# Homework 5 Solution

## *ECE 559: MOS VLSI Design (Fall 2009)*

## *ECE Department, Purdue University*

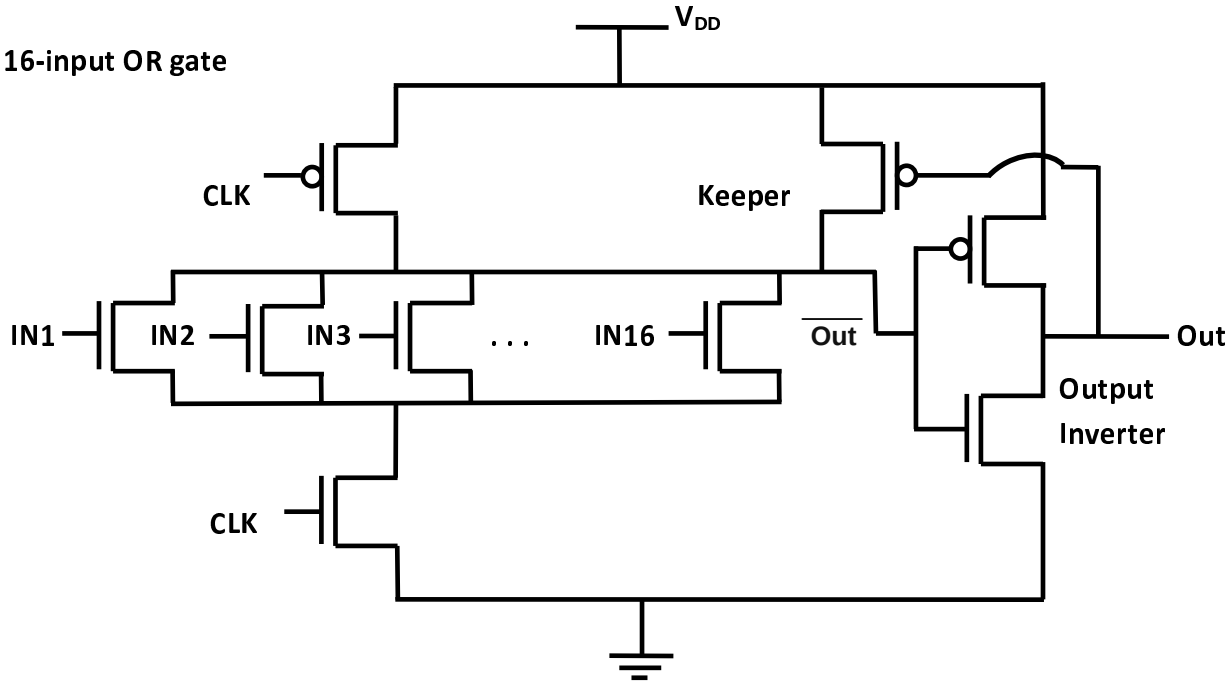
Assigned: 24-Oct-2009 Due: 03-Nov-2009

**Important: Please turn-in all of your codes along with the waveforms (when necessary) during submission of your solution. You may be asked to provide the soft copy of your codes by email if such need arises.**

**Problem 1:** The problem considers the *charge leakage* problem in dynamic logic based circuits. Consider a 16-input OR gate implemented in domino logic with a *PMOS keeper*. Draw and submit the schematic (on paper by hand is OK if you don't use Cadence). Use the following parameters.

L = 300 nm, $V_{DD}$ = 2.5 V, $W_{pre\text{-}charge}$ = 900 nm, $W_{evaluate}$ = 450 nm,
Widths of NMOSs in the PDN, $W_{NMOS\text{-}PDN}$ = 9000 nm.

For any other transistors, allowed *minimum* and *maximum* widths are 450 nm and 1800 nm, respectively. You should consider widths in *50 nm* intervals.



