

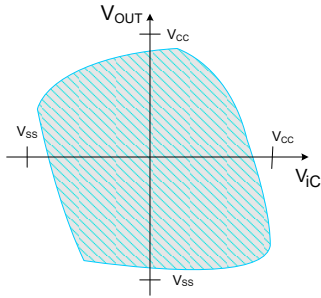
# EE 435

## Lecture 7:

- High Gain Single-Stage Op Amps

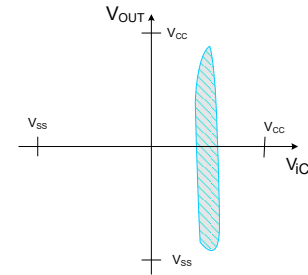
# Signal Swing of Single-Stage Op Amp

What type of signal swing is needed ?



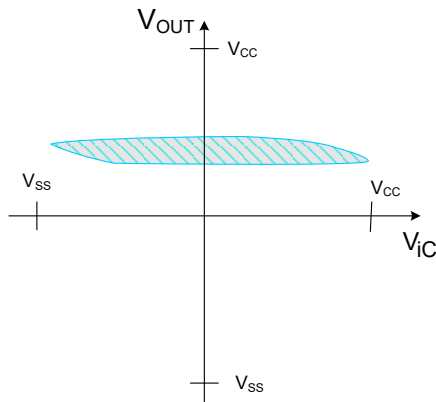
Wide  $V_{iC}$  and  $V_{OUT}$  range

Expected for catalog parts and overall I/O in many applications



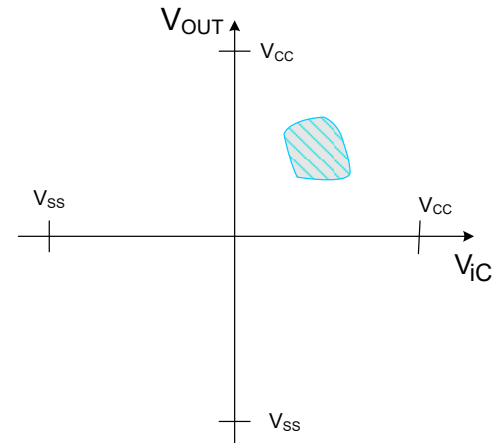
Narrow  $V_{iC}$  and wide  $V_{OUT}$  range

Acceptable when  $V_{iC}$  is fixed



Narrow  $V_{OUT}$  and wide  $V_{iC}$  range

Acceptable when followed by high-gain stage



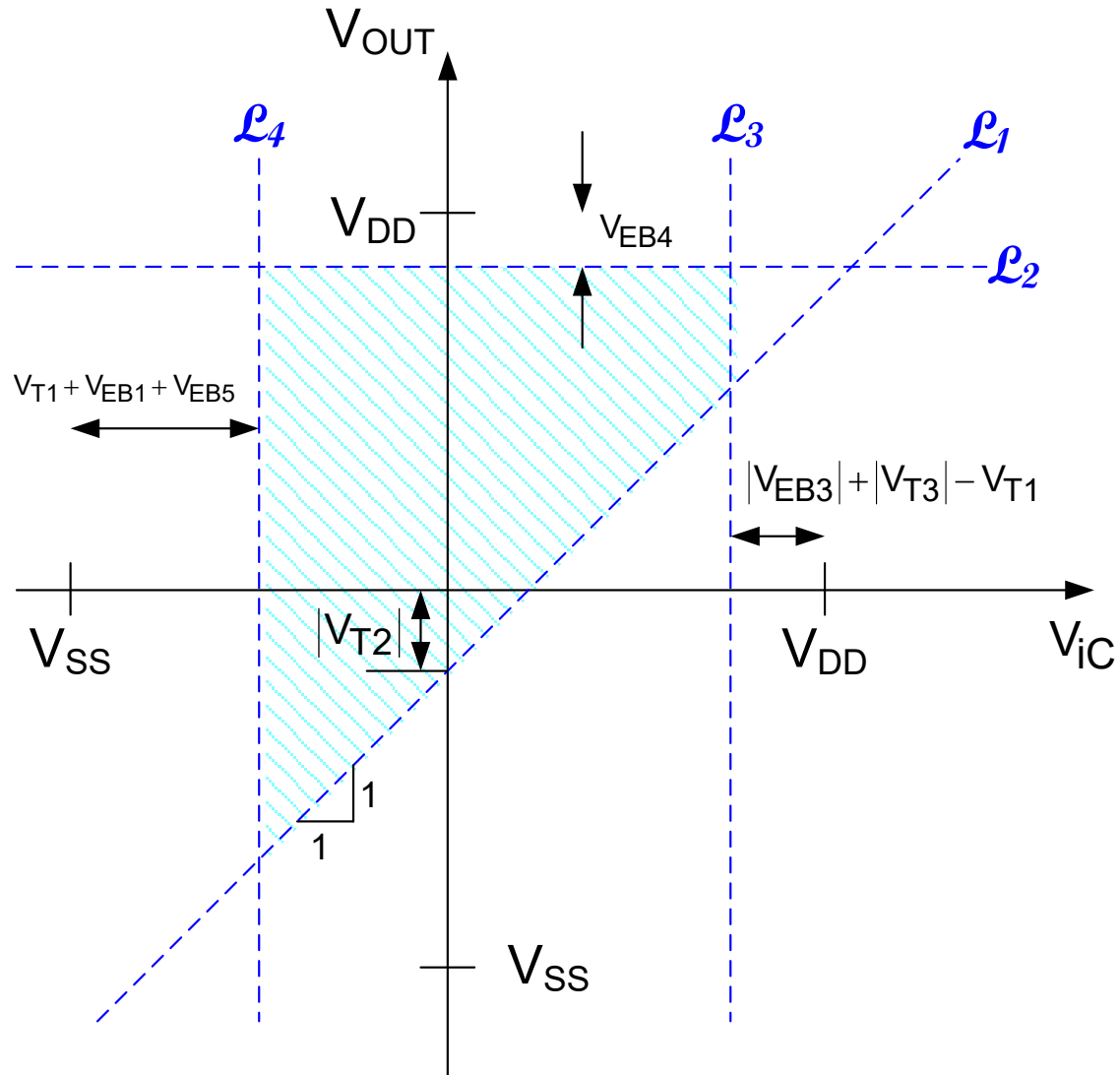
Narrow  $V_{iC}$  and  $V_{OUT}$  range

Acceptable when  $V_{iC}$  fixed and followed by high-gain stage

Review from last lecture:

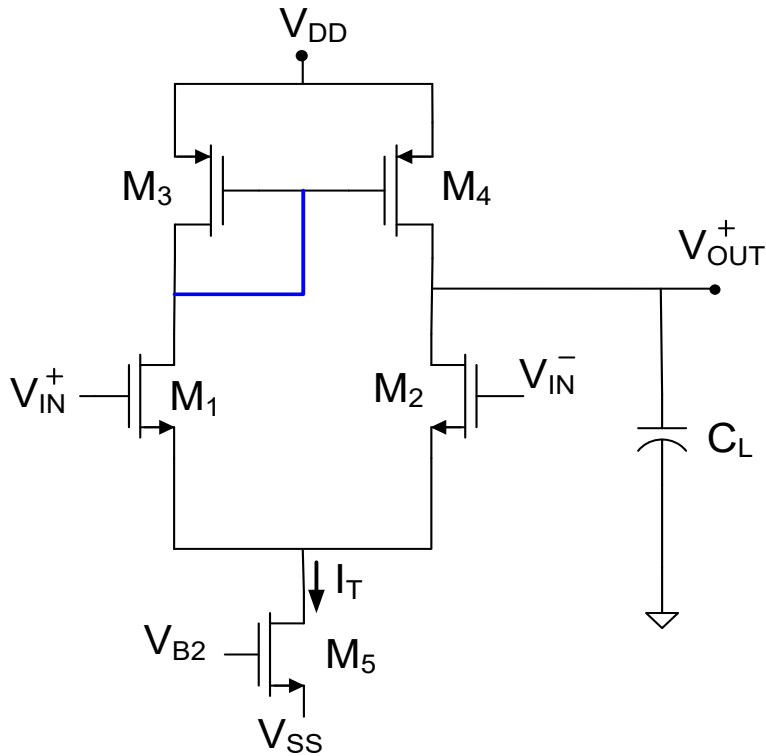
Review from last lecture:

# Signal Swing of Single-Stage Op Amp



Review from last lecture:

# Design space for single-stage op amp



Performance Parameters in Practical Parameter Domain  $\{V_{EB1} V_{EB2} V_{EB5} P\}$ :

$$A_0 = \left[ \frac{1}{\lambda_1 + \lambda_3} \right] \left( \frac{2}{V_{EB1}} \right)$$

$$GB = \left( \frac{P}{V_{DD} C_L} \right) \left[ \frac{1}{V_{EB1}} \right]$$

$$SR = \frac{P}{(V_{DD} - V_{SS}) C_L}$$

$$V_{OUT} < V_{DD} - |V_{EB3}|$$

$$V_{OUT} > V_{ic} - V_{T2}$$

$$V_{ic} < V_{DD} + V_{T1} - |V_{T3}| - |V_{EB3}|$$

$$V_{ic} > V_{T1} + V_{EB1} + V_{EB5} + V_{SS}$$

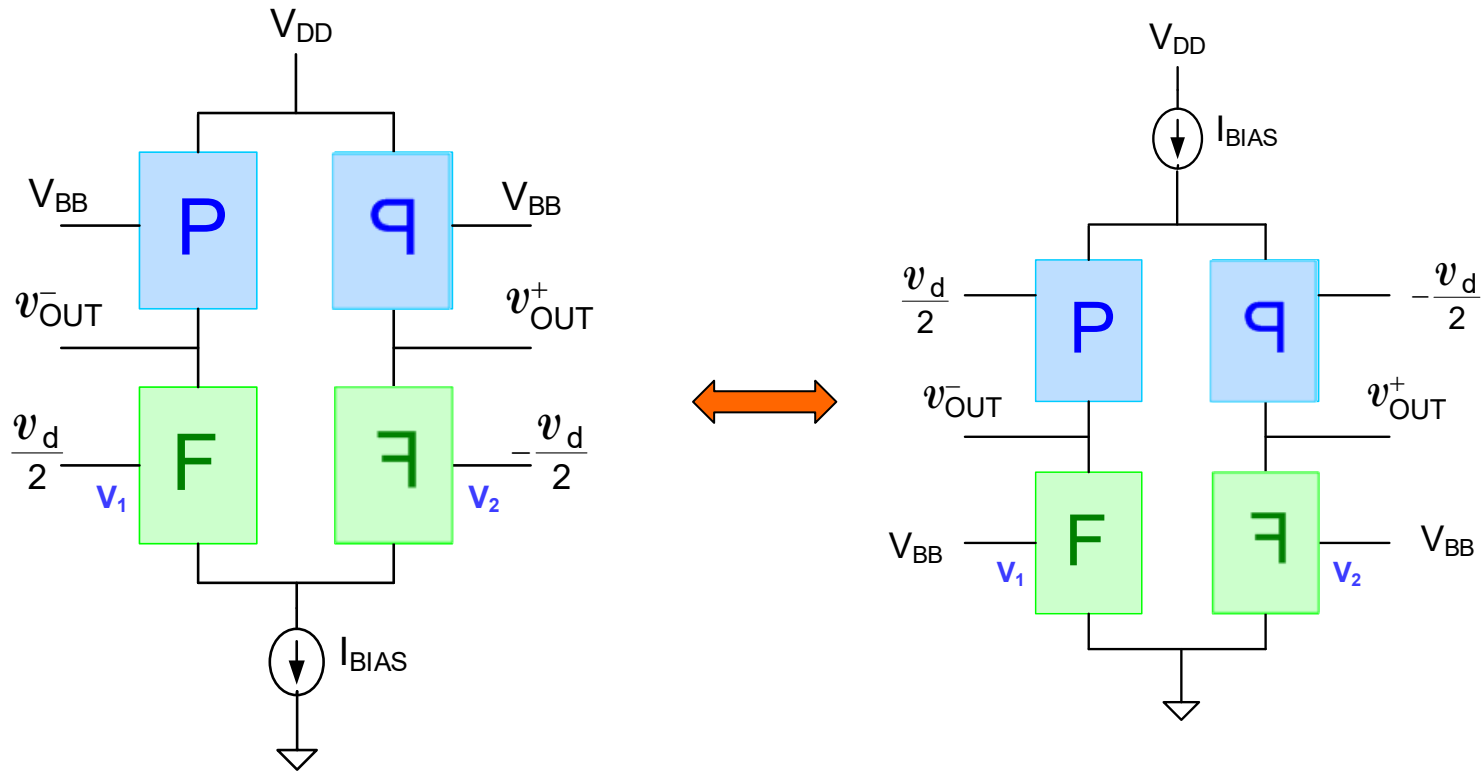
**Simple Expressions (7) in Practical Parameter Domain**

Where we are at:

# Basic Op Amp Design

- Fundamental Amplifier Design Issues
- • Single-Stage Low Gain Op Amps
- Single-Stage High Gain Op Amps
- Other Basic Gain Enhancement Approaches
- Two-Stage Op Amp

# Inputs into Counterpart Circuit or Quarter Circuit

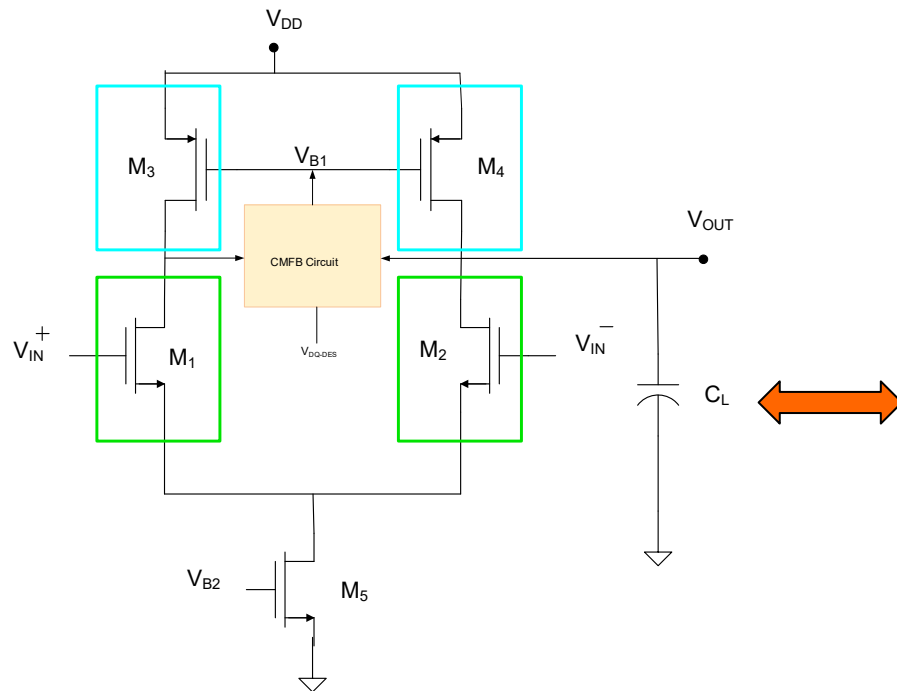


Gain, BW, and GB expressions identical

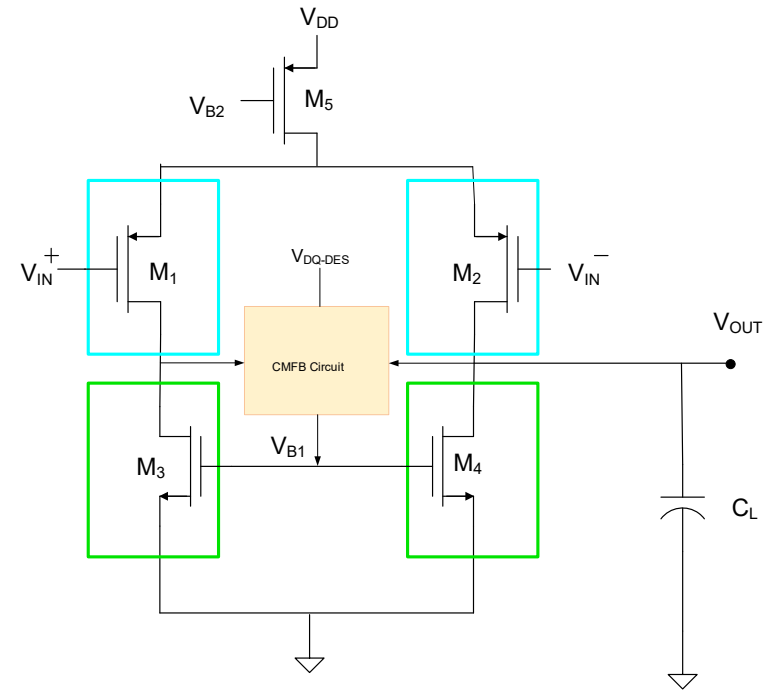
- This is a general concept not related to what type of quarter circuit is used
- Performance may be different because n-channel and p-channel performance different

# Inputs into Counterpart Circuit or Quarter Circuit for single-transistor quarter circuit

5T Op Amp with n-ch inputs



5T Op Amp with p-ch inputs



Gain, BW, and GB expressions identical

- Performance may be different because n-channel and p-channel performance different
- Both are widely used

# Single-stage op amps

Question – is the gain achievable with the single-stage low-gain op amps using a single MOS transistor as a quarter circuit adequate?

$$A_{v0} = \left[ \frac{1}{\lambda_1 + \lambda_3} \right] \left( \frac{1}{V_{EB1}} \right)$$

If  $\lambda_1 = \lambda_3 = .01V^{-1}$  and  $V_{EB1} = .15V$ , then

$$A_{v0} \approx \frac{1}{(.01 + .01)} \frac{1}{0.15} = 333$$

or, in db,  $A_{v0db} = 20 \log_{10} 333 = 50db$

This is inadequate for many applications !

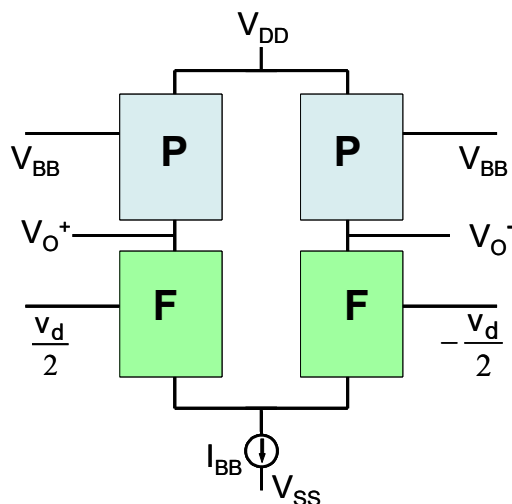
What can be done about it ?



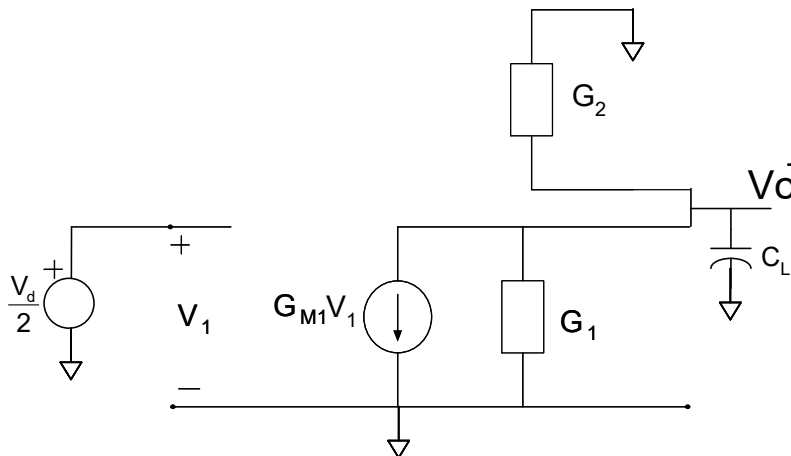


## Recall from a previous lecture:

# Determination of op amp characteristics from quarter circuit characteristics



Small signal differential half-circuit



$$A_V = \frac{V_{O^+}}{V_d} = \frac{-\frac{G_{M1}}{2}}{sC_L + G_1 + G_2}$$



$$A_{VO} = \frac{-G_{M1}}{2(G_1 + G_2)}$$

$$BW = \frac{G_1 + G_2}{C_L}$$

$$GB = \frac{G_{M1}}{2C_L}$$

Recall from a previous lecture:

# Single-Stage High Gain Op Amps

How can the gain of the op amp be increased?

Recall from Quarter-Circuit Concept

$$A_{VO} = \frac{1}{2} \frac{-G_{M1}}{G_1 + G_2}$$

A possible strategy :

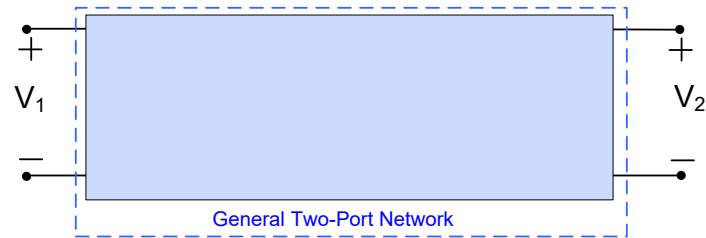
Increase  $G_{M1}$  or Decrease  $G_1$  (and  $G_2$ )  
in Quarter Circuit or Both

Recall from a previous lecture:

# Single-Stage High-Gain Op Amps

- If the output conductance can be decreased without changing the transconductance, the gain can be enhanced
- Will concentrate on quarter-circuits and extend to op amps

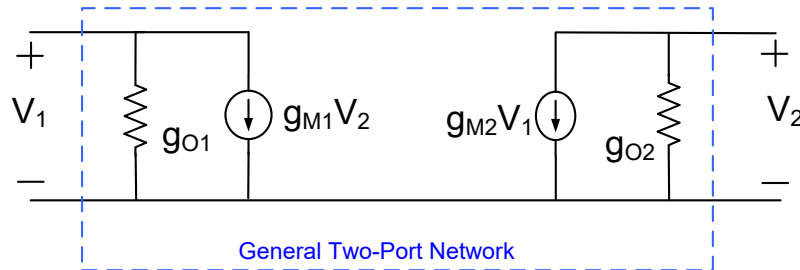
# Determination of 2-port parameters



Determination of  $\{g_{o1}, g_{o2}, g_{M1}, g_{M2}\}$

Method 1 Open-Short Termination Approach

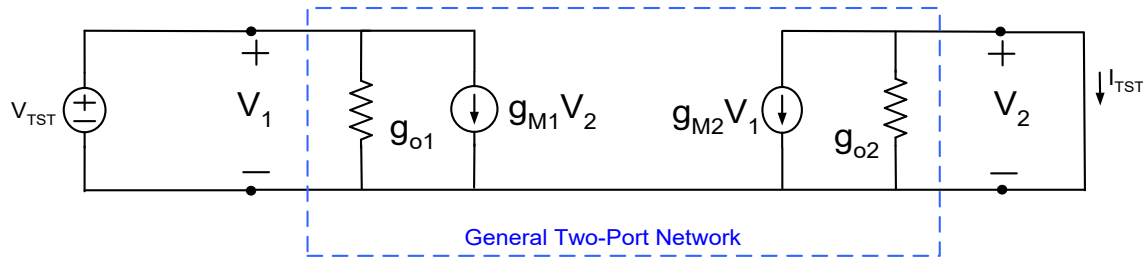
Method 2 Load Termination Approach



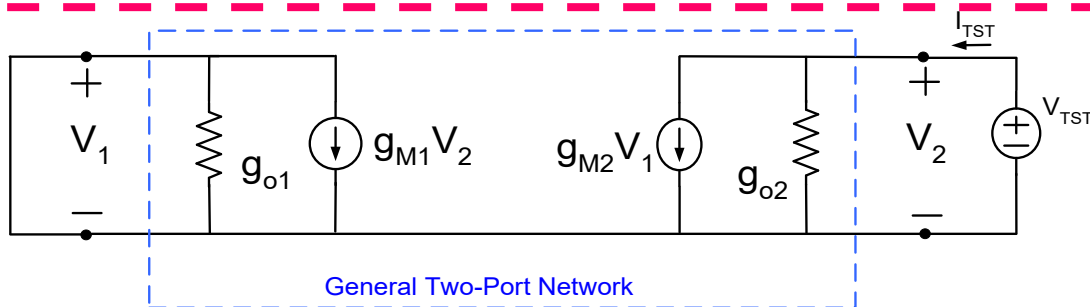
# Determination of 2-port parameters

Determination of  $\{g_{o1}, g_{o2}, g_{M1}, g_{M2}\}$

## Method 1 Open-Short Termination Approach



$$g_{M2} = -\frac{I_{TST}}{V_{TST}}$$



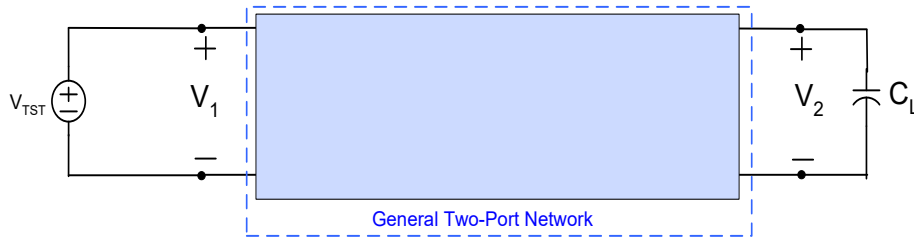
$$g_{o2} = \frac{I_{TST}}{V_{TST}}$$

By structural symmetry, repeat to obtain  $g_{M1}$  and  $g_{o1}$

# Determination of 2-port parameters

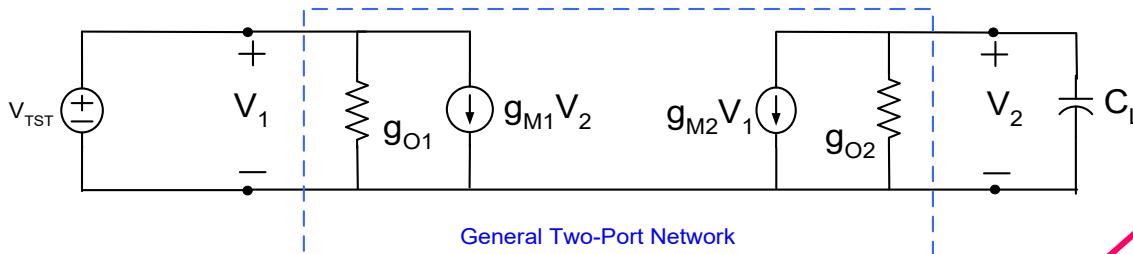
Determination of  $\{g_{o1}, g_{o2}, g_{M1}, g_{M2}\}$

