EE341 Fall 2019 HW 7

Lewis Collum

Updated: December 4, 2019

Inverter Subcircuit

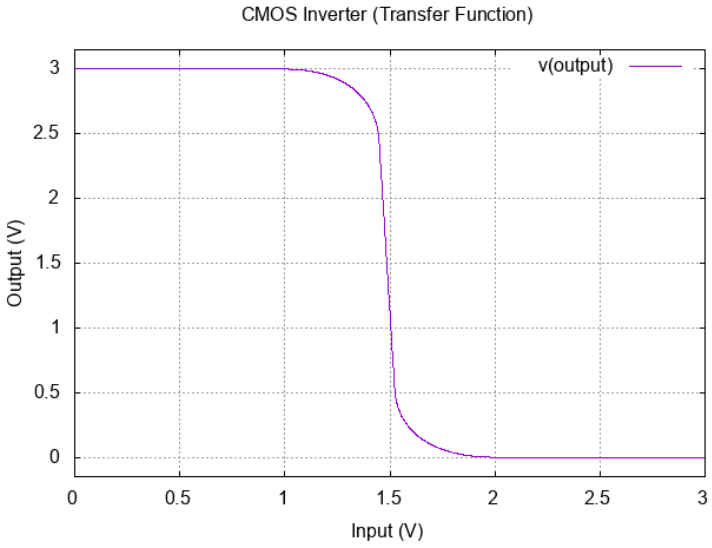
```
.subckt inverter vDD input output
.param sourceLength=1u sourceWidth=16u sinkLength=1u sinkWidth=8u

mSource output input vDD sourceBody source
+ L = sourceLength
+ W = sourceWidth

mSink output input 0 sinkBody sink
+ L = sinkLength
+ W = sinkWidth
.ends



.model source pmos
+ kp = 0.08m
+ vto = -1
+ lambda = 0.2
+ cbd = 100fF
+ cbs = 100fF
+ tox = 50nm

.model sink nmos
+ kp = 0.18m
+ vto = 1
+ lambda = 0.2
+ cbd = 100fF
+ cbs = 100fF
+ tox = 50nm
```

---

(1) Transfer Function of a Single Inverter

```
.title CMOS Inverter (Transfer Function)
.include inverter.cir

vDD vDD 0 DC 3
vIn input 0
xInverter vDD input output inverter

.control
dc vIn 0 3 0.001
set gnuplot_terminal=png/quit
gnuplot $file v(output) vs v(input) ylabel "Output (V)" xlabel "Input (V)"
```

```
.endc
.end
```



Region Characteristics

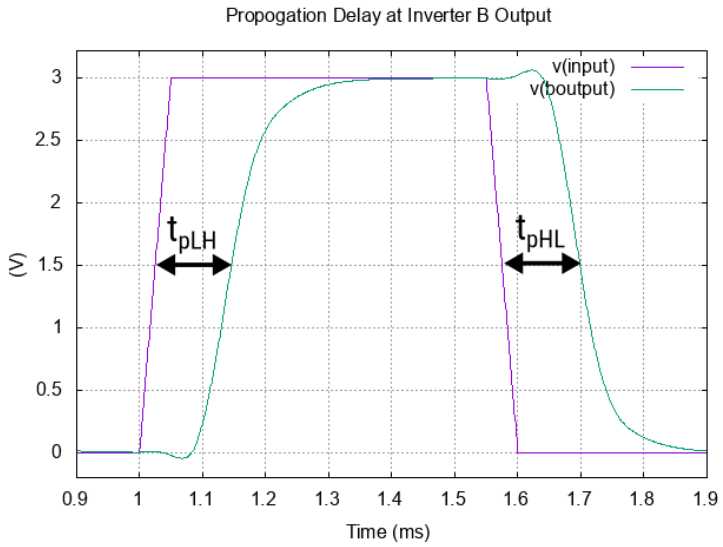
- $v_{in} < 1$:
  - NMOS is **off**.
  - PMOS is **on**.
  - $v_{out} = v_{DD}$
  - Sources current
- $2 < v_{in}$:
  - NMOS is **on**
  - PMOS is **off**
  - $v_{out} = 0$
  - Sinks current
- $1 < v_{in} < 2$:
  - NMOS $v_{gs} < v_{nt}$
  - PMOS $v_{sg} < v_{pt}$

