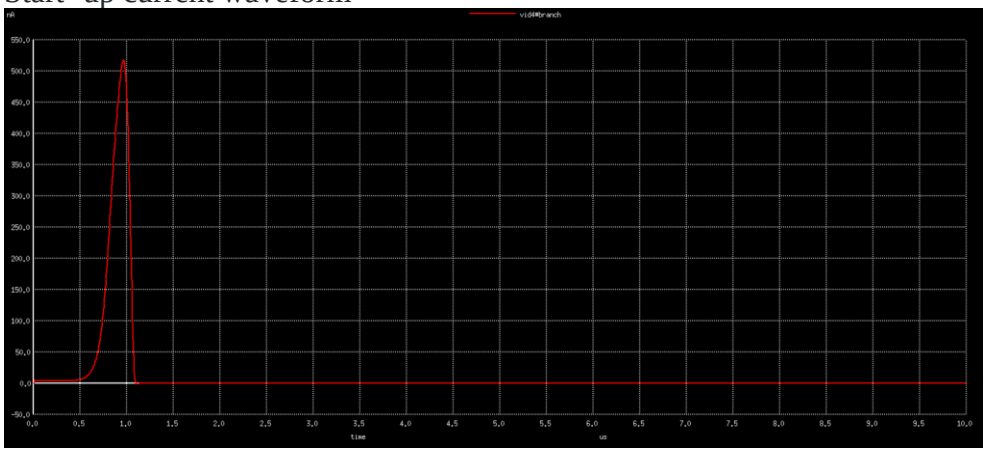
Voltage changes in net1 and net2, it can be observed during start up net2 follows vdd and after start up both settle at regulated value.



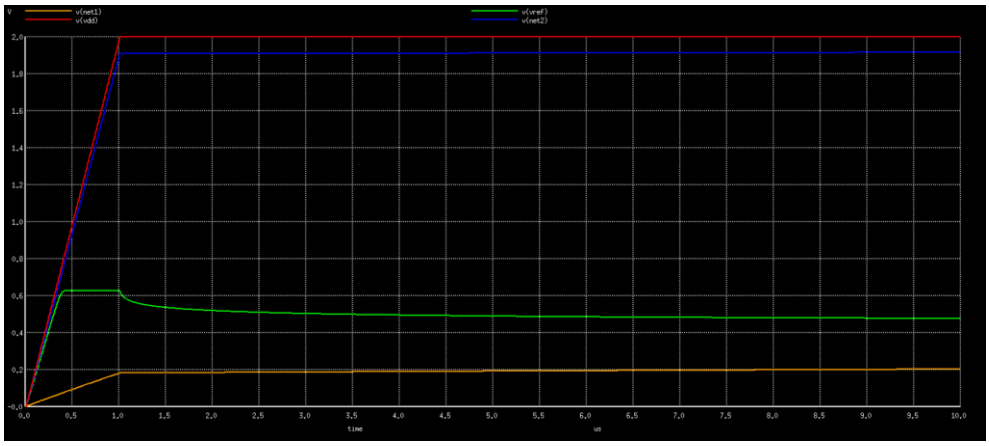
Below waveform shows how on net6 goes high and turns off startup circuit once device reaches desired operating point.



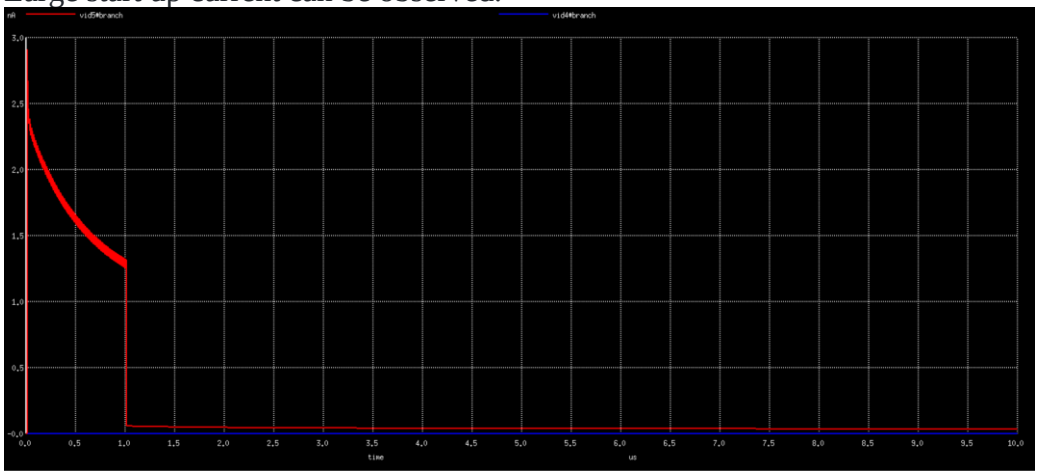
Start -up current waveform



To see effects of not using a start-up circuit, Start-up block was disable in SBCM configuration.