Method 2 Load Termination Approach



Since first-order express the gain  $A(s)$  in form

$$A(s) = \frac{a_0}{sC_L + b_0}$$



observe

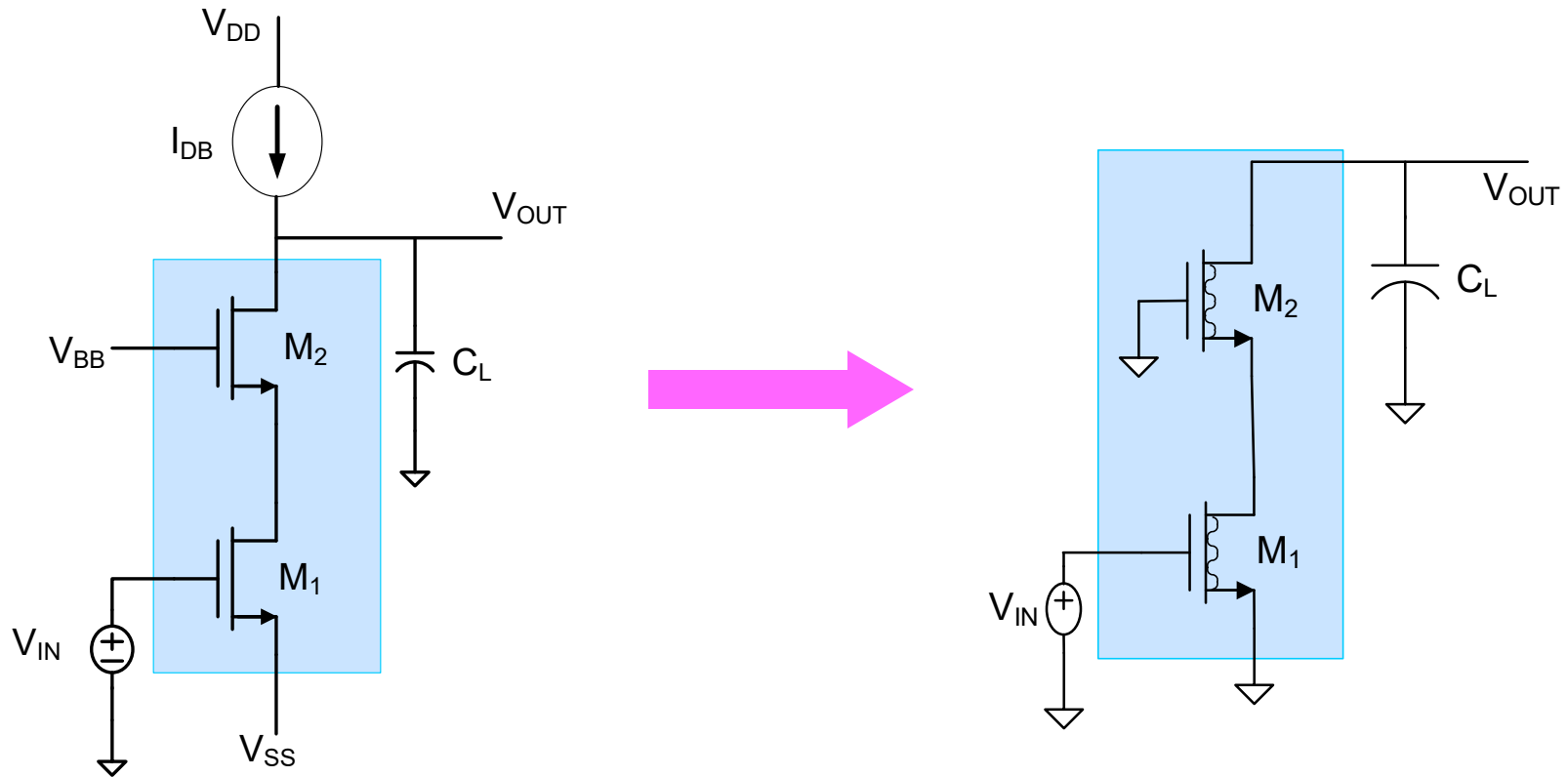
$$V_2(g_{o2} + sC_L) + g_{M2} V_{TST} = 0$$

$$A(s) = \frac{V_2(s)}{V_{TST}(s)} = -\frac{g_{M2}}{sC_L + g_{o2}}$$

(must express with coefficient of  $s$  in den equal to  $C_L$ )

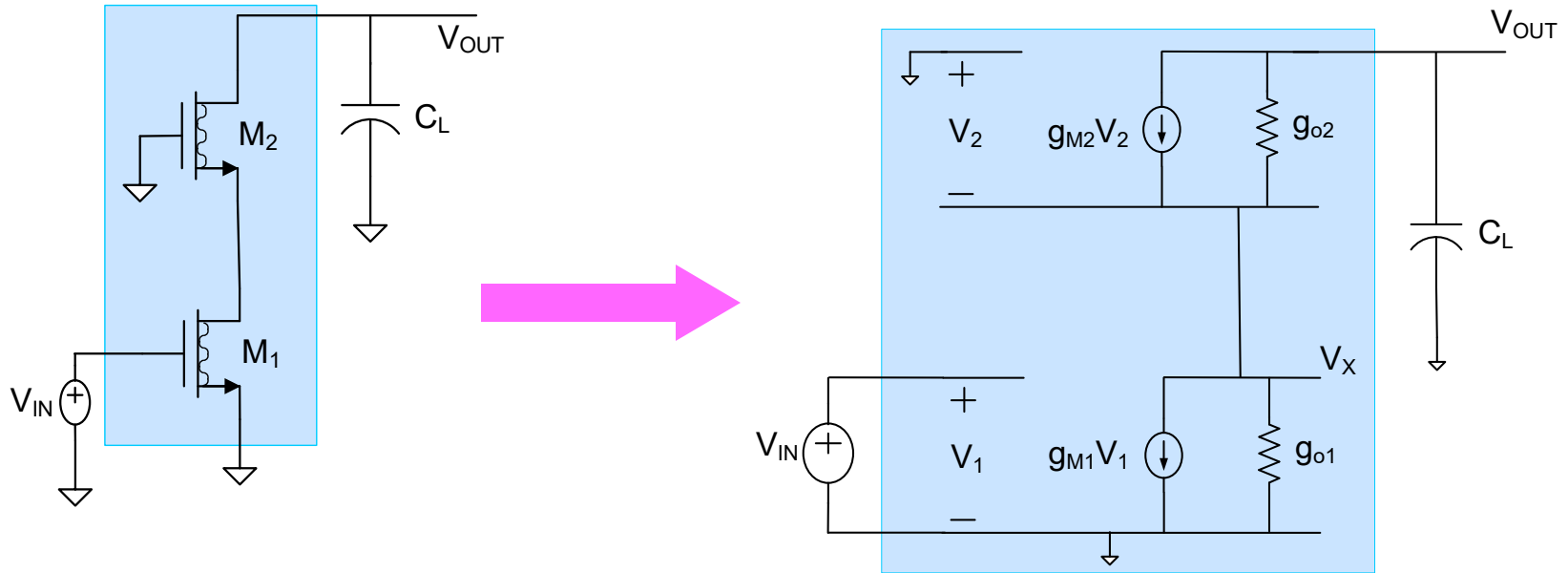
## Background

# Recall Cascode Amplifier (dc current source bias)



## Background

# Analysis of Cascode Amplifier (dc current source bias)



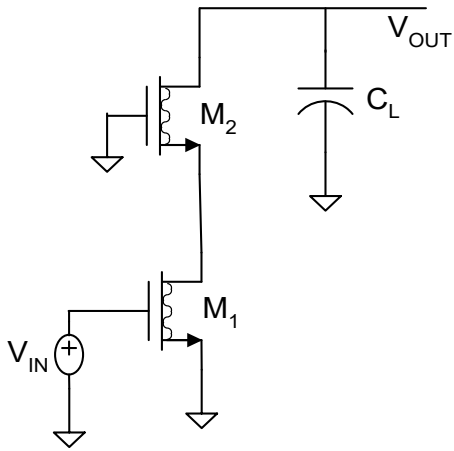
$$\left. \begin{aligned} \mathbf{V_{OUT}(g_{o2} + sC_L) + g_{m2}V_2} &= \mathbf{V_X g_{o2}} \\ \mathbf{V_X(g_{o1} + g_{o2}) + g_{m1}V_1 - g_{m2}V_2} &= \mathbf{V_{OUT}g_{o2}} \\ \mathbf{V_2} &= \mathbf{-V_X} \\ \mathbf{V_1} &= \mathbf{V_{IN}} \end{aligned} \right\}$$

$V_X, V_1$  and  $V_2$  can be eliminated from these 4 equations



## Background

# Analysis of Cascode Amplifier



$$\left. \begin{aligned} \mathbf{V}_{OUT}(\mathbf{g}_{o2} + \mathbf{s}C_L) + \mathbf{g}_{m2}\mathbf{V}_2 &= \mathbf{V}_X\mathbf{g}_{o2} \\ \mathbf{V}_X(\mathbf{g}_{o1} + \mathbf{g}_{o2}) + \mathbf{g}_{m1}\mathbf{V}_1 - \mathbf{g}_{m2}\mathbf{V}_2 &= \mathbf{V}_{OUT}\mathbf{g}_{o2} \\ \mathbf{V}_2 &= -\mathbf{V}_X \\ \mathbf{V}_1 &= \mathbf{V}_{IN} \end{aligned} \right\}$$

$$\left. \begin{aligned} \mathbf{V}_{OUT}(\mathbf{g}_{o2} + \mathbf{s}C_L) - \mathbf{g}_{m2}\mathbf{V}_X &= \mathbf{V}_X\mathbf{g}_{o2} \\ \mathbf{V}_X(\mathbf{g}_{o1} + \mathbf{g}_{o2}) + \mathbf{g}_{m1}\mathbf{V}_{IN} + \mathbf{g}_{m2}\mathbf{V}_X &= \mathbf{V}_{OUT}\mathbf{g}_{o2} \end{aligned} \right\}$$

$$\frac{\mathbf{V}_{OUT}}{\mathbf{V}_{IN}} = \frac{-\mathbf{g}_{m1}(\mathbf{g}_{o2} + \mathbf{g}_{m2})}{\mathbf{s}C_L(\mathbf{g}_{o1} + \mathbf{g}_{o2} + \mathbf{g}_{m2}) + \mathbf{g}_{o1}\mathbf{g}_{o2}} \approx \frac{-\mathbf{g}_{m1}\mathbf{g}_{m2}}{\mathbf{s}C_L\mathbf{g}_{m2} + \mathbf{g}_{o1}\mathbf{g}_{o2}}$$

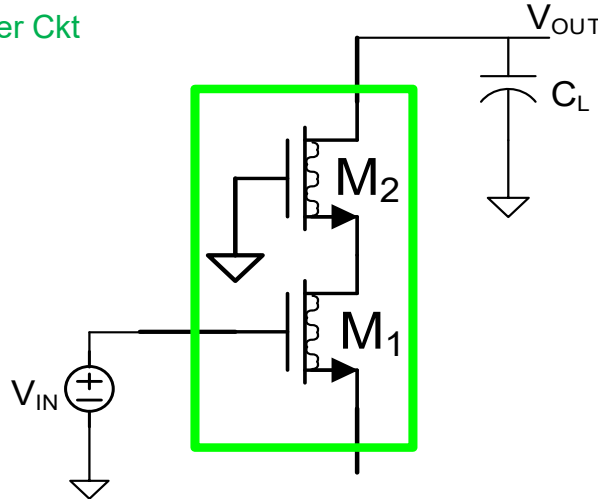
$$\frac{\mathbf{V}_{OUT}}{\mathbf{V}_{IN}} \approx \frac{-\mathbf{g}_{m1}}{\mathbf{s}C_L + \mathbf{g}_{o1}\left(\frac{\mathbf{g}_{o2}}{\mathbf{g}_{m2}}\right)}$$

$\mathbf{g}_{MEQ}$   
 $\mathbf{g}_{OEQ}$

# Comparison of Basic Amplifier and Cascode Amplifier

$$\frac{V_{OUT}}{V_{IN}} = \frac{-G_m}{sC_L + G}$$

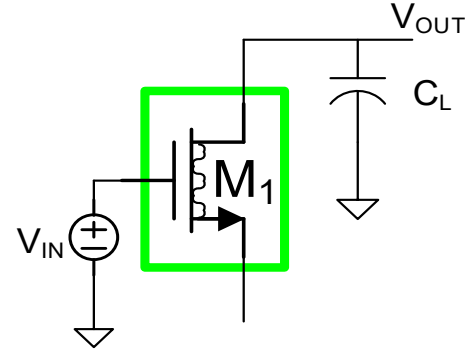
General form for any Quarter Ckt



$$\frac{V_{OUT}}{V_{IN}} \approx \frac{-g_{m1}}{sC_L + g_{o1} \left( \frac{g_{o2}}{g_{m2}} \right)}$$

$$G = g_{OEQ} \cong g_{O1} \left[ \frac{g_{O2}}{g_{m2}} \right]$$

$$G_M = g_{mEQ} \cong g_{m1}$$

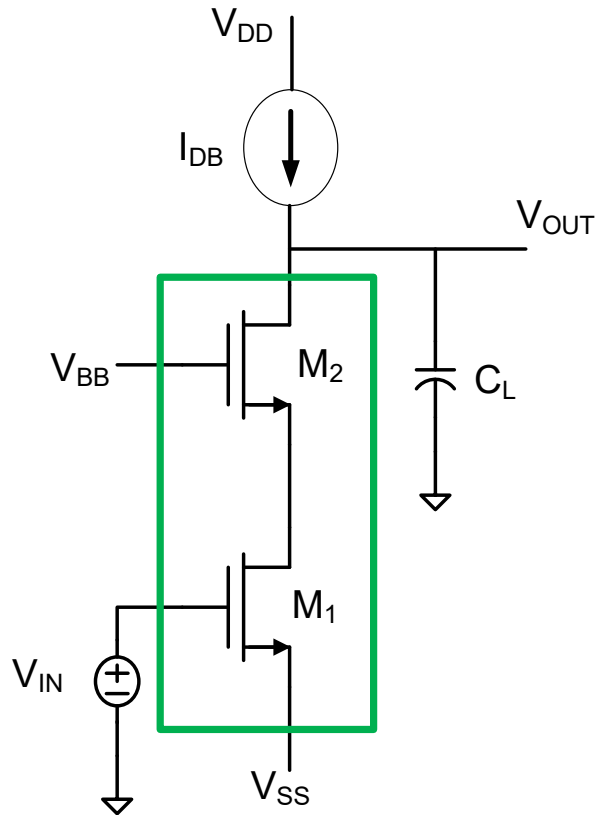


$$\frac{V_{OUT}}{V_{IN}} = \frac{-g_{m1}}{sC_L + g_{o1}}$$

$$G = g_{O1}$$

$$G_M = g_{m1}$$

# High output impedance quarter-circuits



**Cascode Amplifier**

$$G = g_{OEQ} \cong g_{O1} \left[ \frac{g_{O2}}{g_{m2}} \right]$$

$$G_M = g_{mEQ} \cong g_{m1}$$

- **Transconductance appears to be unchanged** (from basic single-device structure)
- **Output conductance appears to have been decreased !**

$$A_V(s) \cong \frac{-g_{m1}}{sC_L + g_{o1} \left[ \frac{g_{o2}}{g_{m2}} \right]}$$

$$A_{V0} \cong \left( \frac{g_{m1}}{g_{o1}} \right) \left[ \frac{g_{m2}}{g_{o2}} \right]$$

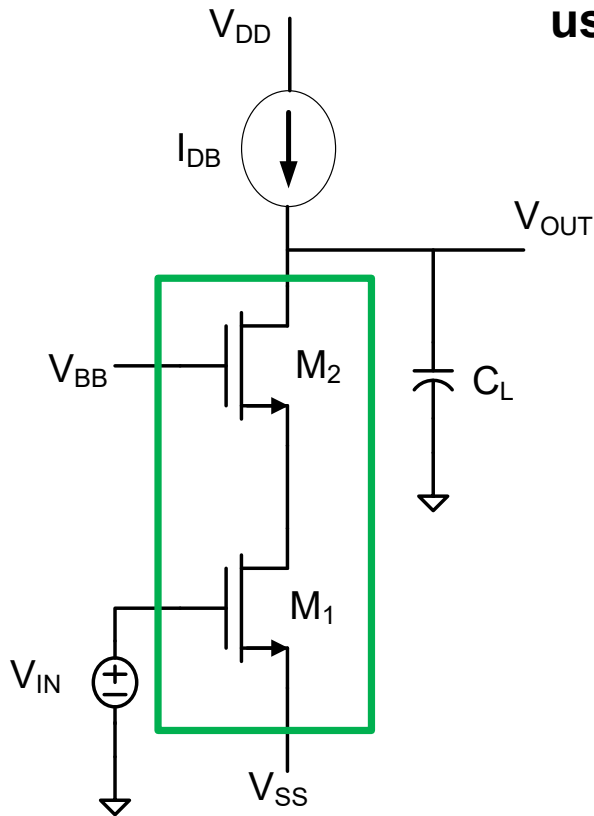
$$GB \cong \frac{g_{m1}}{C_L}$$

But must verify in the practical parameter domain to be sure!

# High output impedance quarter-circuits

How does this compare with previous amplifier using single transistor as quarter circuit?

Cascode amplifier quarter circuit:



**Cascode Amplifier**

$$A_{V0} \cong \left( \frac{g_{m1}}{g_{o1}} \right) \left[ \frac{g_{m2}}{g_{o2}} \right] \quad GB \cong \frac{g_{m1}}{C_L}$$

$$A_{V0} \cong \left[ \frac{2}{\lambda_1 V_{EB1}} \right] \cdot \left[ \frac{2}{\lambda_2 V_{EB2}} \right]$$

$$GB = \left[ \frac{2P}{V_{DD} C_L} \right] \cdot \left[ \frac{1}{V_{EB1}} \right]$$

Single transistor quarter circuit:

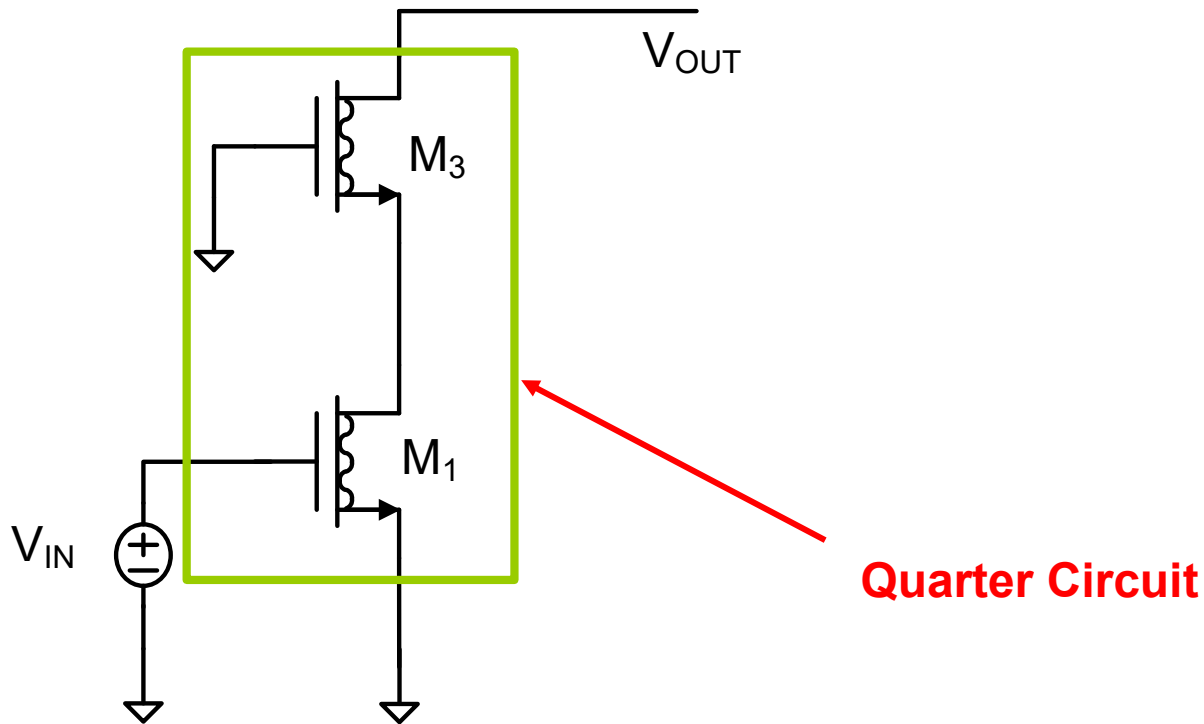
$$A_{V0} = \left[ \frac{2}{\lambda V_{EB}} \right]$$

$$GB = \left( \frac{2P}{V_{DD} C_L} \right) \cdot \left( \frac{1}{V_{EB}} \right)$$

