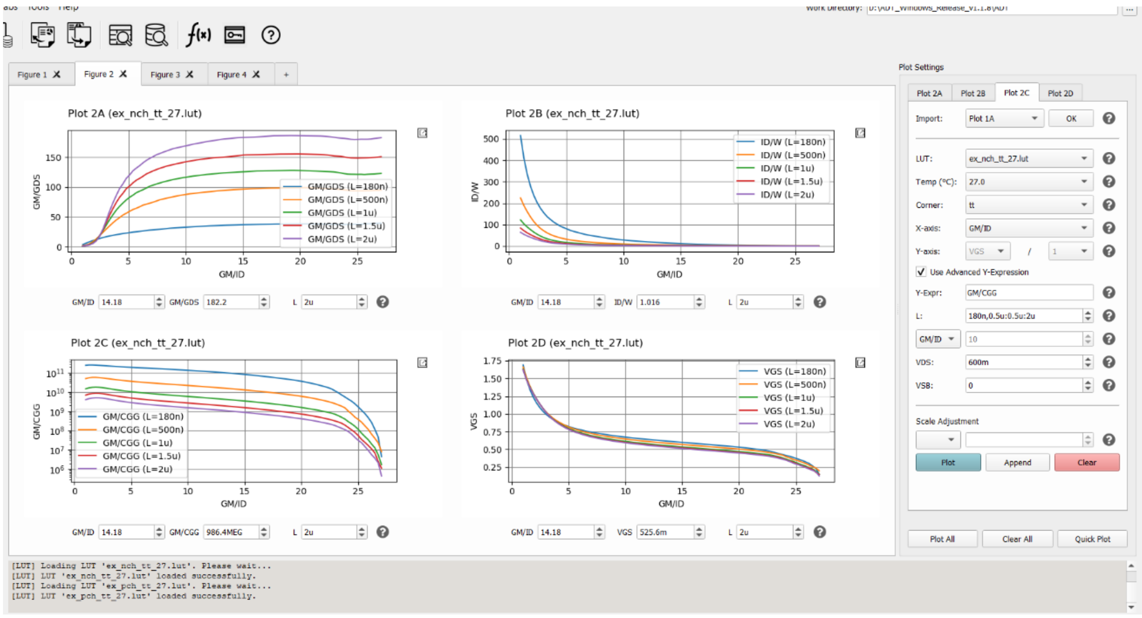
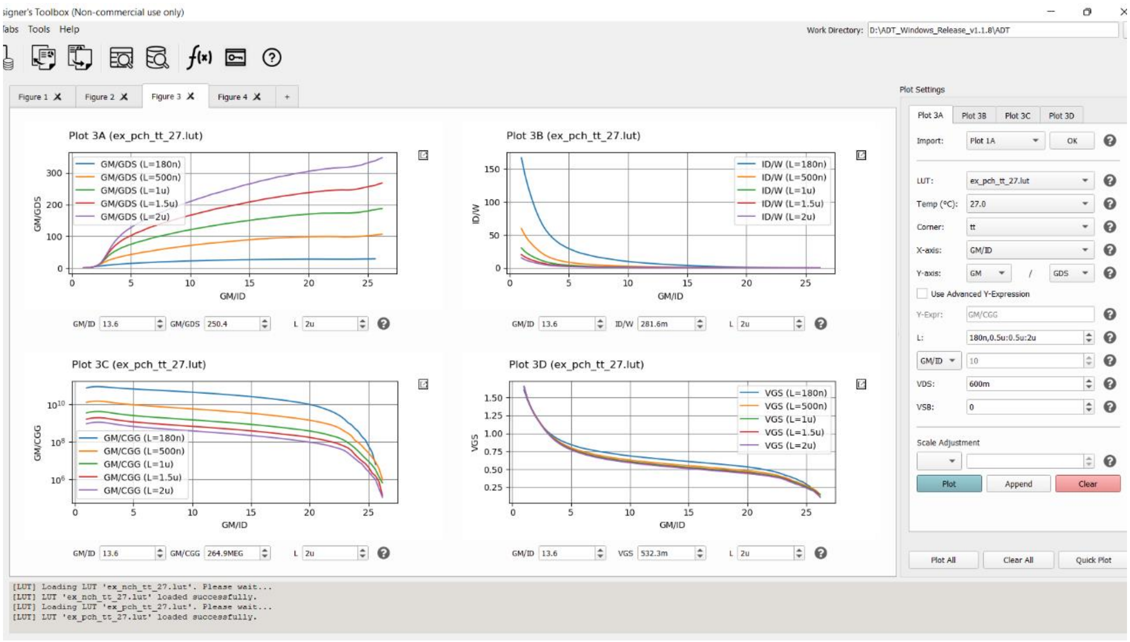
Part1

Question 1:

NMOS Charts

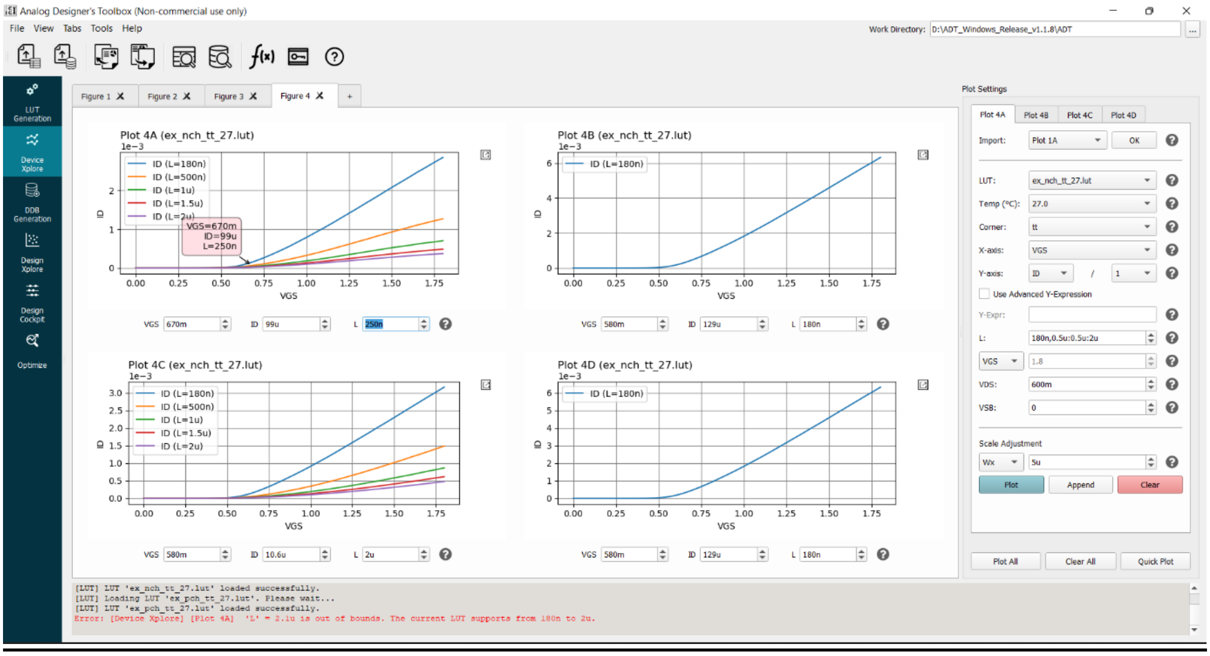


PMOS Charts



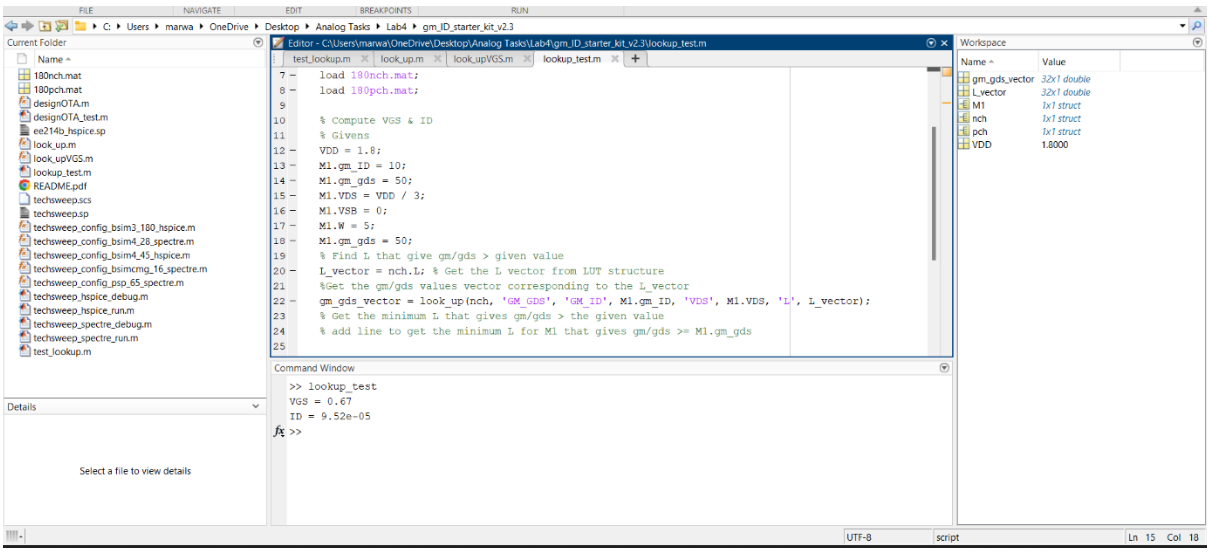
Question 2:

ID vs VGS Plots for NMOS that has gm/ID = 10, gm/gds = 50, VDS = VDD/3, VSB = 0, and W = 5u.



Question 3:

Matlab Results

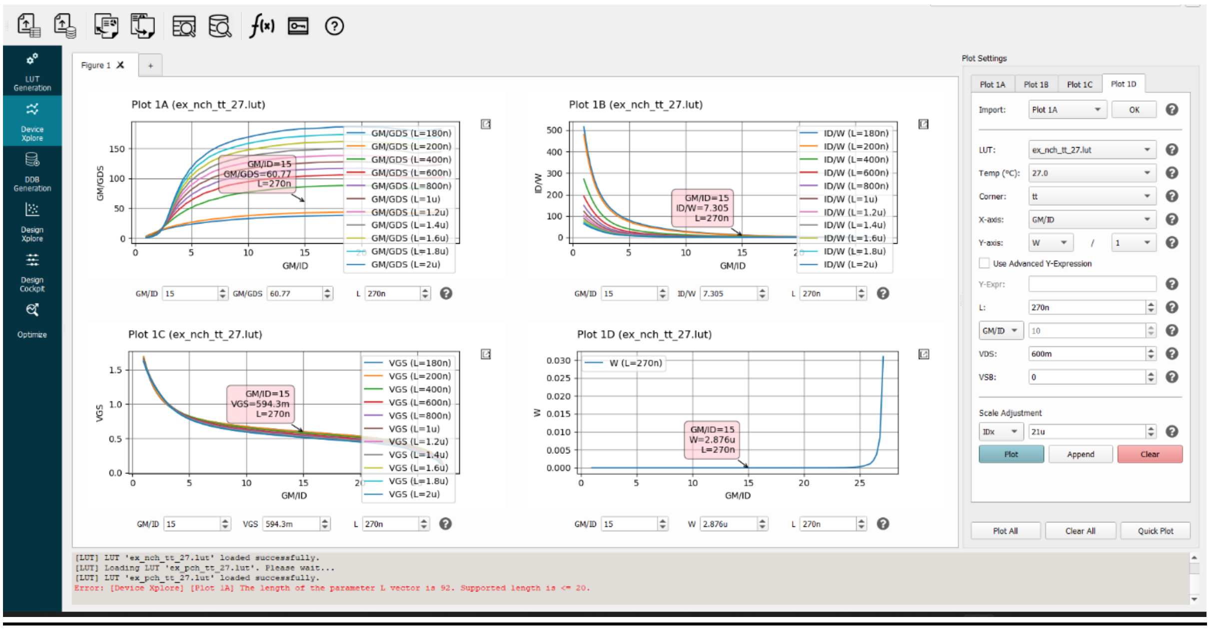


MATLAB	VGS=0.67v	ID=95.2u
ADT	VGS=0.67v	ID=99u

This slight difference in Id Because of using Different LUT in Each Program as ADT Uses LTSpice Generated LUT While Matlab Uses Hspice Generated LUT.

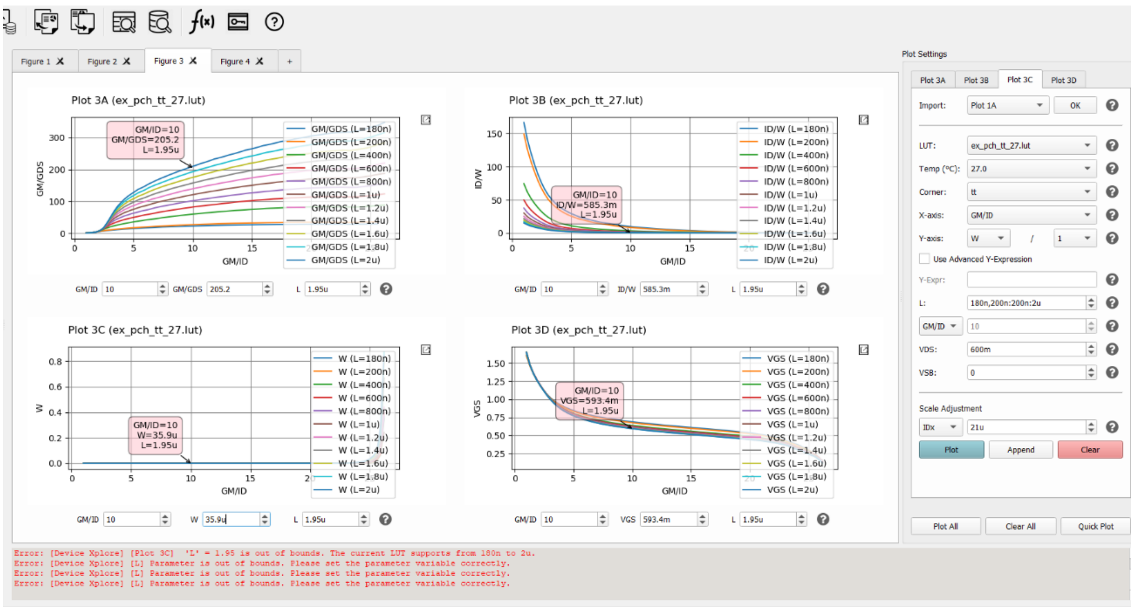
Question 4:

Input Pair Sizing

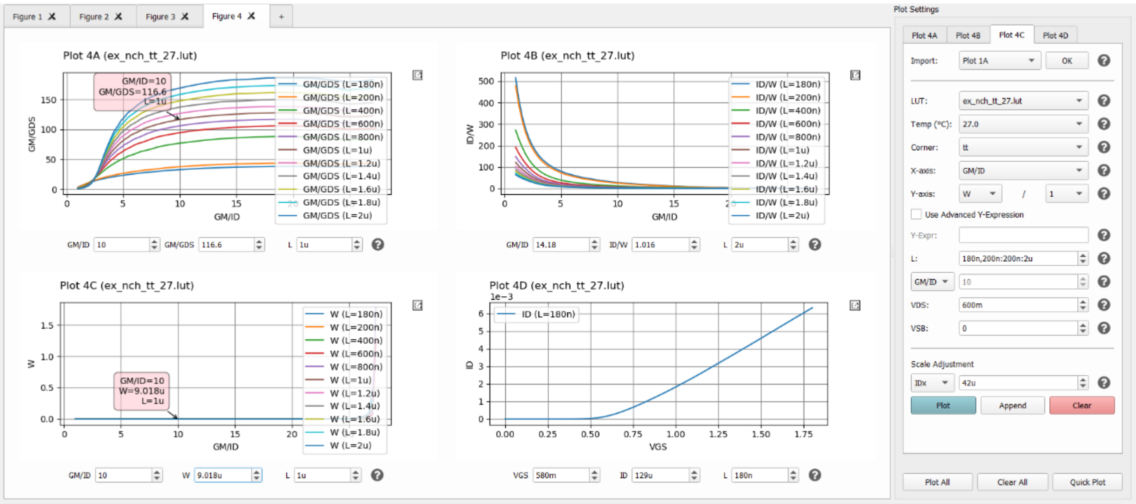


Used Some Hand analysis to get Gm and with the Assumptions got the above points , Vgs Determined to Add CM to bias The devices.

