Lecture 4
Driving Point Impedance (DPI) Method

Prof. Guoyong Shi
shiguoyong@sjtu.edu.cn
School of Micro/Nano-electronics
Shanghai Jiao Tong University
Fall 2015

Foreword

- **The DPI method introduced in this lecture is based on the following paper**

- **A. Ochoa, Jr., "A systematic approach to the analysis of general and feedback circuits and systems using signal flow graphs and driving-point impedance," *IEEE Trans. on CAS-II: Analog and Digital Signal Processing*, vol. 45, no. 2, Feb. 1998, pp. 187-195.**

Outline

- **Driving Point Impedance (DPI) method**
- **Signal Flow Graph (SFG) method**
- **Examples**
  - **Two-stage opamp transfer function**

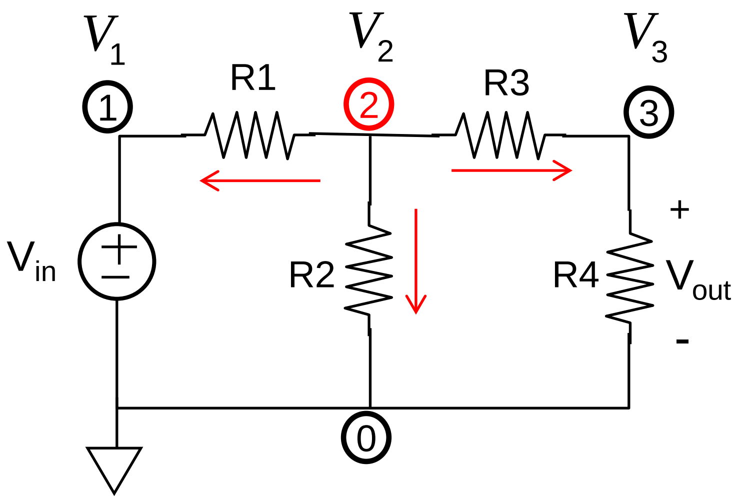
- **Appendix 1: Mason's rule**

Basic Steps for DPI

1. Introduce **internal voltage variables**
2. Derive **current-driven impedances** in Norton form
3. Draw **Signal Flow Graph (SFG)**
4. Derive I/O functions

DPI Intuition

- **DPI is closely related to the <u>modified nodal analysis</u> (MNA) method.**



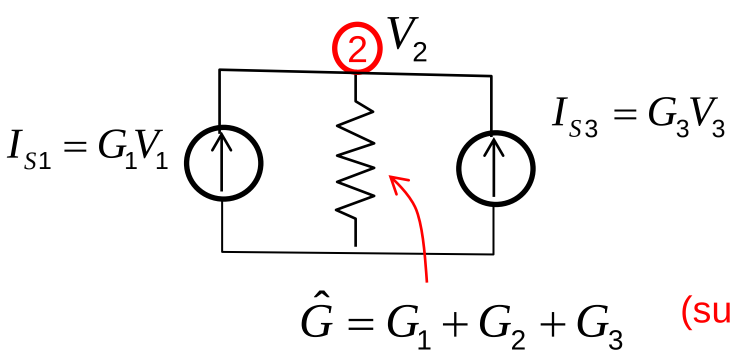
The sum of currents at node "2" (all leaving):

$$G_1(V_2-V_1) + G_2(V_2) + G_3(V_2-V_3) = 0$$

Separating the voltages

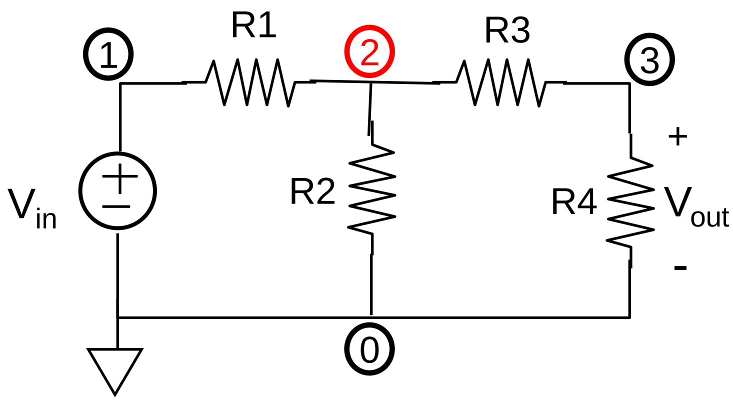
$$-G_1V_1 + (G_1+G_2+G_3)\,V_2 - G_3V_3 = 0$$

viewed as current srcs

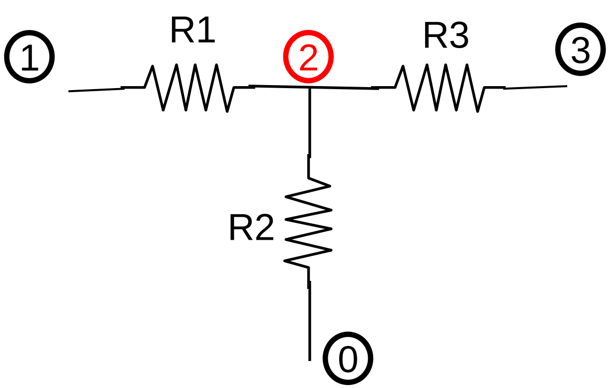


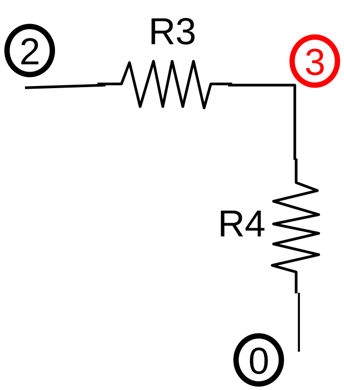
(sum of admittances)

Circuit Example



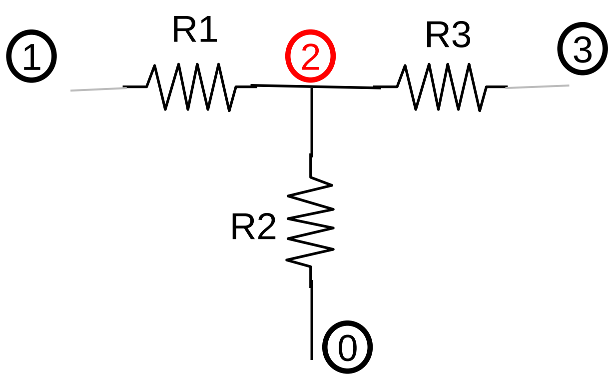
Consider two driving points at nodes "2" and "3"

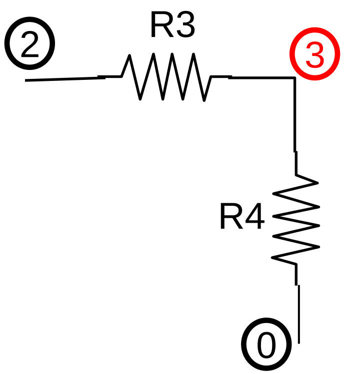




Circuit Conversion

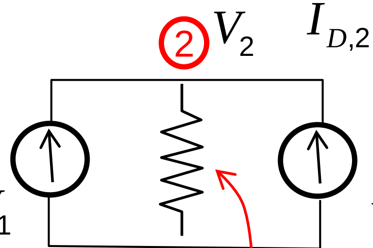
$I_{D,n}$ denotes the driving point (DP) current seen at node "n".





(total DP current)