**Substantial increase in dc gain**

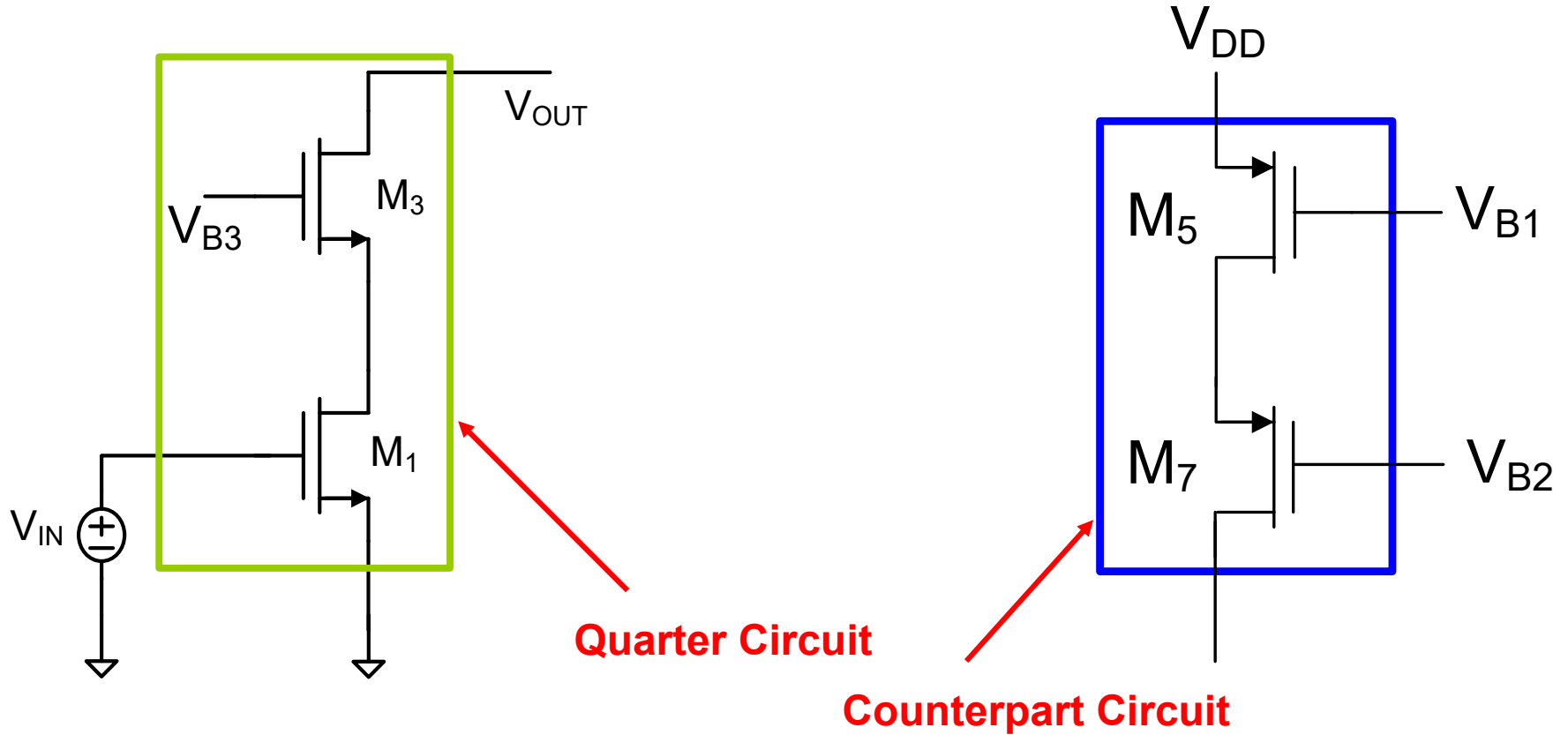
**No improvement in GB but also no deterioration in GB !**

# High output impedance quarter-circuits



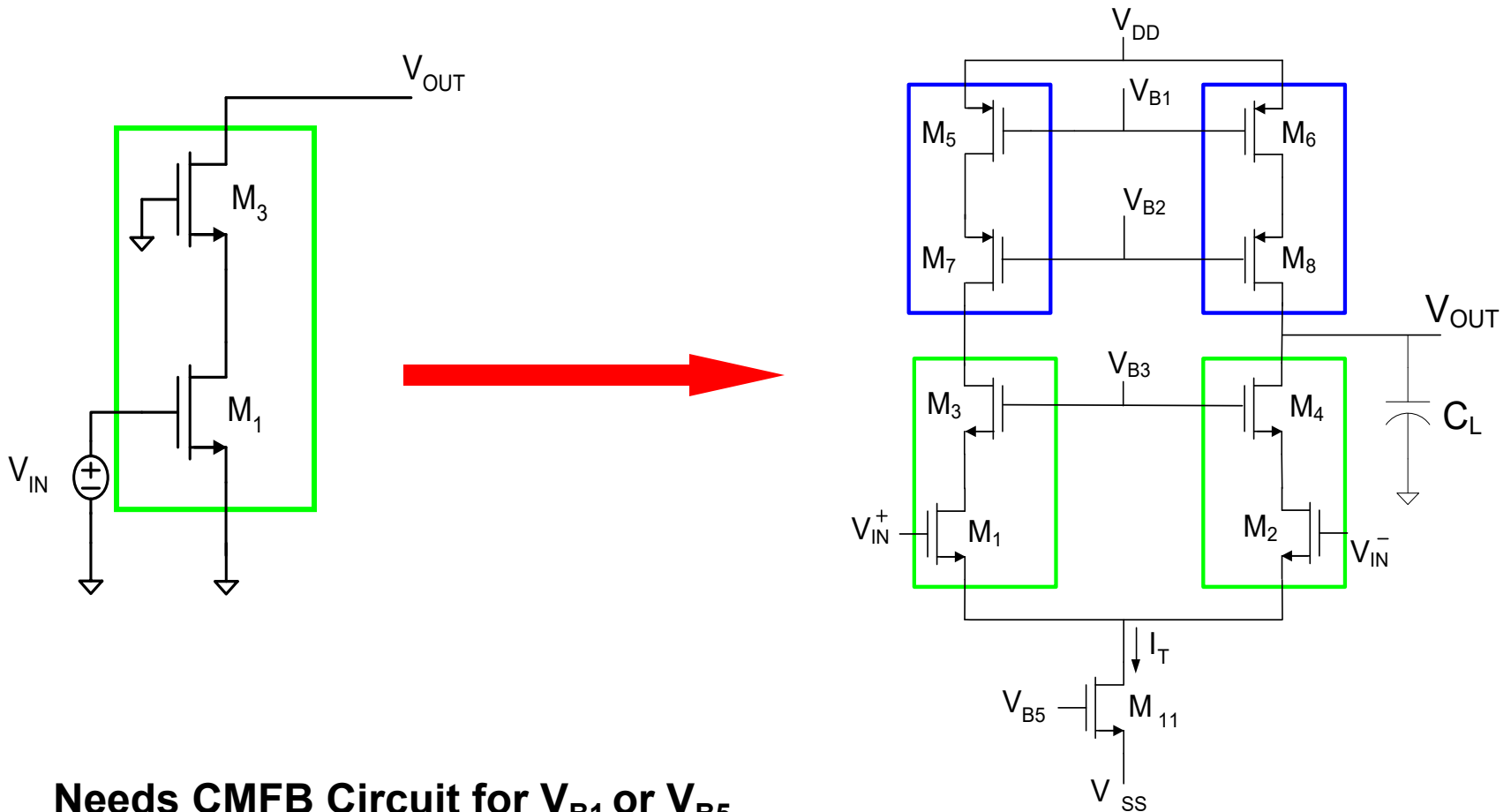
**Cascode Amplifier  
(small-signal equiv)**

# High output impedance quarter-circuits



**Cascode Amplifier**

# Telescopic Cascode Op Amp

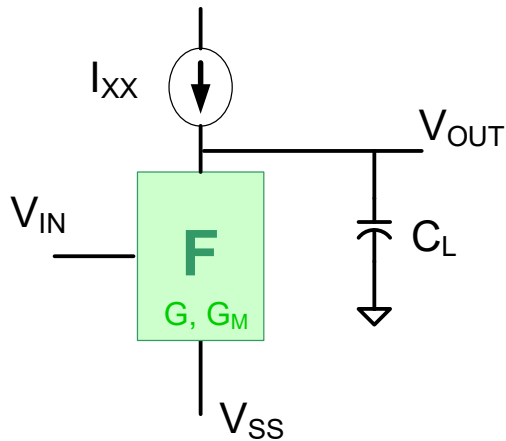


**Needs CMFB Circuit for  $V_{B1}$  or  $V_{B5}$**   
**Either single-ended or differential outputs**  
**Can connect counterpart as current mirror to eliminate CMFB**

# Recall:

## Determination of op amp characteristics from quarter circuit characteristics (for single-ended gain)

Small signal Quarter Circuit



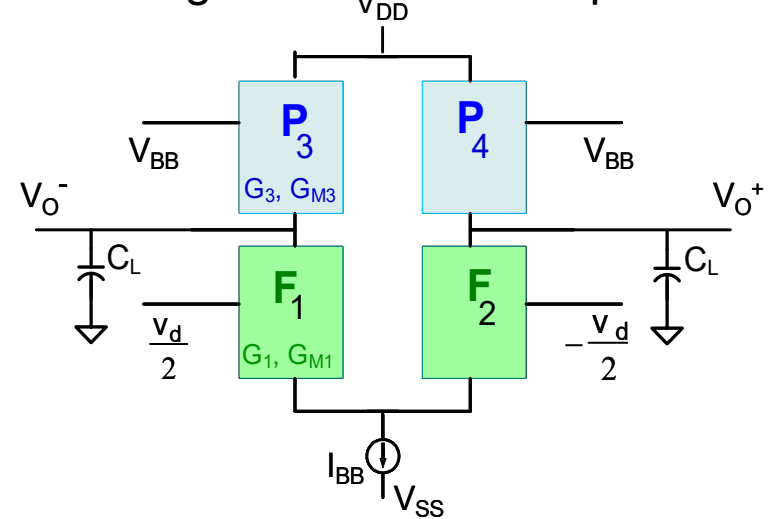
$$A_{V_{OQC}} = -\frac{G_M}{G}$$

$$BW = \frac{G}{C_L}$$

$$GB = \frac{G_M}{C_L}$$

$$A(s) = \frac{-G_M}{sC_L + G}$$

Small signal differential amplifier



$$A_{V_0} = \frac{-\frac{G_{M1}}{2}}{(G_1 + G_3)}$$

$$BW = \frac{G_1 + G_3}{C_L}$$

$$GB = \frac{G_{M1}}{2C_L}$$

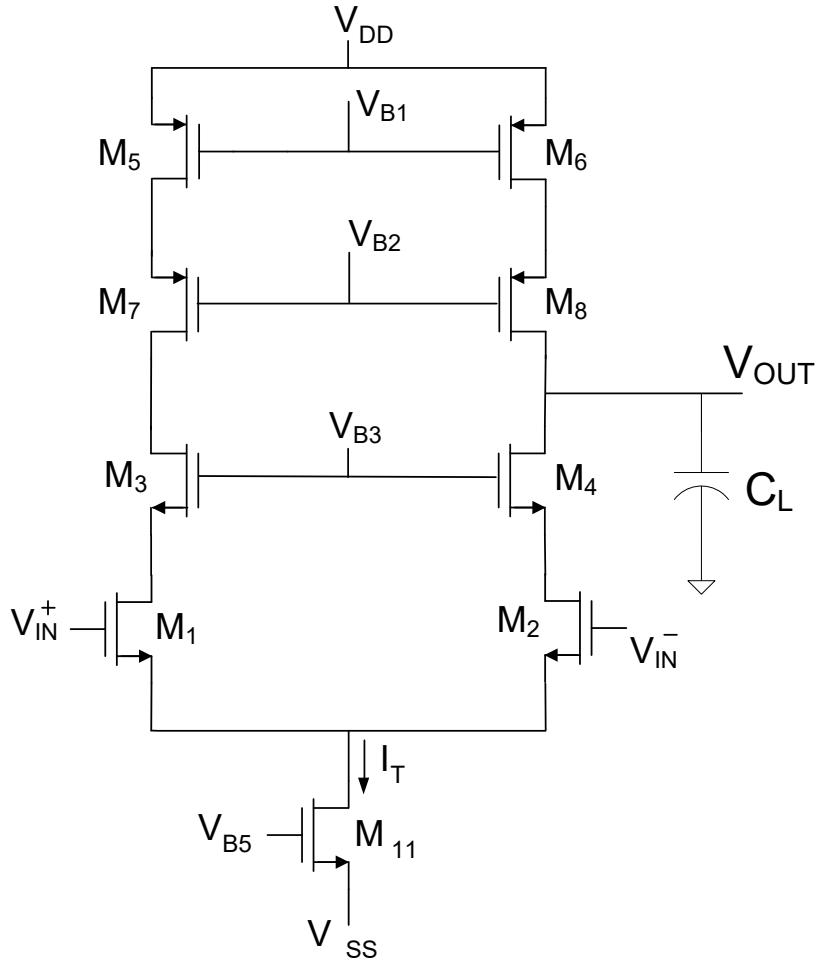
$$A(s) = \frac{-\frac{G_{M1}}{2}}{sC_L + G_1 + G_3}$$



Note: Factor of 4 reduction of single-ended gain (for G<sub>1</sub>=G<sub>2</sub>)



# Telescopic Cascode Op Amp



**Single-ended operation**

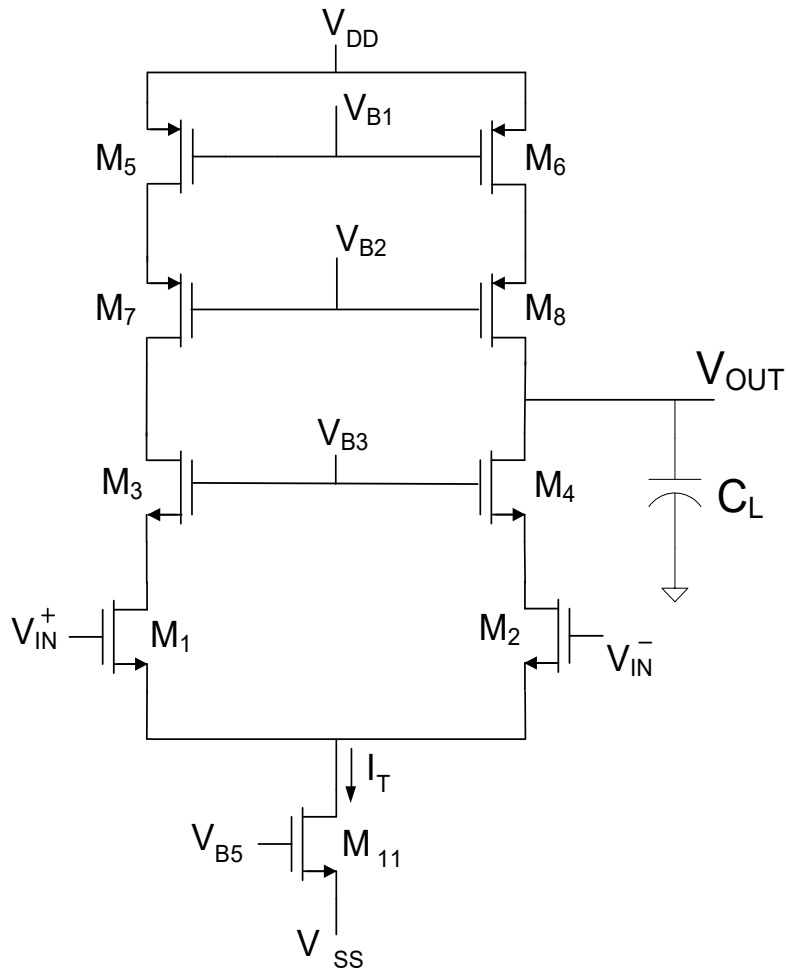
$$g_{oQC} = \underline{\hspace{10em}}$$

$$g_{oCC} = \underline{\hspace{10em}}$$

$$g_{mQC} = \underline{\hspace{10em}}$$

# Telescopic Cascode Op Amp

## Single-ended operation

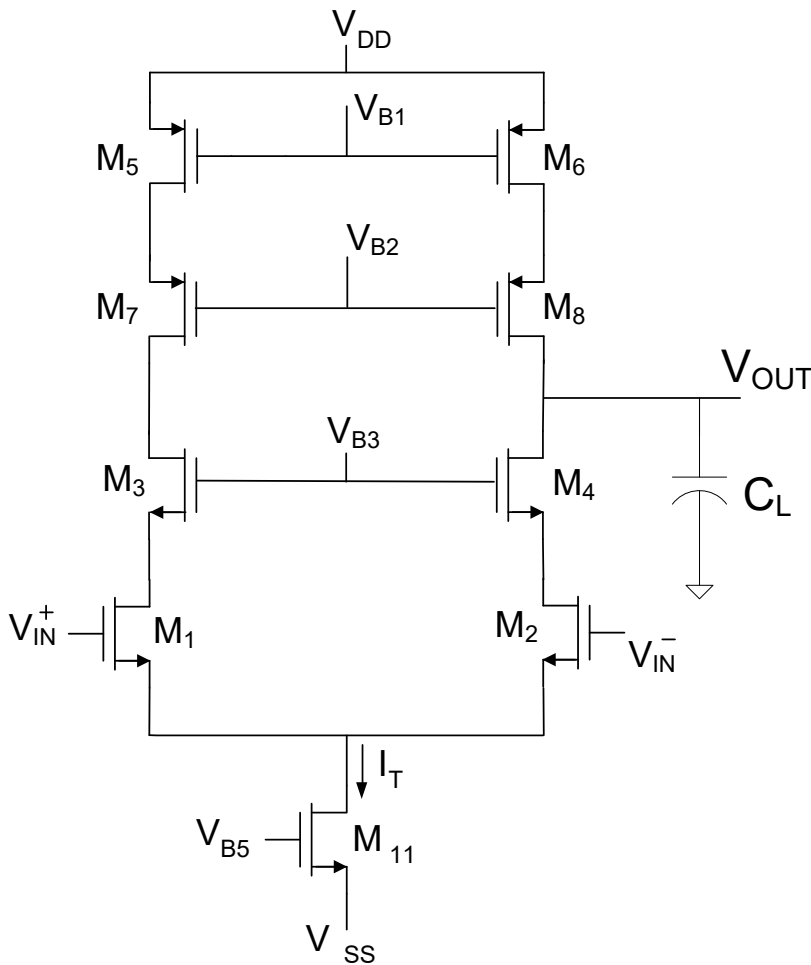


$$A_d(s) = \frac{-\frac{g_{m1}}{2}}{sC_L + g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

$$A_o = \frac{-\frac{g_{m1}}{2}}{g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

$$GB = \frac{g_{m1}}{2C_L}$$

# Telescopic Cascode Op Amp



(CMFB circuit not shown)

## Single-ended operation

$$A_d(s) = \frac{-\frac{g_{m1}}{2}}{sC_L + g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

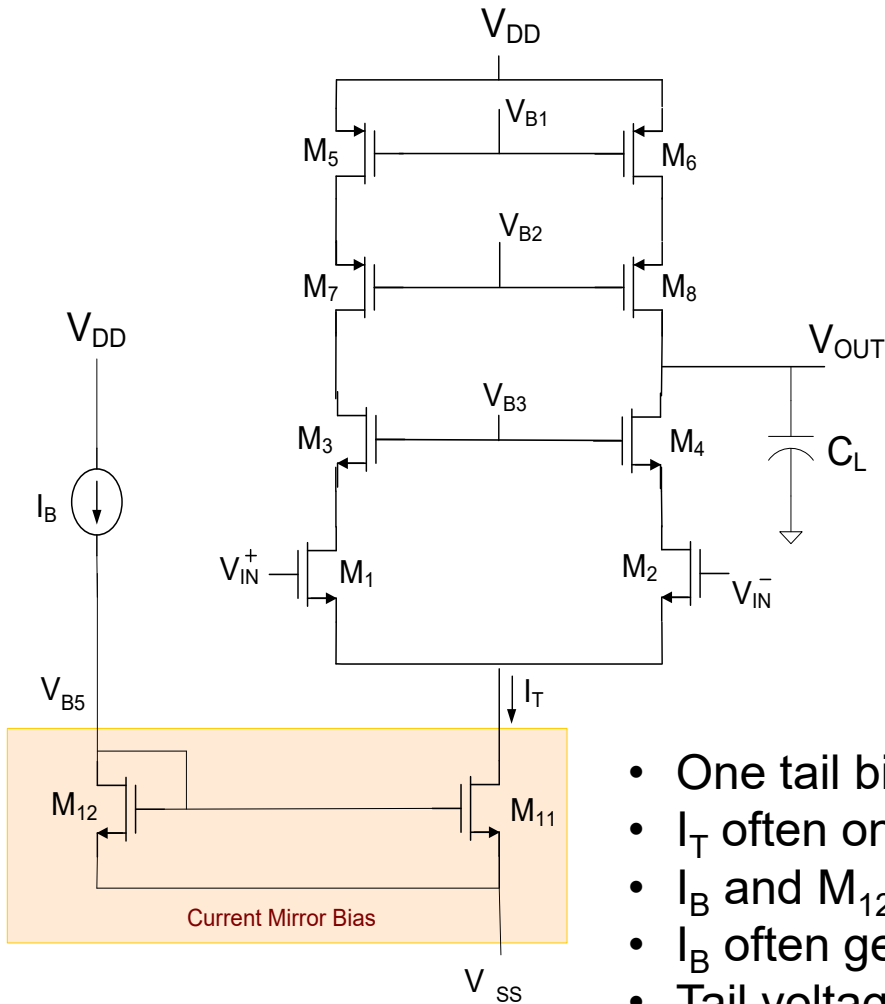
$$A_0 = \frac{-\frac{g_{m1}}{2}}{g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

$$GB = \frac{g_{m1}}{2C_L}$$

- Large improvement in  $A_0$
- No change in GB

This circuit is widely used !!