---

(2) Three Cascade Inverters with a Load Capacitor

```
.title Propogation Delay at Inverter B Output
.include inverter.cir

vDD vDD 0 DC 3
vIn input 0 pulse 0 3 0 0.05ns 0.05ns 0.5ns 1ns

xA vDD input aOutput inverter
xB vDD aOutput bOutput inverter
```

```
xC vDD bOutput output inverter

cLoad output 0 20f

.control
tran 0.001ns 1.9ns 0.9ns
set gnuplot_terminal=png/quit
let millis = time*1e9
setscale millis
gnuplot $file v(input) v(bOutput) xlabel "Time (ns)" ylabel "(V)"
.endc
```



$t_{pHL} = t_{pLH} = t_{pd} \approx 0.12\text{ns}$

---

(3) Inverter B Fan-Outs

For this question I used `meas` to provide accurate timestamps for the input and output crossing 1.5V. I then use these timestamps to calculate the propogation delay.

```
.title Inverter B Fanout
.include inverter.cir
vDD vDD 0 DC 3
vIn input 0 pulse 0 3 0 0.05ns 0.05ns 0.5ns 1ns

xA vDD input aOutput inverter
xB vDD aOutput bOutput inverter
xFanOut vDD bOutput inverterWithOutputCapacitor m=15

.control
tran 0.001ns 1.9ns 0.9ns

meas tran output_tplh WHEN v(bOutput)=1.5 CROSS=1
meas tran input_tplh WHEN v(input)=1.5 CROSS=1
meas tran output_tphl WHEN v(bOutput)=1.5 CROSS=2
```

```
meas tran input_tphl WHEN v(input)=1.5 CROSS=2
echo "$&output_tplh, $&input_tplh, $&output_tphl, $&input_tphl" > fanout_15.csv

set gnuplot_terminal=png/quit
let nanos = time*1e9
setscale nanos
gnuplot $file v(input) v(bOutput) xlabel "Time (ns)" ylabel "(V)"
.endc

.subckt inverterWithOutputCapacitor vDD input
xInverter vDD input capacitorIn inverter
cOutput capacitorIn 0
.ends
```

Table

N	output t_{plh}	input t_{plh}	output t_{phl}	input t_{phl}	t_{plh}	t_{phl}	$t_p[\mathrm{ns}]$
1	1.146e-09	1.025e-09	1.698e-09	1.575e-09	1.207e-10	1.234e-10	1.221e-10
5	1.173e-09	1.025e-09	1.722e-09	1.575e-09	1.479e-10	1.474e-10	1.476e-10
10	1.2e-09	1.025e-09	1.748e-09	1.575e-09	1.752e-10	1.726e-10	1.739e-10
15	1.219e-09	1.025e-09	1.768e-09	1.575e-09	1.94e-10	1.926e-10	1.933e-10

The propogation delay increases as `N` increases. This is because fanout increases the capacitative load on inverter B; capacitative loads take time to charge and that time is reflected in the propogation delay of inverter B. This effect can be seen between the `N=1` case and the `N=15` case below.

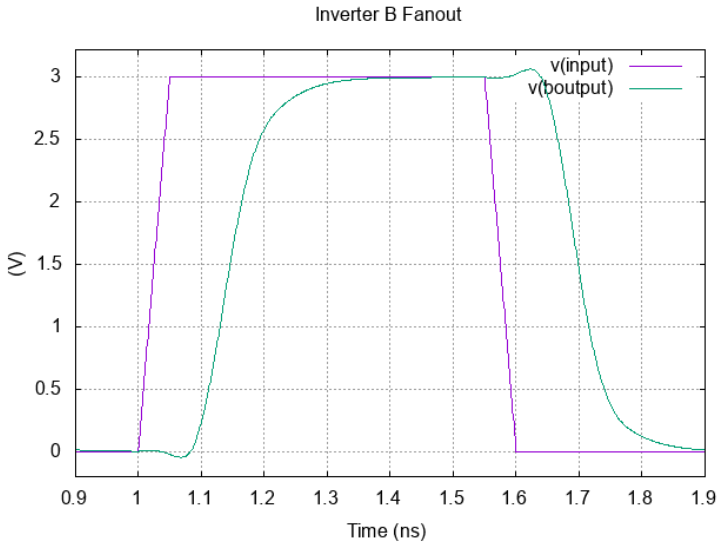


Figure 1: Case N=1.

