

# Homework 1 Solutions

EECS 251B

## 1 System Interconnect

You have been tasked with building a system-interconnect for a system-on-chip with 256 processing cores. You are considering several options for the on-chip system interconnect network: (1) a 16-ary 2-mesh, (2) a 16-ary 2-cube, (3) an 8-ary 2-Cmesh4, and (4) 16-ary 3-fly Clos. The application requires a packet size of 1024 bits.

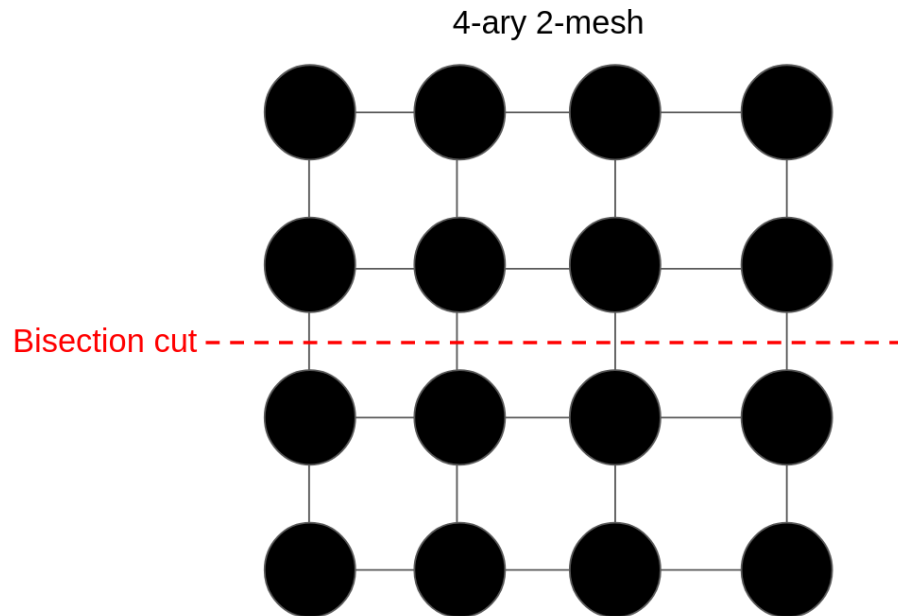
1. a) Calculate the required number of bits per unidirectional channel, for each network option, such that each of the networks can support the ideal throughput of 64 bits/cycle/core under uniform random traffic.

1. a) (1) 16-ary 2-mesh

First, let's calculate the total throughput ( $\Theta_{total}$ ) of the network.

$$\Theta_{total} = N * \Theta_{core} = 256 \text{ cores} * 64 \text{ bits/core/cycle} = 16,384 \text{ bits/cycle} \quad (1)$$

Now we need to determine the bisection bandwidth ( $B_B$ ) and the number of unidirectional bisection channels ( $B_C$ ). A 4-ary 2-mesh network is shown below. Let's solve for its  $B_B$  and  $B_C$  and then generalize the results.



The above figure's bisection cut goes through 4 bidirectional channels. Thus we get  $B_C = 4 * 2 = 8$  unidirectional channels. We can also see that under uniform random traffic this topology should see a

bisection bandwidth that is exactly half of the total throughput. Generalizing our results gives us:

$$B_C = \sqrt{N} * 2 \quad (2)$$

$$B_B = \frac{\Theta_{total}}{2} \quad (3)$$

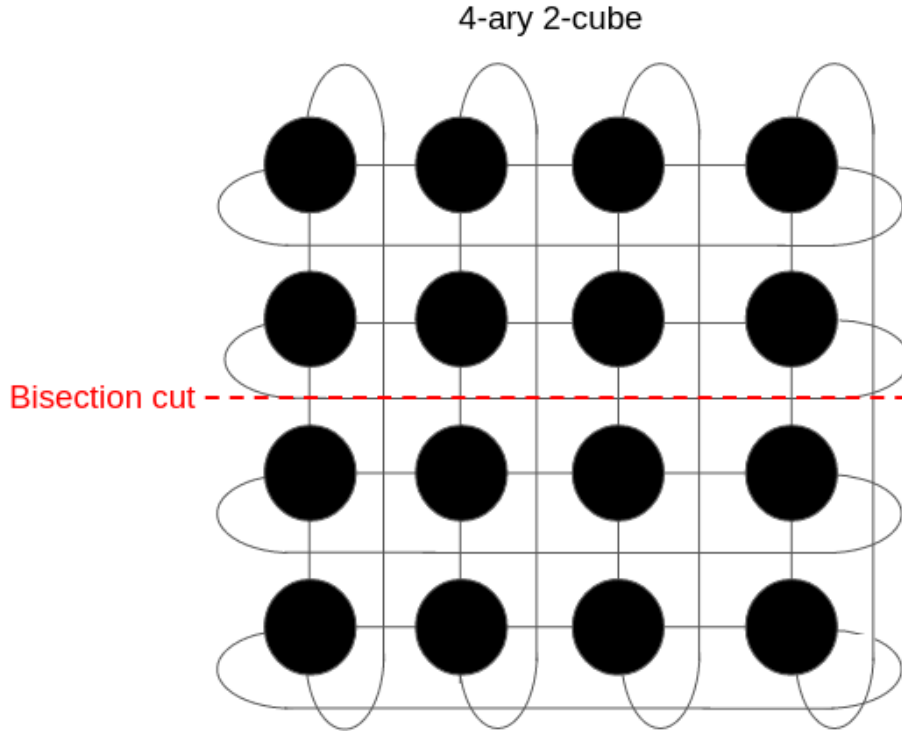
Now we can solve for the unidirectional bandwidth ( $b_C$ ) by dividing the bisection bandwidth ( $B_B$ ) by the number of bisection channels ( $B_C$ ). Plugging in the numbers for a 16-ary 2-mesh gives us:

$$b_C = \frac{B_B}{B_C} = \frac{\frac{16,384}{2}}{16 * 2} = 256 \text{ bits/channel} \quad (4)$$

#### 1. a) (2) 16-ary 2-cube

A similar analysis to "Question 1. a) - 16-ary 2-mesh" is performed for this topology.

A smaller network, 4-ary 2-cube, is shown below



The above figure's bisection cut goes through 8 bidirectional channels. Thus we get  $B_C = 8 * 2 = 16$  unidirectional channels. We can also see that under uniform random traffic this topology should see a bisection bandwidth that is exactly half of the total throughput. Generalizing our results gives us:

$$B_C = \sqrt{N} * 4 \quad (5)$$

$$B_B = \frac{\Theta_{total}}{2} \quad (6)$$

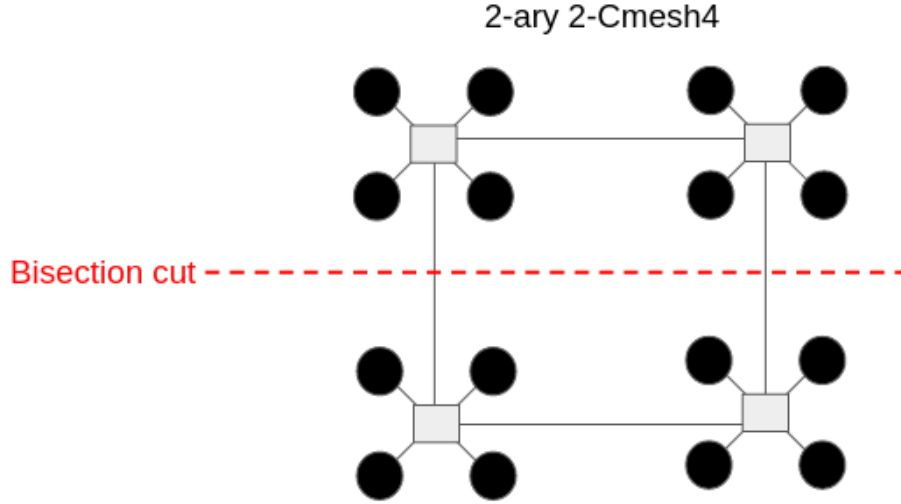
Now we can solve for the unidirectional bandwidth ( $b_C$ ) by dividing the bisection bandwidth ( $B_B$ ) by the number of bisection channels ( $B_C$ ). Plugging in the numbers for a 16-ary 2-mesh gives us:

$$b_C = \frac{B_B}{B_C} = \frac{\frac{16,384}{2}}{16 * 4} = 128 \text{ bits/channel} \quad (7)$$

### 1. a) (3) 8-ary 2-Cmesh4

A similar analysis to "Question 1. a) - 16-ary 2-mesh" is performed for this topology.