$$I_{D,2} = G_1V_1 + G_3V_3$$

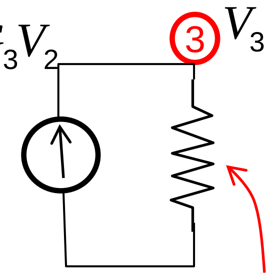


$$I_{S1} = G_1V_1$$

$$I_{S3} = G_3V_3$$

$$\hat{G} = G_1 + G_2 + G_3$$

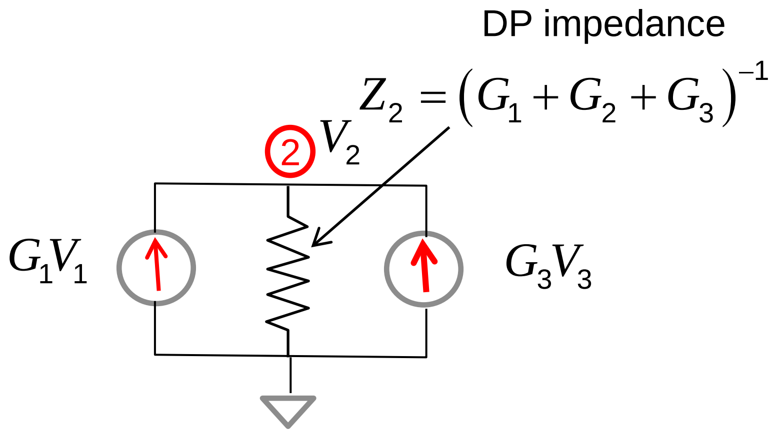
$$I_{D,3} = G_3V_2$$

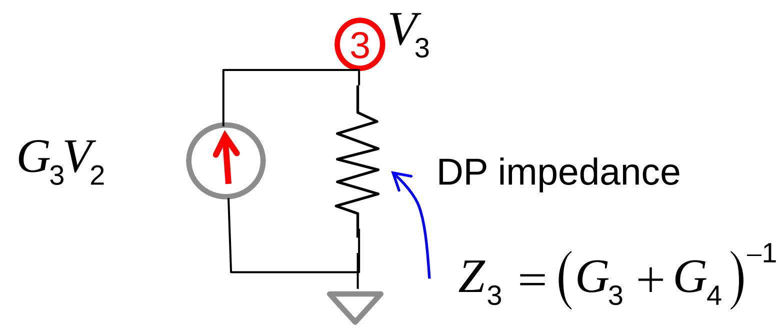


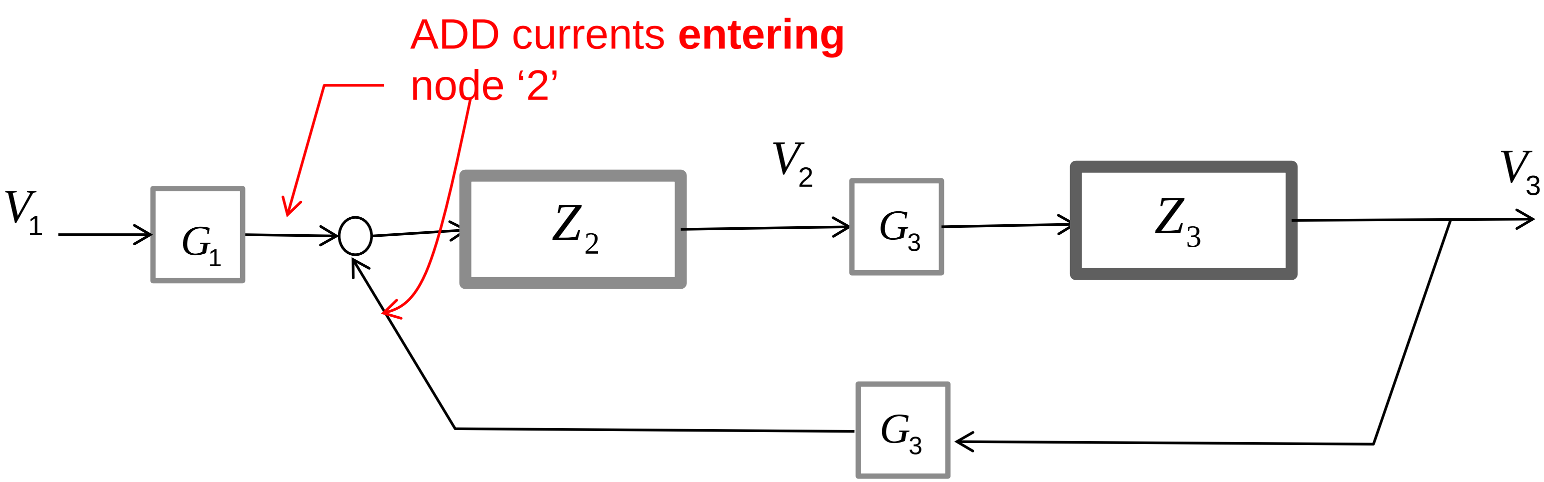
$$\hat{G} = G_3 + G_4$$

(lumped DP admittance)

Drawing SFG



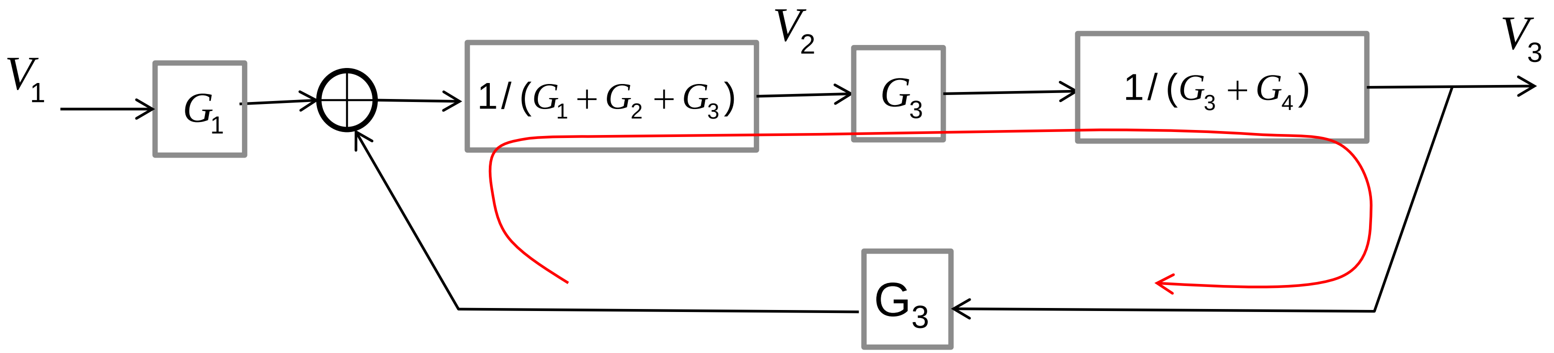




$$I_{D,m} = \sum_n G_{m,n} V_n$$

$G_{m,n}$ = admittance driving node m from node n

Deriving TF from SGF



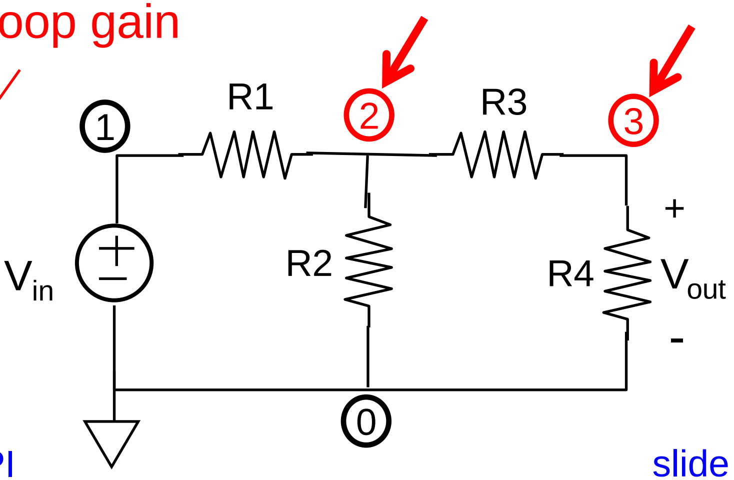
Mason's Rule

feedforward gain

loop gain

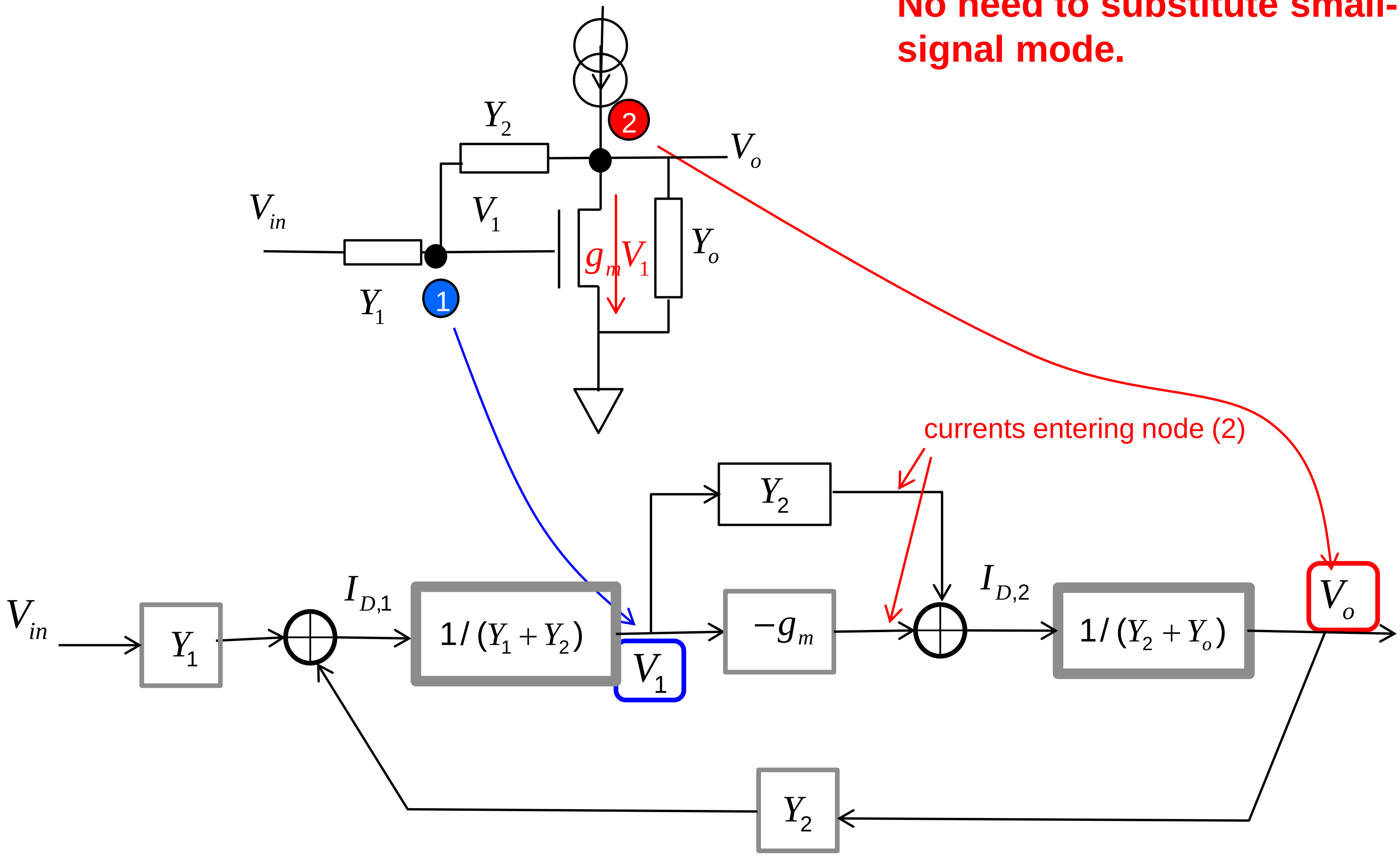
$$\Rightarrow \frac{V_3}{V_1} = \frac{G_1G_3/(G_1+G_2+G_3)(G_3+G_4)}{1-G_3^2/(G_1+G_2+G_3)(G_3+G_4)}$$

$$= \frac{G_1G_3}{(G_1+G_2+G_3)(G_3+G_4)-G_3^2}$$



Example 1

No need to substitute small-signal mode.



