EE214a Project

Amy Yen

Matthew Feldman

Table : Summary of Specs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Gain** | **Bandwidth** | **Power** | **G\*BW/Power** | **CM Output Voltage** |
| **Required** | 30.000 kΩ | 90.0000 MHz | 2.0000 mW |  | [-150 mV, 150 mV] |
| **Achieved** | 30.019 kΩ | 96.6133 MHz | 1.9444 mW |  | -114.45 mV |

Table : Summary of Areas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Core Circuit** | **Bias Circuit** | **VPMOS-bias** | **VNMOS-bias** |
| **Area** | 134.236 µm | 493.4545 µm | 46 µm | 54 µm |
| **% of Core Circuit** | 100% | 367.6% | 34.3% | 42.5% |

Page 0

Page 1

We started by identifying the following key blocks of the circuit.

1. Common Gate Input stage (Mn1, R1, R2)
   1. Provides a low input impedance to capture the input current generated by the Capacitive Micromachined Ultrasonic Transducer (CMUT).
   2. Provides a large trans-impedance gain over the parallel combination of R1 and R2 that converts the input current to a voltage
2. Cascode Gain Stage (Mp4, Mp5, R3, R4)
   1. A CS transistor connected to a CG transistor, both of which are sized such that the Miller Effect is minimized, which speeds up the input node.
   2. Achieves high gain by the parallel combination of R3 and R4
3. Common Source Stage (Mn7, Mp8)
   1. Provides a buffer between the high output-impedance of the cascade stage and the high input impedance of the final CD stage.
   2. Provides a small amount of gain, based on the size of Mp8, which acts as a resistor in small signal
4. Common Drain Stage (Mp10)
   1. Provides a low output impedance to drive the 20kΩ, 250fF load.

Additionally, we noticed that the circuit was divided into four transistor “stacks” and two resistor “stacks” and attempted to design the circuit assuming that no current leaks between them. With this assumption, we know that we can treat the resistor stacks as voltage dividers to set the overdrive voltages on Mp4 and Mn7. We can also balance Mp3 and Mn1 to set the current through Mn2. Similarly, we can balance the current in Mn6 with the current set through Mp4 so that Mp5 operates in a certain range.

Finally, we recognized that there were three properties that we needed to consider to figure out how to size all of the transistors: allocation of power between stacks, allocation of gain between stages, and allocation of time constant between nodes. We tried designing the circuit by hand, using a lot of approximations, with different combinations of two of these properties.

We first used power allocation and gain allocation to design the circuit by hand. This means allocating how much current each stack should carry, as well as the gain we want from each stage using the approximations shown later in this report, in Table 4.

After computing the design parameters of the circuit, we put it in to HSPICE and noticed that the results did not look like what we expected. With a small amount of directed adjustments, we were able to get all of the transistors in saturation, but were still not matching with our expected results.

We then tried using the gm/Id approach to fix the expected performance of each transistor in the circuit, and set the overdrives accordingly. (stuff that Amy did)

Finally, we tried calculating the valid ranges for node and bias voltages in DC and what resistor ratios would give us these values. We also used intuition on how the transistors should be sized to give us the best bandwidth, and used a script to sweep over a variety of possible designs to look for one that gave good performance.

From the three methods above, we identified “candidate” designs that we were able to use as seeds for optimization until we finally got a circuit that met all of the specs

Page 1

Page 2

Page 2

Pages 3-6

Gm/id design methodology

1. Estimated equations for gain and time constant in each stage

|  |  |  |
| --- | --- | --- |
| Node | Time Constant | Assumptions/Simplifications |
| 1 - Input | ~ | * Cin >> Cgs2 |
| 2 - Vx |  | * R2/R1 < 4 |
| 3 - Vy |  | * R3/R4 < 4 |
| 4 - Vz |  |  |
| 5 - Vout |  | Cgs10 << , >> 1 => tau = |

|  |  |  |
| --- | --- | --- |
| Stage | Gain1 | R in terms of gain |
| 1 - CG |  | R2 = 4R1  = gain  => = = gain  => R1 = 5/4\* gain |
| 2 – Cascode |  | R3 = 4R4  = 5  => = 5  => = 5  => R4 = |
| 3 – CS |  |  |
| 4 – CD |  |  |