A smaller network, 2-ary 2-Cmesh4, is shown below



The above figure's bisection cut goes through 2 bidirectional channels. Thus we get  $B_C = 2 * 2 = 4$  unidirectional channels. We can also see that under uniform random traffic this topology should see a bisection bandwidth that is exactly half of the total throughput. Generalizing our results gives us:

$$B_C = \sqrt{\frac{N}{4}} * 2 \quad (8)$$

$$B_B = \frac{\Theta_{total}}{2} \quad (9)$$

Now we can solve for the unidirectional bandwidth ( $b_C$ ) by dividing the bisection bandwidth ( $B_B$ ) by the number of bisection channels ( $B_C$ ). Plugging in the numbers for a 8-ary 2-Cmesh4 gives us:

$$b_C = \frac{B_B}{B_C} = \frac{\frac{16,384}{2}}{\sqrt{64} * 2} = 512 \text{ bits/channel} \quad (10)$$

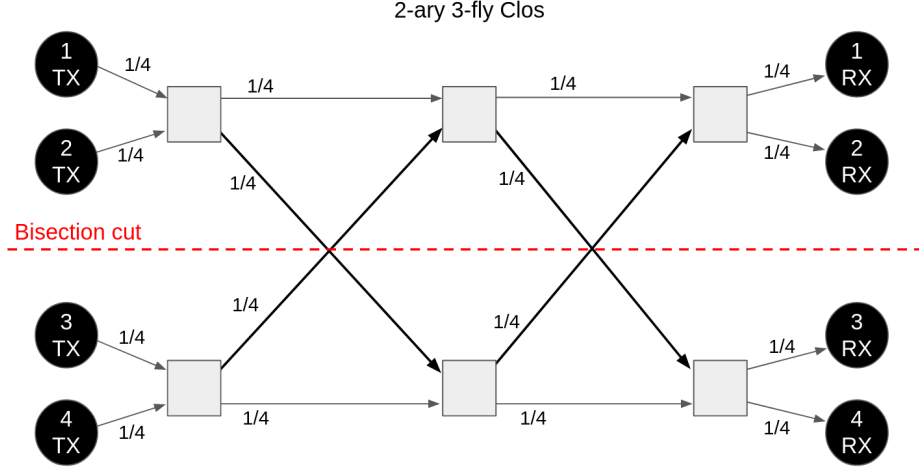
### 1. a) (4) 16-ary 3-fly

A similar analysis to "Question 1. a) - 16-ary 2-mesh" is performed for this topology.

A smaller network, 2-ary 3-fly Clos, is shown below

The above figure's bisection cut goes through 4 bidirectional channels. Thus we get  $B_C = 2 * 2 = 8$  unidirectional channels. We can also see that under uniform random traffic this topology should see a bisection bandwidth that is exactly equal to the total throughput. This is different than the previous topologies.

In the above image, the labeled "1/4" on all the channels represents the portion of the total network throughput that any given channel is seeing under uniform random traffic. Each core inputs 1/4 of the total throughput into the first stage (because we have 4 cores total in this example). The first stage switches will split the 1/4 input throughput from one node to two 1/8 throughput streams onto the outgoing channels. This will give us 1/4 of the total network throughput on both outgoing channels (1/8 from one input and 1/8 from the other). This is repeated until we output 1/4 of the total network throughput to each of the core's receivers. Now we can draw our bisection line, shown in red, and calculate the ratio of total throughput in our bisection cut. It should be very clear that the ratio is 1:1 because we have 4 channels crossing the bisection each carrying 1/4 of the total data throughput.