From Ochoa (1998)

Highlights of Steps

- **Mark the nodes (as the Driving Points)**
- **Capture the Driving Point Impedance (DPI) @ each node.**
- **Collect the currents entering each DPI node.**
- **Complete the SFG**

Example 2: Current Mirror

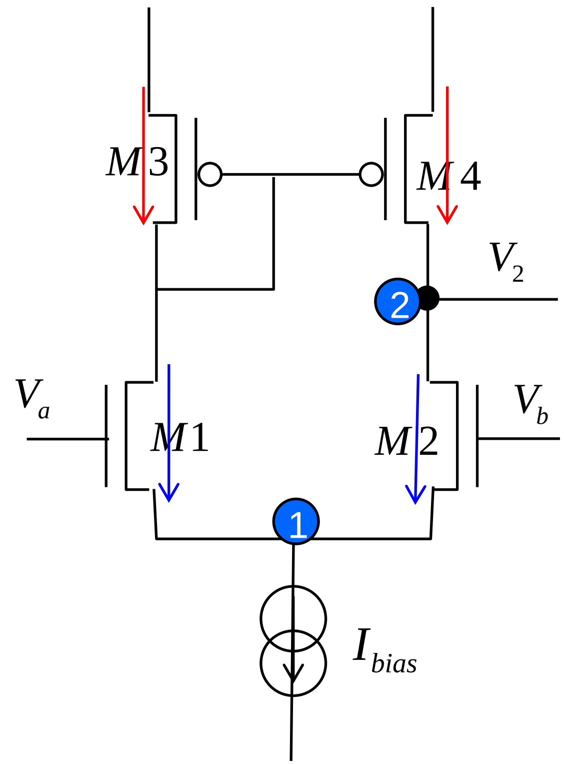
$$I_{M1} = g_{m1}(V_a - V_1); \qquad I_{M2} = g_{m2}(V_b - V_1);$$

$$I_{M4} = I_{M3} = I_{M1}$$

$$I_{M1} + I_{M2} = 0$$ (small-signal $I_{bias} = 0$)

$$\Rightarrow \quad g_{m1}(V_a - V_1) + g_{m2}(V_b - V_1) = 0$$

$$\Rightarrow \quad g_{m1}V_a + g_{m2}V_b = (g_{m1} + g_{m2})V_1$$



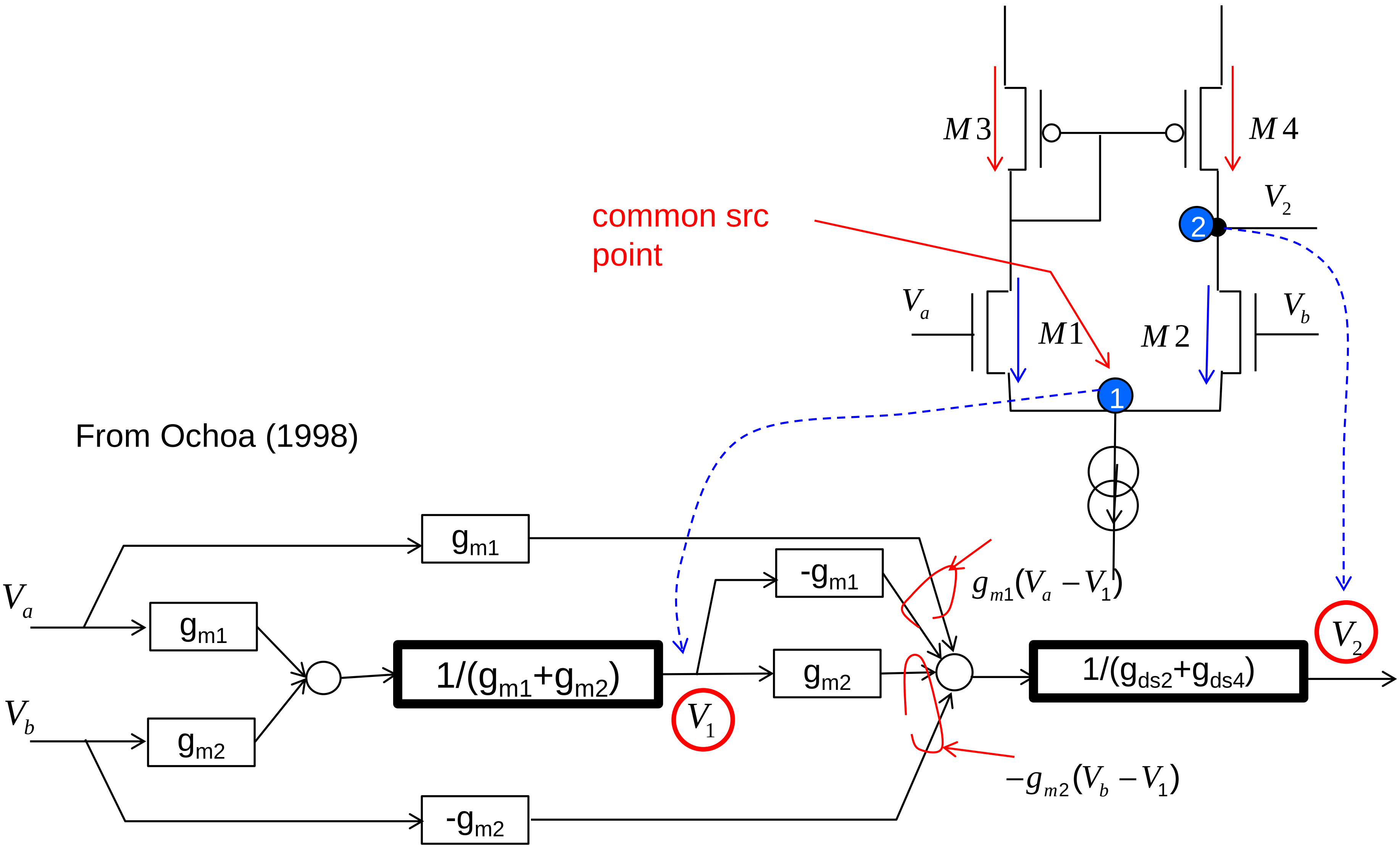
DPI at node "2"

$$V_2 = (I_{M4} - I_{M2})\frac{1}{g_{ds2} + g_{ds4}}$$

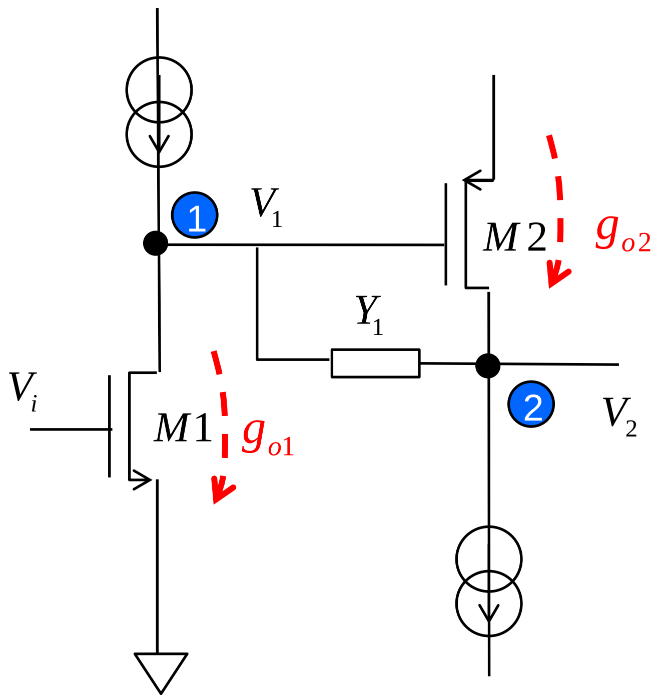
$$= \left[g_{m1}(V_a - V_1) - g_{m2}(V_b - V_1)\right]\frac{1}{g_{ds2} + g_{ds4}}$$

Both I_{M4} and ($- I_{M2}$) entering node "2"