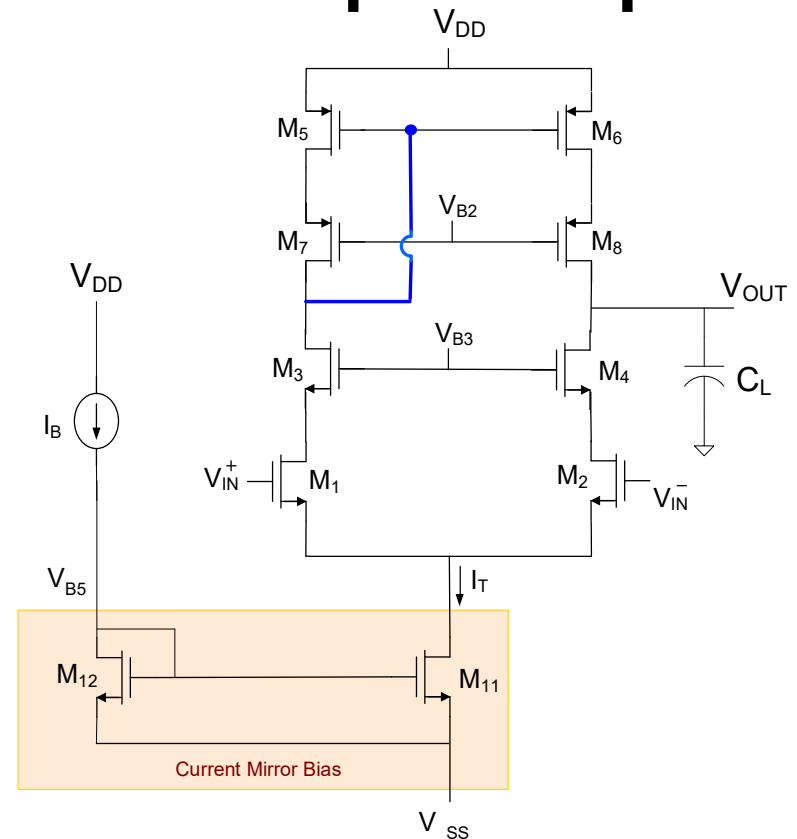
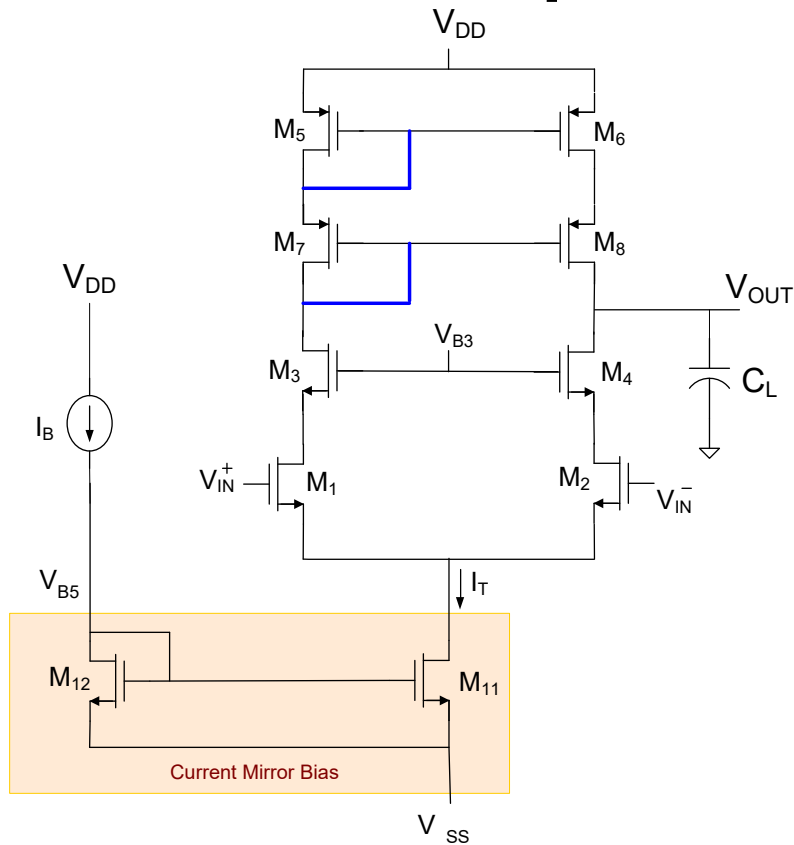
# Telescopic Cascode Op Amp



- One tail bias current generator shown
- $I_T$  often one of many outputs of a current mirror
- $I_B$  and  $M_{12}$  often common to many blocks
- $I_B$  often generated from a reference generator circuit
- Tail voltage bias can also be used

(CMFB circuit not shown)

# Telescopic Cascode Op Amp

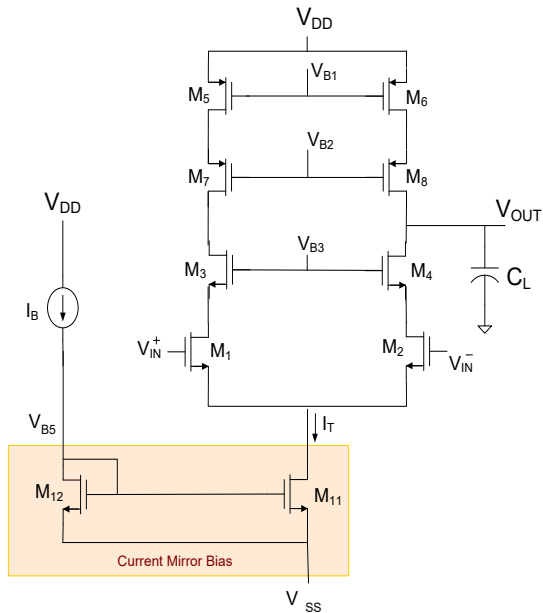


**Standard p-channel Cascode Mirror**

**Wide-Swing p-channel Cascode Mirror**

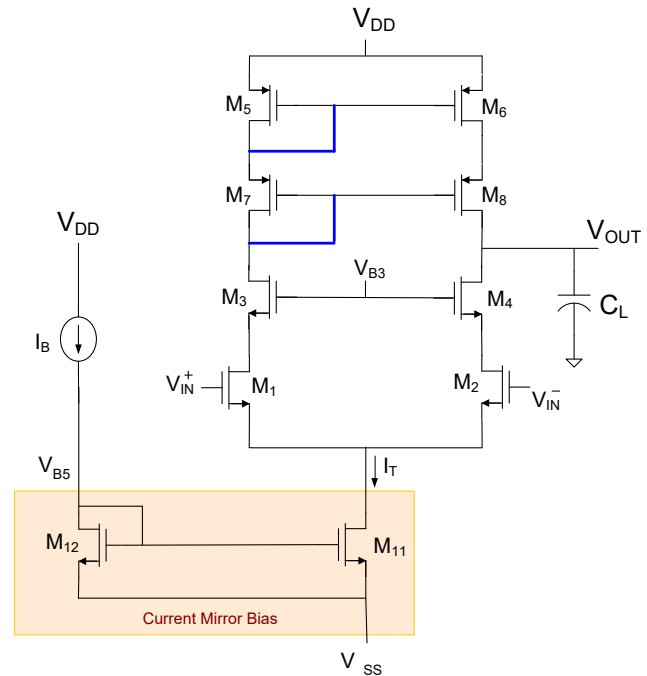
- Current-Mirror p-channel Bias to Eliminate CMFB
- Only single-ended output available
- Doubles dc gain

# Telescopic Cascode Op Amp

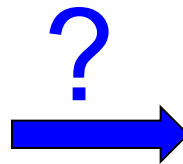


(CMFB circuit needed)

$$A_d(s) = \frac{-\frac{g_{m1}}{2}}{sC_L + g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

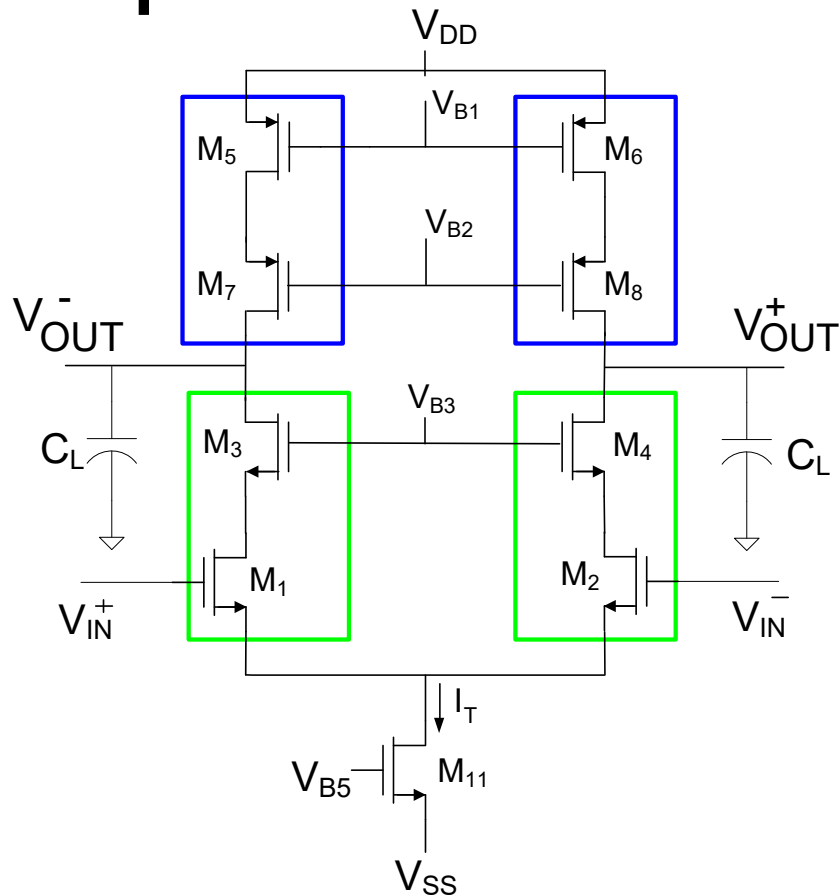


(No CMFB circuit needed)



$$A_d(s) = \frac{-g_{m1}}{sC_L + g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

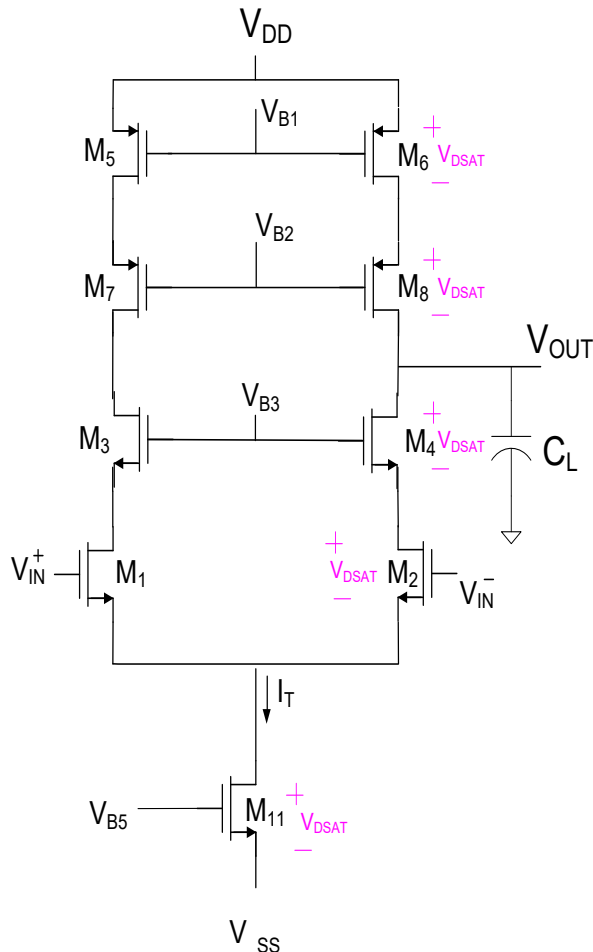
# Telescopic Cascode Op Amp



- Differential Output
- CMFB to establish  $V_{B1}$  or  $V_{B5}$  needed
- Tail current generally generated with current mirror

# Telescopic Cascode Op Amp

## Signal Swing and Power Supply Limitations



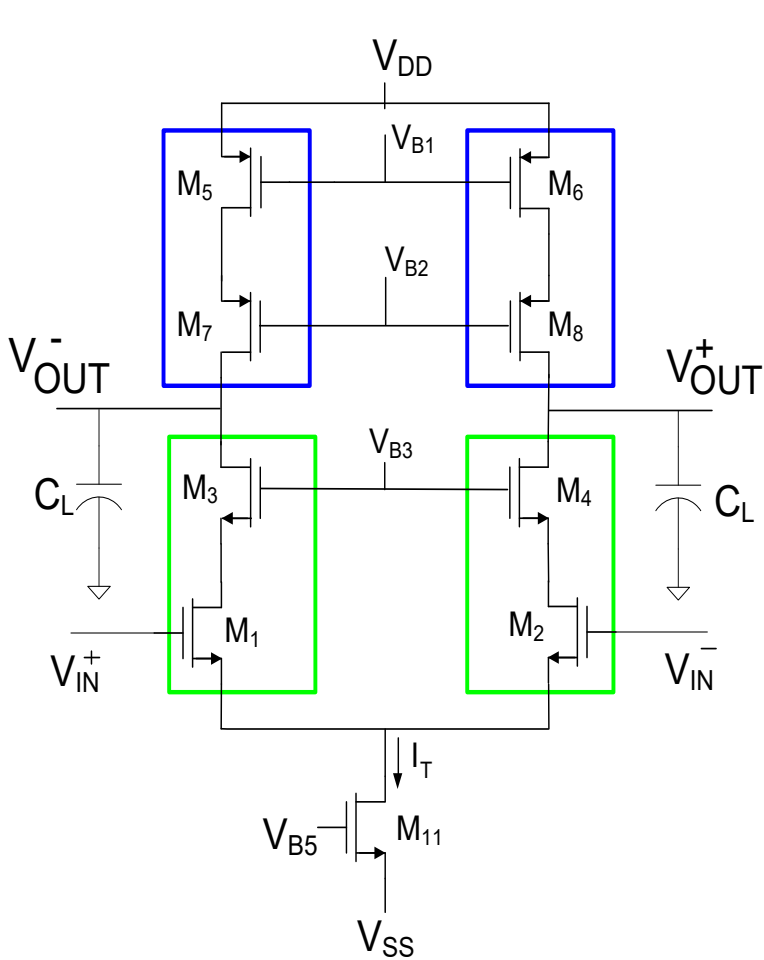
There are a minimum of  $2 V_{DSAT}$  drops between  $V_{OUT}$  and  $V_{DD}$  and a minimum of  $3 V_{DSAT}$  drops between  $V_{OUT}$  and  $V_{SS}$

Thus, there are a minimum of  $5 V_{DSAT}$  drops between  $V_{DD}$  and  $V_{SS}$

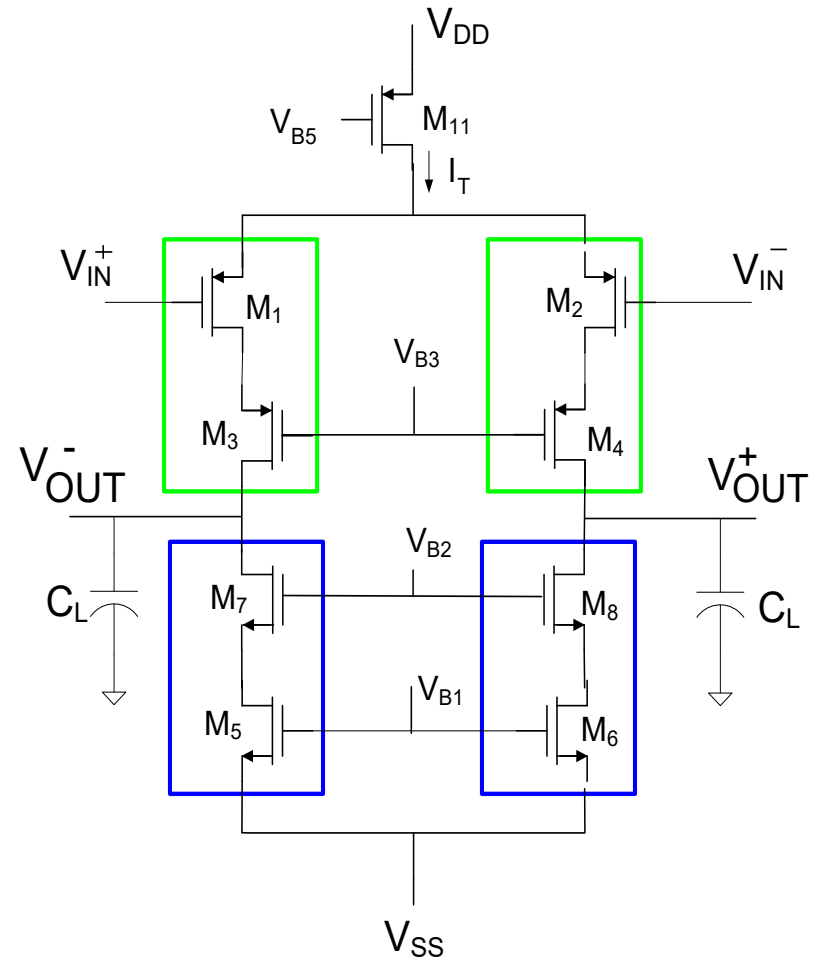
This establishes a lower bound on  $V_{DD} - V_{SS}$  and it will be further reduced by the p-p signal swing required on the output



# Telescopic Cascode Op Amp



n-channel inputs

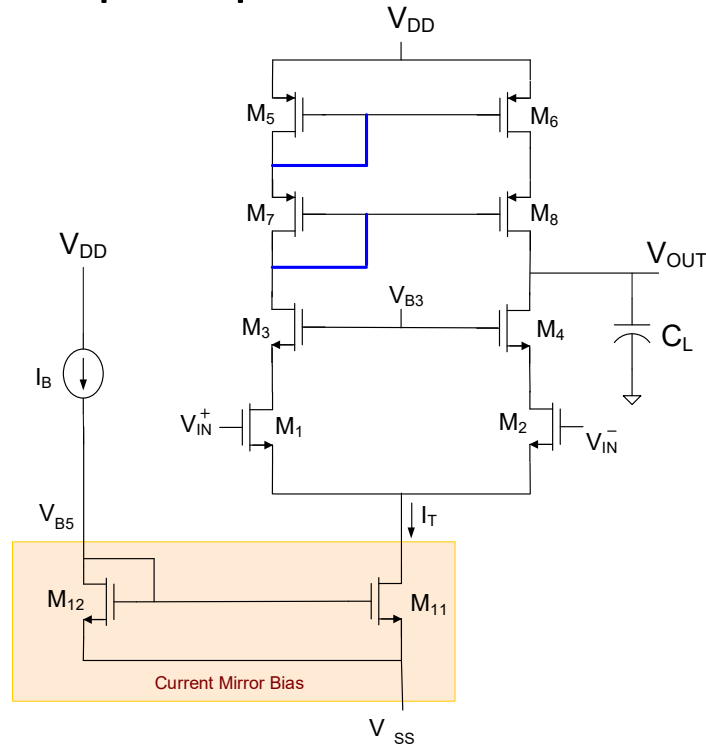


p-channel inputs

How important is it to develop good approximate analysis methods for an op amp with the complexity of the telescopic cascode structure?

And many useful op amp circuits will have more complexity !!

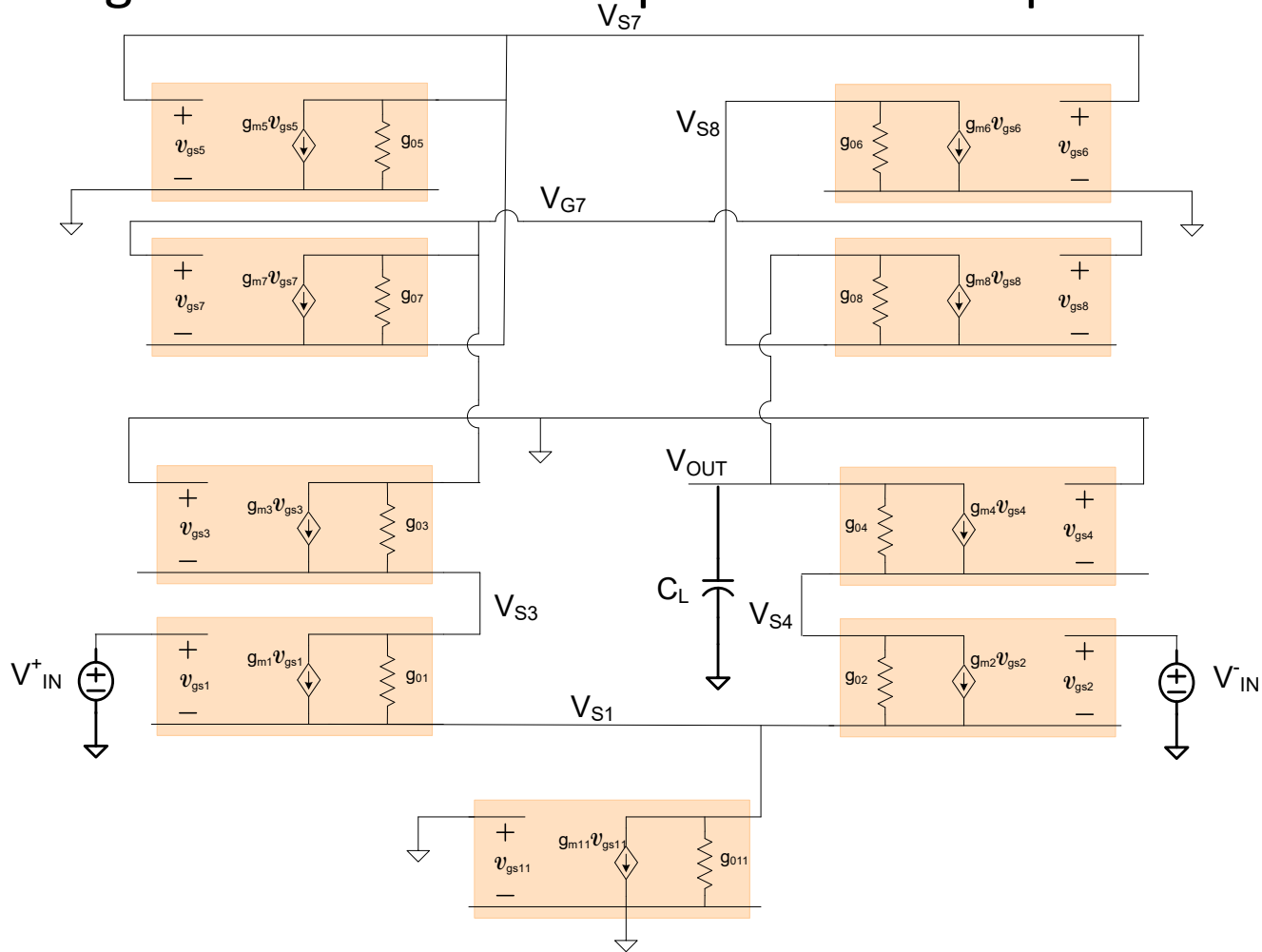
# Telescopic Cascode Op Amp with Mirror-connected Counterpart Circuit



- Previous analysis obtained almost by inspection
- Gain expressions were real simple
- Gain expressions provide major insight into operation
- Some assumptions were made to simplify analysis
  - $V_{ac}=0$  at “approximate axis of symmetry”
  - Matched left and right side transistors
  - Current mirror used to mirror left-side current to right side

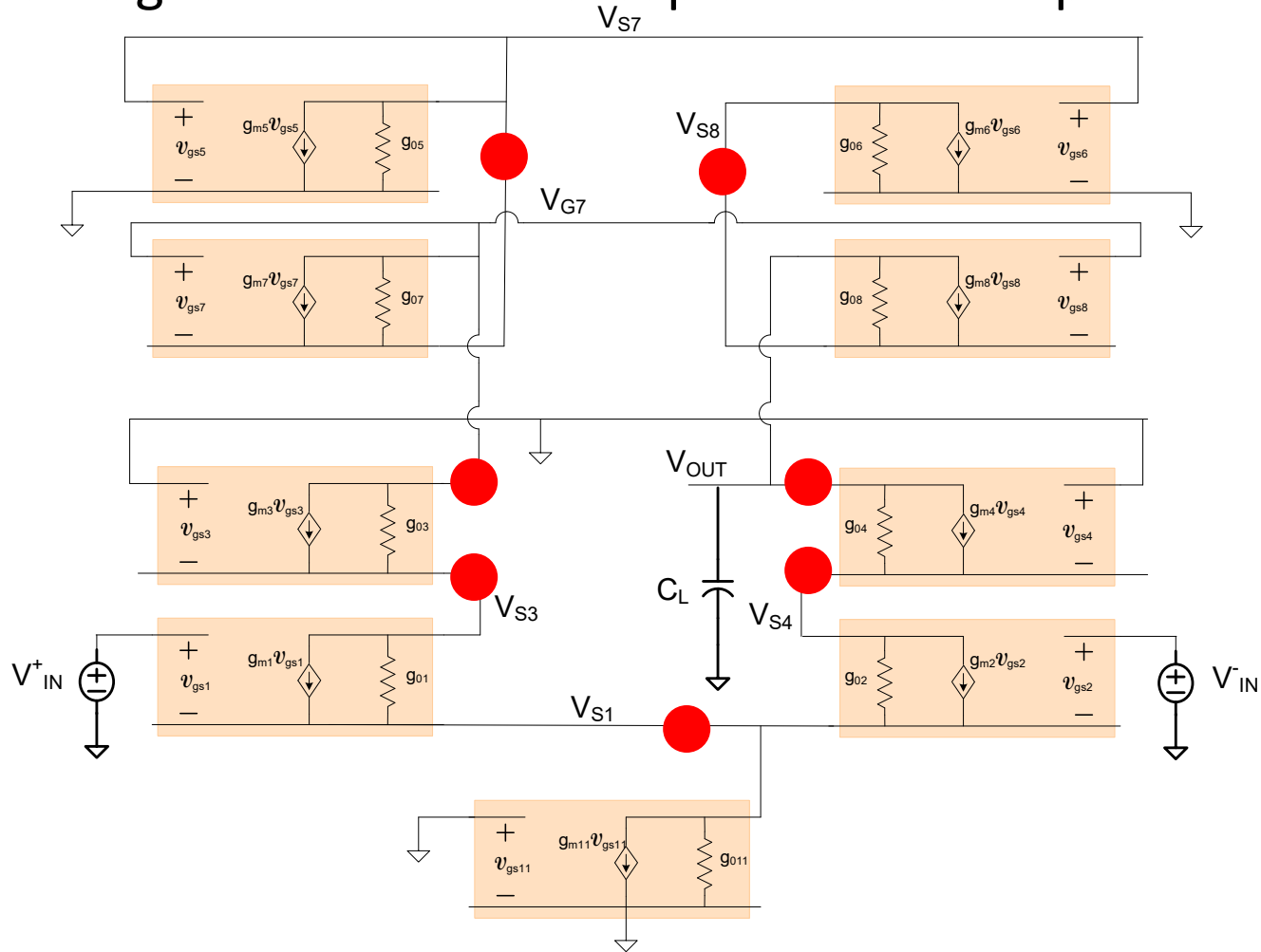
How much more involved would an exact analysis be ?  
Would an exact analysis provide additional insight ?