Generalizing our results gives us:

$$B_C = N \quad (11)$$

$$B_B = \Theta_{total} \quad (12)$$

Now we can solve for the unidirectional bandwidth ( $b_C$ ) by dividing the bisection bandwidth ( $B_B$ ) by the number of bisection channels ( $B_C$ ). Plugging in the numbers for a 16-ary 3-fly gives us:

$$b_C = \frac{B_B}{B_C} = \frac{16,384}{256} = 64 \text{ bits/channel} \quad (13)$$

1. b) **Show the worst-case zero-load latency breakdown for each interconnect network. Assume the flit size of 64b and shortest (core-to-core) channel latency of 1 cycle. For Clos, assume that the middle stages are located at the center of the chip**

#### 1. b) (1) 16-ary 2-mesh

Assuming cut-through flow control, we can calculate the zero-load latency as:

$$T_0 = H * t_r + \frac{D}{v} + \frac{L}{b} \quad (14)$$

Where  $H$  is the hop count,  $t_r$  is the router delay,  $D$  is the total physical distance the signal must traverse,  $v$  is the speed of the signal,  $L$  is the length of the signal in bits, and  $b$  is the channel bandwidth.

We have been given a channel latency of 1 cycle. This latency covers both the routing delay ( $t_r$ ) and the average time of flight delay for a single hop. We also know that our serialization delay is the number of cycles required to serialize the packet across a channel of width  $b_C$  (this is the number we calculated in the previous question). Thus, our latency equation can be rewritten in terms of "cycles":

$$T_0 = H * (1 \text{ cycle per hop}) + (\text{flits per packet}) * (\text{cycles per flit}) \quad (15)$$

The flit size is set to a static 64 bits regardless of the channel width  $b_C$ . With a packet size of 1024 bits, this gives us  $\frac{1024 \text{ bits per packet}}{64 \text{ bits per flit}} = 16$  flits per packet. "cycles per flit" is greater than 1 only when the channel width ( $b_C$ ) is less than the flit size. The routers will allocate buffer space at the flit level and thus channels w/ widths greater than the flit size will be underutilized (i.e. flits are sent sequentially across channels even

for channels with widths greater than the flit size. It is important to note that networks will generally never be designed such that the flit size is less than the channel width, but this question is an exception).

We want the worst-case zero-load latency and thus we have to find the diameter ( $H_{max}$ ) of the network to plug into our latency equation.  $H_{max}$  is the longest minimum-hop path between two nodes. For this topology,  $H_{max}$  is found from one corner node to the opposite corner node. We can now solve for the latency of the 16-ary 2-mesh:

$$H_{max} = (\sqrt{N} * 2) - 1 = 31 \quad (16)$$

$$T_0 = H_{max} + (\text{flits per packet}) * (\text{cycles per flit}) = 31 + 16 * 1 = 47 \text{ cycles} \quad (17)$$

#### 1. b) (2) 16-ary 2-cube

$H_{max}$  for this topology is found by going from one of the corner middle nodes to one of the outside corner nodes on the opposite side of the network. Thus,

$$H_{max} = \sqrt{N} + 1 = 17 \quad (18)$$

$$T_0 = H_{max} + (\text{flits per packet}) * (\text{cycles per flit}) = 17 + 16 * 1 = 33 \text{ cycles} \quad (19)$$

#### 1. b) (3) 8-ary 2-Cmesh4

$H_{max}$  for this topology is found by going from one of the corner node clusters to a node in the opposite corner node cluster.

$$H_{max} = (\sqrt{N/4} * 2) - 1 = 15 \quad (20)$$

$$T_0 = H_{max} + (\text{flits per packet}) * (\text{cycles per flit}) = 15 + 16 * 1 = 31 \text{ cycles} \quad (21)$$

#### 1. b) (4) 16-ary 3-fly

$H_{max}$  for this topology is found by going from any node to any other node.