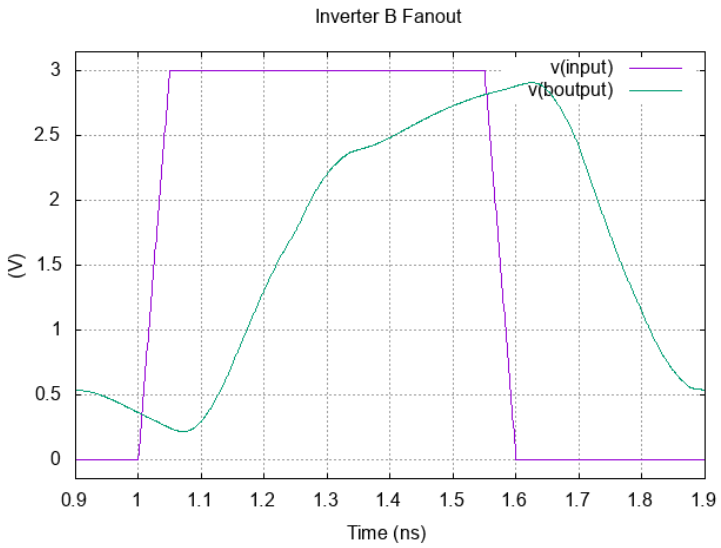


Figure 2: Case N=15.

---

(4) Fix N=15

```
.title Fix N=15
.include inverter.cir
vDD vDD 0 DC 3
vIn input 0 pulse 0 3 0 0.05ns 0.05ns 0.5ns 1ns

xA vDD input aOutput inverter
xB vDD aOutput bOutput inverter sourceWidth=64u sinkWidth=48u
xFanOut vDD bOutput inverterWithOutputCapacitor m=15

.control
tran 0.001ns 1.9ns 0.9ns

meas tran output_tplh WHEN v(bOutput)=1.5 CROSS=1
meas tran input_tplh WHEN v(input)=1.5 CROSS=1
meas tran output_tphl WHEN v(bOutput)=1.5 CROSS=2
meas tran input_tphl WHEN v(input)=1.5 CROSS=2

let propDelay = ((output_tplh - input_tplh) + (output_tphl - input_tphl))/2
echo $&propDelay > out.raw

set gnuplot_terminal=png/quit
let nanos = time*1e9
setscale nanos
gnuplot $file v(input) v(bOutput) xlabel "Time (ns)" ylabel "(V)"
.endc

.subckt inverterWithOutputCapacitor vDD input
xInverter vDD input capacitorIn inverter
```

```
cOutput capacitorIn 0
.ends
```

When $v_{DD} = 4\text{V}$, the propogation delay is 0.127ns. This method works since it makes the output voltage reach its peak at 4V whereas the input peaks at 3V. This makes the rising and falling output voltage slopes steeper and hence reduces the propogation delay.

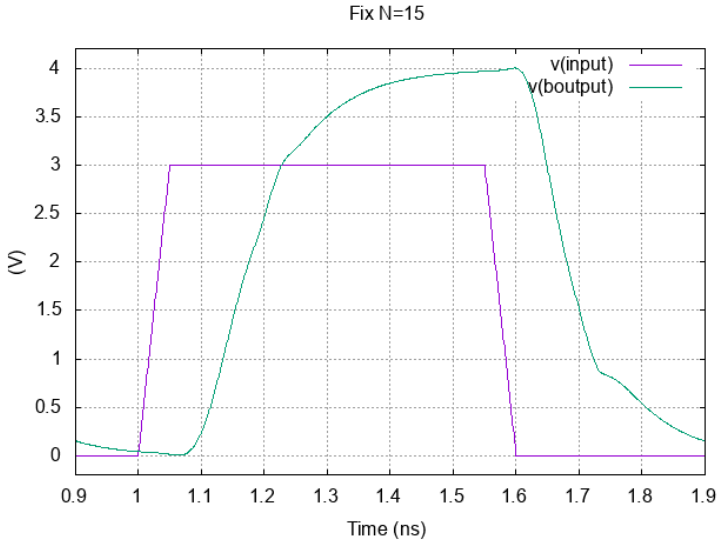


Figure 3: $v_{DD} = 4\text{V}$.

With the PMOS width set to 64u, and the NMOS width set to 48u, the propogation delay is $.139\text{ns}$. This method works since it allows more current to be sourced by inverter B when it is on and more current to be drained when it is off. More current allows the capacitative load to charge and discharge quicker, making the rise and fall times short for interver B.