**Part a):** Determine (by hand calculation) the width of the keeper PMOS ensuring the *correct functionality* of the gate. You can select the characteristics of the output inverter by your own *with explanation*. State your assumptions, if any. Use the following information.

$K_p' = \mu_p C_{ox} = 300e\text{-}6\ A/V^2$ $K_n' = \mu_n C_{ox} = 600e\text{-}6\ A/V^2$

$V_{tp} = -0.5\ V$ $V_{tn} = 0.5\ V.$

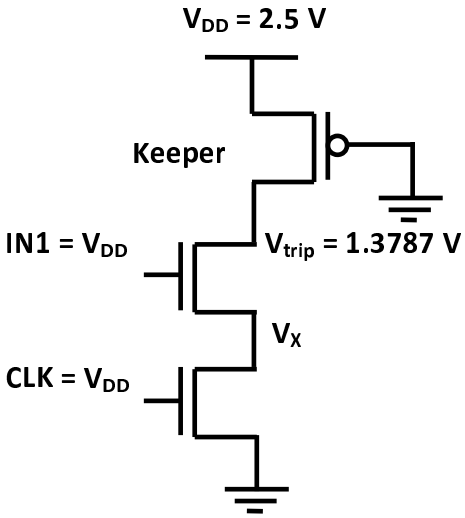
**Solution:** First we need to analyze the dependence of falling delay, rising delay, and charge leakage with different transistor widths at node $\overline{\textbf{Out}}$ . Initially, we will not consider any trade-off due to increased capacitance because of using high PDN NMOS widths at the $\overline{\textbf{Out}}$ node. **It may be significant as we are using quite higher widths (9000 nm) for the PDN MOSs. We will check that by SPICE simulation for part b).**

| | **Worst case condition** | $W_{keeper}$ ↓ | $W_{output,\ p}$ ↑ | $W_{output,\ n}$ ↓ |
|---|---|---|---|---|
| **Falling Delay** | Only one input is switching 0→1, other inputs are at logic 0 | ↓ | ↓ | ↓ |
| **Precharge Time** | All the inputs are at logic 1 | ↑ | ↑ | ↑ |
| **Robustness (Charge Leakage)** | All the inputs are at logic 0 | ↓ | ↑ | ↑ |

Since we are not given any clock period, we will not consider the quantitative values for falling delay or precharge time rather we will consider the robustness and check the condition that the ratioed operation (between keeper PMOS and the PDN and evaluate NMOSs) is performing correctly, i.e., in the worst case of the falling delay (when only one NMOS in the PDN is switching 0→1 and other inputs are at logic 0), the voltage at node $\overline{\textbf{Out}}$ does not goes below the trip-point of the output inverter. This condition will give us the *maximum* value of the keeper that we can set.

As the settings for the output inverter MOSs are concerned, we see that as we increase the width of the PMOS and decrease the width of the NMOS (so that the trip-point of the inverter goes more toward $V_{DD}$), robustness due to charge leakage increases. According to the allowable values for the minimum and maximum widths of the MOSs that are given, we can choose $\mathbf{W_{output,\ p} = 1800\ nm}$ and $\mathbf{W_{output,\ n} = 450\ nm.}$

With these values and other given values, equating the currents when both the transistors are in saturation, we get $\mathbf{V_{trip,\ inv} = 1.3787\ V.}$ Noise margins do not seem to be affected much.




With the values of the threshold voltages provided, all the transistors are in **linear** region. Equating the current through the PDN NMOS and the evaluate NMOS, we get

$$V_X = 1.2529 \text{ V.}$$

Similarly, equating the current through the keeper PMOS and the evaluate NMOS, we get

$$W_{keeper} = 959.67 \text{ nm.} \qquad \text{So} \qquad 950 \text{ nm} \le W_{keeper} \le 950 \text{ nm}$$

With the given values of the threshold voltages of the MOSs, we can be quite sure that the already determined range for the width of keeper would not quite affect the robustness of the circuit due to charge leakage. (Note that robustness decreases with decreasing keeper size.)