# Small-Signal model of Telescopic Cascode Amplifier



A bit tedious to obtain but really straight forward

# Small-Signal model of Telescopic Cascode Amplifier



A bit tedious to obtain but really straight forward

● Non-ground nodes for small-signal nodal analysis

# Analysis of Telescopic Cascode Amplifier

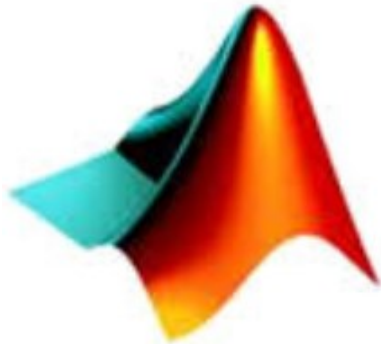
Apply KCL at 7 nodes to obtain a set of 7 independent linear equations

$$\left. \begin{aligned}
 V_{S1} (g_{01} + g_{02} + g_{011}) &= V_{S3} g_{01} + V_{S4} g_{02} + g_{m1} (V_{IN}^+ - V_{S1}) + g_{m2} (V_{IN}^- - V_{S1}) \\
 V_{S3} (g_{01} + g_{03}) + g_{m1} (V_{IN}^+ - V_{S1}) &= g_{01} V_{S1} + g_{03} V_{G7} - g_{m3} V_{S3} \\
 V_{S4} (g_{02} + g_{04}) + g_{m2} (V_{IN}^- - V_{S1}) &= g_{02} V_{S1} + g_{04} V_{OUT} - g_{m4} V_{S4} \\
 V_{OUT} (sC_L + g_{04} + g_{08}) - g_{m4} V_{S4} + g_{m8} (V_{G7} - V_{S8}) &= g_{04} V_{S4} + g_{08} V_{S8} \\
 V_{G7} (g_{07} + g_{03}) + g_{m7} (V_{G7} - V_{S7}) - g_{m3} V_{S3} &= g_{03} V_{S3} + g_{07} V_{S7} \\
 V_{S8} (g_{06} + g_{08}) + g_{m8} V_{S7} &= g_{m8} (V_{G7} - V_{S8}) + V_{OUT} g_{08} \\
 V_{S7} (g_{05} + g_{07}) + g_{m5} V_{S7} &= g_{m7} (V_{G7} - V_{S7}) + g_{07} V_{G7}
 \end{aligned} \right\}$$

A bit tedious to obtain but really straight forward

Time required to obtain set of equations is quite small



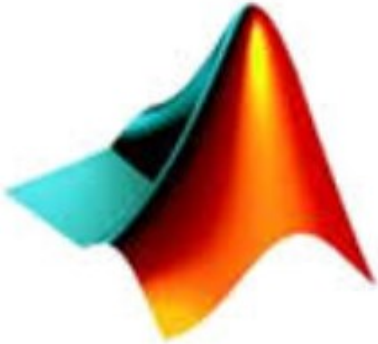


# MATLAB

```
%Complete Analysis of Telescopic Cascode Op Amp
diary('matlabdiary')
clear
clc
fprintf('Complete Analysis of Telescopic Cascode Op Amp')
syms Av s CL VOUT VS1 VS3 VS4 VS7 VS8 VG7 Vinn Vinp g01 g02 g03 g04 g05
g06 g07 g08 g011 gm1 gm2 gm3 gm4 gm5 gm6 gm7 gm8
eqn1 =VS1*(gm1+gm2+g01+g02+g011)==VS3*g01+VS4*g02+Vinp*gm1+gm2*Vinn;
eqn2 = VS3*(gm3+g01+g03)+gm1*Vinp==VS1*(g01+gm1)+VG7*g03;
eqn3 = VS4*(gm4+g02+g04)+Vinn*gm2==VS1*(gm2+g02)+VOUT*g04;
eqn4 = VOUT*(s*CL+g04+g08)+VG7*gm8==VS4*(gm4+g04)+VS8*(gm8+g08);
eqn5 = VG7*(gm7+g07+g03)==VS3*(gm3+g03)+VS7*(gm7+g07);
eqn6 = VS8*(gm8+g06+g08)+VS7*gm8==VG7*gm8+VOUT*g08;
eqn7 = VS7*(gm7+gm5+g05+g07)==VG7*(gm7+g07);
eqn8 = Av==VOUT/Vinp;
sol=solve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8],[VOUT,VS1,VS3,VS4,VS7,
VS8,VG7,Av]);
VOUTX=sol.VOUT
collect(VOUTX,Vinn)
```

Simple program and effort to write program is small





# MATLAB

```
fprintf('Differential Analysis of Telescopic Cascode Op Amp')
syms Av s CL VOUT VS1 VS3 VS4 VS7 VS8 VG7 Vdiff g01 g02 g03 g04 g05 g06 g07 g08 g011 gm1 gm2 gm3
gm4 gm5 gm6 gm7 gm8
eqn1 = VS1*(gm1+gm2+g01+g02+g011)==VS3*g01+VS4*g02+Vdiff*gm1/2-gm2*Vdiff/2;
eqn2 = VS3*(gm3+g01+g03)+gm1*Vdiff/2==VS1*(g01+gm1)+VG7*g03;
eqn3 = VS4*(gm4+g02+g04)-Vdiff*gm2/2==VS1*(gm2+g02)+VOUT*g04;
eqn4 = VOUT*(s*CL+g04+g08)+VG7*gm8==VS4*(gm4+g04)+VS8*(gm8+g08);
eqn5 = VG7*(gm7+g07+g03)==VS3*(gm3+g03)+VS7*(gm7+g07);
eqn6 = VS8*(gm8+g06+g08)+VS7*gm8==VG7*gm8+VOUT*g08;
eqn7 = VS7*(gm7+gm5+g05+g07)==VG7*(gm7+g07);
eqn8 = Av==VOUT/Vdiff;
sol=solve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8],[VOUT,VS1,VS3,VS4,VS7,VS8,VG7,Av]);
VOUTX=sol.VOUT;
GainAv=sol.Av
collect(GainAv,s)
```

Simple program and effort to write program is small







# MATLAB Solution Continued 3:

```
CL*g01*g03*g05*g07*g08*gm4*s + CL*g01*g04*g05*g07*g08*gm3*s + CL*g03*g04*g05*g07*g08*gm1*s +
CL*g02*g03*g04*g05*g07*gm8*s + CL*g02*g03*g04*g05*g08*gm7*s + CL*g02*g03*g04*g07*g08*gm5*s +
CL*g02*g03*g05*g07*g08*gm4*s + CL*g02*g04*g05*g07*g08*gm3*s + CL*g03*g04*g05*g07*g08*gm2*s +
CL*g01*g02*g03*g05*g011*gm8*s + CL*g01*g02*g03*g06*g011*gm7*s + CL*g01*g02*g03*g08*g011*gm5*s +
CL*g01*g03*g04*g06*g011*gm5*s + CL*g01*g03*g05*g06*g011*gm4*s + CL*g01*g02*g03*g07*g011*gm8*s +
CL*g01*g02*g03*g08*g011*gm7*s + CL*g01*g02*g05*g06*g011*gm7*s + CL*g01*g02*g06*g07*g011*gm5*s +
CL*g01*g03*g04*g05*g011*gm8*s + CL*g01*g03*g04*g06*g011*gm7*s + CL*g01*g03*g04*g08*g011*gm5*s +
CL*g01*g03*g05*g08*g011*gm4*s + CL*g01*g03*g06*g07*g011*gm4*s + CL*g01*g02*g05*g07*g011*gm8*s +
CL*g01*g04*g06*g07*g011*gm5*s + CL*g01*g05*g06*g07*g011*gm4*s + CL*g02*g03*g05*g06*g011*gm7*s +
CL*g02*g03*g06*g07*g011*gm3*s + CL*g02*g03*g05*g08*g011*gm7*s + CL*g02*g03*g07*g08*g011*gm5*s +
CL*g02*g03*g05*g07*g011*gm8*s + CL*g02*g03*g05*g08*g011*gm7*s + CL*g02*g03*g07*g08*g011*gm5*s +
CL*g02*g05*g07*g08*g011*gm3*s + CL*g03*g04*g05*g06*g011*gm7*s + CL*g03*g04*g06*g07*g011*gm5*s +
CL*g03*g04*g05*g08*g011*gm7*s + CL*g03*g04*g07*g08*g011*gm3*s + CL*g03*g04*g05*g07*g08*g011*gm4*s +
CL*g03*g04*g06*g07*g08*g011*gm5*s + CL*g03*g04*g07*g08*g011*gm3*s + CL*g03*g04*g05*g07*g08*g011*gm4*s +
CL*g01*g02*g03*g08*gm4*s + CL*g01*g02*g04*g06*gm5*s + CL*g01*g02*g03*g07*gm4*s + CL*g01*g02*g06*gm2*gm4*s +
CL*g01*g03*g07*g08*gm2*gm4*s + CL*g01*g04*g05*g06*gm2*gm7*s + CL*g01*g04*g06*g07*gm2*gm5*s +
CL*g01*g05*g06*g07*gm2*gm4*s + CL*g01*g02*g03*g05*gm7*s + CL*g01*g02*g03*g07*gm5*gm8*s +
CL*g01*g02*g03*g08*gm5*gm7*s + CL*g01*g02*g05*g07*gm3*gm8*s + CL*g01*g02*g05*g08*gm3*gm7*s +
CL*g01*g02*g07*g08*gm3*gm5*s + CL*g01*g03*g04*g06*gm5*gm7*s + CL*g01*g03*g04*g08*gm5*gm7*s +
CL*g01*g03*g05*g08*gm4*s + CL*g01*g03*g07*g08*gm4*gm5*s + CL*g01*g03*g05*g07*gm4*gm8*s +
CL*g01*g03*g08*gm4*gm7*s + CL*g01*g04*g05*g08*gm3*gm7*s + CL*g01*g04*g07*g08*gm3*gm5*s +
CL*g03*g04*g05*g07*gm1*gm8*s + CL*g03*g04*g05*g08*gm1*gm7*s + CL*g03*g04*g07*g08*gm1*gm5*s +
CL*g03*g04*g06*g07*gm1*gm4*s + CL*g03*g05*g06*g07*gm1*gm3*s + CL*g04*g05*g06*g07*gm1*gm3*s +
CL*g01*g02*g04*g05*gm7*gm8*s + CL*g01*g02*g04*g07*gm5*gm8*s + CL*g01*g02*g04*g08*gm5*gm7*s +
CL*g01*g04*g05*g07*gm2*gm8*s + CL*g01*g04*g05*g08*gm2*gm7*s + CL*g01*g04*g07*g08*gm2*gm5*s +
CL*g01*g05*g07*g08*gm2*gm4*s + CL*g02*g03*g04*g06*gm5*gm7*s + CL*g02*g03*g05*g06*gm4*gm7*s +
CL*g02*g03*g06*g07*gm4*gm5*s + CL*g02*g04*g05*g06*gm3*gm7*s + CL*g02*g04*g06*g07*gm3*gm5*s +
CL*g02*g05*g06*g07*gm3*gm4*s + CL*g03*g04*g05*g06*gm2*gm7*s + CL*g03*g04*g06*g07*gm2*gm5*s +
CL*g03*g05*g06*g07*gm2*gm4*s + CL*g04*g05*g06*g07*gm2*gm3*s + CL*g01*g03*g04*g05*gm7*gm8*s +
CL*g01*g03*g04*g07*gm5*gm8*s + CL*g01*g03*g04*g08*gm5*gm7*s + CL*g01*g03*g05*g07*gm4*gm8*s +
CL*g01*g03*g08*gm4*gm7*s + CL*g01*g03*g07*g08*gm4*gm5*s + CL*g01*g04*g05*g08*gm3*gm7*s +
CL*g01*g04*g07*g08*gm3*gm5*s + CL*g01*g05*g07*g08*gm3*gm4*s + CL*g03*g04*g07*g08*gm1*gm5*s +
CL*g03*g04*g05*g07*gm1*gm4*s + CL*g04*g05*g06*g07*gm1*gm3*s + CL*g02*g03*g04*g05*gm7*gm8*s +
CL*g02*g03*g04*g07*gm5*gm8*s + CL*g02*g03*g05*g07*gm4*gm8*s + CL*g02*g03*g06*g07*gm1*gm3*gm5*s +
CL*g01*g04*g08*gm1*gm5*gm7*s + CL*g01*g05*g07*gm1*gm4*gm8*s + CL*g01*g05*g08*gm1*gm4*gm7*s +
CL*g01*g07*g08*gm1*gm4*gm5*s + CL*g02*g03*g08*gm1*gm5*gm7*s + CL*g02*g05*g08*gm1*gm3*gm7*s +
CL*g02*g07*g08*gm1*gm3*gm5*s + CL*g03*g04*g06*gm1*gm5*gm7*s + CL*g03*g05*g06*gm1*gm4*gm7*s +
CL*g03*g06*g07*gm1*gm4*gm5*s + CL*g04*g05*g06*gm1*gm3*gm7*s + CL*g04*g05*g08*gm1*gm3*gm7*s +
CL*g04*g07*gm1*gm3*gm5*s + CL*g05*g07*gm1*gm3*gm4*s + CL*g01*g03*g06*gm2*gm4*gm5*s +
CL*g01*g02*g03*gm4*gm5*gm8*s + CL*g01*g03*g04*gm2*gm5*gm8*s + CL*g01*g03*g05*gm2*gm4*gm8*s +
CL*g01*g03*g06*gm2*gm4*gm7*s + CL*g01*g03*g08*gm2*gm4*gm5*s + CL*g01*g02*g06*gm3*gm5*gm7*s +
CL*g02*g03*g06*gm1*gm5*gm7*s + CL*g02*g05*g06*gm1*gm3*gm7*s + CL*g02*g06*g07*gm1*gm3*gm5*s +
CL*g01*g02*g03*gm4*gm7*gm8*s + CL*g01*g02*g06*gm4*gm5*gm7*s + CL*g01*g03*g04*gm2*gm7*gm8*s +
```

# MATLAB Solution Continued 4:

```
CL*g01*g03*g07*gm2*gm4*gm8*s + CL*g01*g03*g08*gm2*gm4*gm7*s + CL*g01*g04*g06*gm2*gm5*gm7*s +
CL*g01*g05*g06*gm2*gm4*gm7*s + CL*g01*g06*g07*gm2*gm4*gm5*s + CL*g01*g02*g03*gm5*gm7*gm8*s +
CL*g01*g02*g05*gm3*gm7*gm8*s + CL*g01*g02*g07*gm3*gm5*gm8*s + CL*g01*g02*g08*gm3*gm5*gm7*s +
CL*g01*g03*g06*gm4*gm5*gm7*s + CL*g01*g04*g06*gm3*gm5*gm7*s + CL*g01*g05*g06*gm3*gm4*gm7*s +
CL*g01*g06*g07*gm3*gm4*gm5*s + CL*g02*g03*g05*gm1*gm7*gm8*s + CL*g02*g03*g07*gm1*gm5*gm8*s +
CL*g02*g03*g08*gm1*gm5*gm7*s + CL*g02*g05*g07*gm1*gm3*gm8*s + CL*g02*g05*g08*gm1*gm3*gm7*s +
CL*g02*g07*g08*gm1*gm3*gm5*s + CL*g03*g04*g06*gm1*gm5*gm7*s + CL*g03*g05*g06*gm1*gm4*gm7*s +
CL*g03*g06*g07*gm1*gm4*gm5*s + CL*g04*g05*g06*gm1*gm3*gm7*s + CL*g04*g06*g07*gm1*gm3*gm5*s +
CL*g05*g06*g07*gm1*gm3*gm4*s + CL*g01*g02*g04*gm5*gm7*gm8*s + CL*g01*g02*g05*gm4*gm7*gm8*s +
CL*g01*g02*g07*gm4*gm5*gm8*s + CL*g01*g02*g08*gm4*gm5*gm7*s + CL*g01*g04*g05*gm2*gm7*gm8*s +
CL*g01*g04*g07*gm2*gm5*gm8*s + CL*g01*g05*g07*gm2*gm4*gm8*s + CL*g01*g05*g08*gm2*gm4*gm8*s +
CL*g01*g05*g08*gm2*gm4*gm7*s + CL*g01*g07*g08*gm2*gm4*gm5*s + CL*g02*g03*g06*gm4*gm5*gm7*s +
CL*g02*g04*g06*gm3*gm5*gm7*s + CL*g02*g05*g06*gm3*gm4*gm7*s + CL*g02*g06*g07*gm3*gm4*gm5*s +
CL*g03*g04*g06*gm2*gm5*gm7*s + CL*g03*g05*g06*gm2*gm4*gm7*s + CL*g03*g06*g07*gm2*gm4*gm5*s +
CL*g04*g05*g06*gm2*gm3*gm7*s + CL*g04*g06*g07*gm2*gm3*gm5*s + CL*g05*g06*g07*gm2*gm3*gm4*s +
CL*g01*g03*g04*gm5*gm7*gm8*s + CL*g01*g03*g05*gm4*gm7*gm8*s + CL*g01*g03*g07*gm4*gm5*gm8*s +
CL*g01*g03*g08*gm4*gm5*gm7*s + CL*g01*g04*g05*gm3*gm7*gm8*s + CL*g01*g04*g07*gm3*gm5*gm8*s +
CL*g01*g04*g08*gm3*gm5*gm7*s + CL*g01*g07*g08*gm3*gm4*gm5*s + CL*g03*g04*g05*gm1*gm7*gm8*s +
CL*g03*g04*g05*gm1*gm4*gm8*s + CL*g03*g05*g08*gm1*gm4*gm7*s + CL*g03*g07*g08*gm1*gm3*gm5*gm8*s +
CL*g03*g07*g08*gm1*gm3*gm5*s + CL*g04*g05*g07*gm1*gm3*gm8*s + CL*g04*g05*g08*gm1*gm3*gm7*s +
CL*g04*g07*g08*gm1*gm3*gm4*s + CL*g02*g03*g04*gm5*gm7*gm8*s + CL*g02*g03*g04*gm5*gm7*s +
CL*g02*g03*g05*gm4*gm7*gm8*s + CL*g02*g03*g07*gm4*gm5*gm8*s + CL*g02*g03*g08*gm4*gm5*gm7*s +
CL*g02*g04*g05*gm3*gm7*gm8*s + CL*g02*g04*g07*gm3*gm5*gm8*s + CL*g02*g04*g08*gm3*gm5*gm7*s +
CL*g02*g05*g07*gm3*gm4*gm8*s + CL*g02*g05*g08*gm3*gm4*gm7*s + CL*g02*g07*g08*gm3*gm4*gm5*s +
CL*g03*g04*g05*gm2*gm7*gm8*s + CL*g03*g04*g07*gm2*gm5*gm8*s + CL*g03*g04*g08*gm2*gm5*gm7*s +
CL*g03*g05*g07*gm2*gm4*gm8*s + CL*g03*g05*g08*gm2*gm4*gm7*s + CL*g03*g07*g08*gm2*gm4*gm5*s +
CL*g04*g05*g07*gm2*gm3*gm8*s + CL*g04*g05*g08*gm2*gm3*gm7*s + CL*g04*g07*g08*gm2*gm3*gm5*s +
CL*g05*g07*g08*gm2*gm3*gm4*s + CL*g01*g03*g011*gm4*gm5*gm8*s + CL*g01*g02*g011*gm3*gm5*gm7*s +
CL*g01*g04*g011*gm5*gm7*gm8*s + CL*g01*g05*g011*gm4*gm7*gm8*s + CL*g01*g07*g011*gm4*gm5*gm8*s +
CL*g01*g08*g011*gm4*gm5*gm7*s + CL*g02*g03*g011*gm5*gm7*gm8*s + CL*g02*g05*g011*gm3*gm7*gm8*s +
CL*g02*g07*g011*gm3*gm5*gm8*s + CL*g02*g08*g011*gm3*gm5*gm7*s + CL*g03*g06*g011*gm3*gm4*gm5*s +
CL*g03*g06*g07*gm3*gm4*gm5*s + CL*g03*g07*g011*gm3*gm4*gm5*s + CL*g03*g08*gm3*gm4*gm5*s +
CL*g03*g08*gm3*gm4*gm7*s + CL*g04*g05*g011*gm3*gm7*gm8*s + CL*g04*g05*g011*gm3*gm7*gm8*s +
CL*g04*g07*g011*gm3*gm5*gm8*s + CL*g04*g08*gm3*gm5*gm7*s + CL*g05*g07*g011*gm3*gm4*gm8*s +
CL*g05*g08*gm3*gm4*gm7*s + CL*g07*g08*gm3*gm4*gm5*s + CL*g01*g03*gm2*gm4*gm5*gm8*s +
CL*g01*g03*gm2*gm4*gm7*gm8*s + CL*g01*g06*gm2*gm4*gm5*gm7*s + CL*g01*g02*gm3*gm5*gm7*gm8*s +
CL*g01*g06*gm3*gm4*gm5*gm7*s + CL*g02*g03*gm1*gm5*gm7*gm8*s + CL*g02*g05*gm1*gm3*gm7*gm8*s +
CL*g02*g07*gm1*gm3*gm5*gm8*s + CL*g04*g06*gm1*gm3*gm5*gm7*s + CL*g05*g06*gm1*gm3*gm4*gm5*s +
CL*g04*g06*gm1*gm3*gm5*gm7*s + CL*g05*g06*gm1*gm3*gm4*gm7*s + CL*g06*g07*gm1*gm3*gm4*gm5*s +
CL*g07*g08*gm1*gm3*gm4*gm5*s + CL*g01*g04*gm2*gm5*gm7*gm8*s + CL*g01*g05*gm2*gm4*gm7*gm8*s +
CL*g01*g07*gm2*gm4*gm5*gm8*s + CL*g02*g06*gm3*gm4*gm5*gm7*s + CL*g02*g06*gm3*gm4*gm5*gm7*s +
CL*g02*g06*gm3*gm4*gm5*gm7*s + CL*g01*g03*gm4*gm5*gm7*gm8*s + CL*g01*g04*gm3*gm5*gm7*gm8*s +
CL*g01*g05*gm3*gm4*gm7*gm8*s + CL*g01*g07*gm3*gm4*gm5*gm8*s + CL*g03*g04*gm1*gm5*gm7*gm8*s +
CL*g03*g04*gm1*gm5*gm7*gm8*s + CL*g03*g07*gm1*gm4*gm5*gm8*s + CL*g03*g07*gm1*gm4*gm5*gm8*s +
CL*g03*g08*gm1*gm4*gm5*gm7*s + CL*g04*g05*gm1*gm3*gm7*gm8*s + CL*g04*g05*gm1*gm3*gm7*gm8*s +
CL*g04*g08*gm1*gm3*gm5*gm7*s + CL*g05*g07*gm1*gm3*gm4*gm8*s + CL*g05*g08*gm1*gm3*gm4*gm7*s +
CL*g07*g08*gm1*gm3*gm4*gm5*s + CL*g02*g03*gm4*gm5*gm7*gm8*s + CL*g02*g04*gm3*gm5*gm7*gm8*s +
CL*g02*g05*gm3*gm4*gm7*gm8*s + CL*g02*g08*gm3*gm4*gm5*gm7*s + CL*g03*g07*gm2*gm4*gm5*gm8*s +
CL*g03*g07*gm2*gm4*gm5*gm8*s + CL*g03*g07*gm2*gm4*gm5*gm8*s + CL*g04*g07*gm2*gm3*gm5*gm8*s +
CL*g04*g08*gm2*gm3*gm5*gm7*s + CL*g05*g07*gm2*gm3*gm4*gm8*s + CL*g05*g08*gm2*gm3*gm4*gm7*s +
CL*g07*g08*gm2*gm3*gm4*gm5*s + CL*g01*g011*gm4*gm5*gm8*s + CL*g02*g011*gm3*gm5*gm7*gm8*s +
CL*g04*g011*gm3*gm4*gm5*gm7*s + CL*g07*g011*gm3*gm4*gm5*gm8*s + CL*g08*g011*gm3*gm4*gm5*gm7*s +
CL*g02*gm1*gm3*gm5*gm7*gm8*s + CL*g06*gm1*gm3*gm4*gm5*gm7*s + CL*g01*gm2*gm4*gm5*gm7*gm8*s +
CL*g06*gm2*gm3*gm4*gm5*gm7*s + CL*g01*gm3*gm4*gm5*gm7*gm8*s + CL*g03*gm1*gm4*gm5*gm7*gm8*s +
CL*g04*gm1*gm3*gm5*gm7*gm8*s + CL*g07*gm1*gm3*gm4*gm5*gm8*s + CL*g07*gm1*gm3*gm4*gm5*gm8*s +
CL*g08*gm1*gm3*gm4*gm5*gm7*s + CL*g02*gm3*gm4*gm5*gm7*gm8*s + CL*g03*gm2*gm4*gm5*gm8*s +
CL*g04*gm2*gm3*gm4*gm5*gm7*gm8*s + CL*g05*gm2*gm3*gm4*gm7*gm8*s + CL*g07*gm2*gm3*gm4*gm5*gm8*s +
CL*g011*gm3*gm4*gm5*gm7*gm8*s + CL*gm1*gm3*gm4*gm5*gm7*gm8*s + CL*gm1*gm3*gm4*gm5*gm7*gm8*s +
CL*gm2*gm3*gm4*gm5*gm7*gm8*s)) Vinn + (Vinp*g02*g03*g04*g07*gm1*gm8^2 +
Vinp*g02*g03*g07*gm1*gm8^2 + Vinp*g03*g04*g07*g011*gm1*gm8^2 +
Vinp*g02*g03*g04*gm1*gm7*gm8^2 + Vinp*g02*g03*g07*gm1*gm4*gm8^2 + Vinp*g02*g04*g07*gm1*gm3*gm8^2 +
Vinp*g03*g04*g07*gm1*gm2*gm8^2 + Vinp*g02*g03*g011*gm1*gm7*gm8^2 +
Vinp*g02*g07*g011*gm1*gm3*gm8^2 + Vinp*g03*g04*g011*gm1*gm7*gm8^2 +
Vinp*g03*g07*g011*gm1*gm4*gm8^2 + Vinp*g04*g07*g011*gm1*gm3*gm8^2 +
Vinp*g02*g03*gm1*gm4*gm7*gm8^2 + Vinp*g02*g04*gm1*gm3*gm7*gm8^2 + Vinp*g02*g07*gm1*gm3*gm4*gm8^2
```

