IN3170/4170 2024 notes on solutions of the Inspera exam

Disclaimer: These notes are published due to popular demand by the students and are not to be taken as a quality controlled set of solutions. Other approaches might also be correct.

Abstract

These notes on the IN3170/4170 2024 online exam are published due to popular demand by the students and are not to be taken as either a quality controlled set of solutions nor as being nicely presented and formated. I just did my best to quickly dump the notes of the online system here together with the tasks.

Part I: Models

question 1: Strong or Weak Inversion, Conduction or Saturation?

a)

The simplest solution is to use I_S as the current magnitude that separates weak from strong inversion in saturation. Use equation (2.5)

$$I_S = 2 * 1.2 * 3 * 200e - 6 * (26e - 3)^2 = 973\text{nA}$$

the Compendium mentions two rules that are used in literature where either $I_S = 973$nA or $0.48 * I_S = 467$nA (the latter is equivalent to $V_{OV} = 0$) is used as the separator between weak and strong inversion. In both cases 200nA is below and 2μA is above. SO all but the top two currents are weak inversion currents.

b)

The drain current when clearly in saturation if $\lambda = 0$ and for given process parameters and given V_{GS} is a constant and may be bigger than that constant if $\lambda > 0$ (for increasing V_{DS}). The strong inversion approximation can be used here to derive that current. Only if the current is smaller than that saturation current, can the transistor be in conduction.

$$I_D = 0.5/1.2 * 3 * 200e - 6 * (0.7 - 0.4)^2 = 22.5\mu\text{A}$$

So the separation is above 10μA and all the listed currents can be in conduction.

Part II: Linear Circuit Analysis

question 2: Transfer Function

The two Bode plots are resulting from the following transfer functions:

$$H1 = 1000\frac{(1+\frac{j\omega}{10^4})}{(1+\frac{j\omega}{10^3})(1+\frac{j\omega}{10^7})}$$
$$H2 = 100\frac{(1-\frac{j\omega}{10^4})}{(1+\frac{j\omega}{10^3})(1+\frac{j\omega}{10^7})}$$

Part III: Digital Circuits

question 3: Gate Delay

The tasks states to assume that delay is governed purely by the input capacitances of the next gates a gate is connected to. Also an inverter uses a 2.5 times wider pFET than nFET. Thus relative input capacitances for balanced gates are (assuming the gate capacitance of the inverter nFET is 1 and thus the inverter's pFET 2.5):

inverter: 1 + 2.5 =3.5

2 input NAND: 2 + 2.5=4.5

2 input NOR: 1 + 5 =6

3 input NAND 3+2.5= 5.5

resulting worst case (meaning all gates switch) relative gate delays in the marked nodes are:

a= 4.5

b= 6

c= a + (5.5 + 4.5) = 14.5

d= 4.5 + 2*4.5= 13.5

e= c + 3.5=18

f= b + (4.5 + 5.5) + 3.5 = 19.5

It's not really necessary to solve at this level of detail: It's enough to realize that the 2 input nor has more input capacitance than the nand so $a < b$ and then that the three input nand has a bigger input capacitance than the 2 input nand, so $d < c$. Then the unmarked node below d has bigger delay than c (they have identical capacitance but the node below d suffers from the delayed b) and therefore $e < f$

So finally:

$a < b < d < c < e < f$

question 4: FSM PID-Controller

See figure 1. Note that since there are 5 possible sensor inputs there needs to be state transitions for all 5 of these (some may return to the same state or multiple inputs can also lead to the same state). The drawing omits the state transition label for brevity: it's redundant since the state stores the last sensor input, so the heading *leading* to a state is also the heading *stored* in that state. In that sense, each of the 5 transitions has to end up in one of the 5 columns of the state bobbles as organized in the figure given in the task.

question 5: Von Neumann Bottleneck

From the lecture slides on the von Neumann machine: "Since there is only one bus between the memory and the CPU for both instruction and data transfer in a von