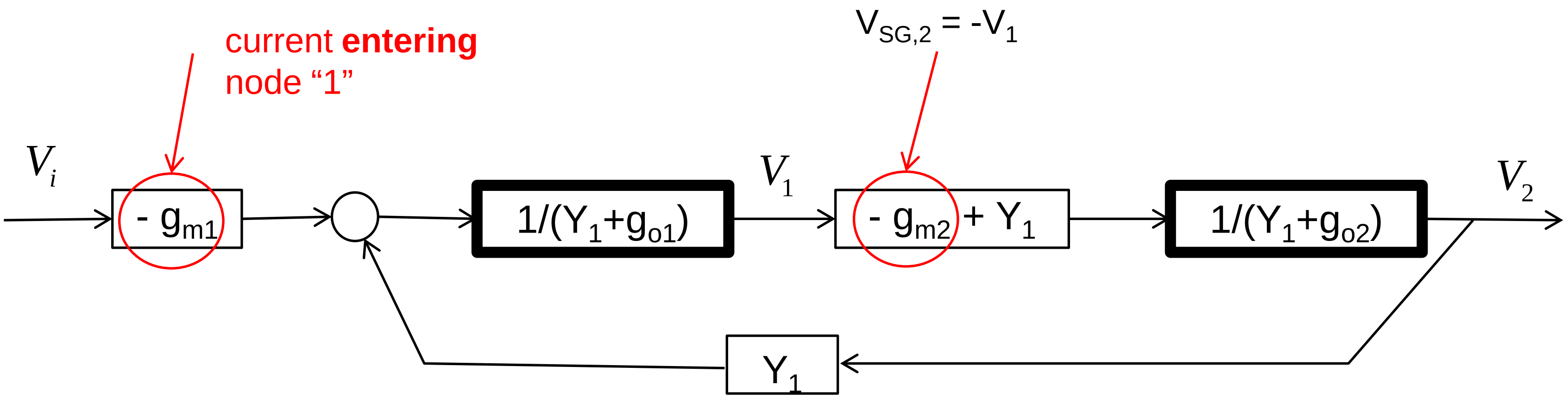
SFG for the Current Mirror



Example 3

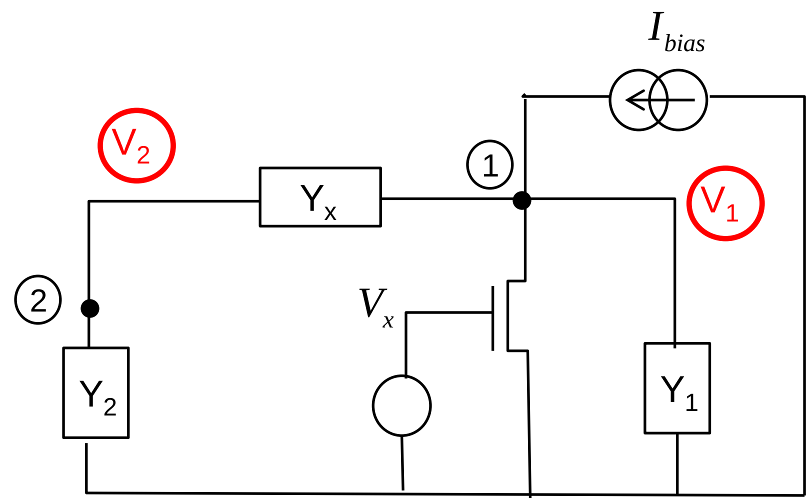


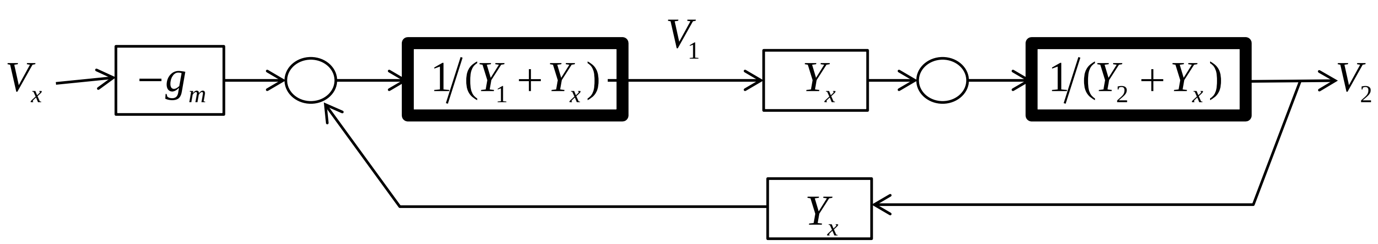
g_{o1} & g_{o2} are output admittance of M1 & M2



From Ochoa (1998)

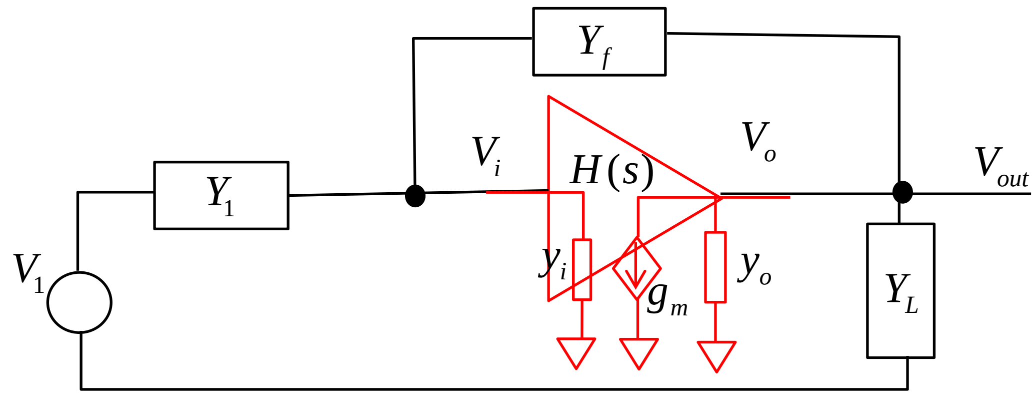
Example 4

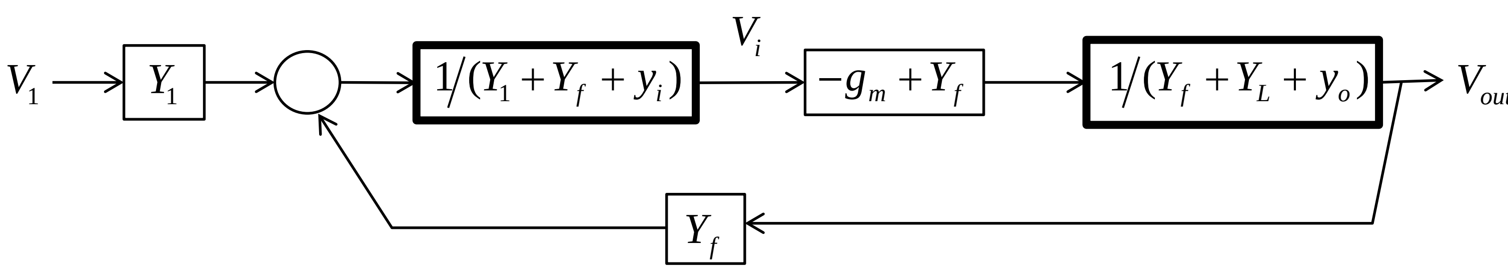




From Ochoa (1998)