# MATLAB Solution Continued 5:

```
+ Vinp*g03*g04*gm1*gm2*gm7*gm8^2 + Vinp*g03*g07*gm1*gm2*gm4*gm8^2 +
Vinp*g04*g07*gm1*gm2*gm3*gm8^2 + Vinp*g02*g011*gm1*gm3*gm7*gm8^2 +
Vinp*g03*g011*gm1*gm4*gm7*gm8^2 + Vinp*g04*g011*gm1*gm3*gm7*gm8^2 +
Vinp*g07*g011*gm1*gm3*gm4*gm8^2 + Vinp*g02*gm1*gm3*gm4*gm7*gm8^2 +
Vinp*g03*gm1*gm2*gm4*gm7*gm8^2 + Vinp*g04*gm1*gm2*gm3*gm7*gm8^2 + Vinp*g07*gm1*gm2*gm3*gm4*gm8^2
+ Vinp*g011*gm1*gm3*gm4*gm7*gm8^2 + Vinp*gm1*gm2*gm3*gm4*gm7*gm8^2 +
Vinp*g02*g03*g04*g05*g06*g07*gm1 + Vinp*g02*g03*g04*g05*g07*g08*gm1 +
Vinp*g02*g03*g04*g05*g06*gm1*gm7 + Vinp*g02*g03*g04*g06*g07*gm1*gm5 +
Vinp*g02*g03*g05*g06*g07*gm1*gm4 + Vinp*g02*g04*g05*g06*g07*gm1*gm3 +
Vinp*g03*g04*g05*g06*g07*gm1*gm2 + Vinp*g02*g03*g04*g05*g06*gm1*gm8 +
Vinp*g02*g03*g04*g05*g07*gm1*gm8 + Vinp*g02*g03*g04*g05*g08*gm1*gm7 +
Vinp*g02*g03*g04*g07*g08*gm1*gm5 + Vinp*g02*g03*g05*g07*g08*gm1*gm4 +
Vinp*g02*g04*g05*g07*g08*gm1*gm3 + Vinp*g03*g04*g05*g07*g08*gm1*gm2 +
Vinp*g02*g03*g04*g06*g07*gm1*gm8 + Vinp*g02*g03*g04*g07*g08*gm1*gm8 +
Vinp*g03*g04*g05*g06*g011*gm1*gm8 + Vinp*g02*g03*g07*g08*g011*gm1*gm8 +
Vinp*g02*g03*g04*g06*g011*gm1*gm8 + Vinp*g02*g03*g07*g08*g011*gm1*gm8 +
Vinp*g03*g04*g06*g07*g011*gm1*gm8 + Vinp*g03*g04*g07*g08*g011*gm1*gm8 +
Vinp*g02*g03*g04*g06*gm1*gm5*gm7 + Vinp*g02*g03*g05*g06*gm1*gm4*gm7 +
Vinp*g02*g03*g06*g07*gm1*gm4*gm5 + Vinp*g02*g04*g05*g06*gm1*gm3*gm7 +
Vinp*g02*g04*g06*g07*gm1*gm3*gm5 + Vinp*g02*g05*g06*g07*gm1*gm3*gm4 +
Vinp*g03*g04*g05*g06*gm1*gm2*gm7 + Vinp*g03*g04*g06*g07*gm1*gm2*gm5 +
Vinp*g03*g05*g06*g07*gm1*gm2*gm4 + Vinp*g04*g05*g06*g07*gm1*gm2*gm3 +
Vinp*g02*g03*g04*g06*gm1*gm5*gm8 + Vinp*g02*g03*g04*g07*gm1*gm5*gm8 +
Vinp*g02*g03*g04*g08*gm1*gm5*gm7 + Vinp*g02*g03*g05*g07*gm1*gm4*gm8 +
Vinp*g02*g03*g05*g08*gm1*gm4*gm5 + Vinp*g02*g03*g07*gm1*gm4*gm8 +
Vinp*g02*g04*g05*g07*gm1*gm3*gm8 + Vinp*g02*g04*g05*g08*gm1*gm3*gm7 +
Vinp*g02*g04*g07*g08*gm1*gm3*gm5 + Vinp*g02*g05*g07*g08*gm1*gm3*gm4 +
Vinp*g03*g04*g05*g06*gm1*gm2*gm8 + Vinp*g03*g04*g05*g08*gm1*gm2*gm7 +
Vinp*g03*g04*g07*g08*gm1*gm2*gm5 + Vinp*g03*g05*g07*g08*gm1*gm2*gm4 +
Vinp*g04*g05*g07*g08*gm1*gm2*gm3 + Vinp*g02*g03*g04*g06*gm1*gm7*gm8 +
Vinp*g02*g03*g06*g07*gm1*gm4*gm8 + Vinp*g02*g04*g06*g07*gm1*gm3*gm8 +
Vinp*g03*g04*g06*g07*gm1*gm2*gm8 + Vinp*g02*g03*g04*g08*gm1*gm7*gm8 +
Vinp*g02*g03*g05*g06*gm1*gm5*gm8 + Vinp*g04*g05*g06*g011*gm1*gm3*gm8 +
Vinp*g03*g05*g06*g011*gm1*gm4*gm8 + Vinp*g04*g05*g06*g011*gm1*gm3*gm8 +
Vinp*g02*g03*g08*g011*gm1*gm7*gm8 + Vinp*g02*g07*g08*g011*gm1*gm3*gm8 +
Vinp*g03*g04*g06*g011*gm1*gm7*gm8 + Vinp*g03*g06*g07*g011*gm1*gm4*gm8 +
Vinp*g04*g06*g07*g011*gm1*gm3*gm8 + Vinp*g03*g04*g08*g011*gm1*gm7*gm8 +
Vinp*g03*g07*g08*g011*gm1*gm7*gm8 + Vinp*g04*g07*g08*g011*gm1*gm7*gm8 +
Vinp*g02*g03*g06*gm1*gm4*gm5*gm8 + Vinp*g02*g04*g06*gm1*gm3*gm5*gm8 +
Vinp*g05*g06*g011*gm1*gm3*gm4*gm8 + Vinp*g02*g08*g011*gm1*gm3*gm7*gm8 +
Vinp*g03*g06*g011*gm1*gm4*gm7*gm8 + Vinp*g04*g06*g011*gm1*gm3*gm7*gm8 +
```





# MATLAB Solution Continued 7:

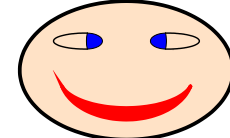
g01\*g03\*g04\*g06\*g08\*gm2\*gm7 + g01\*g03\*g06\*g07\*g08\*gm2\*gm4 + g02\*g03\*g04\*g06\*g07\*gm1\*gm8 +  
g01\*g02\*g03\*g06\*g08\*gm5\*gm7 + g01\*g02\*g05\*g06\*g08\*gm3\*gm7 + g01\*g02\*g06\*g07\*g08\*gm3\*gm5 +  
g02\*g03\*g05\*g06\*g08\*gm1\*gm7 + g02\*g03\*g06\*g07\*g08\*gm1\*gm5 + g02\*g05\*g06\*g07\*g08\*gm1\*gm3 +  
g01\*g02\*g03\*g04\*g08\*gm7\*gm8 + g01\*g02\*g04\*g06\*g08\*gm5\*gm7 + g01\*g02\*g04\*g07\*g08\*gm3\*gm8 +  
g01\*g02\*g05\*g06\*g08\*gm4\*gm7 + g01\*g02\*g06\*g07\*g08\*gm4\*gm5 + g01\*g04\*g05\*g06\*g08\*gm2\*gm7 +  
g01\*g04\*g06\*g07\*g08\*gm2\*gm5 + g01\*g05\*g06\*g07\*g08\*gm2\*gm4 + g02\*g03\*g04\*g07\*g08\*gm1\*gm8 +  
g01\*g02\*g03\*g04\*g011\*gm5\*gm8 + g01\*g03\*g04\*g06\*g08\*gm5\*gm7 + g01\*g03\*g05\*g06\*g08\*gm4\*gm7 +  
g01\*g03\*g06\*g07\*g08\*gm4\*gm5 + g01\*g04\*g05\*g06\*g08\*gm3\*gm7 + g01\*g04\*g06\*g07\*g08\*gm3\*gm5 +  
g01\*g04\*g06\*g07\*g08\*gm1\*gm7 + g03\*g04\*g05\*g06\*g08\*gm1\*gm7 + g03\*g04\*g06\*g07\*g08\*gm1\*gm5 +  
g04\*g05\*g06\*g07\*g08\*gm1\*gm4 + g04\*g05\*g06\*g07\*g08\*gm1\*gm3 + g02\*g03\*g04\*g06\*g08\*gm5\*gm7 +  
g02\*g03\*g05\*g06\*g08\*gm4\*gm7 + g02\*g03\*g06\*g07\*g08\*gm4\*gm5 + g02\*g04\*g05\*g06\*g08\*gm3\*gm7 +  
g02\*g04\*g06\*g07\*g08\*gm3\*gm5 + g02\*g05\*g06\*g07\*g08\*gm3\*gm4 + g03\*g04\*g05\*g06\*g08\*gm2\*gm7 +  
g03\*g04\*g06\*g07\*g08\*gm2\*gm5 + g03\*g05\*g06\*g07\*g08\*gm2\*gm4 + g04\*g05\*g06\*g07\*g08\*gm2\*gm3 +  
g01\*g02\*g03\*g04\*g011\*gm7\*gm8 + g01\*g02\*g04\*g06\*g011\*gm5\*gm7 + g01\*g02\*g04\*g05\*gm1\*gm8 +  
g01\*g02\*g04\*g07\*g011\*gm5\*gm8 + g01\*g02\*g04\*g08\*g011\*gm5\*gm7 + g01\*g03\*g06\*g08\*g011\*gm4\*gm5 +  
g02\*g03\*g04\*g06\*g011\*gm5\*gm7 + g02\*g04\*g05\*g06\*g011\*gm3\*gm7 + g02\*g04\*g06\*g07\*gm1\*gm3\*gm5 +  
g01\*g02\*g06\*g08\*g011\*gm5\*gm7 + g01\*g03\*g06\*g08\*g011\*gm4\*gm7 + g02\*g03\*g04\*g05\*gm1\*gm8 +  
g02\*g04\*g05\*g06\*gm1\*gm3\*gm8 + g02\*g04\*g07\*gm3\*gm5\*gm8 + g01\*g02\*g04\*g07\*gm3\*gm5\*gm8 +  
g01\*g02\*g04\*g07\*gm3\*gm5\*gm8 + g01\*g02\*g04\*g08\*gm3\*gm5\*gm7 + g02\*g03\*g04\*g05\*gm1\*gm7\*gm8 +  
g02\*g03\*g04\*g07\*gm1\*gm5\*gm8 + g02\*g04\*g05\*g08\*gm1\*gm3\*gm7 + g01\*g02\*g04\*g05\*gm1\*gm3\*gm8 +  
g01\*g03\*g06\*g08\*gm2\*gm5\*gm7 + g02\*g03\*g04\*g08\*gm1\*gm5\*gm7 + g02\*g04\*g05\*g08\*gm1\*gm3\*gm7 +  
g01\*g03\*g06\*g08\*gm2\*gm4\*gm7 + g02\*g04\*g07\*g08\*gm1\*gm3\*gm5 + g01\*g02\*g04\*g06\*gm3\*gm7\*gm8 +  
g01\*g02\*g06\*g08\*gm1\*gm3\*gm5 + g02\*g03\*g04\*g06\*gm1\*gm7\*gm8 + g02\*g04\*g06\*g07\*gm1\*gm3\*gm8 +  
g02\*g06\*g07\*g08\*gm1\*gm3\*gm5 + g01\*g02\*g04\*g08\*gm3\*gm7\*gm8 + g01\*g02\*g06\*g08\*gm2\*gm4\*gm5 +  
g01\*g04\*g06\*g08\*gm1\*gm7\*gm8 + g02\*g04\*g07\*g08\*gm1\*gm3\*gm8 + g01\*g03\*g06\*g08\*gm4\*gm5\*gm7 +  
g01\*g04\*g06\*g08\*gm3\*gm5\*gm7 + g01\*g05\*g06\*g08\*gm3\*gm4\*gm7 + g01\*g06\*g07\*g08\*gm3\*gm4\*gm5 +  
g03\*g04\*g06\*g08\*gm1\*gm5\*gm7 + g03\*g05\*g06\*g08\*gm1\*gm4\*gm7 + g03\*g06\*g07\*g08\*gm1\*gm4\*gm5 +  
g04\*g05\*g06\*g08\*gm1\*gm3\*gm7 + g04\*g06\*g07\*g08\*gm1\*gm3\*gm5 + g05\*g06\*g07\*g08\*gm1\*gm3\*gm4 +  
g02\*g03\*g06\*g08\*gm4\*gm5\*gm7 + g02\*g04\*g06\*g08\*gm3\*gm5\*gm7 + g02\*g05\*g06\*g08\*gm3\*gm4\*gm7 +  
g02\*g06\*g07\*g08\*gm3\*gm4\*gm5 + g03\*g04\*g06\*g08\*gm2\*gm5\*gm7 + g03\*g05\*g06\*g08\*gm2\*gm4\*gm7 +  
g03\*g06\*g07\*g08\*gm2\*gm3\*gm7 + g04\*g05\*g06\*g08\*gm2\*gm3\*gm7 + g04\*g06\*g07\*g08\*gm2\*gm3\*gm5 +  
g05\*g06\*g07\*g08\*gm2\*gm3\*gm4 + g01\*g02\*g04\*g011\*gm5\*gm7\*gm8 + g02\*g04\*g05\*gm1\*gm3\*gm5\*gm7 +  
g02\*g04\*g08\*g011\*gm3\*gm5\*gm7 + g01\*g06\*g08\*g011\*gm3\*gm5\*gm7 + g02\*g06\*g08\*g011\*gm3\*gm5\*gm7 +  
g03\*g06\*g08\*g011\*gm4\*gm5\*gm7 + g04\*g06\*g08\*g011\*gm3\*gm5\*gm7 + g05\*g06\*g08\*g011\*gm3\*gm4\*gm7 +  
g06\*g07\*g08\*g011\*gm3\*gm5\*gm7 + g02\*g04\*g06\*gm1\*gm3\*gm5\*gm7 + g02\*g04\*g06\*gm1\*gm3\*gm5\*gm7 +  
g01\*g02\*g04\*gm3\*gm5\*gm7\*gm8 + g02\*g03\*g04\*gm1\*gm5\*gm7\*gm8 + g02\*g04\*g05\*gm1\*gm3\*gm7\*gm8 +  
g02\*g04\*g07\*gm1\*gm3\*gm5\*gm8 + g02\*g04\*g08\*gm1\*gm3\*gm5\*gm7 + g02\*g04\*g06\*gm1\*gm3\*gm7\*gm8 +  
g02\*g06\*g08\*gm1\*gm3\*gm5\*gm7 + g01\*g06\*g08\*gm2\*gm4\*gm5\*gm7 + g02\*g04\*g08\*gm1\*gm3\*gm7\*gm8 +  
g01\*g06\*g08\*gm2\*gm4\*gm5\*gm7 + g03\*g06\*g08\*gm1\*gm4\*gm5\*gm7 + g04\*g06\*g08\*gm1\*gm3\*gm5\*gm7 +  
g05\*g06\*g08\*gm1\*gm3\*gm4\*gm7 + g03\*g06\*g08\*gm2\*gm3\*gm4\*gm5 + g04\*g06\*g08\*gm2\*gm3\*gm4\*gm7 +  
g06\*g07\*g08\*gm2\*gm3\*gm4\*gm5 + g02\*g04\*g011\*gm3\*gm5\*gm7\*gm8 + g06\*g08\*g011\*gm3\*gm4\*gm5\*gm7 +  
g02\*g04\*gm1\*gm3\*gm5\*gm7\*gm8 + g06\*g08\*gm2\*gm3\*gm4\*gm5\*gm7 + g06\*g08\*gm2\*gm3\*gm4\*gm5\*gm7 +  
CL\*g01\*g02\*g03\*g04\*g05\*g06\*s + CL\*g01\*g02\*g03\*g04\*g05\*g08\*s + CL\*g01\*g02\*g03\*g04\*g05\*g06\*g07\*s +  
CL\*g01\*g02\*g03\*g05\*g06\*g07\*s + CL\*g01\*g02\*g03\*g04\*g07\*g08\*s + CL\*g01\*g02\*g04\*g05\*g06\*g07\*s +  
CL\*g01\*g02\*g03\*g05\*g06\*g07\*s + CL\*g01\*g02\*g03\*g05\*g06\*g011\*s + CL\*g01\*g03\*g04\*g05\*g07\*g08\*s +  
CL\*g02\*g03\*g04\*g05\*g07\*g08\*s + CL\*g01\*g02\*g03\*g05\*g08\*g011\*s + CL\*g01\*g02\*g03\*g06\*g07\*g011\*s +  
CL\*g01\*g03\*g04\*g05\*g06\*g011\*s + CL\*g01\*g02\*g03\*g07\*g08\*g011\*s + CL\*g01\*g02\*g05\*g06\*g07\*g011\*s +  
CL\*g01\*g03\*g04\*g05\*g08\*g011\*s + CL\*g01\*g03\*g04\*g06\*g07\*g011\*s + CL\*g01\*g02\*g05\*g07\*g08\*g011\*s +  
CL\*g01\*g03\*g04\*g07\*g08\*g011\*s + CL\*g01\*g04\*g05\*g06\*g07\*g011\*s + CL\*g02\*g03\*g05\*g06\*g07\*g011\*s +  
CL\*g01\*g04\*g05\*g07\*g08\*g011\*s + CL\*g02\*g03\*g05\*g07\*g08\*g011\*s + CL\*g03\*g04\*g05\*g06\*g07\*g011\*s +  
CL\*g03\*g04\*g05\*g07\*g08\*g011\*s + CL\*g01\*g02\*g03\*g04\*g06\*gm5\*s + CL\*g01\*g02\*g03\*g05\*g06\*gm4\*s +  
CL\*g01\*g03\*g04\*g05\*g06\*gm2\*s + CL\*g01\*g02\*g03\*g04\*g05\*gm8\*s + CL\*g01\*g02\*g03\*g04\*g06\*gm7\*s +  
CL\*g01\*g02\*g03\*g04\*g08\*gm2\*s + CL\*g01\*g03\*g04\*g06\*g07\*gm2\*s + CL\*g01\*g03\*g04\*g06\*g07\*gm2\*s +  
CL\*g01\*g02\*g03\*g06\*g07\*gm5\*s + CL\*g01\*g02\*g05\*g06\*g07\*gm3\*s + CL\*g02\*g03\*g05\*g06\*g07\*gm1\*s +  
CL\*g01\*g02\*g03\*g04\*g07\*gm8\*s + CL\*g01\*g02\*g03\*g04\*g08\*gm7\*s + CL\*g01\*g02\*g03\*g05\*g06\*g07\*gm4\*s +  
CL\*g01\*g02\*g04\*g05\*g06\*gm7\*s + CL\*g01\*g02\*g04\*g06\*g07\*gm5\*s + CL\*g01\*g02\*g05\*g06\*g07\*gm4\*s +  
CL\*g01\*g03\*g04\*g07\*g08\*gm2\*s + CL\*g01\*g04\*g05\*g06\*g07\*gm2\*s + CL\*g01\*g02\*g03\*g05\*g07\*gm8\*s +  
CL\*g01\*g02\*g03\*g05\*g08\*gm7\*s + CL\*g01\*g02\*g03\*g07\*g08\*gm5\*s + CL\*g01\*g02\*g03\*g07\*g08\*gm3\*s +