**MATLAB Code**

```
Vdd=2.5; Vtp=-0.5; Vtn=0.5;

kpd = 300;      % in uA/V2
knd = 600;      % in uA/V2

Wpo = 1800; Wno = 450;

Wn=9000; Wne=450;

% determine the threshold voltage of the output inverter
Vtinv = fsolve(@(x) kpd*Wpo*(x-Vdd-Vtp)^2 - knd*Wno*(x-Vtn)^2, Vdd/2)

% determine the intermediate node voltage bwteen the PDN and evaluate NMOS
Vx = fsolve(@(x) Wn*((Vdd-x-Vtn)*(Vtinv-x) - (Vtinv-x)^2/2) - Wne*((Vdd-
Vtn)*x - x^2/2), Vtinv)

% determine the keeper size
Wkeeper = fsolve(@(x) kpd*x*((-Vdd-Vtp)*(Vtinv-Vdd) - (Vtinv-Vdd)^2/2) -
knd*Wne*((Vdd-Vtn)*Vx - Vx^2/2), Vtinv)
```

**Part b)** Perform SPICE simulation to determine the width of the keeper PMOS and the widths of the output inverter MOSs that achieve the *fastest operation* of the gate, however, you should ensure the *correct functionality* of the gate. Clearly state your procedure and assumptions, if any. Use

Rise/Fall Time = 0.01 ns, Delay Time = 0.01 ns, Clock Period = 200 ns (with 50% duty cycle), the high/low state of the other input signals you can choose at your convenience.

**Solution:** We have to check first the worst case condition for the falling delay at the $\overline{\textbf{Out}}$ node as we are using transistors with quite higher widths (9000 nm) for the PDN MOSs and the condition might change because of the increased capacitance at the $\overline{\textbf{Out}}$ node. It is verified by SPICE simulation that the **worst case falling delay at $\overline{\textbf{Out}}$ node is happening when all the inputs are switching from 0→ 1.**

From the table as in the solution of part a), we have already seen that as we increase the width of the PMOS and decrease the width of the NMOS of the output inverter (so that the trip-point of the inverter goes more toward $V_{DD}$), falling delay at node $\overline{\textbf{Out}}$ decreases. Since we are asked for the *fastest operation* of the gate, according to the allowable values for the minimum and maximum widths of the MOSs that are given, we can choose

$$\mathbf{W_{output,\,p} = 1800\ nm} \text{ and } \mathbf{W_{output,\,n} = 450\ nm.}$$

We have to go on increasing the width of the keeper PMOS to check when the circuit starts operating fine, i.e., the charge leakage during the evaluation period (when all the inputs are at logic 0) keeps the voltage at node $\overline{\textbf{Out}}$ above the trip-point of the inverter making the keeper PMOS on.

Since we are using 250 nm technology library, the threshold voltages of the transistors are large enough that the charge leakage is not quite significant. According to this expectation, we have got the *fastest operation* of the circuit with *minimum keeper size*, i.e.,

$$\mathbf{W_{keeper} = 450\ nm.}$$

The voltage at node $\overline{\textbf{Out}}$ goes down only to **2.4964 V** at the end. The worst case delays at the **Out** node are as follows:

**precharge time = 5.3386E-09 sec,**

**rising delay = 1.4954E-09 sec (when only one input is switching from 0→ 1),**

**rising delay = 2.3088E-09 sec (when all the inputs are switching from 0→ 1).**

## SPICE Code

```
* HW 5, Problem 1 Solution, Part b)
* ECE 559, Fall 2009, Purdue University

.GLOBAL VDD!

.lib "$CDK_DIR/models/hspice/public/publicModel/tsmc25N" NMOS
.lib "$CDK_DIR/models/hspice/public/publicModel/tsmc25P" PMOS

.PARAM VDD  = 2.5v
.PARAM VREF = 1.25v
.PARAM L    = 300n

.PARAM WPP  = 900n
.PARAM WN   = 9000n
.PARAM WNE  = 450n

.PARAM WPO  = 1800n
.PARAM WNO  = 450n

.PARAM WPK  = 450n

.PARAM VDDBAT  = 0.0v

.PARAM VDDINH  = 'VDD+VDDBAT'
.PARAM VDDINL  = 'VDDBAT'

.PARAM DT   = .01n
.PARAM RT   = .01n
.PARAM FT   = .01n
.PARAM ST   = 100n

.TEMP    25.0000

.OPTION  POST
```

```
* precharge PMOS
MPP OUTB CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L'
PD='2*WPP+5*L' PS='2*WPP+5*L' M=1

* NMOSs in the pull-down network
MN1 OUTB IN1 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN2 OUTB IN2 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN3 OUTB IN3 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN4 OUTB IN4 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN5 OUTB IN5 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN6 OUTB IN6 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN7 OUTB IN7 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN8 OUTB IN8 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN9 OUTB IN9 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN10 OUTB IN10 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN11 OUTB IN11 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN12 OUTB IN12 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN13 OUTB IN13 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN14 OUTB IN14 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN15 OUTB IN15 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN16 OUTB IN16 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L'
PD='2*WN+5*L' PS='2*WN+5*L' M=1
```

```
* evaluate NMOS
MNE OUTX CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1

* output inverter
MPO OUT OUTB VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1
MNO OUT OUTB 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1

* keeper
MPK OUTB OUT VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1

VVDD! VDD! 0 DC=VDD

.TRAN 'ST/10' '6*ST' START=0

.PRINT TRAN V(IN1) V(IN2) V(CLK) V(OUTX) V(OUTB) V(OUT)

* CLK signal
VCLK CLK 0 PAT (VDD 0 0n RT FT ST b010101 r=1)

VIN1 IN1 0 PAT (VDDINH VDDINL DT RT FT ST b010110 r=1)
VIN2 IN2 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN3 IN3 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN4 IN4 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN5 IN5 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN6 IN6 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN7 IN7 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN8 IN8 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN9 IN9 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN10 IN10 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN11 IN11 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN12 IN12 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN13 IN13 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN14 IN14 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN15 IN15 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
```

```
VIN16  IN16  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)

* measure the worst case precharge time
.MEASURE TRAN PROPAGATION_PRECHARGE
+ TRIG v(CLK)  VAL=VREF TD=ST FALL=2
+ TARG v(OUT)  VAL=VREF TD=ST FALL=2

* measure the rising delay when only one input is switching
.MEASURE TRAN PROPAGATION_RISING_ONLY_ONE
+ TRIG v(IN1)  VAL=VREF TD=ST RISE=1
+ TARG v(OUT)  VAL=VREF TD=ST RISE=1

* measure the rising delay when all the inputs are switching
.MEASURE TRAN PROPAGATION_RISING_ALL_ONE
+ TRIG v(IN1)  VAL=VREF TD=ST RISE=2
+ TARG v(OUT)  VAL=VREF TD=ST RISE=2

.END
```

**Note:** You can verify the correct functionality of the circuit by looking at the output of the .PRINT statement. Look at when the signals OUTB and OUT get flipped due to charge leakage for some keeper width.

**Part c)** Decrease the threshold voltages of the NMOSs in the pull-down network. You can change the effective threshold voltages of the NMOSs by adding a battery at the inputs in addition to the input voltages applied so that gate-overdrive $V_{GS}$ - $V_T$ changes. You don't need to change the threshold voltage in the library. Use battery voltage, $V_B$ = 0.8 V.

Repeat Part b). You can consider breaking down the 16-inputs in multiple cascaded stages as necessary. Explain why or why not. Draw and submit your final schematic (on paper by hand is OK if you don't use Cadence). Explain the differences in results with the Part b).