It can be observed that net 1 doesn’t go above 0.2V and Vref remains at a lower stable point. Large start up current can be observed.



Part 10: Layout of components

Now after getting our final netlist, we have to design the layout for our BGR. Layout is drawing the masks used in fabrication. We are going to use the Magic VLSI tool for our layout design.

Getting started with Magic

Magic is an open-source VLSI layout editor tool. To launch magic open terminal and write the following command.

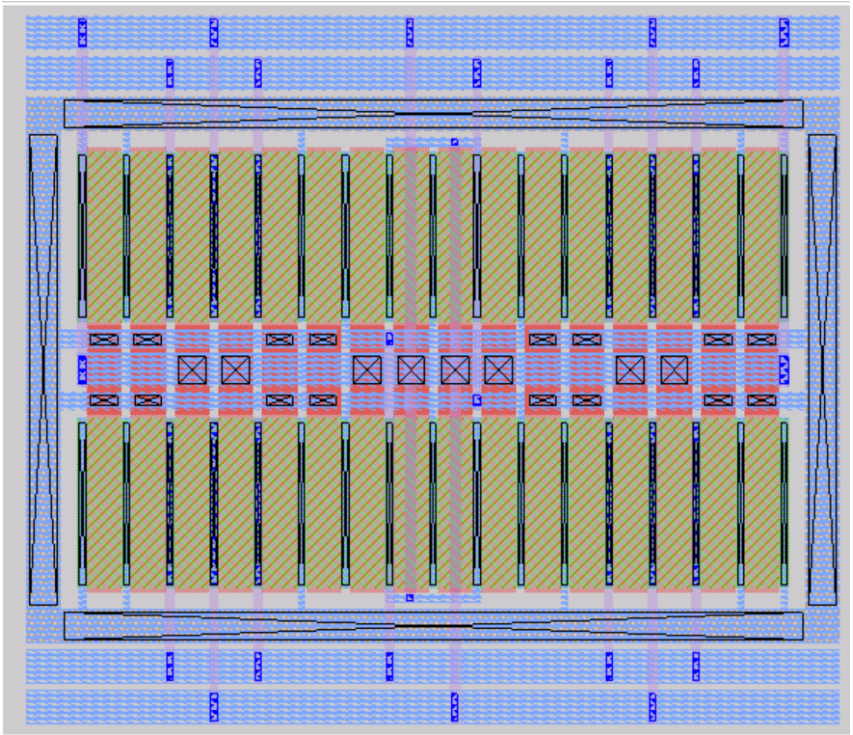
$ magic -T /home/<path for sky130A.tech>/sky130A.tech
Now it will open up two windows, those are tkcon.tcl where we design and edit our Layout

Now device wise we have the following devices in our circuit.

- PFETS
- NFETS
- Resistor Bank
- BJTs

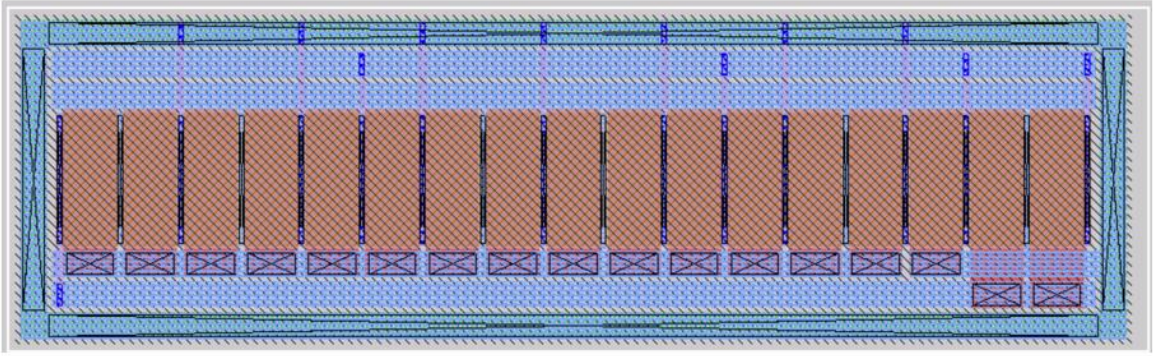
Design of NFET

In our circuit we are using LVT type NFETs. placed the nfets in such a way that it follows common centroid matching