1. Estimate R1,R2,R3,R4

To find R3 and R4

N\_out = 0V and Vov > 0.15 from spec

Vth10 ~ 1V from playing with spice

* Vg10 = Vov + Vth10 = 0.15 + 1 = 1.15 🡪 assume 1.2V for some margin

Ids7 = Ids8 =>

1/2\*kn\*(W7/L7)(Vy-Vss-Vt)^2 = 1/2\*kp\*(W8/L8)(Vdd - Vz - Vt)^2

Let L7 and L8 be min 1u, and W7 and W8 be 1u also to mitigate miller effect

=> solve for Vy = -1.43

Vy = Vss \* (R3/(R3+R4)) 🡨 Vy = voltage divider of Vss assuming no current through R

=> -1.43 = -2.5 \* (R3/(R3+R4))

=> R3/(R3+R4) = 0.572

=> R3/R4 = 1.346

Through iterations, found R3/R4 = 90k/68k yielded the best result

To find R1 and R2:

Vg2 = 0, Vov2 = -Vs2 – Vth2 > 0.15 => Vs2 < -0.15 - 0.5 = -0.65

Sweep 0 < Vx < 1.8:

< 1.8 or else M4 won't have enough overdrive: Vov4 = Vdd - Vx - Vt4 > 0.15 => Vx < 2.5 - 0.5 - 0.15 = 1.85

> -0.5 or else M2 goes into triode region: Vds2 > Vov2 => Vx – Vs2 > 0.15

=> Vx > 0.15 + Vs2 = 0.15 - 0.65 = -0.5

But Vx can’t be lower than 0 b/c the voltage divider cannot go lower than ground

=> 0 < Vx < 1.8

Through iterations, found Vx = 0.9V best

=> Vx = Vdd \* (R1/(R1+R2))

=> 0.9 = 2.5 \* (R1/(R1+R2)) => (R1/(R1+R2) = 0.36

=> R1 = 0.36R1 + 0.36R2 => 0.64\*R1 = 0.36\*R2 => R1/R2 ~ 0.6

Through iterations, found R1/R2 = 9.4k/15.4k = 0.6 best

1. Start with output stage. Pick gm/id of 8 (8-10 valid range, 8 is the best one from iterations) for MN10 based on the gm/id v.s. ft and v.s. gain graphs presented in discussion section. Time constant ~ gm10/Cl = allocated 110% of target 90MHz -> solve for gm10. From gm10 and gm/id of 8, solve for id10. From gm and Id, solve for Vov10. Pick min L of 1u for MN10 (CD gain stage). From L, Vov, and id, get W10.
2. MN9 has same Id as MN10 (b/c within same branch). Since MN9 is a bias xtor, pick small L (2u) and min W (2u). Calculate Vov9 = sqrt((2\*Ids9)/(kp\*(W9/L9))).
3. Next tackle input stage. Pick gm/id of 3 for MN2 (small gm/id for this stage b/c this is a CG with close to unity current gain). Time constant on the input node ~ Cin/gm2 = allocated 6% of total 110Mhz -> solve for gm2 = Cin \* 110Mhz/0.06. From gm2 and gm/id of 3, solve for id2. From id and gm, solve for Vov2. Pick min L of 1 us for MN2 (CG gain stage). From L, Vov, and id, get W2.
4. Id2 = Id1 = Id3 (within same branch).
   1. Vov1 = Vov9 -> find W1 from Id1 and Vov1
   2. For MP3, since it’s a bias xtor, pick small w=2u and small L=1u. From Id3, w3, and l3, we can solve for Vov3 (how much voltage needed for pmos biasing)
5. 2nd stack, MN6 is a bias xtor. Pick l6=2u, w6=6u (big enough to match our allocated Ids %). Vov6 = Vov9. From Vov, l, and w, solve for Id6.
6. Id4 = Id5 = Id6
   1. For MP4, pick Gm/id=5 (smaller compared to last stage, 5 is the best pick through iterations) -> find gm4 from Id4 and gm/id. From Id and gm, find Vov4. Pick L4=1u. From L, Vov, Id, solve for w4.
   2. For MP5, we want gm5 ~ gm4 (both in cascade). From gm and Id, solve for Vov. Pick L5=1u. From L, Vov, and Id, solve for w5.
7. Third stack Id = remaining Id after what the other 3 branches + 4 R’s have already used up = Id8 = id 7
   1. For MP8, we know its Vg8 = Vov10. So Vov8 = Vs8 – Vg8 – Vt = Vov10 – Vdd – Vt. Pick L8=1u. From L, Id and Vov, solve for w8.
   2. For MN7, pick gm/id = 5 (assume stack3+2 have comparable allocation). With Id know, solve for gm7. From gm and Id, solve for Vov7. Pick L=1u. From L, Vov, Id, solve for w7.