**Solution:** In this case as well, **w**e have to check first the worst case condition for the falling delay at the $\overline{\textbf{Out}}$ node as we are using transistors with quite higher widths (9000 nm) for the PDN MOSs and the condition might change because of the increased capacitance at the $\overline{\textbf{Out}}$ node. It is verified by SPICE simulation that the **worst case falling delay at $\overline{\textbf{Out}}$ node is happening when all the inputs are switching from 0→ 1.**

In this case we are decreasing the threshold voltages of the PDN MOSs. So charge leakage may be significant enough to bring the voltage at the node $\overline{\textbf{Out}}$ down below the trip-point of the output inverter making the keeper PMOS off.

Since we are asked for the *fastest operation* of the gate, according to the allowable values for the minimum and maximum widths of the MOSs that are given, we can choose

$\mathbf{W_{output,\,p}}$ **= 1800 nm** and $\mathbf{W_{output,\,n}}$ **= 450 nm.**

We have to go on increasing the width of the keeper PMOS to check when the circuit starts operating fine, i.e., the charge leakage during the evaluation period (when all the inputs are at logic 0) keeps the voltage at node $\overline{\textbf{Out}}$ above the trip-point of the inverter making the keeper PMOS on. We found

$\mathbf{W_{keeper}}$ **= 750 nm**

for proper operation of the circuit. The voltage at node $\overline{\textbf{Out}}$ goes down to **1.6010 V** at the end. The worst case delays at the **Out** node are as follows:

**precharge time = 6.2974E-09 sec,**

**rising delay = 1.9542E-09 sec (when only one input is switching from 0→1),**

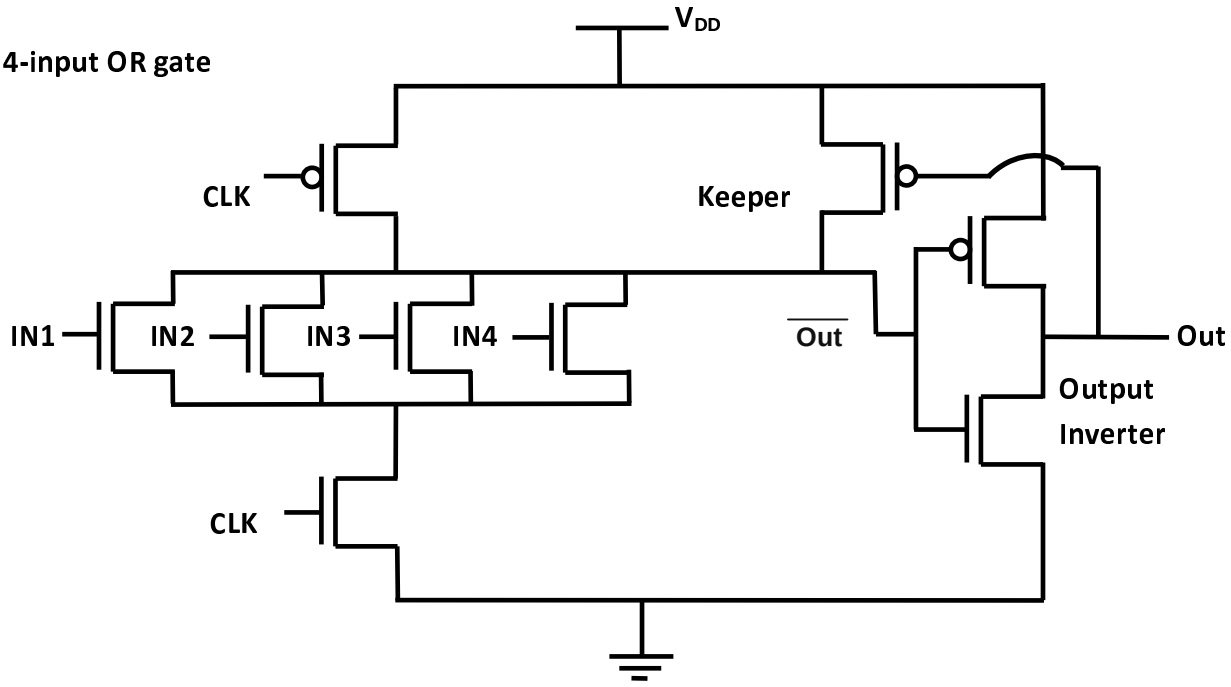
**rising delay = 4.5258E-09 sec (when all the inputs are switching from 0→1).**