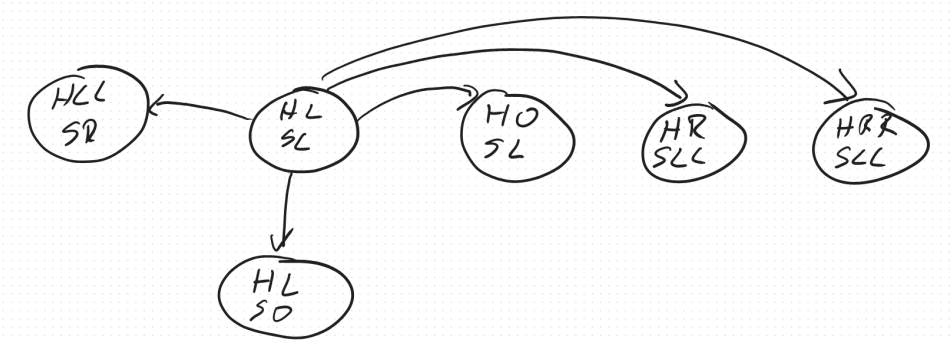


Figure 1: Solution question 4

Neumann architecture (actually two: one for the address and one for the data), this bus can be a main speed limiting factor (von Neumann bottleneck)."

Part IV: Analog Circuits

question 6: Find Lambda*L from CS

A possible Matlab script:

```
UT=26e-3
Vt=0.6
RL=210e6
A=20
Vg=0.45
W=1200e-9
L=600e-9
uCox=300e-6
k=uCox*W/L
Is=2*k*UT^2
Id=Is*exp((Vg-Vt)/UT)
gm=Id/UT
%gm = 9.7411e-08
RsL=A/gm

%RsL= rds*RL/(RL+rds)
%RsL*rds + RsL*RL=rds*RL
%RsL*RL = rds*(RL -RsL)
rds=RsL*RL/(RL-RsL)
%rds = 9.2028e+09

%from: rds=L/(lambdaL*Id)
lambdaL=L/(rds*Id)

% lambdaL=2.5742e-08
% lambdaL/L=0.0429
```

question 7: SF Transfer Function

G_M and R_O can directly be deduced from I_D and the given device parameters. Since the drain current and gate voltage (input point of operation) V_I is given, one can then deduce the output point of operation V_O and from I_D and V_O the appropriate R_L.

From there one can use the general transconductance amplifier model in chapter 5 for p_1, p_2, z_1, as indicated in the task text. See equation (5.26) in version beta 0.15 of the compendium. From that equation one needs to solve a quadratic equation to get the exact poles.

```
uCox=320e-6 % A/V^2
WL=1
n=1
lambda=0.05
%Vt=0.3 % that had been the intended value but due to a typo
%in the exam it became 1.2
Vt=1.2
Vdd=1.8
VI=1.2
UT=26e-3
k=WL*uCox
Is=2*n*k*UT^2
%Is = 4.3264e-07
Id=50e-9

%Id=Is*exp((VI-Vt-n*VO)/n/UT)
%log(Id)-log(Is)=(VI-Vt-n*VO)/n/UT
VO=((log(Is)-log(Id))*n*UT-Vt+VI)/n
%VO=0.0561 ; % a bit unfortunately low VO.
%Would have been much better with Vt=0.3
RL=VO/Id
%RL=1.1221e+06

gm=Id/UT
%gm =1.9231e-06
rds=1/lambda/Id
%rds = 400000000

Rsig=100e3
Cgs=10e-15
Cgd=1e-15
CL=100e-15
RO=rds/(1+gm*rds)
%RO = 5.1932e+05
CsL=CL
Cfb=Cgs
CI=Cgd
RsL=RO*RL/(RO+RL)
RsI=Rsig
GMA=-gm
A=RsI*RsL*((Cfb+CsL)*(Cfb+CI)-Cfb^2)
B=RsL*(Cfb+CsL)+RsI*(CI+Cfb*(1+GMA*RsL))

p1=(-B+sqrt(B^2-4*A))/2/A
```

```
%p1 = -2.6012e+07

p2=(-B-sqrt(B^2-4*A))/2/A
%p2 =-9.7557e+08

z1=GMA/Cfb
%z1 = -1.9231e+08
GM=GMA;
%end
```