Pmos Current Mirror Load Sizing



Tail Current Source Sizing



Gm/id assumptions	For the input stage we used large gm/id as its gm Contributes at The gain and less random mismatch , while for the load and tail cs we used small gm/id as there gm doesn't Contribute at Gain and to have smaller area to get small capacitances from them and to have a less effect of VDS on Current As Va increase.
Ro assumptions	We used 5ro for The load To have A Less VDS Effect at the current from The load
Tail Current Source L assumptions	As L increase Ro tail cs increase so Cm gain decrease and cm rejection ratio increase , and have an accurate current Value as VDS Effect Decrease as Ro increase .

Question 5:

FILE NAVIGATE EDIT BREAKPOINTS RUN

```matlab
%OTA Synthesis Function
function OTA = designOTA(specs)
%Additional Specs
VDD = 1.8;

%load LUTs
load 180nch.mat;
load 180pch.mat;

%Input Pair
OTA.M1.gm = specs.GBW * specs.CL * 2 * pi;
% assume ro(load) = 5 * ro(input) --> ro(input) = 6/5 R_total
DC_Gain_mag = 10^(specs.AVDC / 20); % Convert from dB to mag
Rout = DC_Gain_mag / OTA.M1.gm; % Compute the equivalent output resistance of OTA (Rout)

OTA.M1.ro =(6/5)*Rout; % Complete the line to compute the ro of M1

OTA.M1.gds = 1 / OTA.M1.ro;
OTA.M1.VDS = VDD/3;
OTA.M1.gm_gds = OTA.M1.gm / OTA.M1.gds;
OTA.M1.gm_ID = 15; % assumption

OTA.M1.ID =(OTA.M1.gm/OTA.M1.gm_ID);% Complete the line to get the current of M1

% Search for the minimum L that gives gm / gds > specified value
L_vector = nch.L;
gm_gds_vector = look_up(nch, 'GM_GDS', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L', L_vector);
OTA.M1.L =min(L_vector( gm_gds_vector >= OTA.M1.gm_gds));
 %Complete the line to get the minimum L that gives gm/gds >= OTA.M1.gm_gds
% Compute ID/W to get the W value
OTA.M1.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L', OTA.M1.L);
OTA.M1.W = OTA.M1.ID / OTA.M1.ID_W;
```

```matlab
 %Complete the line to get the minimum L that gives gm/gds >= OTA.M1.gm_gds
% Compute ID/W to get the W value
OTA.M1.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L', OTA.M1.L);
OTA.M1.W = OTA.M1.ID / OTA.M1.ID_W;

%CM_Load
OTA.M3.ID = OTA.M1.ID;

OTA.M3.ro =(5*OTA.M1.ro); %Complete the line to get the ro of the CM load

OTA.M3.gds = 1 / OTA.M3.ro;
OTA.M3.VDS = VDD/3;
OTA.M3.gm_ID = 10;
OTA.M3.gm = OTA.M3.gm_ID * OTA.M3.ID;
OTA.M3.gm_gds = OTA.M3.gm / OTA.M3.gds;
gm_gds_vector = look_up(pch, 'GM_GDS', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', L_vector);
OTA.M3.L = min(L_vector(gm_gds_vector > OTA.M3.gm_gds));
OTA.M3.ID_W =look_up(pch, 'ID_W', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', OTA.M3.L); %Complete the line to get the ID/W of M3
OTA.M3.W = OTA.M3.ID / OTA.M3.ID_W;

% Tail bias
OTA.M5.L = 1; %assumption
OTA.M5.ID = 2 * OTA.M1.ID;
OTA.M5.VDS = VDD/3;
OTA.M5.gm_ID = 10; %assumption
% Get ID/W to compute W
OTA.M5.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M5.gm_ID, 'VDS', OTA.M5.VDS, 'L', OTA.M5.L);
OTA.M5.W = OTA.M5.ID / OTA.M5.ID_W;

% get CMIN bias value
OTA.M1.VGS =look_upVGS(nch, 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L',OTA.M1.L);
 %Complete the line to get the VGS of M1
OTA.M1.VG =OTA.M1.VGS + OTA.M5.VDS;
%Complete the line to get the DC CM input of OTA
```

Question 6: Test Script

Desktop ▸ Analog Tasks ▸ Lab4 ▸ gm_ID_starter_kit_v2.3

Editor - C:\Users\marwa\OneDrive\Desktop\Analog Tasks\Lab4\gm_ID_starter_kit_v2.3\designOTA_test.m

test_lookup.m | look_up.m | look_upVGS.m | lookup_test.m | designOTA_test.m | designOTA2.m | designOTA_test2.m | +