Separating the 16 NMOSs into multiple stages will allow us to reduce the keeper size below 750 nm as we have to combat with less leakage due to less number of transistors. **Reducing the keeper size will incur less falling delay at the node $\overline{\text{Out}}$ .** Some simulation results for different number of PDN MOSs are given below.

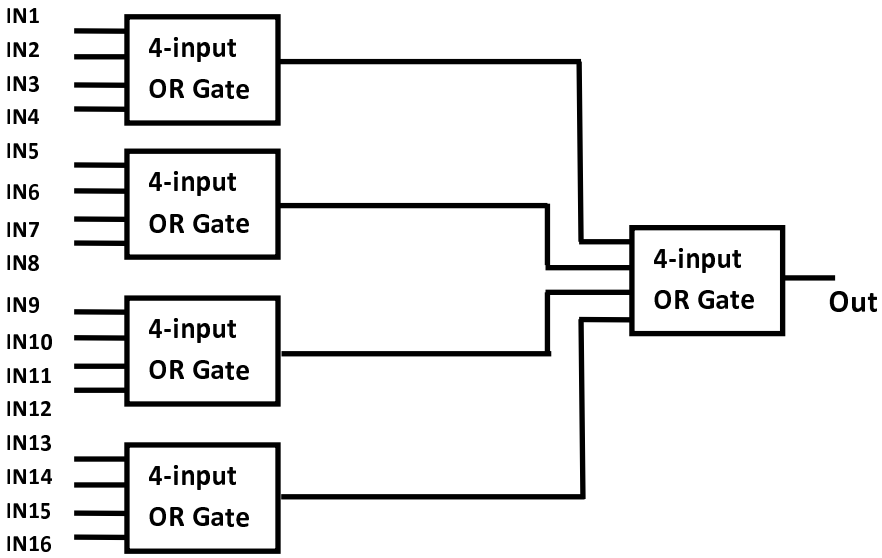
| # PDN MOSs | $W_{keeper}$ (nm) | Precharge Time (sec) | Rising Delay (sec) IN1: 0→1, IN2-16: 0 | Rising Delay (sec) IN1-16: 0→1 |
|---|---|---|---|---|
| 16 | 750 | 6.4652E-09 | 1.9542E-09 | 4.5258E-09 |
| 8 | 700 | 3.0842E-09 | 1.0219E-09 | 2.1621E-09 |
| 4 | 600 | 1.8420E-09 | 7.1010E-10 | 1.1304E-09 |
| 2 | 550 | 9.6077E-10 | 5.3195E-10 | 6.4161E-10 |

From the table, we can decide of **separating the 16-inputs of the OR gate into two stages – first stage contains four four-input OR gates operating in parallel and the second stage contains one four-input OR gate with each input receiving an output from each OR gate from the first stage.**

We can check that any other configurations (e.g., fist stage containing two eight-input OR gates and the second stage containing one two-input OR gate, similar configuration with all 2-input OR gates, any serial way of connecting gates etc.) would incur more delay than that of the aforesaid case.



**16-input OR Gate**




With the cascaded configuration, we got

$$W_{keeper} = 600 \text{ nm}$$

for proper operation of the circuit. The voltage at node $\overline{\text{Out}}$ for the gate at the first stage goes down to **1.5728 V** at the end. The worst case delays at the **Out** node of the OR gate at second stage are as follows:

**precharge time = 2.0566E-09 sec,**

**rising delay = 9.5256E-10 sec (when only one input is switching from 0→1),**

**rising delay = 1.4037E-09 sec (when all the inputs are switching from 0→1).**

Accordingly, we have got faster operation than that of the case when all the 16 PDN MOSs were together under one single gate.