$$H_{max} = 3 \quad (22)$$

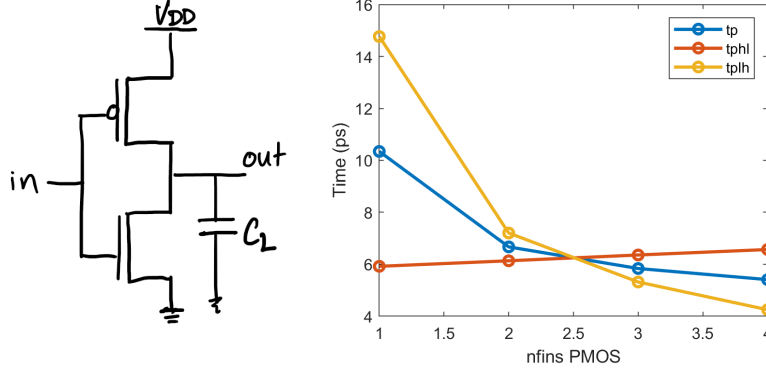
$$T_0 = H_{max} + (\text{flits per packet}) * (\text{cycles per flit}) = 3 + 16 * 1 = 19 \text{ cycles} \quad (23)$$

## 2 Transistor Sizing

### a) Inverter Sizing

The testbench should have an NMOS of fixed fin count and a PMOS of variable fin count. A step or pulse source should be injected with the delay measured by SPICE for both transition edges.

Starting with a 2-fin NMOS, 1fF loading, and 1ps input transition time, an equal  $t_p$  occurs at a ratio of 2:2.5. This is close to a **1:1 ratio**. In general, with other values of the fixed NMOS fins, the resulting 1:1 ratio is also achieved. Note that the propagation delay is as low as **just under 2ps** with much lower



output capacitance.

### b) Inverter Intrinsic Propagation Delay

Simulating the 2-fin inverter with no load,  $t_{pHL} = 911fs$ ,  $t_{pLH} = 949fs$ , and  $t_p = 930fs$ .

### c) Optimum Inverter Fanout

To find the optimal fanout, use the equation:

$$f_{opt} = e^{1 + \frac{\gamma}{f_{opt}}}$$

Finding  $\gamma$  requires two measurements: 1)  $delay_1$  = the delay from a FO1-loaded 2 fin inverter, and 2)  $delay_2$  = the intrinsic propogation delay above:

$$\begin{aligned} delay_1 &= \ln 2 R_{eq} C_{inv} \left( \gamma + \frac{C_{inv}}{C_{inv}} \right) = 1.91ps \\ delay_2 &= \ln 2 R_{eq} C_{inv} (\gamma) = 930fs \\ delay_1 - delay_2 &= \ln 2 R_{eq} C_{inv} \\ \gamma &= \frac{delay_2}{delay_1 - delay_2} = 0.9529 \end{aligned}$$

This  $\gamma$  is close to 1 (typical approximation). Solving the fanout equation yields  $f_{opt} = \mathbf{3.55}$ . This is close to the typical wisdom of optimal fanout of 4.

#### d) NAND2 Sizing

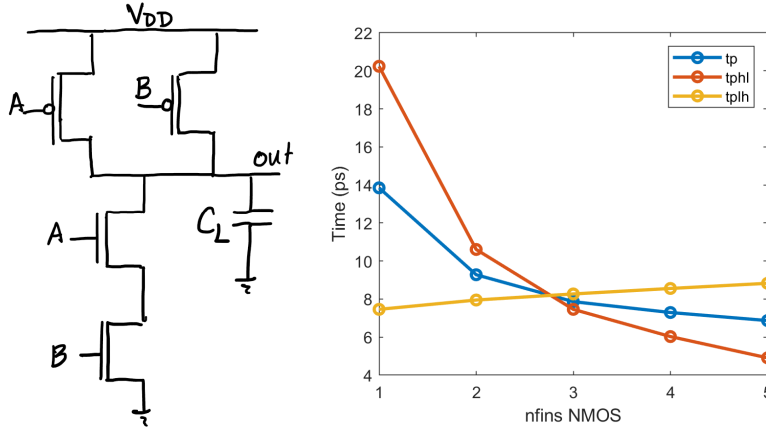
A stack of 2 NMOS transistors would be equivalent to having a length of  $2L$ . For the same resistance, we need the  $I_{DSat}$  to remain the same. Using  $E_C L = 0.7V$ ,  $V_{TH} = 0.2$ :