Source of error on Ids stack

1) stack4 % error due to over simplification of tau = Cl/gm equation.

2) stack1 and stack2 both had some current leaked to R1-R4; so not all xtor’s Ids within the same stack are the same.

We assumed no current leaking into R1/R2 stack. So bias voltage for MP4 is just the voltage divider set by Vx. But this turned out to be not entirely accurate in simulation, so MP4 ended up with inaccurate biasing voltage. -> also explains the %error on MN7’s gm + W (same argument of Vy divided over R3/R4 for MN7 biasing – worse yet b/c R3/R4 bigger)

3) stack3 = Ids\_total\_for\_xtor – Ids\_stack4 – Ids\_stack2 – Ids\_stack3. So the prior % error is carried over here.

Source of error on mosfet’s Vov

1. MP3: initial pick of 2u for speed but resulted in a lot of current leaking to R stack. Enlarged W3 until minimum leakage.
2. Usage of equations such as: Vov = (2\*Ids)/gm and Vov = (sqrt(2\*Ids)/(kp\*(W/L)) assume xtor is in saturation already. The initial hand calculation plugged directly into spice showed a few xtors in linear state. Had to make manual adjustment to put all xtors in saturation state.

Source of error on mosfet’s gm: gm/id selection based on nmos’ gm/id x ft + gm/id x gmro graph is rough estimate. Error can result from either the graph itself or our selection point.

Source of error on mosfet’s W: usage of this equation, W = (2\*Ids\*L)/(kp\*Vov^2) assumes xtor is in saturation already. Some turn out to be in linear region; need to make adjustment to put into saturation.

Pages 3-6

Page 7

Page 7

Page 8

Page 8

Page 9

This project proved to be a challenging, many-hour task. In order to find a working design for the circuit, we went through numerous stages that flowed from analysis to confusion to discovery to frustration and repeat.

One of the valuable take-aways from this project was what circuit design is actually like. Up to this point in our academic career, we have only taken circuits classes that focus on analysis rather than design. This was our first experience that required us to use analysis as a tool in the feedback loop of the design process. We now have a deeper appreciation of how analysis simplifications and assumptions can turn a vastly complex circuit into fundamental components that can be managed by hand. Furthermore, while these assumptions help to get us a first order design, we also learned to appreciate their limitations and the more nuanced effects that cause the hand-calculated design to differ from what happens in the actual circuit when all effects are taken into account.

However, one of the drawbacks of the project was that, at times, it felt like the techniques that were taught in class did not really help too much with design, except for having some intuition on which knobs control what in a circuit. There seemed to be a lot of “educated” tweaking that we used to fix the circuit, but could not replicate the exact numbers on paper because of channel length modulation, body effect, loading effect, and other non-ideal factors. While the homework throughout the course broke parts of this design process into fundamental components, there still seemed to be a huge gap between those problems and this design project.