**rising delay = 1.9542E-09 sec (when only one input is switching from 0→1),**

**rising delay = 4.5258E-09 sec (when all the inputs are switching from 0→1).**

## SPICE Code

```
* HW 5, Problem 1 Solution, Part c), Cascaded
* ECE 559, Fall 2009, Purdue University

.GLOBAL VDD!

.lib "$CDK_DIR/models/hspice/public/publicModel/tsmc25N" NMOS
.lib "$CDK_DIR/models/hspice/public/publicModel/tsmc25P" PMOS

.PARAM VDD  = 2.5v
.PARAM VREF = 1.25v
.PARAM L    = 300n

.PARAM WPP  = 900n
.PARAM WN   = 9000n
.PARAM WNE  = 450n

.PARAM WPO  = 1800n
.PARAM WNO  = 450n

.PARAM WPK  = 600n

.PARAM VDDBAT  = 0.8v

.PARAM VDDINH  = 'VDD+VDDBAT'
.PARAM VDDINL  = 'VDDBAT'

.PARAM DT   = .01n
.PARAM RT   = .01n
.PARAM FT   = .01n
```

```
.PARAM ST   = 100n

.TEMP    25.0000
.OPTION  POST

* ========== First Stage, 1st set ============================
* precharge PMOS
MPP OUTB CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L' PD='2*WPP+5*L' PS='2*WPP+5*L' M=1

* NMOSs in the pull-down network
MN1 OUTB IN1 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN2 OUTB IN2 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN3 OUTB IN3 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1
MN4 OUTB IN4 OUTX 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

* evaluate NMOS
MNE OUTX CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1

* output inverter
MPO OUT OUTB VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1
MNO OUT OUTB 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1

* keeper
MPK OUTB OUT VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1
* ============================================================
```

```
* ========== First Stage, 2nd set ============================

* precharge PMOS

MPP2 OUTB2 CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L' PD='2*WPP+5*L' PS='2*WPP+5*L' M=1


* NMOSs in the pull-down network

MN5 OUTB2 IN5 OUTX2 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN6 OUTB2 IN6 OUTX2 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN7 OUTB2 IN7 OUTX2 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN8 OUTB2 IN8 OUTX2 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1


* evaluate NMOS

MNE2 OUTX2 CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1


* output inverter

MPO2 OUT2 OUTB2 VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1

MNO2 OUT2 OUTB2 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1


* keeper

MPK2 OUTB2 OUT2 VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1

* ============================================================


* ========== First Stage, 3rd set ============================

* precharge PMOS

MPP3 OUTB3 CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L' PD='2*WPP+5*L' PS='2*WPP+5*L' M=1
```

```
* NMOSs in the pull-down network

MN9 OUTB3 IN9 OUTX3 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN10 OUTB3 IN10 OUTX3 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN11 OUTB3 IN11 OUTX3 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN12 OUTB3 IN12 OUTX3 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1


* evaluate NMOS

MNE3 OUTX3 CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1


* output inverter

MPO3 OUT3 OUTB3 VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1

MNO3 OUT3 OUTB3 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1


* keeper

MPK3 OUTB3 OUT3 VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1

* ============================================================


* ========== First Stage, 4th set ============================

* precharge PMOS

MPP4 OUTB4 CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L' PD='2*WPP+5*L' PS='2*WPP+5*L' M=1


* NMOSs in the pull-down network