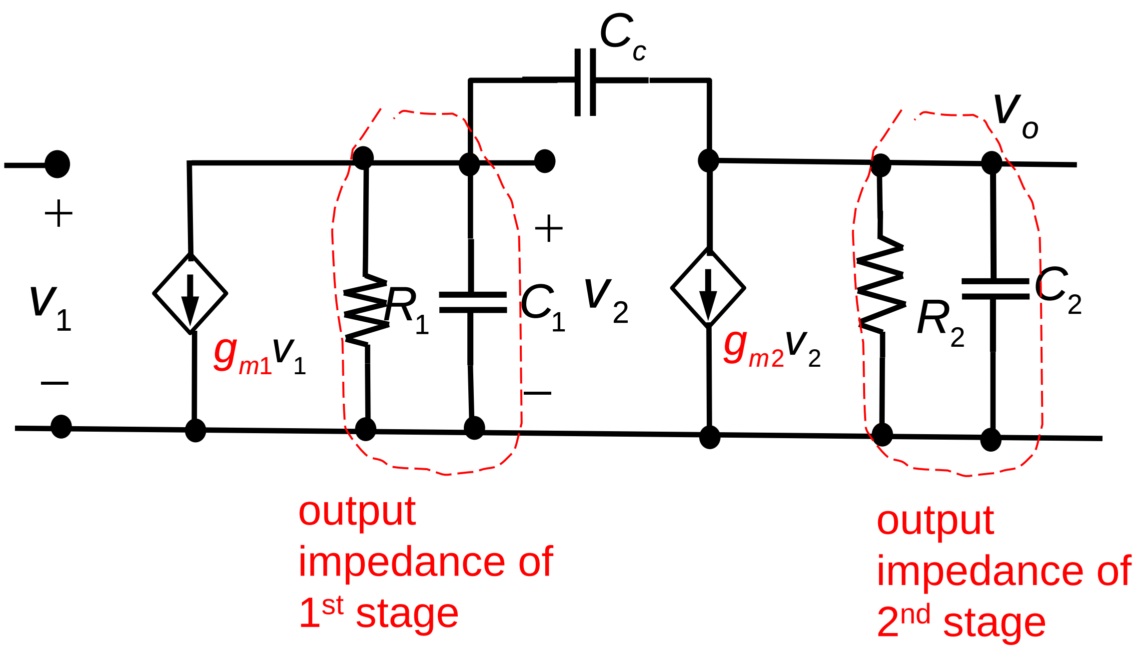
Example 5



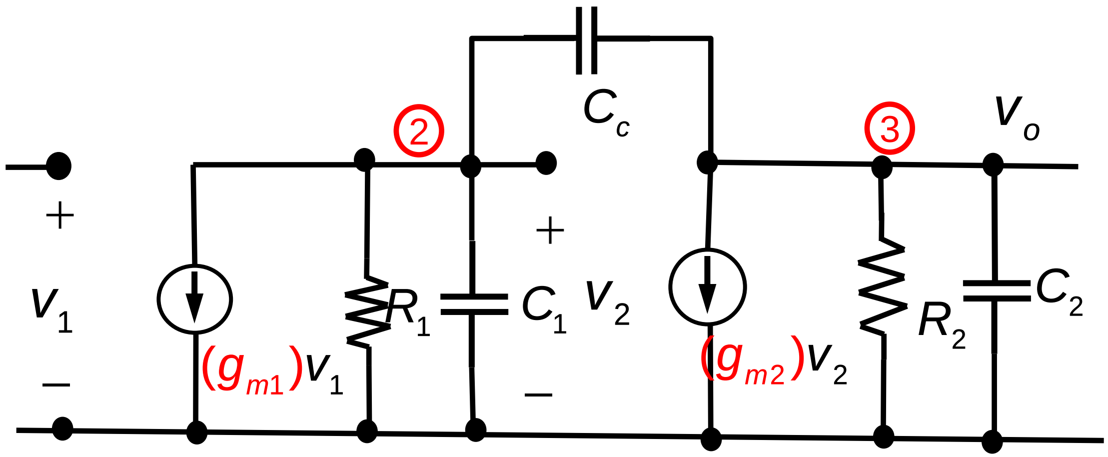


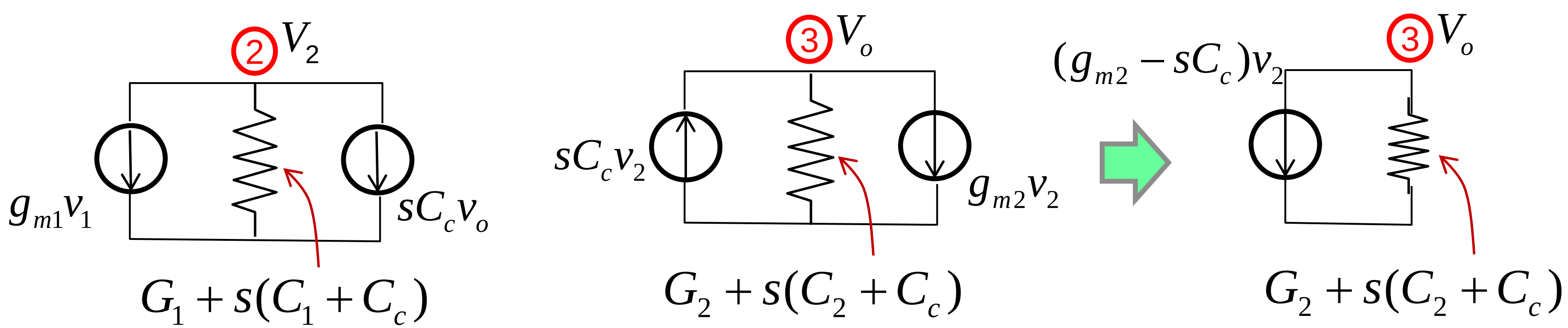
From Ochoa (1998)

Two-Stage Opamp Analysis

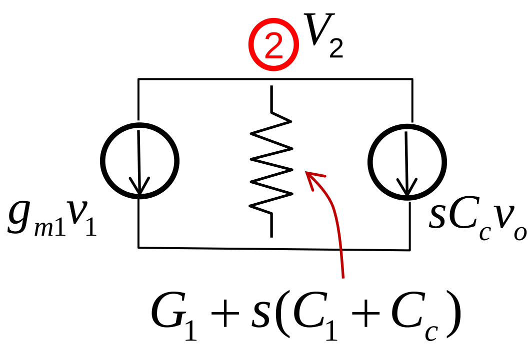


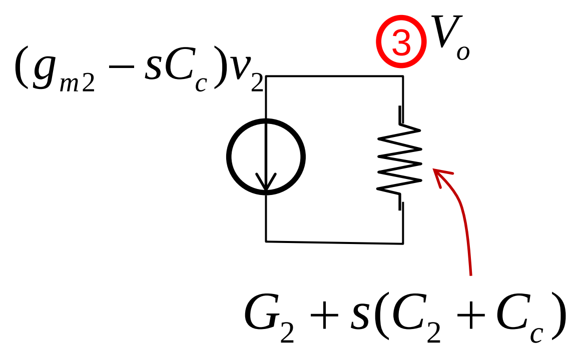
DPI Analysis

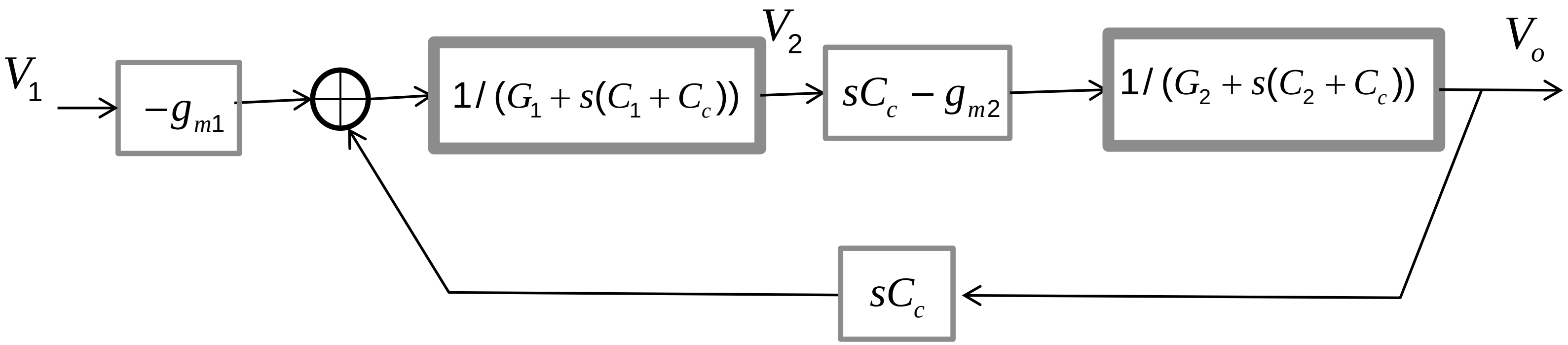




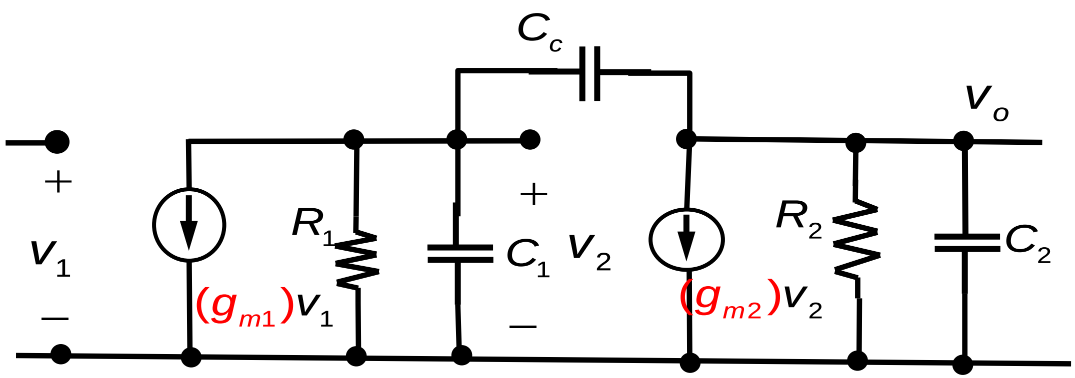
Two-stage opamp (SFG)







Two-stage opamp -- I/O TF



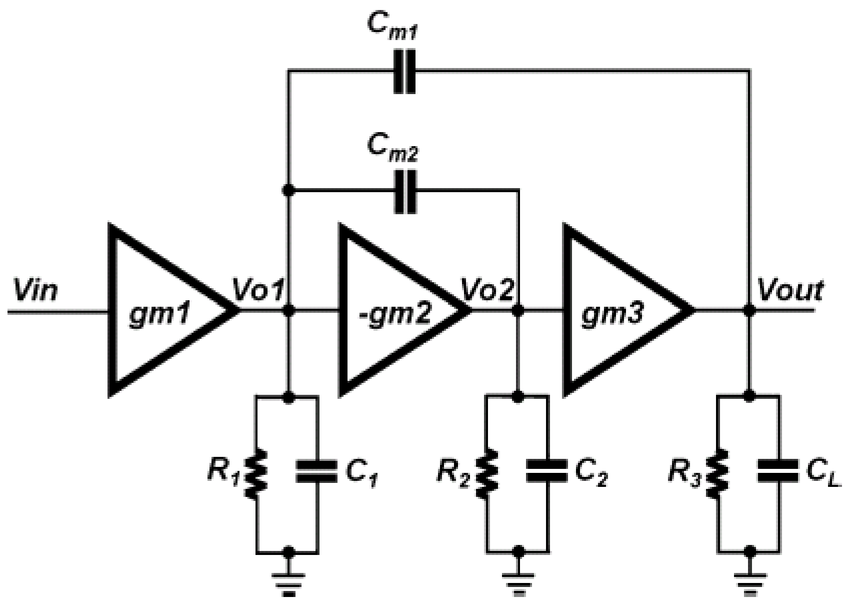
One zero in RHP

$$\frac{V_o}{V_1} = \frac{g_{m1}(g_{m2} - sC_c)}{[G_1 + s(C_1 + C_c)][G_2 + s(C_2 + C_c)] - sC_c(sC_c - g_{m2})}$$

$$D(s) \equiv G_1G_2 + s[G_1C_2 + G_2C_1 + C_c(G_1 + G_2 + g_{m2})] + s^2[C_1C_2 + (C_1 + C_2)C_c]$$

Two poles in LHP

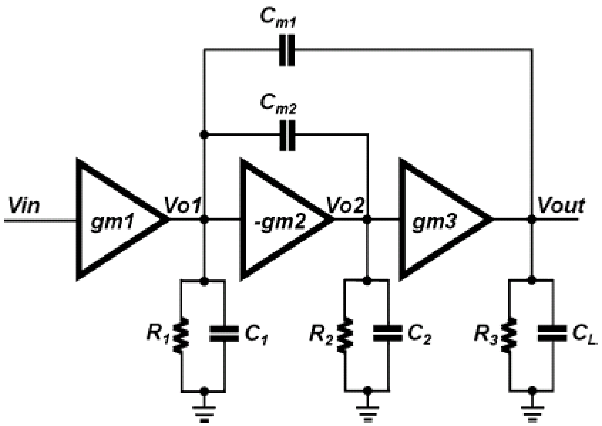
RNMC opamp



Reversed nested Miller compensation (RNMC) amplifier

Y. –J. Kim and S. –H. Lee, “A 10-b 120-MS/s 45 nm CMOS ADC using a re-configurable three-stage switched amplifier,” Analog Integrated Circuits and Signal Processing, vol. 72, 75-87, 2012.