$$\begin{aligned}
 I_{DSatWinv} &= I_{DSatWNAND2} \\
 \frac{W_{INV}}{L} \frac{u_{eff} C_{ox} E_C L}{2} \frac{(V_{GS} - V_{TH})^2}{(V_{GS} - V_{TH}) + E_C L} &= \frac{W_{NAND2}}{2L} \frac{u_{eff} C_{ox} E_C 2L}{2} \frac{(V_{GS} - V_{TH})^2}{(V_{GS} - V_{TH}) + E_C 2L} \\
 \frac{W_{INV}}{1} \frac{E_C L}{2} \frac{1}{(V_{GS} - V_{TH}) + E_C L} &= \frac{W_{NAND2}}{2} \frac{E_C 2L}{2} \frac{1}{(V_{GS} - V_{TH}) + E_C 2L} \\
 \frac{W_{NAND2}}{W_{INV}} &= \frac{E_C L * 4(V_{GS} - V_{THN} + E_C 2L)}{2E_C 2L(V_{GS} - V_{THN} + E_C L)} \\
 &= \frac{0.7 * 4 * ((0.7 - 0.2) + 1.4)}{1.4 * 2 * ((0.7 - 0.2) + 0.7)} \\
 &= 1.58
 \end{aligned}$$

Using  $E_C L = 180mV$  and  $V_{TH} = 284mV$  from 1b:  $\frac{W_{NAND2}}{W_{INV}} = 1.30$ .

Using  $E_C L = 100mV$  and  $V_{TH} = 317mV$  from 1b when fitting only  $E_C L$ :  $\frac{W_{NAND2}}{W_{INV}} = 1.21$ .

None of these are quantized to an integer multiple of fins. From SPICE simulations sweeping  $W_n$  while holding the bottom input high, we see that the optimal fin count is about  $2.7/2=1.35$  larger than the inverter fin count:



Sources of discrepancy may stem from imperfect fitting to the used models in problem 1 and NAND2 gate resistances following a different trajectory that does not reach the expected  $Req$  values as discussed in lecture.

#### e) Logical Effort and Intrinsic Delay of NAND2

The logical effort is:

$$g = \frac{C_{inNAND2}}{C_{inINV}}$$

Using symmetrical effort sizing, the NMOS fin width should be 3 (1.5 normalized):

$$g = \frac{1.5 * fins + 1 * fins}{1 * fins + 1 * fins} = 1.25$$

Intrinsic delay: SPICE simulation of the unloaded symmetrically sized NAND2 gate yields  $t_{pHL} = 1.43ps$ ,  $t_{pLH} = 1.34fs$ , and  $t_p = \mathbf{1.38ps}$ .

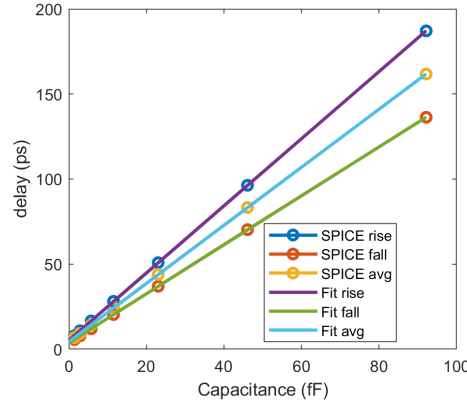
## f) NAND2 Logical Effort and Intrinsic Delay from .lib