MN13 OUTB4 IN13 OUTX4 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1
```

```
MN14 OUTB4 IN14 OUTX4 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN15 OUTB4 IN15 OUTX4 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN16 OUTB4 IN16 OUTX4 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1


* evaluate NMOS

MNE4 OUTX4 CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1


* output inverter

MPO4 OUT4 OUTB4 VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1

MNO4 OUT4 OUTB4 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1


* keeper

MPK4 OUTB4 OUT4 VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1

* ============================================================


* ========== 2nd/Final Stage ==================================

* precharge PMOS

MPP5 OUTB5 CLK VDD! VDD! TSMC25P L='L' W='WPP' AD='WPP*2.5*L' AS='WPP*2.5*L' PD='2*WPP+5*L' PS='2*WPP+5*L' M=1


* NMOSs in the pull-down network

MN1C OUTB5 OUT OUTX5 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN2C OUTB5 OUT2 OUTX5 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

MN3C OUTB5 OUT3 OUTX5 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1
```

```
MN4C OUTB5 OUT4 OUTX5 0 TSMC25N L='L' W='WN' AD='WN*2.5*L' AS='WN*2.5*L' PD='2*WN+5*L' PS='2*WN+5*L' M=1

* evaluate NMOS
MNE5 OUTX5 CLK 0 0 TSMC25N L='L' W='WNE' AD='WNE*2.5*L' AS='WNE*2.5*L' PD='2*WNE+5*L' PS='2*WNE+5*L' M=1

* output inverter
MPO5 OUT5 OUTB5 VDD! VDD! TSMC25P L='L' W='WPO' AD='WPO*2.5*L' AS='WPO*2.5*L' PD='2*WPO+5*L' PS='2*WPO+5*L' M=1
MNO5 OUT5 OUTB5 0 0 TSMC25N L='L' W='WNO' AD='WNO*2.5*L' AS='WNO*2.5*L' PD='2*WNO+5*L' PS='2*WNO+5*L' M=1

* keeper
MPK5 OUTB5 OUT5 VDD! VDD! TSMC25P L='L' W='WPK' AD='WPK*2.5*L' AS='WPK*2.5*L' PD='2*WPK+5*L' PS='2*WPK+5*L' M=1
* ============================================================

VVDD! VDD! 0 DC=VDD

.TRAN 'ST/10' '6*ST' START=0

.PRINT TRAN V(IN1) V(IN2) V(CLK) V(OUTX) V(OUTB) V(OUT) V(OUTB5) V(OUT5)

* CLK signal
VCLK CLK 0 PAT (VDD 0 0n RT FT ST b010101 r=1)

VIN1 IN1 0 PAT (VDDINH VDDINL DT RT FT ST b010110 r=1)
VIN2 IN2 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN3 IN3 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN4 IN4 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN5 IN5 0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
```

```
VIN6    IN6  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN7    IN7  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN8    IN8  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN9    IN9  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN10  IN10  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN11  IN11  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN12  IN12  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN13  IN13  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN14  IN14  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN15  IN15  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)
VIN16  IN16  0 PAT (VDDINH VDDINL DT RT FT ST b000110 r=1)

* measure the worst case precharge time
.MEASURE TRAN PROPAGATION_PRECHARGE
+ TRIG v(CLK)  VAL=VREF TD=ST FALL=2
+ TARG v(OUT5) VAL=VREF TD=ST FALL=2

* measure the rising delay when only one input is switching
.MEASURE TRAN PROPAGATION_RISING_ONLY_ONE
+ TRIG v(IN1)  VAL=VREF TD=ST RISE=1
+ TARG v(OUT)  VAL=VREF TD=ST RISE=1

* measure the rising delay when all the inputs are switching
.MEASURE TRAN PROPAGATION_RISING_ALL_ONE
+ TRIG v(IN1)  VAL=VREF TD=ST RISE=2
+ TARG v(OUT)  VAL=VREF TD=ST RISE=2

.END
```

**Note:** You can verify the correct functionality of the circuit by looking at the output of the .PRINT statement. Look at when the signals OUTB and OUT get flipped due to charge leakage for some keeper width.