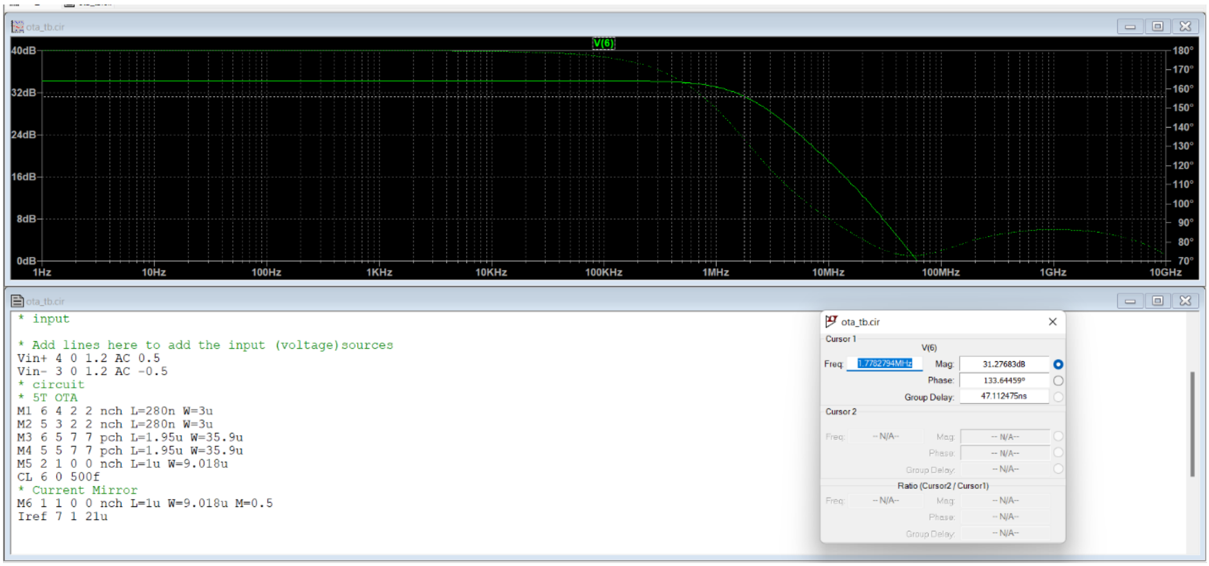
```matlab
% OTA Design Script
% Write the SPECS
clear all;
AVDC =34; % complete the line to add the gain SPEC
GBW =100e6 ;% complete the line to add the GBW SPEC
CL = 500e-15;% complete the line to add the CL SPEC
specs = struct('AVDC', AVDC,...
'CL', CL,...
'GBW', GBW);

OTA = designOTA(specs);
% Print the solution
fprintf('**** OTA Design ****\n\n');
fprintf('Input Pair:\n');
fprintf('    L = %.2f um\n    W=%.2f um\n    ViCM=%.4f V\n    Id=%.4f um \n\n',OTA.M1.L,OTA.M1.W,OTA.M1.VG,OTA.M1.ID*1e6);
fprintf('CM Load:\n');
fprintf('    L = %.2f um\n    W=%.2f um\n    Id=%.4f um\n    \n\n',OTA.M3.L,OTA.M3.W,OTA.M3.ID*1e6);
fprintf('Tail Current Source:\n');
fprintf('    L = %.2f um\n    W=%.2f um\n    Id=%.4f um\n    \n\n',OTA.M5.L,OTA.M5.W,OTA.M5.ID*1e6);
```

Matlab Results (No Parasitic Caps)

```
>> designOTA_test
**** OTA Design ****

Input Pair:
    L = 0.28 um
    W=3.01 um
    ViCM=1.1928 V
    Id=20.9440 um

CM Load:
    L = 1.70 um
    W=31.44 um
    Id=20.9440 um


Tail Current Source:
    L = 1.00 um
    W=9.06 um
    Id=41.8879 um

fx >>
```

Question 7:

Netlist & Simulation Results (No Parasitic Caps)



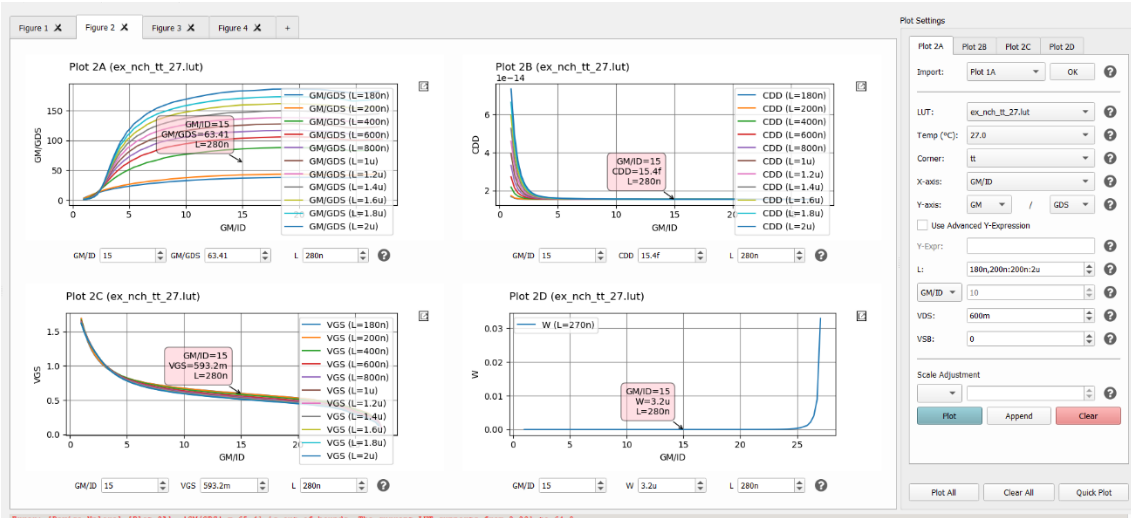
	Specs	Spice Results
GBW	100 Mega hz	1.77*Mega*51.2 = 92 Mega hz
DC Gain	34 db	34.3db

The Dc Gain Is Achieved but there is an error at the GBW this is due neglecting the parasitic caps , but it will effect the GBW as there order of magnitude is the same as Cl (Femto).