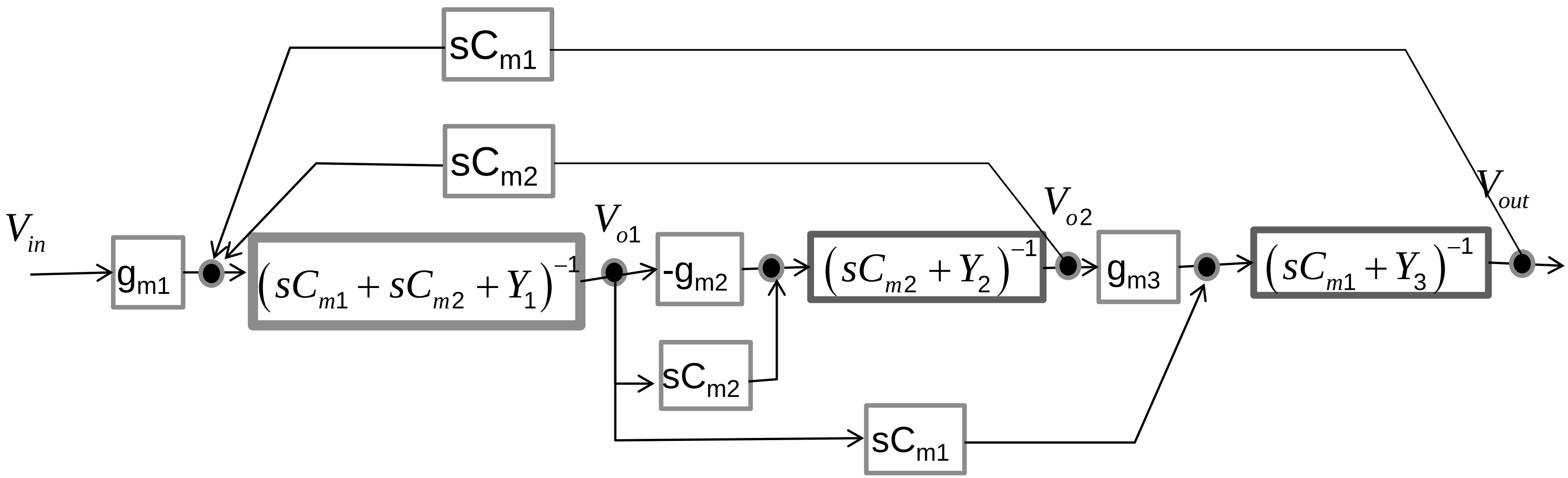
DPI Analysis



$$Y_1 = 1/R_1 + sC_1$$

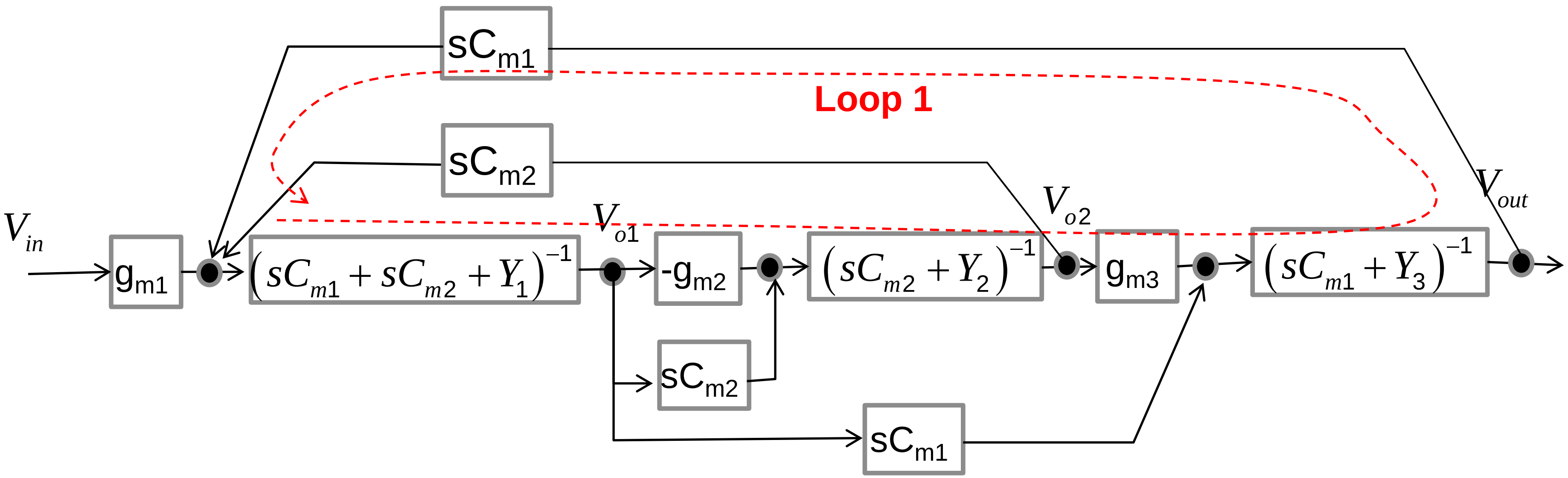
$$Y_2 = 1/R_2 + sC_2$$

$$Y_3 = 1/R_3 + sC_L$$



Three loops in the SFG

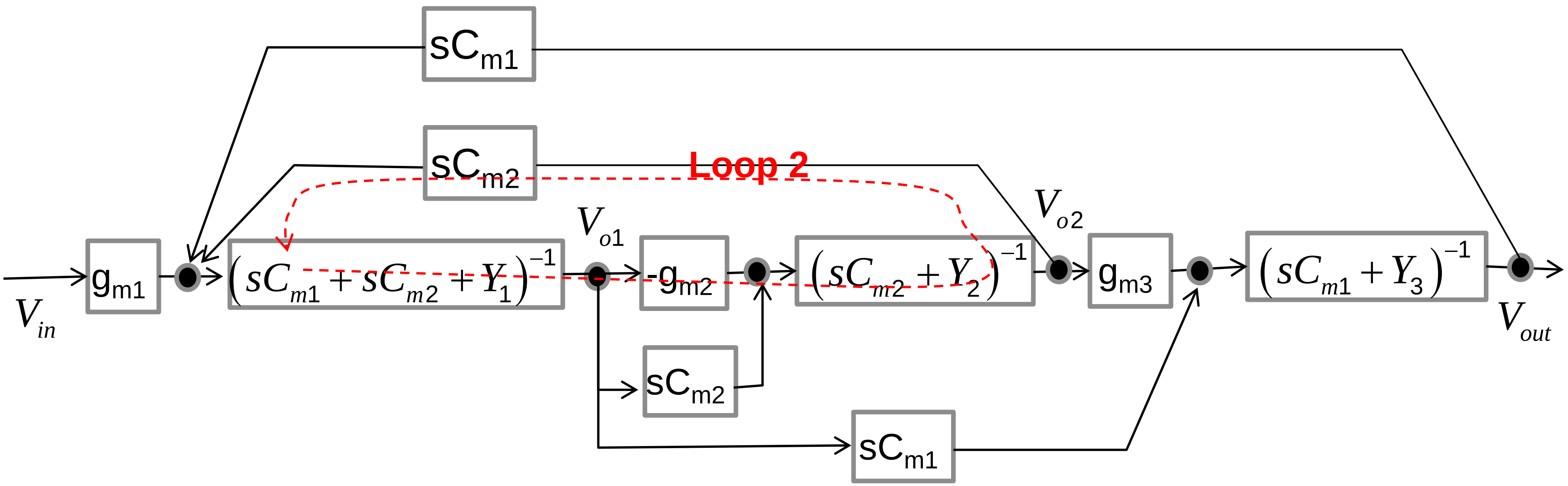
Transfer Function



Forward path: $$P=\frac{g_{m1}g_{m3}(sC_{m2}-g_{m2})+g_{m1}sC_{m1}(sC_{m2}+Y_2)}{(sC_{m1}+sC_{m2}+Y_1)(sC_{m2}+Y_2)(sC_{m1}+Y_3)}$$

Loop 1: $$L_1=\frac{sC_{m1}g_{m3}(sC_{m2}-g_{m2})}{(sC_{m1}+sC_{m2}+Y_1)(sC_{m2}+Y_2)(sC_{m1}+Y_3)}$$