From the NAND2x2 cell from the TT library and extrapolating for pin A, use the `cell_rise` and `cell_fall` times. Index 1 (row) is the input transition time in ps, index 2 (column) is the capacitance in fF:

```
cell_fall (delay_template 7x7 x1) {
  index 1 ("5, 10, 20, 40, 80, 160, 320");
  index 2 ("1.44, 2.88, 5.76, 11.52, 23.04, 46.08, 92.16");
  values ( \
    "5.68567, 7.80473, 11.9661, 20.3491, 36.9879, 70.2487, 136.313", \
    "6.75933, 8.88777, 13.125, 21.5053, 38.0899, 71.3456, 137.311", \
    "8.34562, 10.8441, 15.156, 23.499, 40.2018, 73.347, 139.426", \
    "10.5202, 13.6981, 18.9513, 27.6791, 44.3997, 77.5414, 143.878", \
    "12.9208, 17.4459, 24.2987, 35.0985, 52.6797, 85.9863, 152.217", \
    "14.7737, 21.4324, 31.1568, 45.413, 67.1879, 102.533, 168.949", \
    "14.5634, 24.1866, 38.4263, 58.7031, 87.7034, 131.336, 202.258" \
  );
}

cell_rise (delay_template 7x7 x1) {
  index 1 ("5, 10, 20, 40, 80, 160, 320");
  index 2 ("1.44, 2.88, 5.76, 11.52, 23.04, 46.08, 92.16");
  values ( \
    "7.80536, 10.8263, 16.6254, 28.1067, 50.8377, 96.3349, 187.104", \
    "9.35012, 12.252, 17.9993, 29.5165, 52.2326, 97.6007, 188.727", \
    "12.5161, 15.4135, 21.1455, 32.5012, 55.4738, 100.884, 191.633", \
    "17.3971, 21.3043, 27.5431, 38.9515, 61.6827, 107.297, 198.178", \
    "24.5504, 30.0314, 38.6787, 51.6461, 74.3285, 119.783, 210.842", \
    "34.981, 42.6968, 55.0358, 73.0705, 99.8227, 145.617, 236.428", \
    "50.7793, 61.1333, 78.2723, 104.187, 142.019, 196.206, 287.382" \
  );
}
```

Extrapolating for the first row:



The found intrinsic values are:  $t_{pLH} = 5.23\text{ps}$ ,  $t_{pHL} = 3.72\text{ps}$ ,  $t_p = 4.47\text{ps}$  (note this is higher than Spice sims due to a difference in transition time: 1ps vs. 5ps).

To find the logical effort, find the pin capacitances of the INVx2 input gate (pin A) and the NAND2x2 gate (either pin A or B), as these gates have similar drive strength. Plugging in the values:

$$g = \frac{C_{NAND2}}{C_{INV}} = \frac{1.38\text{fF}}{0.83\text{fF}} = 1.66$$

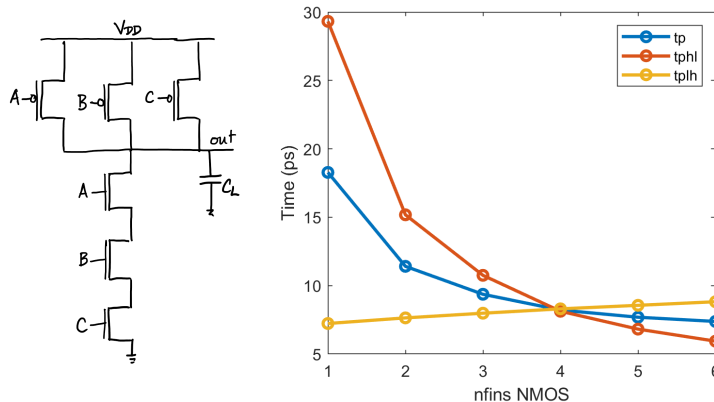
## g) NAND3 Sizing

$$\begin{aligned}
 I_{DSatWinv} &= I_{DSatWNAND3} \\
 \frac{W_{INV}}{L} \frac{u_{eff} C_{ox} E_C L}{2} \frac{(V_{GS} - V_{TH})^2}{(V_{GS} - V_{TH}) + E_C L} &= \frac{W_{NAND3}}{3L} \frac{u_{eff} C_{ox} E_C 3L}{3} \frac{(V_{GS} - V_{TH})^2}{(V_{GS} - V_{TH}) + E_C 3L} \\
 \frac{W_{INV}}{1} \frac{E_C L}{2} \frac{1}{(V_{GS} - V_{TH}) + E_C L} &= \frac{W_{NAND3}}{3} \frac{E_C 3L}{3} \frac{1}{(V_{GS} - V_{TH}) + E_C 3L} \\
 \frac{W_{NAND3}}{W_{INV}} &= \frac{E_C L * 9 * ((V_{GS} - V_{THN}) + E_C 3L)}{6 E_C L ((V_{GS} - V_{THN}) + E_C L)} \\
 &= \frac{0.7 * 9 * ((0.7 - 0.2) + 2.1)}{2.1 * 2 * ((0.7 - 0.2) + 0.7)} \\
 &= 3.25
 \end{aligned}$$

Using  $E_C L = 180\text{mV}$  and  $V_{TH} = 284\text{mV}$  from 1b:  $\frac{W_{NAND2}}{W_{INV}} = 2.40$ .