Design of PFET

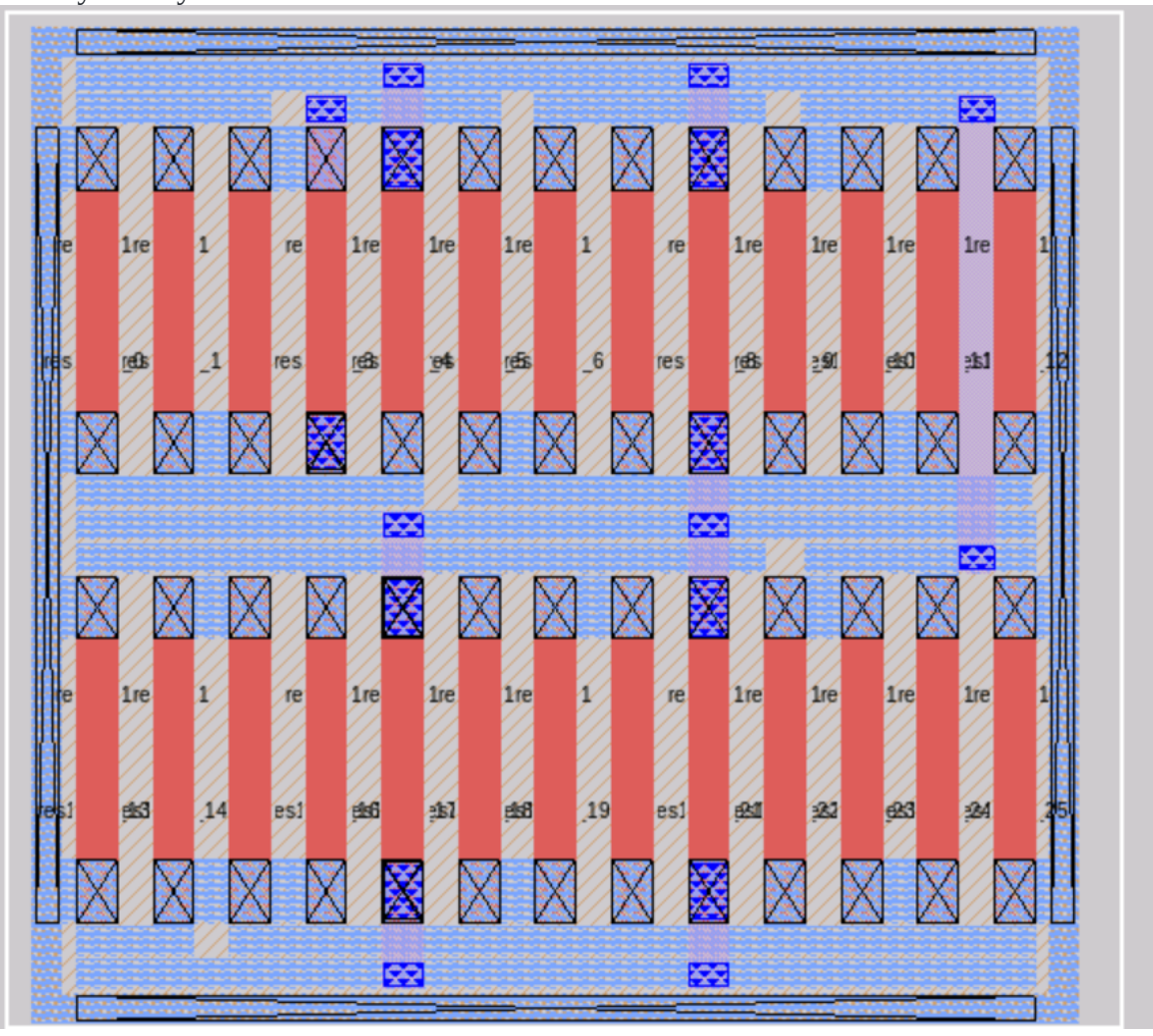
In our circuit we are using LVT type PFETs. These are matched from centre.