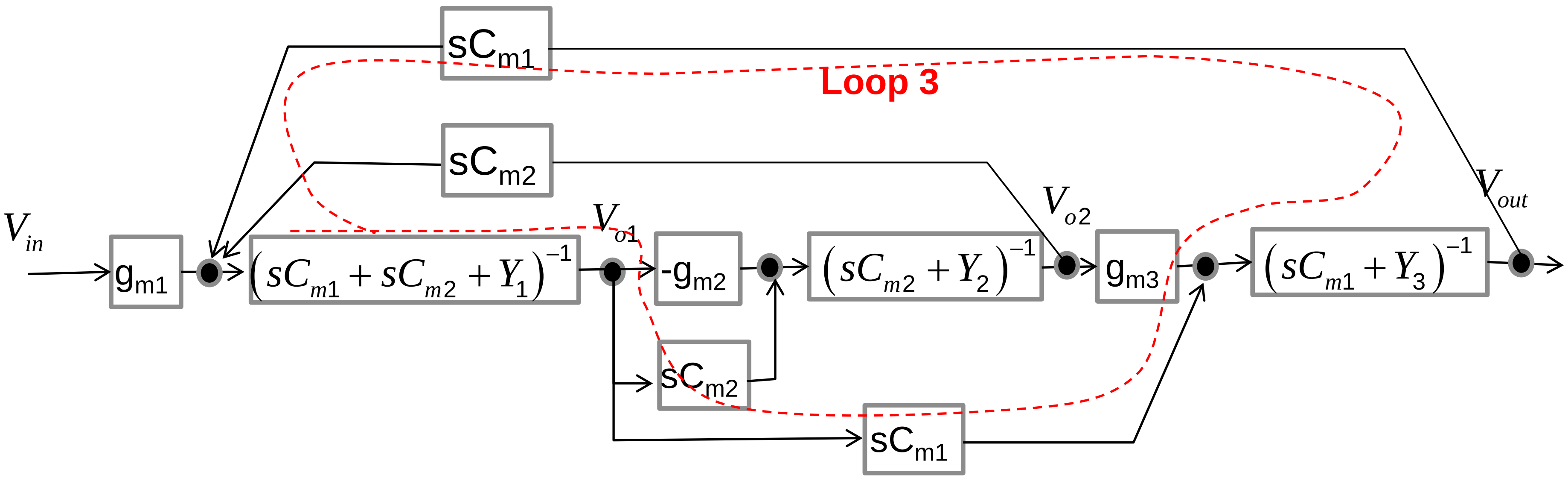
Transfer Function



Loop 2:

$$L_2 = \frac{sC_{m2}(sC_{m2} - g_{m2})}{(sC_{m1} + sC_{m2} + Y_1)(sC_{m2} + Y_2)}$$

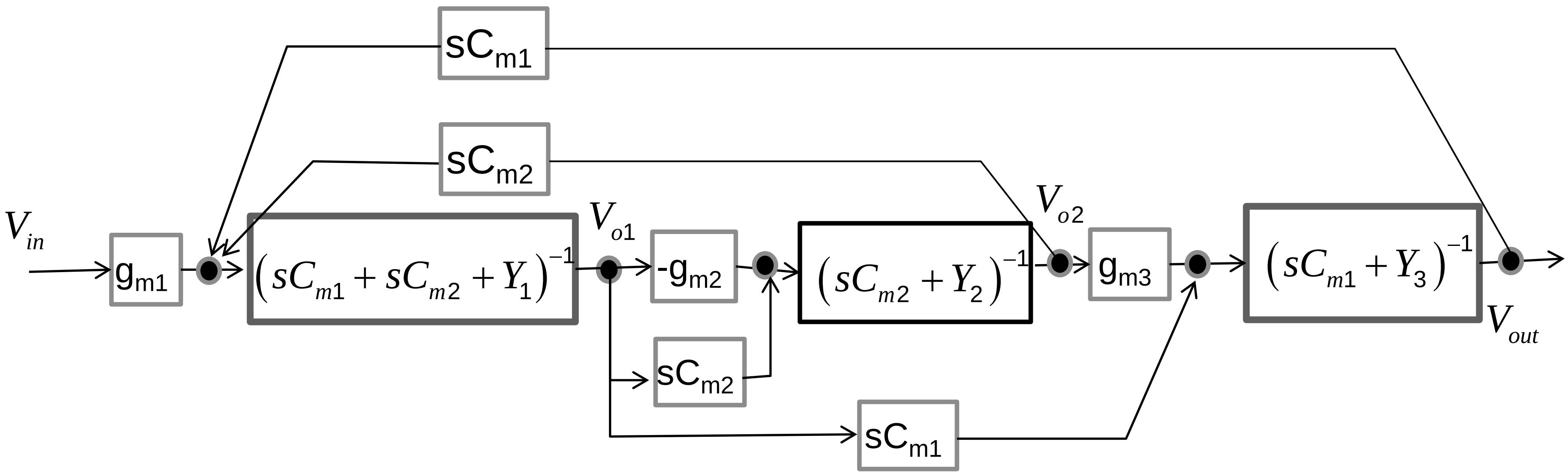
Transfer Function



Loop 3:

$$L_3 = \frac{sC_{m1} sC_{m1}}{(sC_{m1} + sC_{m2} + Y_1)(sC_{m1} + Y_3)}$$

Transfer Function



All three loops **touch** each other; hence no cross-product terms of L_k.
By **Mason's rule**:

$$H(s) = \frac{P(s)}{1 - L_1(s) - L_2(s) - L_3(s)}$$

References

- **A. Ochoa, Jr., "A systematic approach to the analysis of general and feedback circuits and systems using signal flow graphs and driving-point impedance," *IEEE Trans. on CAS-II: Analog and Digital Signal Processing*, vol. 45, no. 2, Feb. 1998, pp. 187-195.**

Appendix 1

- **Manson’s Rule**

Mason's Rule

P_k = the kth forward path

L_i = loop-gain of the ith loop

$L_i L_j$ = loop-gain product of two "non-touching" loops;

...

Δ_k = the kth cofactor (determined by loops)

$$H(s) = \frac{y_{out}}{u_{in}} = \frac{\sum_k P_k \Delta_k}{\Delta}$$

Δ_k = co(Δ, P_k) : Determinant of all loops nontouching the path P_k

(Δ_k is called ***cofactor*** of Δ w.r.t. P_k)

If no loops, then Δ = 1

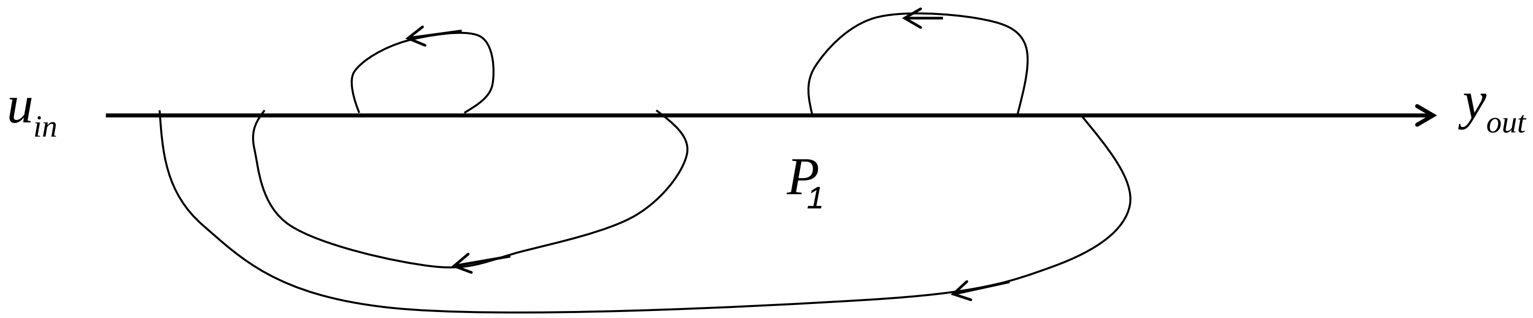
$$\Delta = 1 - \sum_i L_i + \sum_{i,j} L_i L_j - \sum_{i,j,k} L_i L_j L_k + \cdots$$

Δ = Determinant of all loops of the SFG

Cross-product of all nontouching (or node-disjoint) loop-gains

Src: http://en.wikipedia.org/wiki/Mason%27s_rule

Mason Rule – Special Case



Special Case: Only one forward path (P1) while all loops touching the forward path P1. Then, the Manson's rule is very simple:

$$\frac{y_{out}}{u_{in}} = \frac{P_1}{\Delta}$$

P_1 = gain of the forward path

$$\Delta_1 = 1$$

This case is highly common in application.

New Perspective of Mason's Rule

In the traditional Mason's rule: The procedure of finding numerator is "different" from finding the denominator.

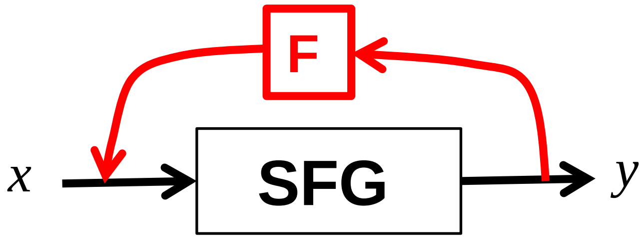
$$H(s) = \frac{\sum_k P_k \Delta_k}{\Delta}$$

Among all nontouching loops, those "touching path P_k must be removed".

Requiring extra cost to check!

Find all nontouching loops

Reformulation of Mason's Rule



Making F part of the loops.