Using  $E_C L = 100\text{mV}$  and  $V_{TH} = 317\text{mV}$  from 1b when fitting only  $E_C L$ :  $\frac{W_{NAND2}}{W_{INV}} = 2.12$ .





From SPICE simulation,  $t_{pHL} = 1.74\text{ps}$ ,  $t_{pLH} = 1.70\text{fs}$ , and  $t_p = \mathbf{1.72\text{ps}}$  for a gate ratio of  $\approx 2 : 1$ . Logical effort from the SPICE results is:

$$g = \frac{2 * f_{ins} + 1 * f_{ins}}{1 * f_{ins} + 1 * f_{ins}} = 1.5$$

## h) NOR2 Sizing

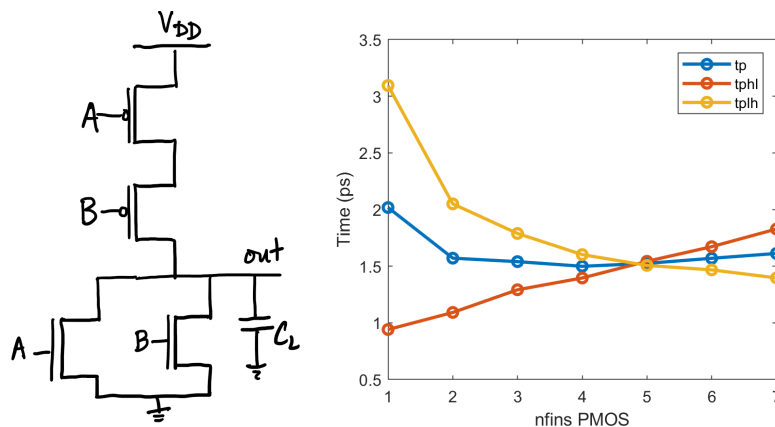
$$\frac{W_{NOR2}}{W_{INV}} = \frac{E_{CL} * 4(V_{GS} - V_{THN} + E_{CL}2L)}{2E_{CL}2L(V_{GS} - V_{THN} + E_{CL}2L)}$$

For the homework provided  $E_{CL}$  and  $V_{TH}$ :  $\frac{W_{NOR2}}{W_{INV}} = 1.58$ .

For fitted  $E_{CL} = -262\text{mV}$ ,  $V_{THP} = -266\text{mV}$ :  $\frac{W_{NOR2}}{W_{INV}} = 1.28$ .

For fitted  $E_{CL} = -110\text{mV}$ ,  $V_{THP} = -314\text{mV}$ :  $\frac{W_{NOR2}}{W_{INV}} = 1.22$ .

Using the optimal ratio of PMOS:NMOS width obtained in part (a) (2.5:2), the expected PMOS width is somewhere between 1.525 and 1.975 times that of the NMOS (quantized), resulting in about the same logical effort as NAND2 of 1.25. However, SPICE (below) shows a 2.5:1 PMOS:NMOS fin ratio to be symmetric sizing. Reasons for this are likely the same as those in part d.



The intrinsic delay of the 5 fin PMOS, 2 fin NMOS device with 1fF loading and 1ps input transition time is  $t_{pHL} = 1.54\text{ps}$ ,  $t_{pLH} = 1.51\text{fs}$ , and  $t_p = \mathbf{1.52\text{ps}}$ . Logical effort from the SPICE-derived optimal sizing is:

$$g = \frac{2.5 * f_{ins} + 1 * f_{ins}}{1f_{ins} + 1f_{ins}} = 1.75$$