Design of Resistor

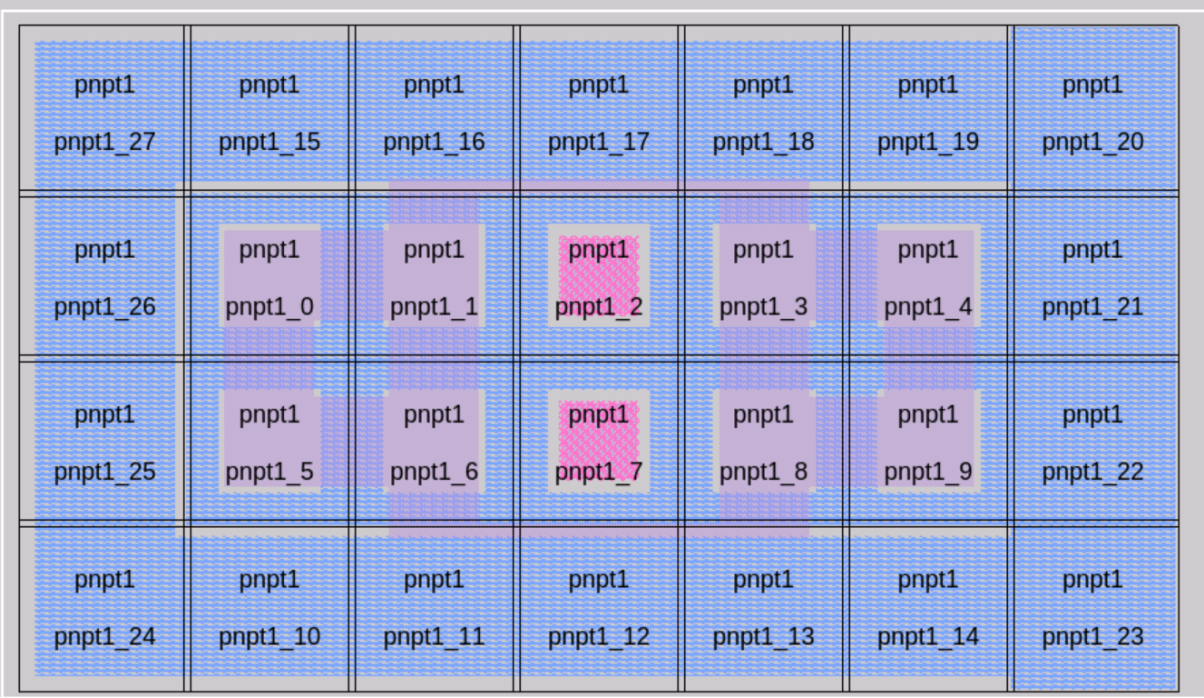
In our desing we are using poly resistors of W=1.41 and L=7.8.

9 poly resistors form R2 and 2 on corners form R1 and 2 dummy are placed. And matched with symmetry at the centre.



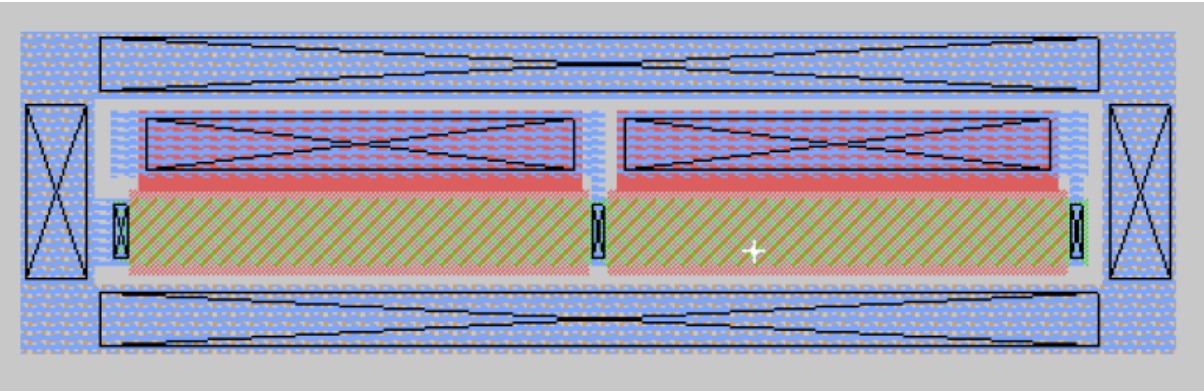
Design of PNP (BJT)

PNPT1_2 is Q1 and PNPT1_7 is Q3 at centre ,4 transistor on left and right are 8 Q2 rest are dummy.