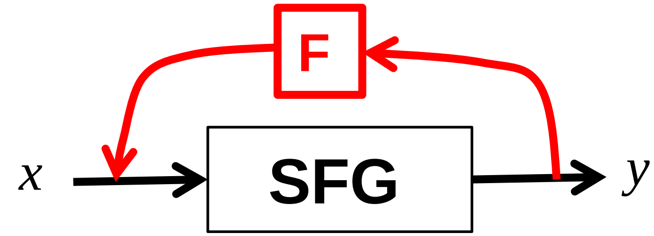
Main idea: Try to find just all nontouching loops.
This can greatly simplify computer implementation.

Form a "closed" network by adding I/O as a reversed network element "F".

How is it valid ?

We let $$H = \frac{1}{F} = \frac{\sum_k P_k \Delta_k}{\Delta}$$ where F is the reverse of H.

$$\Rightarrow \quad \Delta - \sum_k (FP_k)\Delta_k = 0$$



Set 1 — Set 2

Two set of loops:

Set 1: Those excluding F;
Set 2: Those including F.

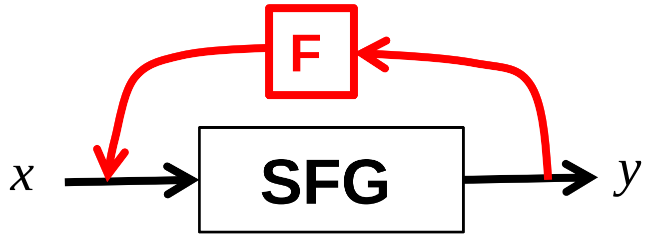
Note that **each (FP_k) is a loop**.
And the loop (FP_k) is nontouching any loop in Δ_k.

W. K. Chen, Applied Graph Theory, Amsterdam, The Netherlands: North-Holland, 1971.

New Algorithm for Mason's Rule

For computer implementation:

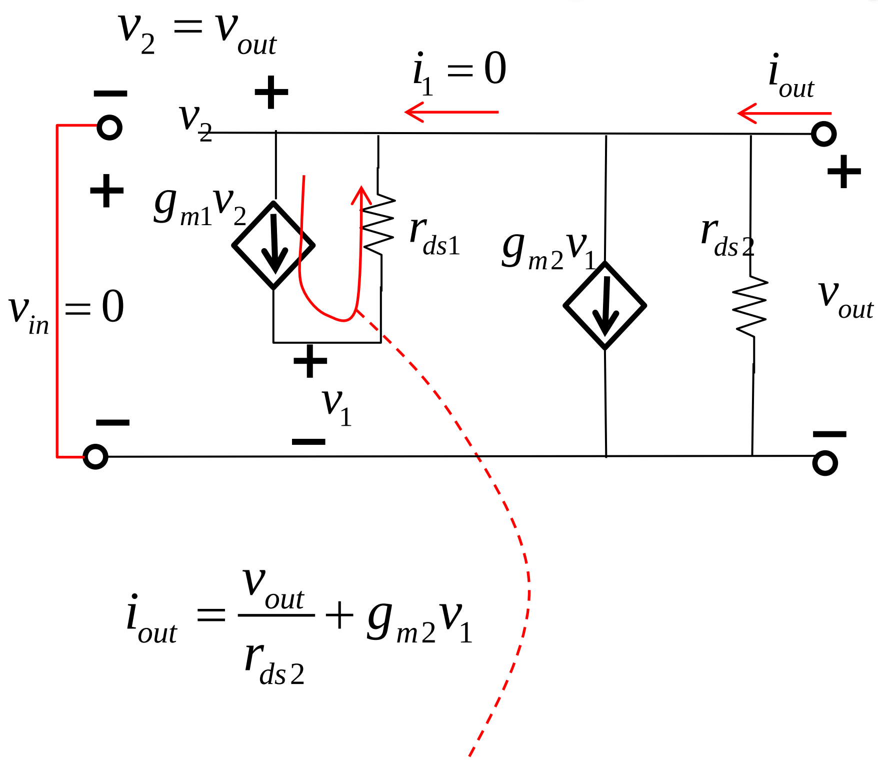
1) Enumerate all loops;
2) Form the kth order node-disjoint loops for k = 1, 2, 3, ... and generate all cross-product terms;
3) Sort the product terms with F (say, T1) and without F (say, T2);
4) Dividing the two partial sums to get the transfer function, i.e., 1/F = -T1/T2.

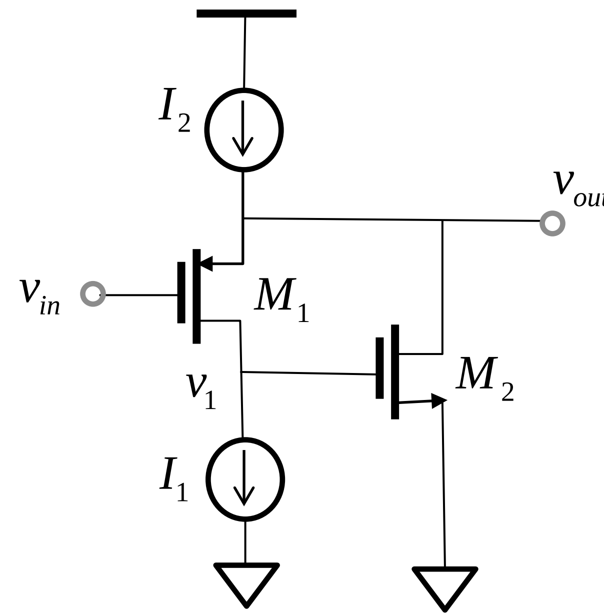


D. B. Johnson, "Finding all the elementary circuits of a directed graph," SIAM Journal of Computing, vol. 4, no. 1, March 1975, pp. 77-84.

Appendix - Extra Examples

Example A1 (Direct solve)





$$i_{out} = \frac{v_{out}}{r_{ds2}} + g_{m2}v_1$$

Find the output impedance

$$-g_{m1}v_{out} = \frac{v_{out} - v_1}{r_{ds1}} \Rightarrow v_1 = (1 + g_{m1}r_{ds1})v_{out}$$

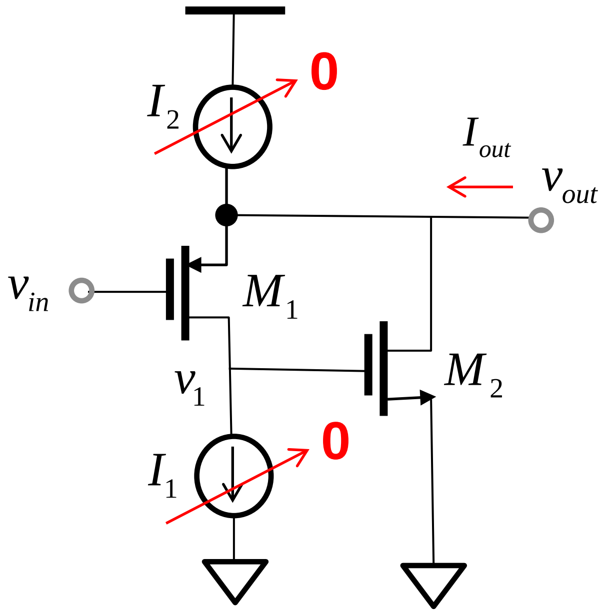
$$\Rightarrow i_{out} = \frac{v_{out}}{r_{ds2}} + g_{m2}v_1 = \left(\frac{1}{r_{ds2}} + g_{m2}(1 + g_{m1}r_{ds1}) \right) v_{out} \Rightarrow R_{out} = \frac{r_{ds2}}{1 + (1 + g_{m1}r_{ds1})g_{m2}r_{ds2}}$$

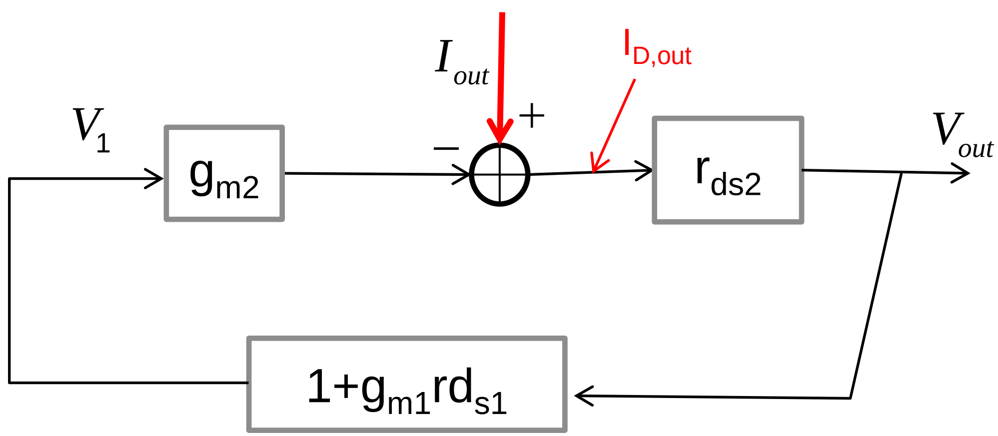
(Courtesy P. E. Allen's Lecture)

Example A1 (by DPI-SFG)

$$r_{ds1}(g_{m1}v_{out}) = v_1 - v_{out}$$

$$\Rightarrow \quad v_1 = (1 + g_{m1}r_{ds1})v_{out}$$





$$v_{out} = r_{ds2}\left(i_{out} - g_{m2}v_1\right)$$

(See the small-signal ckt in the prev. page)