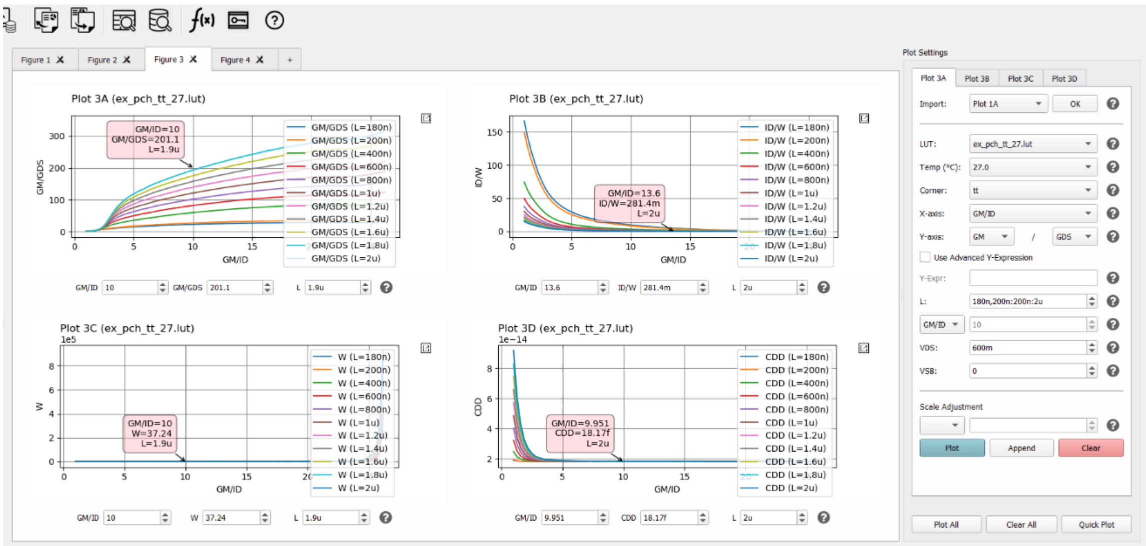
PART 2 :

Question 1

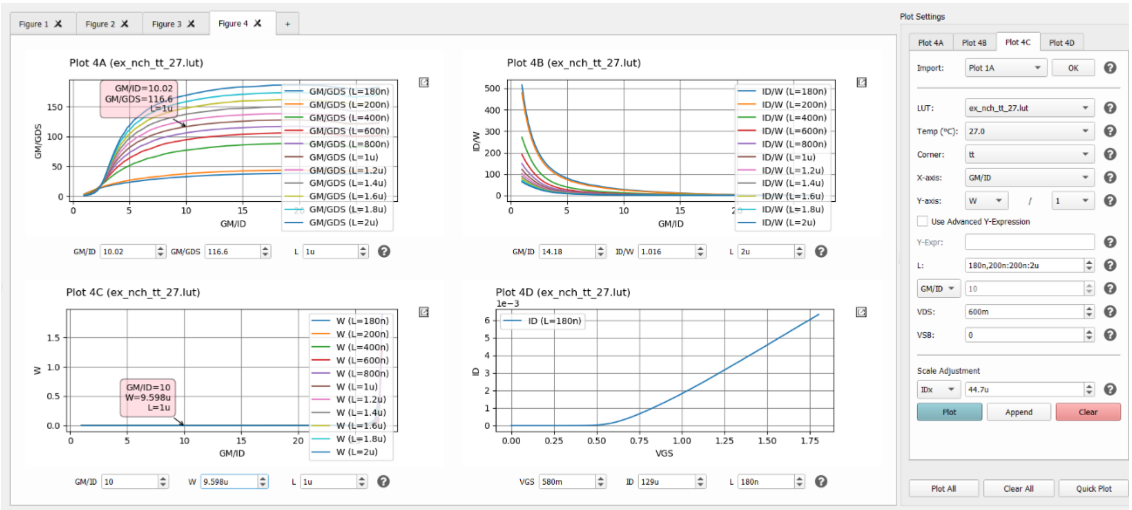
Input Pair Sizing While Taking Parasitic Caps in calculations



Current load Sizing While Taking Parasitic Caps in calculations



Tail Current Source Sizing While Taking Parasitic Caps in calculations



The Methodology I Used Was Calculating Both CDD for Input and Current Mirror Load Then Calculating Gm From GBW Equation And Continue Design As usual .