# MATLAB Solution Continued 9:

```
CL*g03*g05*g07*g011*gm4*gm8*s + CL*g03*g05*g08*g011*gm4*gm7*s + CL*g03*g07*g08*g011*gm4*gm5*s +
CL*g04*g05*g07*g011*gm3*gm8*s + CL*g04*g05*g08*g011*gm3*gm7*s + CL*g04*g07*g08*g011*gm3*gm5*s +
CL*g05*g07*g08*g011*gm3*gm4*s + CL*g01*g03*g06*gm2*gm4*gm5*s + CL*g01*g02*g03*gm4*gm5*gm8*s +
CL*g01*g03*g04*gm2*gm5*gm8*s + CL*g01*g03*g05*gm2*gm4*gm8*s + CL*g01*g03*g06*gm2*gm4*gm7*s +
CL*g01*g03*g08*gm2*gm4*gm5*s + CL*g01*g02*g06*gm3*gm5*gm7*s + CL*g02*g03*g06*gm1*gm5*gm7*s +
CL*g02*g05*g06*gm1*gm3*gm7*s + CL*g02*g06*g07*gm1*gm3*gm5*s + CL*g01*g02*g03*gm4*gm7*gm8*s +
CL*g01*g02*g06*gm4*gm5*gm7*s + CL*g01*g03*g04*gm2*gm7*gm8*s + CL*g01*g03*g07*gm2*gm4*gm8*s +
CL*g01*g03*g08*gm2*gm4*gm7*s + CL*g01*g04*g06*gm2*gm5*gm7*s + CL*g01*g05*g06*gm2*gm4*gm7*s +
CL*g01*g06*g07*gm2*gm4*gm5*s + CL*g01*g02*g03*gm5*gm7*gm8*s + CL*g01*g02*g05*gm3*gm7*gm8*s +
CL*g01*g02*g07*gm3*gm5*gm8*s + CL*g01*g02*g08*gm3*gm5*gm7*s + CL*g01*g03*g06*gm4*gm5*gm7*s +
CL*g01*g04*g06*gm3*gm5*gm7*s + CL*g01*g05*g06*gm3*gm4*gm7*s + CL*g01*g06*g07*gm3*gm4*gm5*s +
CL*g02*g03*g05*gm1*gm7*gm8*s + CL*g02*g03*g07*gm1*gm5*gm8*s + CL*g02*g03*g08*gm1*gm5*gm7*s +
CL*g02*g05*g07*gm1*gm3*gm8*s + CL*g02*g05*g08*gm1*gm3*gm7*s + CL*g02*g07*g08*gm1*gm3*gm5*s +
CL*g03*g04*g06*gm1*gm5*gm7*s + CL*g03*g05*g06*gm1*gm4*gm7*s + CL*g03*g06*g07*gm1*gm4*gm5*s +
CL*g04*g05*g06*gm1*gm3*gm7*s + CL*g04*g06*g07*gm1*gm3*gm5*s + CL*g05*g06*g07*gm1*gm3*gm4*s +
CL*g01*g02*g04*gm5*gm7*gm8*s + CL*g01*g02*g05*gm4*gm7*gm8*s + CL*g01*g02*g07*gm4*gm5*gm8*s +
CL*g01*g04*g08*gm2*gm5*gm7*s + CL*g01*g05*g07*gm2*gm4*gm8*s + CL*g01*g05*g08*gm2*gm4*gm7*s +
CL*g01*g07*g08*gm2*gm4*gm5*s + CL*g02*g03*g06*gm4*gm5*gm7*s + CL*g02*g04*g06*gm3*gm5*gm7*s +
CL*g02*g05*g06*gm3*gm4*gm7*s + CL*g02*g06*g07*gm3*gm4*gm5*s + CL*g03*g04*g06*gm2*gm5*gm7*s +
CL*g03*g05*g06*gm2*gm4*gm7*s + CL*g03*g06*g07*gm2*gm3*gm8*s + CL*g04*g05*g06*gm1*gm3*gm5*s +
CL*g04*g06*g07*gm1*gm3*gm8*s + CL*g05*g06*g07*gm1*gm3*gm5*s + CL*g01*g03*g04*gm5*gm7*gm8*s +
CL*g01*g03*g05*gm4*gm7*gm8*s + CL*g01*g03*g08*gm4*gm5*gm7*s + CL*g01*g04*g08*gm3*gm5*gm7*s +
CL*g01*g05*g07*gm3*gm4*gm8*s + CL*g01*g05*g08*gm3*gm4*gm5*s + CL*g01*g07*g08*gm3*gm4*gm5*s +
CL*g03*g04*g05*gm1*gm7*gm8*s + CL*g03*g04*g07*gm1*gm5*gm8*s + CL*g03*g04*g08*gm1*gm5*gm7*s +
CL*g03*g05*g07*gm1*gm4*gm8*s + CL*g03*g05*g08*gm1*gm4*gm5*s + CL*g03*g07*g08*gm1*gm4*gm5*s +
CL*g04*g05*g08*gm2*gm3*gm7*s + CL*g04*g07*g08*gm2*gm3*gm5*s + CL*g05*g07*g08*gm2*gm3*gm4*s +
CL*g01*g03*g011*gm4*gm5*gm8*s + CL*g01*g02*g011*gm5*gm7*gm8*s + CL*g01*g03*g011*gm4*gm7*gm8*s +
CL*g01*g06*g011*gm4*gm5*gm7*s + CL*g02*g06*g011*gm3*gm5*gm7*s + CL*g01*g04*g011*gm5*gm7*gm8*s +
CL*g01*g05*g011*gm4*gm7*gm8*s + CL*g01*g07*g011*gm4*gm5*gm8*s + CL*g01*g08*g011*gm4*gm5*gm7*s +
CL*g02*g03*g011*gm5*gm7*gm8*s + CL*g02*g05*g011*gm3*gm7*gm8*s + CL*g02*g07*g011*gm3*gm5*gm8*s +
CL*g02*g08*g011*gm3*gm5*gm7*s + CL*g03*g06*g011*gm4*gm5*gm7*s + CL*g04*g06*g011*gm4*gm5*gm7*s +
CL*g05*g06*g011*gm3*gm4*gm7*s + CL*g06*g07*g011*gm3*gm4*gm5*s + CL*g03*g04*g011*gm5*gm7*gm8*s +
CL*g03*g05*g011*gm4*gm7*gm8*s + CL*g03*g07*g011*gm4*gm5*gm8*s + CL*g03*g08*g011*gm4*gm5*gm7*s +
CL*g04*g05*g011*gm3*gm7*gm8*s + CL*g04*g07*g011*gm3*gm5*gm8*s + CL*g04*g08*g011*gm3*gm5*gm7*s +
CL*g05*g07*g011*gm3*gm4*gm8*s + CL*g05*g08*g011*gm3*gm4*gm7*s + CL*g07*g08*g011*gm3*gm4*gm5*s +
CL*g01*g03*gm2*gm4*gm5*gm8*s + CL*g02*g06*gm1*gm3*gm5*gm7*s + CL*g01*g03*gm2*gm4*gm7*gm8*s +
CL*g01*g06*gm3*gm4*gm5*gm7*s + CL*g01*g07*gm3*gm4*gm5*gm8*s + CL*g01*g08*gm3*gm4*gm5*gm7*s +
CL*g02*g03*gm1*gm5*gm7*gm8*s + CL*g02*g05*gm1*gm3*gm7*gm8*s + CL*g02*g07*gm1*gm3*gm5*gm8*s +
CL*g02*g08*gm1*gm3*gm5*gm7*s + CL*g03*g06*gm1*gm4*gm5*gm7*s + CL*g04*g06*gm1*gm3*gm5*gm7*s +
CL*g05*g06*gm1*gm3*gm4*gm7*s + CL*g06*g07*gm1*gm3*gm4*gm5*s + CL*g01*g02*gm2*gm4*gm5*gm7*gm8*s +
CL*g01*g04*gm2*gm3*gm4*gm5*gm8*s + CL*g01*g05*gm2*gm3*gm4*gm7*gm8*s + CL*g01*g07*gm2*gm3*gm4*gm5*gm8*s +
CL*g01*g08*gm2*gm3*gm4*gm5*gm7*gm8*s + CL*g02*gm1*gm3*gm4*gm5*gm7*gm8*s + CL*gm2*gm3*gm4*gm5*gm7*gm8*s)
```

We now (finally) have a complete analytic expression !





# Individual product terms have product of 7 small-signal parameters

- Approximately 2000 factors in output characteristics
- Approximately 14,000 small-signal parameter appearances
- GB expression much longer

$$\frac{V_{in}p \cdot g_{08} \cdot g_{m1} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} + V_{in}p \cdot g_{06} \cdot g_{m1} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m7} \cdot g_{m8} + V_{in}p \cdot g_{08} \cdot g_{m1} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m7} \cdot g_{m8} + V_{in}p \cdot g_{m1} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8}}{(g_{01} \cdot g_{02} \cdot g_{03} \cdot g_{04} \cdot g_{07} \cdot g_{m8}^2 + g_{01} \cdot g_{02} \cdot g_{03} \cdot g_{04} \cdot g_{m7} \cdot g_{m8}^2 + g_{01} \cdot g_{02} \cdot g_{04} \cdot g_{07} \cdot g_{m3} \cdot g_{m8}^2 + g_{02} \cdot g_{03} \cdot g_{04} \cdot g_{07} \cdot g_{m1} \cdot g_{m8}^2 + g_{01} \cdot g_{02} \cdot g_{04} \cdot g_{m3} \cdot g_{m7} \cdot g_{m8}^2 + g_{02} \cdot g_{03} \cdot g_{04} \cdot g_{m1} \cdot g_{m7} \cdot g_{m8}^2 +$$

$$CL \cdot g_{02} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{03} \cdot g_{m2} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{04} \cdot g_{m2} \cdot g_{m3} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{05} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{07} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m8} \cdot s + CL \cdot g_{08} \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot s + CL \cdot g_{011} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{m1} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s + CL \cdot g_{m2} \cdot g_{m3} \cdot g_{m4} \cdot g_{m5} \cdot g_{m7} \cdot g_{m8} \cdot s)$$

# MATLAB simulation results using Symbolic Math Toolbox

## Differential Analysis with $V_d = V_{in}^+ - V_{in}^-$

### MATLAB Solution:

```
GainAv =
(g01*g03*g04*g07*gm2*gm8^2 + g02*g03*g04*g07*gm1*gm8^2 + g02*g03*g07*g011*gm1*gm8^2 +
g03*g04*g07*g011*gm1*gm8^2 + g01*g03*g04*gm2*gm7*gm8^2 + g01*g03*g07*gm2*gm4*gm8^2 +
g01*g04*g07*gm2*gm3*gm8^2 + g02*g03*g04*gm1*gm7*gm8^2 + g02*g03*g07*gm1*gm4*gm8^2 +
g02*g04*g07*gm1*gm3*gm8^2 + 2*g03*g04*g07*gm1*gm2*gm8^2 + g02*g03*g011*gm1*gm7*gm8^2 +
g02*g07*g011*gm1*gm3*gm8^2 + g03*g04*g011*gm1*gm7*gm8^2 + g03*g07*g011*gm1*gm4*gm8^2 +
g04*g07*g011*gm1*gm3*gm8^2 + g01*g03*gm2*gm4*gm7*gm8^2 + g01*g04*gm2*gm3*gm7*gm8^2 +
g01*g07*gm2*gm3*gm4*gm8^2 + g02*g03*gm1*gm4*gm7*gm8^2 + g02*g04*gm1*gm3*gm7*gm8^2 +
g02*g07*gm1*gm3*gm4*gm8^2 + 2*g03*g04*gm1*gm2*gm7*gm8^2 + 2*g03*g07*gm1*gm2*gm4*gm8^2 +
2*g04*g07*gm1*gm2*gm3*gm8^2 + g02*g011*gm1*gm3*gm7*gm8^2 + g03*g011*gm1*gm4*gm7*gm8^2 +
g04*g011*gm1*gm3*gm7*gm8^2 + g07*g011*gm1*gm3*gm4*gm8^2 + g01*gm2*gm3*gm4*gm7*gm8^2 +
g02*gm1*gm3*gm4*gm7*gm8^2 + 2*g03*gm1*gm2*gm4*gm7*gm8^2 + 2*g04*gm1*gm2*gm3*gm7*gm8^2 +
2*g07*gm1*gm2*gm3*gm4*gm8^2 + g011*gm1*gm3*gm4*gm7*gm8^2 + 2*gm1*gm2*gm3*gm4*gm7*gm8^2 +
g01*g03*g04*g05*g06*g07*gm2 + g02*g03*g04*g05*g06*g07*gm1 + g01*g03*g04*g05*g07*g08*gm2 +
g02*g03*g04*g05*g07*g08*gm1 + g01*g03*g04*g05*g06*g011*gm2 + g01*g03*g04*g05*g08*g011*gm2 +
g01*g03*g04*g06*g07*g011*gm2 + g01*g03*g04*g07*g08*g011*gm2 + g01*g04*g05*g06*g07*g011*gm2 +
g01*g04*g05*g07*g08*g011*gm2 + g03*g04*g05*g06*g07*g011*gm2 + g03*g04*g05*g07*g08*g011*gm2 +
g01*g03*g04*g05*g06*gm2*gm7 + g01*g03*g04*g06*g07*gm2*gm5 + g01*g03*g05*g06*g07*gm2*gm4 +
g01*g04*g05*g06*g07*gm2*gm3 + g02*g03*g04*g05*g06*gm1*gm7 + g02*g03*g04*g06*g07*gm1*gm5 +
g02*g03*g05*g06*g07*gm1*gm4 + g02*g04*g05*g06*g07*gm1*gm3 + 2*g03*g04*g05*g06*g07*gm1*gm2 +
g01*g03*g04*g05*g08*gm2*gm7 + g02*g03*g04*g05*g06*gm1*gm8 + g01*g03*g04*g05*g07*gm2*gm8 +
g01*g03*g04*g05*g08*gm2*gm5 + g01*g03*g05*g07*g08*gm2*gm4 + g01*g04*g05*g07*g08*gm2*gm3 +
g02*g03*g04*g05*g07*gm1*gm8 + g02*g03*g04*g05*g08*gm1*gm7 + g02*g03*g04*g05*g08*gm1*gm4 +
2*g03*g04*g05*g07*g08*gm1*gm2 + g01*g03*g04*g06*g07*gm2*gm8 + g02*g03*g04*g06*g07*gm1*gm8 +
g01*g03*g04*g06*g07*g08*gm2*gm5 + g01*g03*g04*g07*g08*gm2*gm4 + g02*g03*g04*g06*g07*gm1*gm8 +
g02*g03*g04*g07*g08*gm1*gm2 + g01*g03*g04*g05*g011*gm2*gm8 + g01*g03*g04*g06*g011*gm2*gm7 +
g01*g03*g04*g08*g011*gm2*gm5 + g01*g03*g05*g08*g011*gm2*gm4 + g01*g03*g06*g07*g011*gm2*gm4 +
g01*g03*g04*g07*g011*gm2*gm8 + g01*g03*g04*g08*g011*gm2*gm7 + g01*g03*g07*g08*g011*gm2*gm4 +
g01*g04*g05*g06*g011*gm2*gm7 + g01*g04*g06*g07*g011*gm2*gm5 + g01*g05*g06*g07*g011*gm2*gm4 +
g02*g03*g05*g06*g011*gm1*gm8 + g01*g04*g05*g07*g011*gm2*gm8 + g01*g04*g05*g08*g011*gm2*gm7 +
g01*g04*g07*g08*g011*gm2*gm5 + g01*g05*g07*g08*g011*gm2*gm4 + g02*g03*g06*g07*g011*gm1*gm8 +
g03*g04*g05*g06*g011*gm1*gm8 + g03*g04*g05*g06*g011*gm2*gm7 + g03*g04*g06*g07*g011*gm2*gm5 +
g03*g04*g05*g07*g011*gm2*gm8 + g03*g04*g05*g08*g011*gm2*gm7 + g03*g04*g06*g07*g011*gm1*gm8 +
g03*g04*g07*g08*g011*gm2*gm5 + g03*g05*g07*g08*g011*gm2*gm4 + g04*g05*g07*g08*g011*gm2*gm3 +
g03*g04*g07*g08*g011*gm1*gm8 + g01*g03*g04*g06*gm2*gm5*gm7 + g01*g03*g05*g06*gm2*gm4*gm7 +
g01*g03*g06*g07*gm2*gm3*gm5 + g01*g04*g05*g06*gm2*gm3*gm7 + g01*g04*g06*g07*gm2*gm3*gm5 +
g01*g05*g06*g07*gm2*gm3*gm4 + g02*g03*g04*g06*gm1*gm5*gm7 + g02*g03*g05*g06*gm1*gm4*gm7 +
g02*g03*g06*g07*gm1*gm4*gm5 + g02*g04*g05*g06*gm1*gm3*gm7 + g02*g04*g06*g07*gm1*gm3*gm5 +
g02*g05*g06*g07*gm1*gm3*gm4 + 2*g03*g04*g05*g06*gm1*gm2*gm7 + 2*g03*g04*g06*g07*gm1*gm2*gm5 +
2*g03*g05*g06*g07*gm1*gm2*gm4 + 2*g04*g05*g06*g07*gm1*gm2*gm3 + g01*g03*g04*g06*gm2*gm5*gm8 +
g01*g03*g05*g06*gm2*gm4*gm8 + g01*g04*g05*g06*gm2*gm3*gm8 + g02*g03*g04*g06*gm1*gm5*gm8 +
g02*g03*g05*g06*gm1*gm4*gm8 + g02*g04*g05*g06*gm1*gm3*gm8 + 2*g03*g04*g05*g06*gm1*gm2*gm8 +
g01*g03*g04*g05*gm2*gm7*gm8 + g01*g03*g04*g08*gm2*gm5*gm7 + g01*g03*g04*g08*gm2*gm4*gm5 +
g01*g03*g05*g07*gm2*gm4*gm8 + g01*g03*g05*g08*gm2*gm4*gm7 + g01*g03*g07*g08*gm2*gm4*gm5 +
g01*g04*g05*g07*gm2*gm3*gm8 + g01*g04*g05*g08*gm2*gm3*gm7 + g01*g04*g07*g08*gm2*gm3*gm5 +
g01*g05*g07*g08*gm2*gm3*gm4 + g02*g03*g04*g05*gm1*gm7*gm8 + g02*g03*g04*g07*gm1*gm5*gm8 +
g02*g03*g04*g08*gm1*gm5*gm7 + g02*g03*g05*g07*gm1*gm4*gm8 + g02*g03*g05*g08*gm1*gm4*gm7 +
g02*g03*g07*g08*gm1*gm4*gm5 + g02*g04*g05*g07*gm1*gm3*gm8 + g02*g04*g05*g08*gm1*gm3*gm7 +
g02*g04*g07*g08*gm1*gm3*gm5 + g02*g05*g07*g08*gm1*gm3*gm4 + 2*g03*g04*g05*g07*gm1*gm2*gm8 +
2*g03*g04*g05*g08*gm1*gm2*gm5 + 2*g03*g04*g07*g08*gm1*gm2*gm5 + 2*g03*g05*g07*g08*gm1*gm2*gm4 +
2*g04*g05*g07*g08*gm1*gm2*gm3 + g01*g03*g04*g06*gm2*gm7*gm8 + g01*g03*g06*g07*gm2*gm4*gm8 +
g01*g04*g05*g06*g07*gm2*gm3*gm8 + g02*g03*g04*g06*gm1*gm7*gm8 + g02*g03*g06*g07*gm1*gm4*gm8 +
g02*g04*g06*g07*gm1*gm3*gm8 + 2*g03*g04*g06*g07*gm1*gm2*gm8 + g01*g03*g06*g011*gm2*gm4*gm5 +
g01*g03*g04*g08*gm2*gm7*gm8 + g01*g03*g07*g08*gm2*gm4*gm8 + g01*g04*g07*g08*gm2*gm3*gm8 +
g02*g03*g04*g08*gm1*gm7*gm8 + g02*g03*g07*g08*gm1*gm4*gm8 + g02*g04*g07*g08*gm1*gm3*gm8 +
2*g03*g04*g07*g08*gm1*gm2*gm8 + g01*g03*g04*g011*gm2*gm5*gm8 + g01*g03*g05*g011*gm2*gm4*gm8 +
g01*g03*g06*g011*gm2*gm4*gm7 + g01*g03*g08*g011*gm2*gm4*gm5 + g01*g03*g04*g011*gm2*gm7*gm8 +
g01*g03*g07*g011*gm2*gm4*gm8 + g01*g03*g08*g011*gm2*gm4*gm7 + g01*g04*g06*g011*gm2*gm5*gm7 +
g01*g05*g06*g011*gm2*gm4*gm7 + g01*g06*g07*g011*gm2*gm4*gm5 + g02*g03*g06*g011*gm1*gm5*gm8 +
g02*g05*g06*g011*gm1*gm3*gm8 + g01*g04*g05*g011*gm2*gm7*gm8 + g01*g04*g07*g011*gm2*gm5*gm8 +
g01*g04*g08*g011*gm2*gm5*gm7 + g01*g05*g07*g011*gm2*gm4*gm8 + g01*g05*g08*g011*gm2*gm4*gm7 +
g01*g07*g08*g011*gm2*gm4*gm5 + g02*g03*g06*g011*gm1*gm7*gm8 + g02*g06*g07*g011*gm1*gm3*gm8 +
g03*g04*g06*g011*gm1*gm5*gm8 + g03*g04*g06*g011*gm2*gm5*gm7 + g03*g05*g06*g011*gm1*gm4*gm8 +
g03*g05*g06*g011*gm2*gm4*gm7 + g03*g06*g07*g011*gm2*gm4*gm5 + g04*g05*g06*g011*gm1*gm3*gm8 +
g04*g05*g06*g011*gm2*gm3*gm7 + g04*g06*g07*g011*gm2*gm3*gm5 + g05*g06*g07*g011*gm2*gm3*gm4 +
g02*g03*g08*g011*gm1*gm7*gm8 + g02*g07*g08*g011*gm1*gm3*gm8 + g03*g04*g05*g011*gm2*gm7*gm8 +
```