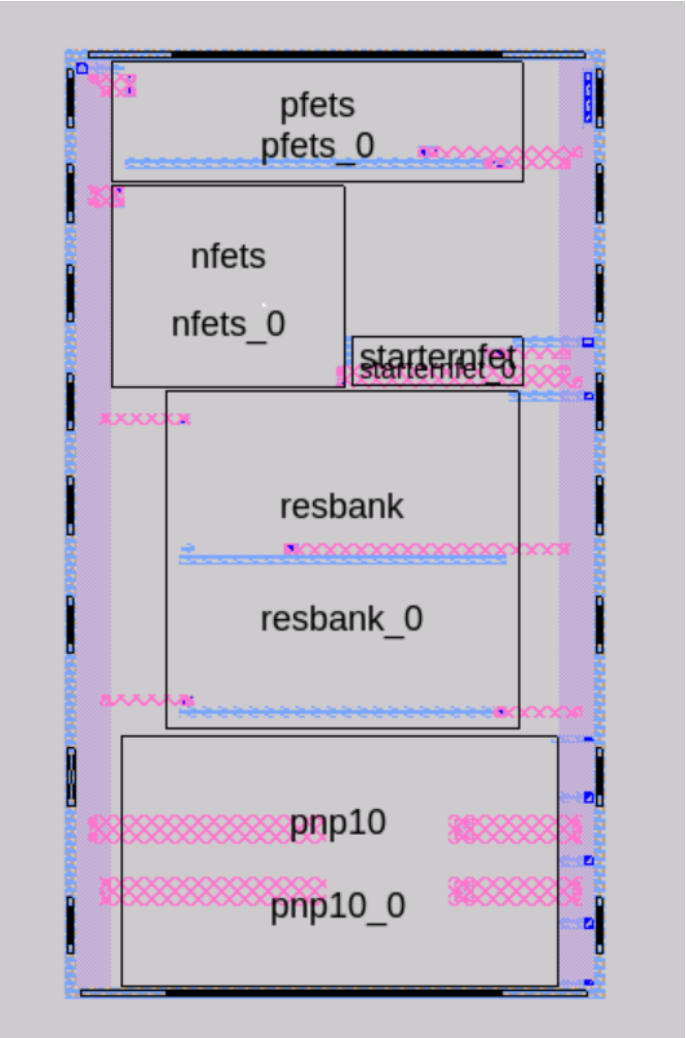
Design of STARTERNFET

We placed the the two w=1, l=7 NFETs



Top level design

To obtain the top level design, we have placed all the blocks together, routed it.

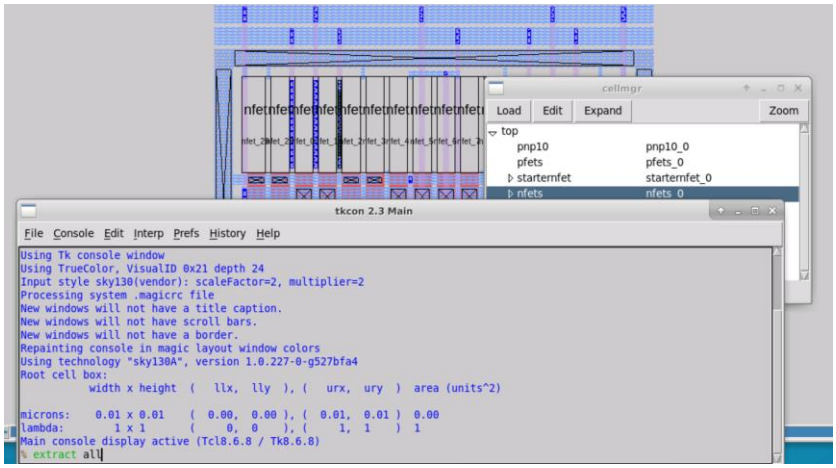


Part 11: Top level extraction and LVS

In order to do the parasitics extraction follow the steps given below:

- Open the Layout top.mag file >Options>Cell manger> select individual block and click open
- In the magic command window type extract, all. This extracts the layout connectivity information into a ".ext." file.
- Generally, if there are any warnings at this point, they would be because of wrong connections or short circuits or etc. But these are warnings and need not be errors.
- Now, we need to convert the ".ext" file to a ".spice" file to use for simulations.

- Use following commands:
    - Ext2sim label on
    - Ext2sim
    - Ext2spice scale off
    - Ext2spice hierarchy off
    - Ext2spice
- Run the above command for all the blocks and then finally on top level.