<u>Question 2:</u>

Matlab Function

```
%OTA Synthesis Function
function OTA = designOTA2(specs)
%Additional Specs
VDD = 1.8;

%load LUTs
load 180nch.mat;
load 180pch.mat;
OTA.M1.VDS = VDD/3;
OTA.M3.VDS = VDD/3;
OTA.M1.gm_ID = 15;
OTA.M3.gm_ID = 10;
% Getting CDD For Both input and load stage
CDD__GM_NMOS = look_up(nch, 'GM_CDD', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS);
CDD__GM_PMOS = look_up(pch, 'GM_CDD', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS);
%Input Pair
tolarance=100;
Current_Gm=0;
Prev_Gm=0;
CDD_NMOS=0;
CDD_PMOS=0;


% Tolarance 8%
while tolarance>0.08
    Prev_Gm=Current_Gm;
    %calculating Gm While adding Parastics to Calculations
OTA.M1.gm = specs.GBW * (specs.CL+CDD_NMOS+CDD_PMOS) * 2 * pi;
```

Desktop ▸ Analog Tasks ▸ Lab4 ▸ gm_ID_starter_kit_v2.3

Editor - C:\Users\marwa\OneDrive\Desktop\Analog Tasks\Lab4\gm_ID_starter_kit_v2.3\designOTA2.m

test_lookup.m | look_up.m | look_upVGS.m | lookup_test.m | designOTA.m | designOTA_test.m | designOTA2.m | designOTA_test2.m

```matlab
OTA.M1.gm = specs.GBW * (specs.CL+CDD_NMOS+CDD_PMOS) * 2 * pi;
% After Completing the desgin without parastics we add the parstics that
% effect the bandwidth
CDD_NMOS=(OTA.M1.gm/CDD__GM_NMOS);
CDD_PMOS=(OTA.M1.gm/CDD__GM_PMOS);
Current_Gm=OTA.M1.gm;
% assume ro(load) = 5 * ro(input) --> ro(input) = 6/5 R_total
DC_Gain_mag = 10^(specs.AVDC / 20); % Convert from dB to mag
Rout = DC_Gain_mag / OTA.M1.gm; % Compute the equivalent output resistance of OTA (Rout)

OTA.M1.ro =(6/5)*Rout; % Complete the line to compute the ro of M1

OTA.M1.gds = 1 / OTA.M1.ro;
OTA.M1.VDS = VDD/3;
OTA.M1.gm_gds = OTA.M1.gm / OTA.M1.gds;
OTA.M1.gm_ID = 15; % assumption

OTA.M1.ID =(OTA.M1.gm/OTA.M1.gm_ID);% Complete the line to get the current of M1

% Search for the minimum L that gives gm / gds > specified value
L_vector = nch.L;
gm_gds_vector = look_up(nch, 'GM_GDS', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L', L_vector);
OTA.M1.L =min(L_vector( gm_gds_vector >= OTA.M1.gm_gds));
 %Complete the line to get the minimum L that gives gm/gds >= OTA.M1.gm_gds
% Compute ID/W to get the W value
OTA.M1.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L', OTA.M1.L);
OTA.M1.W = OTA.M1.ID / OTA.M1.ID_W;

```


eDrive ▸ Desktop ▸ Analog Tasks ▸ Lab4 ▸ gm_ID_starter_kit_v2.3

Editor - C:\Users\marwa\OneDrive\Desktop\Analog Tasks\Lab4\gm_ID_starter_kit_v2.3\designOTA2.m

test_lookup.m | look_up.m | look_upVGS.m | lookup_test.m | designOTA.m | designOTA_test.m | designOTA2.m | designOTA_test2.m


```matlab
%CM_Load
OTA.M3.ID = OTA.M1.ID;

OTA.M3.ro =(5*OTA.M1.ro); %Complete the line to get the ro of the CM load

OTA.M3.gds = 1 / OTA.M3.ro;
OTA.M3.VDS = VDD/3;
OTA.M3.gm_ID = 10;
OTA.M3.gm = OTA.M3.gm_ID * OTA.M3.ID;
OTA.M3.gm_gds = OTA.M3.gm / OTA.M3.gds;
gm_gds_vector = look_up(pch, 'GM_GDS', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', L_vector);
OTA.M3.L = min(L_vector(gm_gds_vector > OTA.M3.gm_gds));
OTA.M3.ID_W =look_up(pch, 'ID_W', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', OTA.M3.L); %Complete the line to get the ID
OTA.M3.W = OTA.M3.ID / OTA.M3.ID_W;

% Tail bias
OTA.M5.L = 1; %assumption
OTA.M5.ID = 2 * OTA.M1.ID;
OTA.M5.VDS = VDD/3;
OTA.M5.gm_ID = 10; %assumption
% Get ID/W to compute W
OTA.M5.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M5.gm_ID, 'VDS', OTA.M5.VDS, 'L', OTA.M5.L);
OTA.M5.W = OTA.M5.ID / OTA.M5.ID_W;

% get CMIN bias value
OTA.M1.VGS =look_upVGS(nch, 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L',OTA.M1.L);
 %Complete the line to get the VGS of M1
OTA.M1.VG =OTA.M1.VGS + OTA.M5.VDS;
```


Editor - C:\Users\marwa\OneDrive\Desktop\Analog Tasks\Lab4\gm_ID_starter_kit_v2.3\designOTA2.m

test_lookup.m | look_up.m | look_upVGS.m | lookup_test.m | designOTA.m | designOTA_test.m | designOTA2.m | designOTA_test2.m

OTA.M3.gm_ID = 10;
OTA.M3.gm = OTA.M3.gm_ID * OTA.M3.ID;
OTA.M3.gm_gds = OTA.M3.gm / OTA.M3.gds;
gm_gds_vector = look_up(pch, 'GM_GDS', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', L_vector);
OTA.M3.L = min(L_vector(gm_gds_vector > OTA.M3.gm_gds));
OTA.M3.ID_W =look_up(pch, 'ID_W', 'GM_ID', OTA.M3.gm_ID, 'VDS', OTA.M3.VDS, 'L', OTA.M3.L); %Complete the line to get the ID
OTA.M3.W = OTA.M3.ID / OTA.M3.ID_W;