# MATLAB Solution Continued 1:

```
g03*g04*g06*g011*gm1*gm7*gm8 + g03*g04*g07*g011*gm2*gm5*gm8 + g03*g04*g08*g011*gm2*gm5*gm7 +
g03*g05*g07*g011*gm2*gm4*gm8 + g03*g05*g08*g011*gm2*gm4*gm7 + g03*g06*g07*g011*gm1*gm4*gm8 +
g03*g07*g08*g011*gm2*gm4*gm5 + g04*g05*g07*g011*gm2*gm3*gm8 + g04*g05*g08*g011*gm2*gm3*gm7 +
g04*g06*g07*g011*gm1*gm3*gm8 + g04*g07*g08*g011*gm2*gm3*gm5 + g05*g07*g08*g011*gm2*gm3*gm4 +
g03*g04*g08*g011*gm1*gm7*gm8 + g03*g07*g08*g011*gm1*gm4*gm8 + g04*g07*g08*g011*gm1*gm3*gm8 +
g01*g03*g06*gm2*gm4*gm5*gm7 + g01*g04*g06*gm2*gm3*gm5*gm7 + g01*g05*g06*gm2*gm3*gm4*gm7 +
g01*g06*g07*gm2*gm3*gm4*gm5 + g02*g03*g06*gm1*gm4*gm5*gm7 + g02*g04*g06*gm1*gm3*gm5*gm7 +
g02*g05*g06*gm1*gm3*gm4*gm7 + g02*g06*g07*gm1*gm3*gm4*gm5 + 2*g03*g04*g06*gm1*gm2*gm5*gm7 +
2*g03*g05*g06*gm1*gm2*gm4*gm7 + 2*g03*g06*g07*gm1*gm2*gm4*gm5 + 2*g04*g05*g06*gm1*gm2*gm3*gm7 +
2*g04*g06*g07*gm1*gm2*gm3*gm5 + 2*g05*g06*g07*gm1*gm2*gm3*gm4 + g01*g03*g06*gm2*gm4*gm5*gm8 +
g01*g04*g06*gm2*gm3*gm5*gm8 + g01*g05*g06*gm2*gm3*gm4*gm8 + g02*g03*g06*gm1*gm4*gm5*gm8 +
g02*g04*g06*gm1*gm3*gm5*gm8 + g02*g05*g06*gm1*gm3*gm4*gm8 + 2*g03*g04*g06*gm1*gm2*gm5*gm8 +
2*g03*g05*g06*gm1*gm2*gm4*gm8 + 2*g04*g05*g06*gm1*gm2*gm3*gm8 + g01*g03*g04*gm2*gm5*gm7*gm8 +
g01*g03*g05*gm2*gm4*gm7*gm8 + g01*g03*g07*gm2*gm4*gm5*gm8 + g01*g03*g08*gm2*gm4*gm5*gm7 +
g01*g04*g05*gm2*gm3*gm7*gm8 + g01*g04*g07*gm2*gm3*gm5*gm8 + g01*g04*g08*gm2*gm3*gm5*gm7 +
g01*g05*g07*gm2*gm3*gm4*gm8 + g01*g05*g08*gm2*gm3*gm4*gm7 + g01*g07*g08*gm2*gm3*gm4*gm5 +
g02*g03*g04*gm1*gm5*gm7*gm8 + g02*g03*g05*gm1*gm4*gm7*gm8 + g02*g03*g07*gm1*gm4*gm5*gm8 +
g02*g03*g08*gm1*gm4*gm5*gm7 + g02*g04*g05*gm1*gm3*gm7*gm8 + g02*g04*g07*gm1*gm3*gm5*gm8 +
g02*g04*g08*gm1*gm3*gm5*gm7 + g02*g05*g07*gm1*gm3*gm4*gm8 + g02*g05*g08*gm1*gm3*gm4*gm7 +
g02*g07*g08*gm1*gm3*gm4*gm5 + 2*g03*g04*g05*gm1*gm2*gm7*gm8 + 2*g03*g04*g07*gm1*gm2*gm5*gm8 +
2*g03*g04*g08*gm1*gm2*gm5*gm7 + 2*g03*g05*g07*gm1*gm2*gm4*gm8 + 2*g03*g05*g08*gm1*gm2*gm3*gm7 +
2*g04*g07*g08*gm1*gm2*gm3*gm5 + 2*g05*g07*g08*gm1*gm2*gm3*gm4 + g01*g03*g06*gm2*gm4*gm7*gm8 +
g01*g04*g06*gm2*gm3*gm7*gm8 + g02*g04*g06*gm1*gm3*gm4*gm8 + g02*g03*g06*gm1*gm2*gm7*gm8 +
g02*g04*g06*gm1*gm3*gm7*gm8 + g02*g06*g07*gm1*gm3*gm4*gm8 + 2*g03*g04*g06*gm1*gm2*gm7*gm8 +
2*g03*g06*g07*gm1*gm3*gm4*gm8 + g01*g07*g08*gm2*gm3*gm4*gm8 + g02*g03*g08*gm1*gm4*gm7*gm8 +
g02*g04*g08*gm1*gm3*gm7*gm8 + g02*g07*g08*gm1*gm3*gm4*gm8 + 2*g03*g04*g08*gm1*gm2*gm7*gm8 +
2*g03*g07*g08*gm1*gm2*gm4*gm8 + g01*g03*g07*gm1*gm2*gm5*gm8 + g01*g03*g08*gm1*gm2*gm4*gm5*gm8 +
g01*g04*g011*gm2*gm4*gm7*gm8 + g01*g05*g011*gm2*gm4*gm7*gm8 + g01*g07*g011*gm2*gm4*gm5*gm8 +
g01*g08*g011*gm2*gm4*gm5*gm7 + g02*g06*g011*gm1*gm3*gm5*gm8 + g03*g06*g011*gm1*gm4*gm5*gm8 +
g03*g06*g011*gm2*gm3*gm5*gm8 + g04*g06*g011*gm2*gm3*gm5*gm7 + g05*g06*g011*gm1*gm3*gm4*gm8 +
g02*g08*g011*gm1*gm3*gm7*gm8 + g03*g04*g011*gm2*gm5*gm7*gm8 + g03*g04*g011*gm2*gm5*gm7*gm8 +
g03*g06*g011*gm1*gm4*gm7*gm8 + g03*g07*g011*gm2*gm4*gm5*gm8 + g03*g08*g011*gm2*gm4*gm5*gm7 +
g04*g05*g011*gm2*gm3*gm7*gm8 + g04*g06*g011*gm1*gm3*gm7*gm8 + g04*g07*g011*gm2*gm3*gm5*gm8 +
g04*g08*g011*gm2*gm3*gm5*gm7 + g05*g07*g011*gm2*gm3*gm4*gm8 + g05*g08*g011*gm2*gm3*gm4*gm7 +
g06*g07*g011*gm1*gm3*gm4*gm8 + g07*g08*g011*gm2*gm3*gm4*gm5 + g03*g08*g011*gm1*gm4*gm7*gm8 +
g04*g08*g011*gm1*gm3*gm7*gm8 + g07*g08*g011*gm1*gm3*gm4*gm8 + g01*g06*gm2*gm3*gm4*gm5*gm7 +
g02*g06*gm1*gm3*gm4*gm5*gm7 + 2*g03*g06*gm1*gm2*gm4*gm5*gm7 + 2*g04*g06*gm1*gm2*gm3*gm5*gm7 +
2*g05*g06*gm1*gm2*gm3*gm4*gm5 + 2*g06*g07*gm1*gm2*gm3*gm4*gm5 + g01*g06*gm2*gm3*gm4*gm5*gm8 +
g02*g06*gm1*gm3*gm4*gm5*gm8 + 2*g03*g06*gm1*gm2*gm4*gm5*gm8 + 2*g04*g06*gm1*gm2*gm3*gm5*gm8 +
2*g05*g06*gm1*gm2*gm3*gm4*gm8 + g01*g03*gm2*gm4*gm5*gm7*gm8 + g01*g04*gm2*gm3*gm5*gm7*gm8 +
g01*g05*gm2*gm3*gm4*gm7*gm8 + g01*g07*gm2*gm3*gm4*gm5*gm8 + g01*g08*gm2*gm3*gm4*gm5*gm7 +
g02*g03*gm1*gm4*gm5*gm7*gm8 + g02*g04*gm1*gm3*gm5*gm7*gm8 + g02*g05*gm1*gm3*gm4*gm7*gm8 +
g02*g07*gm1*gm3*gm4*gm5*gm8 + g02*g08*gm1*gm3*gm4*gm5*gm7 + 2*g03*g04*gm1*gm2*gm5*gm7*gm8 +
2*g03*g05*gm1*gm2*gm4*gm7*gm8 + 2*g04*g05*gm1*gm2*gm3*gm5*gm8 + 2*g04*g07*gm1*gm2*gm3*gm5*gm7 +
2*g05*g07*gm1*gm2*gm3*gm4*gm8 + 2*g05*g08*gm1*gm2*gm3*gm4*gm7 + 2*g07*g08*gm1*gm2*gm3*gm4*gm5 +
g01*g06*gm2*gm3*gm4*gm5*gm7 + 2*g03*g07*gm1*gm2*gm4*gm5*gm8 + 2*g03*g08*gm1*gm2*gm3*gm4*gm5*gm7 +
2*g04*g07*gm1*gm2*gm3*gm5*gm8 + 2*g04*g08*gm1*gm2*gm3*gm5*gm7 + 2*g05*g07*gm1*gm2*gm3*gm4*gm5 +
2*g05*g08*gm1*gm2*gm3*gm4*gm7 + 2*g06*g07*gm1*gm2*gm3*gm4*gm5*gm8 + 2*g06*g08*gm1*gm2*gm3*gm4*gm5*gm7 +
2*g07*gm1*gm2*gm3*gm4*gm5*gm8 + 2*g08*gm1*gm2*gm3*gm4*gm5*gm7 + 2*g06*gm1*gm2*gm3*gm4*gm5*gm7*gm8 +
2*g08*gm1*gm2*gm3*gm4*gm7*gm8 + g011*gm2*gm3*gm4*gm5*gm7*gm8 + 2*gm1*gm2*gm3*gm4*gm5*gm7*gm8)
```

/

```
((2*CL*g01*g02*g03*g04*g05*g06 + 2*CL*g01*g02*g03*g04*g05*g08 + 2*CL*g01*g02*g03*g04*g06*g07 +
2*CL*g01*g02*g03*g05*g06*g07 + 2*CL*g01*g02*g03*g04*g07*g08 + 2*CL*g01*g02*g04*g05*g06*g07 +
2*CL*g01*g02*g03*g05*g07*g08 + 2*CL*g01*g03*g04*g05*g06*g07 + 2*CL*g01*g02*g04*g05*g07*g08 +
2*CL*g02*g03*g04*g05*g06*g07 + 2*CL*g01*g02*g03*g05*g06*g011 + 2*CL*g01*g03*g04*g05*g07*g08 +
2*CL*g02*g03*g04*g05*g07*g08 + 2*CL*g01*g02*g03*g05*g08*g011 + 2*CL*g01*g02*g03*g06*g07*g011 +
2*CL*g01*g03*g04*g05*g06*g011 + 2*CL*g01*g02*g03*g07*g08*g011 + 2*CL*g01*g02*g05*g06*g07*g011 +
2*CL*g01*g03*g04*g05*g08*g011 + 2*CL*g01*g03*g04*g06*g07*g011 + 2*CL*g01*g02*g05*g07*g08*g011 +
```





# MATLAB Solution Continued 3:

```
2*CL*g01*g02*g05*g011*gm7*gm8 + 2*CL*g01*g02*g07*g011*gm5*gm8 + 2*CL*g01*g02*g08*g011*gm5*gm7 +
2*CL*g01*g03*g04*g011*gm7*gm8 + 2*CL*g01*g03*g07*g011*gm4*gm8 + 2*CL*g01*g03*g08*g011*gm4*gm7 +
2*CL*g01*g04*g06*g011*gm5*gm7 + 2*CL*g01*g05*g06*g011*gm4*gm7 + 2*CL*g01*g06*g07*g011*gm4*gm5 +
2*CL*g02*g03*g06*g011*gm5*gm7 + 2*CL*g02*g05*g06*g011*gm3*gm7 + 2*CL*g02*g06*g07*g011*gm3*gm5 +
2*CL*g01*g04*g05*g011*gm7*gm8 + 2*CL*g01*g04*g07*g011*gm5*gm8 + 2*CL*g01*g04*g08*g011*gm5*gm7 +
2*CL*g01*g05*g07*g011*gm4*gm8 + 2*CL*g01*g05*g08*g011*gm4*gm7 + 2*CL*g01*g07*g08*g011*gm4*gm5 +
2*CL*g02*g03*g05*g011*gm7*gm8 + 2*CL*g02*g03*g07*g011*gm5*gm8 + 2*CL*g02*g03*g08*g011*gm5*gm7 +
2*CL*g02*g05*g07*g011*gm3*gm8 + 2*CL*g02*g05*g08*g011*gm3*gm7 + 2*CL*g02*g07*g08*g011*gm3*gm5 +
2*CL*g03*g04*g06*g011*gm5*gm7 + 2*CL*g03*g05*g06*g011*gm4*gm7 + 2*CL*g03*g06*g07*g011*gm4*gm5 +
2*CL*g04*g05*g06*g011*gm3*gm7 + 2*CL*g04*g06*g07*g011*gm3*gm5 + 2*CL*g05*g06*g07*g011*gm3*gm4 +
2*CL*g03*g04*g05*g011*gm7*gm8 + 2*CL*g03*g04*g07*g011*gm5*gm8 + 2*CL*g03*g04*g08*g011*gm5*gm7 +
2*CL*g03*g05*g07*g011*gm4*gm8 + 2*CL*g03*g05*g08*g011*gm4*gm7 + 2*CL*g03*g07*g08*g011*gm4*gm5 +
2*CL*g04*g05*g07*g011*gm3*gm8 + 2*CL*g04*g05*g08*g011*gm3*gm7 + 2*CL*g04*g07*g08*g011*gm3*gm5 +
2*CL*g05*g07*g08*g011*gm3*gm4 + 2*CL*g01*g03*g05*gm2*gm4*gm5 + 2*CL*g01*g02*g03*gm4*gm5*gm8 +
2*CL*g01*g03*g04*gm2*gm5*gm8 + 2*CL*g01*g03*g05*gm2*gm4*gm8 + 2*CL*g01*g03*g06*gm2*gm4*gm7 +
2*CL*g01*g03*g08*gm2*gm4*gm5 + 2*CL*g02*g02*g06*gm1*gm3*gm5 + 2*CL*g02*g03*g06*gm1*gm5*gm7 +
2*CL*g02*g05*g06*gm1*gm3*gm7 + 2*CL*g02*g06*g07*gm1*gm3*gm5 + 2*CL*g01*g02*g03*gm4*gm7*gm8 +
2*CL*g01*g02*g06*gm4*gm5*gm7 + 2*CL*g01*g03*g04*gm2*gm7*gm8 + 2*CL*g01*g03*g07*gm2*gm4*gm8 +
2*CL*g01*g03*g08*gm2*gm4*gm7 + 2*CL*g01*g04*g06*gm2*gm4*gm7 + 2*CL*g01*g05*g06*gm2*gm4*gm7 +
2*CL*g01*g06*g07*gm2*gm4*gm5 + 2*CL*g01*g02*g03*gm5*gm7*gm8 + 2*CL*g01*g02*g05*gm3*gm7*gm8 +
2*CL*g01*g02*g07*gm3*gm5*gm8 + 2*CL*g01*g02*g08*gm3*gm5*gm7 + 2*CL*g01*g03*g06*gm4*gm5*gm7 +
2*CL*g01*g04*g06*gm3*gm5*gm7 + 2*CL*g01*g05*g06*gm3*gm4*gm7 + 2*CL*g01*g06*g07*gm3*gm4*gm5 +
2*CL*g02*g03*g05*gm1*gm7*gm8 + 2*CL*g02*g03*g07*gm1*gm5*gm8 + 2*CL*g02*g03*g08*gm1*gm5*gm7 +
2*CL*g02*g05*g07*gm1*gm3*gm8 + 2*CL*g02*g05*g08*gm1*gm3*gm7 + 2*CL*g02*g07*g08*gm1*gm3*gm5 +
2*CL*g03*g04*g06*gm1*gm5*gm7 + 2*CL*g03*g05*g06*gm1*gm4*gm7 + 2*CL*g03*g06*g07*gm1*gm4*gm5 +
2*CL*g04*g05*g06*gm1*gm3*gm8 + 2*CL*g04*g06*g07*gm1*gm3*gm7 + 2*CL*g04*g08*gm1*gm3*gm4 +
2*CL*g01*g02*g04*gm5*gm7*gm8 + 2*CL*g01*g02*g05*gm4*gm7*gm8 + 2*CL*g01*g02*g07*gm4*gm5*gm8 +
2*CL*g01*g04*g08*gm4*gm5*gm7 + 2*CL*g03*g04*g07*gm1*gm3*gm8 + 2*CL*g03*g04*g08*gm1*gm3*gm7 +
2*CL*g03*g05*g07*gm1*gm3*gm4 + 2*CL*g03*g05*g08*gm1*gm3*gm7 + 2*CL*g03*g07*g08*gm1*gm3*gm5 +
2*CL*g04*g05*g07*gm1*gm3*gm8 + 2*CL*g04*g05*g08*gm1*gm3*gm7 + 2*CL*g04*g07*gm1*gm3*gm4 +
2*CL*g04*g08*gm1*gm3*gm7 + 2*CL*g05*g06*gm1*gm3*gm4 + 2*CL*g05*g07*gm1*gm3*gm8 +
2*CL*g05*g08*gm1*gm3*gm7 + 2*CL*g06*g07*gm1*gm3*gm4 + 2*CL*g06*g08*gm1*gm3*gm7 +
2*CL*g07*g08*gm1*gm3*gm4 + 2*CL*g02*g04*gm3*gm5*gm7 + 2*CL*g02*g05*gm3*gm4*gm7 +
2*CL*g02*g07*gm3*gm4*gm5*gm8 + 2*CL*g02*g08*gm3*gm4*gm5*gm7 + 2*CL*g03*g04*gm2*gm5*gm8 +
2*CL*g03*g05*gm2*gm4*gm7*gm8 + 2*CL*g03*g07*gm2*gm4*gm5*gm8 + 2*CL*g03*g08*gm2*gm4*gm5*gm7 +
```



## MATLAB Solution Continued 5:

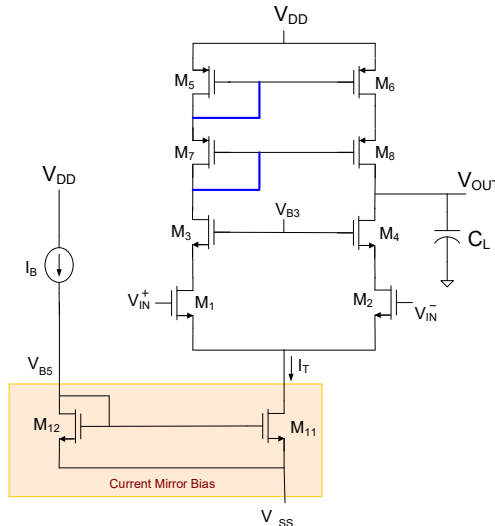
```
2*g01*g02*g04*g06*g011*gm5*gm7 + 2*g01*g02*g04*g05*g011*gm7*gm8 + 2*g01*g02*g04*g07*g011*gm5*gm8
+ 2*g01*g02*g04*g08*g011*gm5*gm7 + 2*g01*g03*g06*g08*g011*gm4*gm5 +
2*g02*g03*g04*g06*g011*gm5*gm7 + 2*g02*g04*g05*g06*g011*gm3*gm7 + 2*g02*g04*g06*g07*g011*gm3*gm5
+ 2*g01*g02*g06*g08*g011*gm5*gm7 + 2*g01*g03*g06*g08*g011*gm4*gm7 +
2*g02*g03*g04*g05*g011*gm7*gm8 + 2*g02*g03*g04*g07*g011*gm5*gm8 + 2*g02*g03*g04*g08*g011*gm5*gm7
+ 2*g02*g04*g05*g07*g011*gm3*gm8 + 2*g02*g04*g05*g08*g011*gm3*gm7 +
2*g02*g04*g07*g08*g011*gm3*gm5 + 2*g01*g04*g06*g08*g011*gm5*gm7 + 2*g01*g05*g06*g08*g011*gm4*gm7
+ 2*g01*g06*g07*g08*g011*gm4*gm5 + 2*g02*g03*g05*g06*g08*g011*gm5*gm7 +
2*g02*g05*g06*g08*g011*gm3*gm7 + 2*g02*g06*g07*g08*g011*gm3*gm5 + 2*g03*g04*g06*g08*g011*gm5*gm7
+ 2*g03*g05*g06*g08*g011*gm4*gm7 + 2*g03*g06*g07*g08*g011*gm4*gm5 +
2*g04*g05*g06*g08*g011*gm3*gm7 + 2*g04*g06*g07*g08*g011*gm3*gm5 + 2*g05*g06*g07*g08*g011*gm3*gm4
+ 2*g01*g02*g04*g06*gm3*gm5*gm7 + 2*g02*g03*g04*g06*gm1*gm5*gm7 + 2*g02*g04*g05*g06*gm1*gm3*gm7
+ 2*g02*g04*g06*g07*gm1*gm3*gm5 + 2*g01*g02*g04*g06*gm3*gm5*gm8 + 2*g01*g03*g06*g08*gm2*gm4*gm5
+ 2*g02*g03*g04*g05*gm1*gm7*gm8 + 2*g02*g03*g04*g07*gm1*gm5*gm8 + 2*g02*g03*g04*g08*gm1*gm5*gm7
+ 2*g02*g04*g05*g07*gm1*gm3*gm8 + 2*g02*g04*g05*g08*gm1*gm3*gm7 + 2*g02*g04*g07*g08*gm1*gm3*gm5
+ 2*g01*g02*g04*g06*gm3*gm7*gm8 + 2*g01*g03*g06*g08*gm2*gm4*gm7 + 2*g02*g03*g04*g06*gm1*gm7*gm8
+ 2*g02*g04*g06*g07*gm1*gm3*gm8 + 2*g01*g02*g06*g08*gm3*gm5*gm7 + 2*g02*g03*g06*g08*gm1*gm5*gm7
+ 2*g02*g05*g06*g08*gm1*gm3*gm7 + 2*g02*g06*g07*g08*gm1*gm3*gm5 + 2*g01*g02*g06*g08*gm2*gm4*gm7
+ 2*g01*g06*g07*g08*gm2*gm4*gm5 + 2*g02*g03*g04*g08*gm1*gm7*gm8 + 2*g02*g04*g07*g08*gm1*gm3*gm8
+ 2*g01*g03*g06*g08*gm4*gm5*gm7 + 2*g01*g04*g06*g08*gm3*gm5*gm7 + 2*g01*g05*g06*g08*gm2*gm4*gm7
+ 2*g01*g06*g07*g08*gm2*gm4*gm5 + 2*g02*g03*g04*g08*gm1*gm7*gm8 + 2*g02*g04*g07*g08*gm1*gm3*gm8
+ 2*g01*g05*g06*g08*gm3*gm4*gm7 + 2*g01*g06*g07*g08*gm3*gm5*gm7 + 2*g03*g04*g06*g08*gm1*gm4*gm7
+ 2*g03*g05*g06*g08*gm1*gm4*gm5 + 2*g04*g05*g06*g08*gm1*gm3*gm7 + 2*g04*g06*g07*g08*gm1*gm3*gm5
+ 2*g05*g06*g07*g08*gm1*gm3*gm4 + 2*g02*g03*g06*g08*gm4*gm5*gm7 + 2*g02*g04*g06*g08*gm3*gm5*gm7
+ 2*g02*g05*g06*g08*gm3*gm4*gm7 + 2*g02*g06*g07*g08*gm3*gm4*gm5 + 2*g03*g04*g06*g08*gm2*gm5*gm7
+ 2*g03*g05*g06*g08*gm2*gm4*gm7 + 2*g03*g06*g07*g08*gm2*gm4*gm5 + 2*g04*g05*g06*g08*gm2*gm3*gm7
+ 2*g04*g06*g07*g08*gm2*gm3*gm5 + 2*g05*g06*g07*g08*gm2*gm3*gm4 + 2*g01*g02*g04*g011*gm5*gm7*gm8
+ 2*g02*g04*g05*g011*gm3*gm5*gm7 + 2*g02*g03*g04*g011*gm5*gm7*gm8 +
2*g02*g04*g05*g011*gm3*gm7*gm8 + 2*g02*g04*g07*g011*gm3*gm5*gm8 + 2*g02*g04*g08*g011*gm3*gm5*gm7
+ 2*g01*g06*g08*g011*gm4*gm5*gm7 + 2*g02*g06*g08*g011*gm3*gm5*gm7 +
2*g03*g06*g08*g011*gm4*gm5*gm7 + 2*g04*g06*g08*g011*gm3*gm5*gm7 + 2*g05*g06*g08*g011*gm3*gm4*gm7
+ 2*g06*g07*g08*g011*gm3*gm4*gm5 + 2*g02*g04*g06*gm1*gm3*gm5*gm7 + 2*g02*g04*g06*gm1*gm3*gm5*gm8
+ 2*g01*g02*g04*gm3*gm5*gm7*gm8 + 2*g02*g03*g04*gm1*gm5*gm7*gm8 + 2*g02*g04*g05*gm1*gm3*gm7*gm8
+ 2*g02*g04*g07*gm1*gm3*gm5*gm8 + 2*g02*g04*g08*gm1*gm3*gm5*gm7 + 2*g02*g04*g06*gm1*gm3*gm7*gm8
+ 2*g02*g06*g08*gm1*gm3*gm5*gm7 + 2*g01*g06*g08*gm2*gm4*gm5*gm7 + 2*g02*g04*g08*gm1*gm3*gm7*gm8
+ 2*g05*g06*g08*gm1*gm4*gm5*gm7 + 2*g03*g06*g08*gm1*gm4*gm5*gm7 + 2*g04*g06*g08*gm1*gm3*gm5*gm7
+ 2*g05*g06*g08*gm3*gm4*gm5 + 2*g06*g07*g08*gm1*gm3*gm4*gm5 + 2*g02*g06*g08*gm3*gm4*gm5*gm7
+ 2*g03*g06*g08*gm2*gm4*gm5*gm7 + 2*g04*g06*g08*gm2*gm3*gm5*gm7 + 2*g05*g06*g08*gm2*gm3*gm4*gm7
+ 2*g06*g07*g08*gm2*gm3*gm4*gm5 + 2*g02*g04*g011*gm3*gm5*gm7*gm8 +
2*g06*g08*g011*gm3*gm4*gm5*gm7 + 2*g02*g04*g04*gm1*gm3*gm5*gm7*gm8 + 2*g06*g08*gm1*gm3*gm4*gm5*gm7 +
2*g06*g08*gm2*gm3*gm4*gm5*gm7 )
```

- Difference Mode Gain has only approximately 1100 product terms
- Difference Mode Gain has approximately 7700 small-signal parameters in expression
  - Extremely difficult to get insight into how this relatively simple circuit performs from this solution
- This does not even include the common-mode gain expression !



# Where this started

## Telescopic Cascode Op Amp with Mirror-connected Counterpart Circuit



$$A_d(s) = \frac{-g_{m1}}{sC_L + g_{o1} \frac{g_{o3}}{g_{m3}} + g_{o5} \frac{g_{o7}}{g_{m7}}}$$

$$A_d(s) = \frac{-g_{m7}g_{m1}g_{m3}}{sC_L g_{m7}g_{m3} + g_{m7}g_{o3}g_{o1} + g_{m3}g_{o5}g_{o7}}$$

- Some assumptions were made to simplify analysis
  - $V_{ac}=0$  at “approximate axis of symmetry”
  - Matched left and right side transistors
  - Current mirror used to mirror left-side current to right side
- Difference Mode Gain has only approximately 1100 product terms
- Difference Mode Gain has approximately 7700 small-signal parameters in expression

How many product terms are present in the simplified analysis?

How many small-signal parameters are in simplified expression?

- Simplified Difference Mode Gain has 4 product terms
- Simplified Difference Mode Gain has 12 small-signal parameters in expression

# Be careful about what you ask for, you may be able to get it !

- How important is it to develop good approximate analysis methods for an op amp of this complexity?
- And many useful op amp circuits will have much more complexity !!

Will you impress your boss if you use an “exact” analysis?





Stay Safe and Stay Healthy !

**End of Lecture 7**