question 8: Current Mirror I/O Voltages

Also see figure 2

```
function [VO, VI, VOexactNOTmin, VIexactNOTmin]=WillsonVoltages(Vgs,Vsat)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
VIexactNOTmin=1;
VOexactNOTmin=0;
VO=Vgs+Vsat;
VI=Vgs+Vgs;
end
```

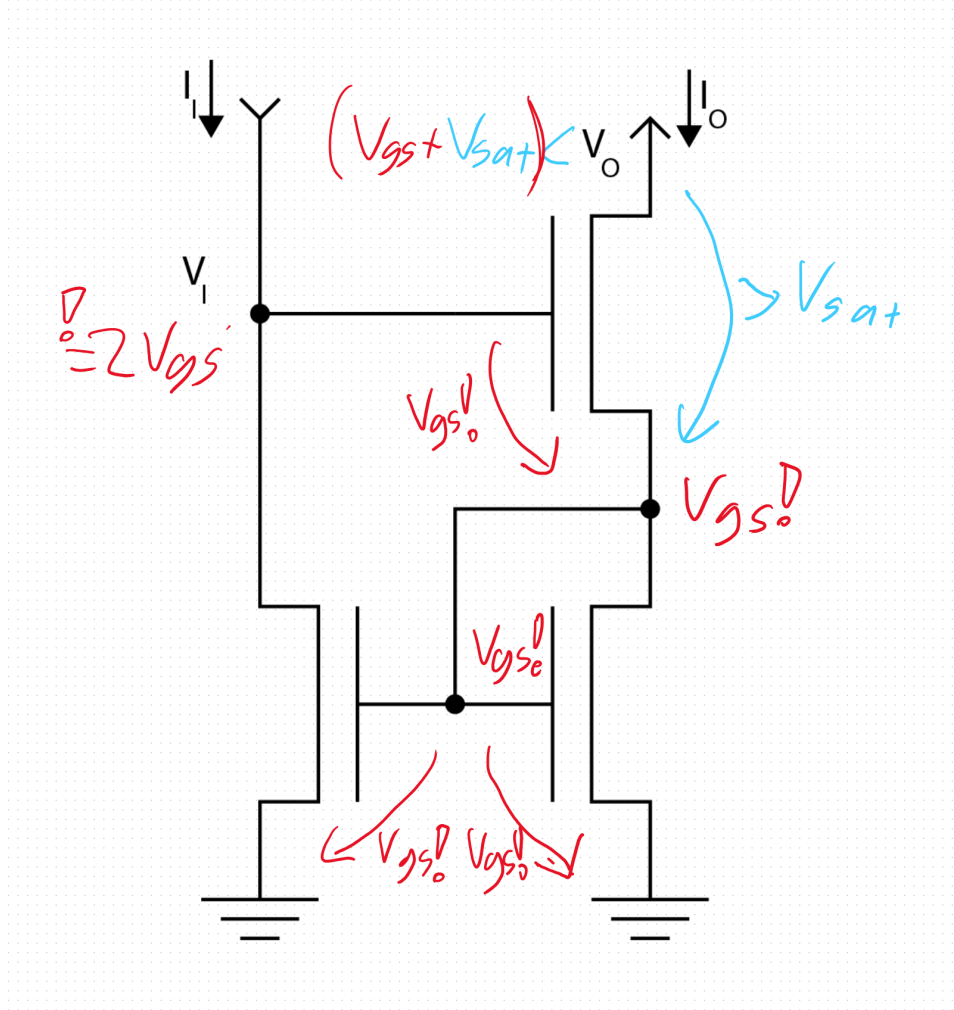


Figure 2: Solution question 8