Page 9

Appendix I

Appendix I

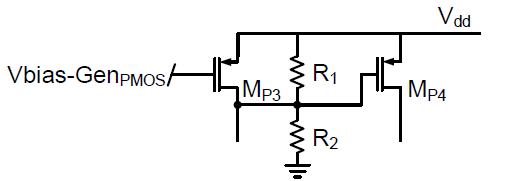
Next, we looked at the valid voltage ranges for each of the nodes in DC, considering that no transistor should have an overdrive voltage less than 0.15V.

1. V\_in
2. V\_x
3. V\_y
4. V\_z
5. -0.15 < V\_o < 0.15 (based on circuit specifications)

|  |  |  |  |
| --- | --- | --- | --- |
| Assumptions | Independent Variables | Derived quantities | Performance Characteristics |
| * Length[1..10] | * I\_branch[1...4]   Random for now   * Gain\_stage[1..4]   CG, cascade  CS,CD ~ 1 always   * $R3 = x\*$R2;   10-10k, increment of 1k   * All gm’s except (4, 7, 8, 10):   0.0000005-0.003 | * Width[1..10] * Cgs[1..10] * Miller Gain (Mn10) * Gm[1..10] * Vov[1..10] | * DC Gain * 3db Bandwidth * Power |

First resistive divider

Calculating bounds on the ratio of R2/R1:



Assume that the current through Mp3 is matched with the current through Mn2, such that no current flows between the R1/R2 resistive divider and into the transistor stack. Assume that there is no body effect on Vt. In order to keep Mp4 in saturation, we must have the following condition:

In other words

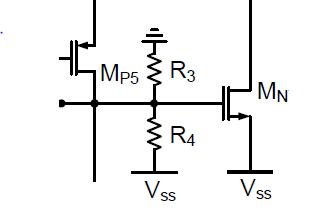
Then R1||R2 = <ratio>/(<1+ratio>R1) = Gain\_CG

R1 = 5/4 Gain\_CG

# First Cgs4

# Second resistor divider

Similarly for R3/R4:



In other words

Use same values as first divider, since GBW product will be less affected by gain since there is a gm factor in this stage.

# third xtor stack

Gain = gm7/gm8

# fourth xtor stack

Script overview

Choose current allocations -> Calculate gm’s -> Calculate Vov’s -> Calculate sizings

1. Find gain equation at each of the 4 stages
   1. Ignore ro, assume ro >> R1-4
2. Find RC time constant at each node
   1. Ignore Cgd b/c Cgd = Cov << Cgs = 2/3WLCox + Cov
   2. Ignore bulk terminal caps b/c hard to estimate
3. Power ≤ 2mW, Vdd - Vss = 5V→ Itot ≤ = 400uA
4. Allocate gain per stage
5. Find R1/R2 ratio = 4
6. Assume R3/R4 = 3
7. Calculate gm from gain and R’s
8. Find power dissipated by the R’s to figure out how much left for xtor stacks
9. Allocate current through 4 stacks based on #5
10. For each xtor,
    1. calculate Vov from its Ids and gm
    2. calculate W from Ids, Vov
    3. calculate Cgs
11. Calculate time constants at each stage. 1/sum must meet our target bandwidth

We know that Vout = 0V and M10 overdrive is about 1. This means Vz must be bigger than 1.150V. Let’s set it equal to 1.2V.

Vz = 1.2V

And we want W7/W8 = 1 so that the miller effect is minimized

Vy = -1.43V

Which means we can get the R3R4 ratio

R3/R4 = 1.346

We want M4 to be balanced with M6. We know that M5 has some body effect, so let’s say it’s Vt is about .9. This means the node between 4 and 5 has to be about 1.05

For M4 to be in saturation, we cannot let Vx drop below 1.05-.5 = 0.55. or above 1.8

We want M4 and M5 to have same size to minimize miller

Vnbias must be between -1.85 and -0.15 to keep M1 and M2 both saturated. But it must also be less than 0.5 + Vy = -0.93

Vpbias must be between 1.85 and Vx+.5

Want biggest gm possible, so make a lot of current go through M2 and give M2 smallest possible overdrive. This means Vn should be as low as possible, but W1 should be as big as possible

We want gm10 big to get good BW on output node, so W10 should be big

To match M4 and M6 currents