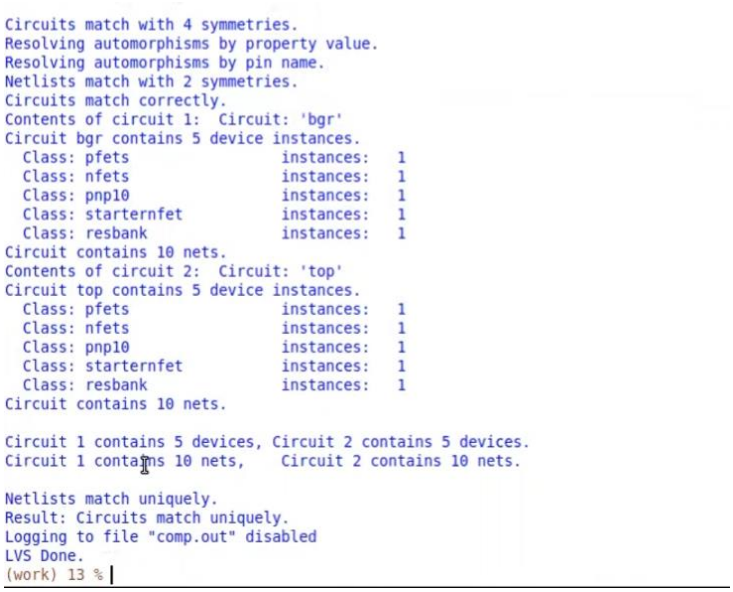

```
microns:    0.01 x 0.01    (  0.00,  0.00 ), (  0.01,  0.01 )  0.00
lambda:        1 x 1       (     0,  0    ), (     1,  1    )  1
Main console display active (Tcl8.6.8 / Tk8.6.8)
% extract all
Extracting nfet into nfet.ext:
Extracting nfets into nfets.ext:
% ext2sim label on
% ext2sim
exttosim finished.
% ext2spice scale off
% ext2spice hierarchy off
%
% ext2spice
exttospice finished.
%
```

```
* SPICE3 file created from nfets.ext - technology: sky130A

.subckt nfets net1 qp1 gnd rp1 net2
X0 rp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X1 gnd gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X2 gnd gnd qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X3 gnd gnd rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X4 qp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X5 rp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X6 net2 net1 rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X7 net1 net1 qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X8 rp1 net1 net2 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X9 qp1 net1 net1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X10 gnd gnd qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X11 net1 net1 qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X12 gnd gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X13 qp1 net1 net1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X14 gnd gnd rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X15 gnd gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X16 gnd gnd rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X17 qp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X18 qp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X19 gnd gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X20 net1 net1 qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X21 qp1 net1 net1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X22 net1 net1 qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X23 qp1 net1 net1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X24 gnd gnd qp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X25 rp1 gnd gnd gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X26 net2 net1 rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X27 rp1 net1 net2 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X28 net2 net1 rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X29 rp1 net1 net2 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X30 net2 net1 rp1 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
X31 rp1 net1 net2 gnd sky130_fd_pr__nfet_01v8_lvt ad=0p pd=0u as=0p ps=0u w=5e+06u l=1e+06u
C0 qp1 net1 9.71fF
C1 rp1 net2 7.43fF
C2 rp1 net1 4.06fF
```

Observe the parasitic added.

LVS:

To run LVS open NETGEN and then type LVS “TOP.sp bgr” “TOP.spice top ” /home/Abhijeet/cad_vds/eda-technology/sky130/tech/netgen/sky130_setup.tcl and run the LVS



And you should get netlist match uniquely.

Post layout Simulation:

In the top-level file include all the individually extracted spice file of all the block and then run ngspice simulation to obtain post layout result with effects of parasitic.

Conclusion and Opinion

The workshop is well paced program with good details on how to approach and start analog design. The hands-on experience is valuable in order to understand and retain the knowledge shared over two days. Step by step approach with connectivity between each module helps in learning concept from every angle. Would have preferred to have spent more time in understanding the calculation and analysis to determine the value of all the components such as resistors, BJT and FET's. Over all good experience and insight into availability of opensource tools to further learn more concepts

References

- https://github.com/ VrushabhDamle/sky130PLLdesignWorkshop
- https://github.com/google/skywater-pdk-libs-sky130_fd_pr.git
- https://www.vsdiat.com/
- https://www.vlsisystemdesign.com/
- https://github.com/silicon-vlsi-org/eda-technology/
- https://github.com/vsdip/vsdopen2021_bgr
- http://opencircuitdesign.com/magic/download.html
- http://opencircuitdesign.com/magic/