% Tail bias
OTA.M5.L = 1; %assumption
OTA.M5.ID = 2 * OTA.M1.ID;
OTA.M5.VDS = VDD/3;
OTA.M5.gm_ID = 10; %assumption
% Get ID/W to compute W
OTA.M5.ID_W = look_up(nch, 'ID_W', 'GM_ID', OTA.M5.gm_ID, 'VDS', OTA.M5.VDS, 'L', OTA.M5.L);
OTA.M5.W = OTA.M5.ID / OTA.M5.ID_W;

% get CMIN bias value
OTA.M1.VGS =look_upVGS(nch, 'GM_ID', OTA.M1.gm_ID, 'VDS', OTA.M1.VDS, 'L',OTA.M1.L);
 %Complete the line to get the VGS of M1
OTA.M1.VG =OTA.M1.VGS + OTA.M5.VDS;


```matlab
%Complete the line to get the DC CM input of OTA
% Calculating Tolarance
tolarance= ((Current_Gm - Prev_Gm)/Current_Gm);

end

end
```

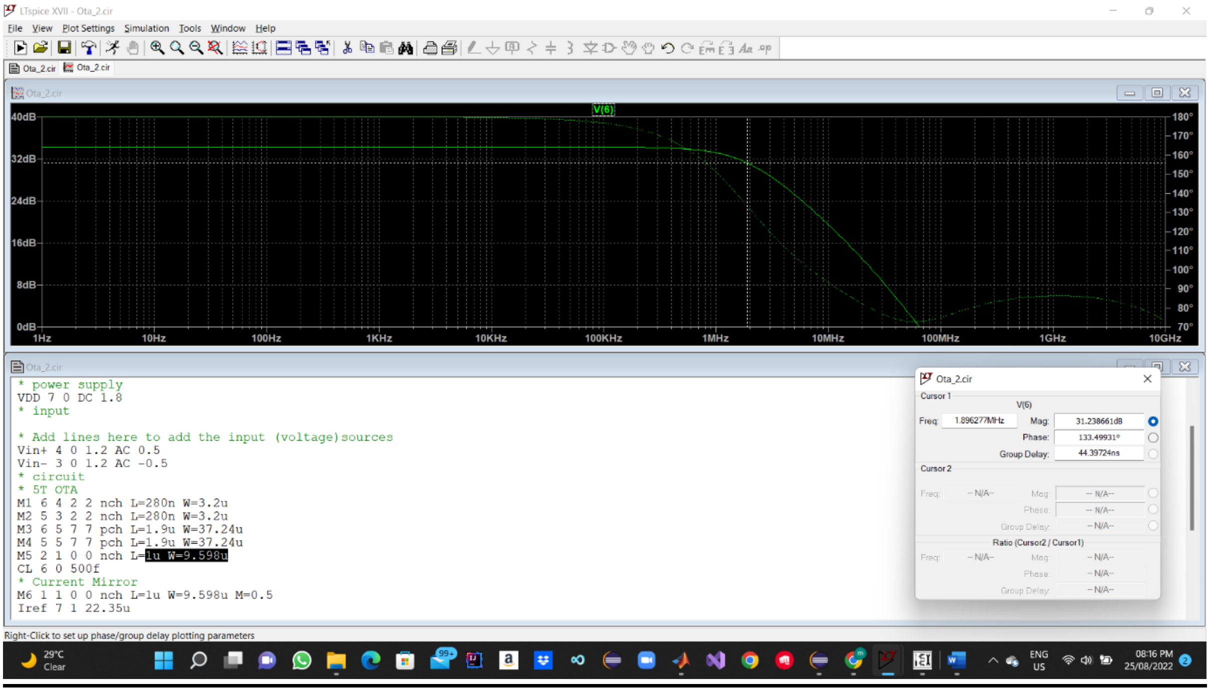
Command Window

The Function Takes The Effect of Parasitic Caps in Consideration it Iterates Until it Founds The best Sizing values With 8% Percent tolerance to stop at the gm Value.

ADT With Parasitic Caps	MATLAB Part1 No Parasitic Caps	MATLAB Part2 With Parasitic Caps
<u>Input Pair:</u> W= 3.2u L=280n <u>Current Mirror Load :</u> W= 37.24u L=1.9u <u>Tail Current Source :</u> W=9.598u L=1u	Command Window >> designOTA_test **** OTA Design **** Input Pair: L = 0.28 um W=3.01 um ViCM=1.1928 V CM Load: L = 1.70 um W=31.44 um Tail Current Source: L = 1.00 um W=9.06 um fx >>	Command Window >> designOTA_test2 **** OTA Design With Parastics and Tolarence 8 percent **** Input Pair: L = 0.28 um W=3.06 um ViCM=1.1928 V Id=21.2358 um CM Load: L = 1.70 um W=31.88 um Id=21.2358 um Tail Current Source: L = 1.00 um W=9.19 um Id=42.4717 um fx >>

There Are Difference Between ADT And Matlab Results As Matlatb LUTs Are made on Hspice While ADT LUTs are made on LTspice , MATLAB Function 2 Has Tolerance 8% Stoping Iterations Condition , When The gm is changed while Calculating it from The GBW as Adding caps in calculations .

<u>Question 3:</u>



	Specs	Spice Results
GBW	100 Mega hz	1.89*Mega*51.286 = 97 Mega hz
DC Gain	34 db	34.2db

As we took the Parasitic Caps Into Consideration We Have A better Accuracy for the specs as the error become 3% only .