Task #1: Dynamic power - tapered buffers for clock tree

Task parameters:

- $f_{\text{clk}} = 500\text{MHz}$
- $V_{\text{DD}} = 1.2V$
- $\alpha_{\text{clk}} = 1$
- $C_{\text{H-tree}} = 20.536pF = 57044C$
- $C = 0.36\text{fF}$, the input capacitance of an X1 inverter in the $65\mu m$ process.

---

a)
$P_{\text{dyn}} = \alpha_{\text{clk}} \cdot f_{\text{clk}} \cdot C \cdot V_{\text{DD}}{}^2 = 1 \cdot 500\text{MHz} \cdot 20.536pF \cdot 1.2V^2 = 14.8\text{mW}$.

b)
The tapered buffer designed in *lab 4* had the following characteristics

- N = 6
- F = 4.92

The total capacitance of the tapered buffer is thus the sum of the input and output capacitances scaled by F cumulatively for each consecutive inverter, starting with X4

$$C_{\text{taper}} = \sum_{i=0}^{5} 2 \cdot 4C \cdot 4.92^i = 29069C = 10.5pF$$

$P_{\text{dyn}} = 1 \cdot 500\text{MHz} \cdot 10.5pF \cdot 1.2V^2 = 7.5\text{mW}$.

c)
The 10 stage tapered buffer has the following characteristics

- N = 10
- F = 2.61

The total capacitance in the tapered buffer is, using the same reasoning as in 1b)

$$C_{\text{taper}} = \sum_{i=0}^{9} 2 \cdot 4C \cdot 2.61^i = 72885.2C = 26.2pF$$

$P_{\text{dyn}} = 1 \cdot 500\text{MHz} \cdot 26.2pF \cdot 1.2V^2 = 18.9\text{mW}$.

In this configuration the tapered buffer has a *path delay* of $D = 10 \cdot 2.61 + 10 = 36.1$, or 180.5ps in absolute terms (with $\tau = 5ps$).

d) The width of the unit nMOS transistor is $0.1\mu m$ in the $65\mu m$ process. The width of the unit pMOS is double that. Therefore, simplified, the total width of an X1 inverter is $0.3\mu m$ while the channel length is constantly $0.06\mu m$. Thus, for a single stage i in the tapered buffer, the area is WLX, where X is the scale relative to the unit inverter. Since the smallest inverter is X4 and the scaling factor is F, the scale $X = 4 \cdot F^i$.

The area of the 6-stage buffer is

$$A_{\text{6-stage}} = WL \sum_{\text{i}=0}^{5} 4 \cdot 4.92^i = 261\mu m^2$$

whereas the area of the 10-stage buffer is

$$A_{\text{10-stage}} = WL \sum_{\text{i}=0}^{9} 4 \cdot 2.61^i = 656\mu m^2$$

e) The results of this exercise are summarised in table 1. From this we can surmise that, due to the insensitivity of the delay to number of stages, a significant increase in the number of stages gives a minuscule increase in delay from the optimum value. There is, however, a large area and power penalty, as each extra stage is exponentially bigger, taking more power to switch. As a final note, the dynamic power of the tapered buffer is far larger than that of the load itself, unsurprisingly, as the final inverter of the tapered buffer alone has a diffusion capacitance comparable to that of the load in magnitude.

Table 1: *Summarised results for task 1.*

	My own tapered buffer from lab 4	A 10-stage tapered buffer with F = 2.61	The load capacitances themselves
Delay	177ps	180ps	n/a
Dynamic Power	7.5mW	18.9mW	14.8mW
Tapered buffer area (only transistor area)	$262\mu m^2$	$656\mu m^2$	n/a

Task #2: Dynamic power - activity factors

Since the *AOI21* and *OAI21* gates are really just cascaded two-input gates with an output inverter, with switching probabilities P given by table 5.1 (W&H), P and α can be worked out successively, from input to output. This has been done for the aforementioned compound gates with the results in figures 1 and 2, respectively.

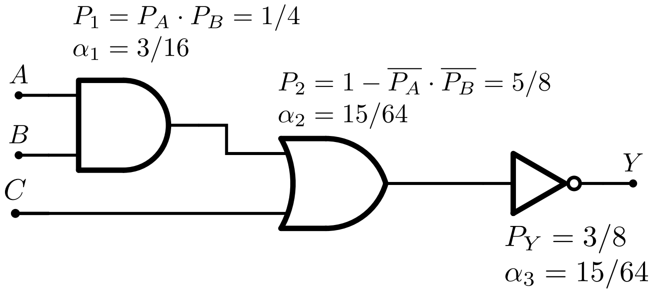
$P_1 = P_A \cdot P_B = 1/4$
$\alpha_1 = 3/16$
A
B
C
$P_2 = 1 - \overline{P_A} \cdot \overline{P_B} = 5/8$
$\alpha_2 = 15/64$
Y
$P_Y = 3/8$
$\alpha_3 = 15/64$

Figure 1: *AOI21 Gate*

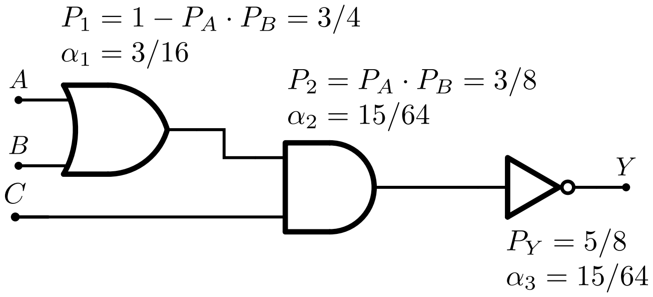

Figure 2: *OAI21 Gate*

Task #3: Static power - power gating

a) The normalised delay of a circuit depends on the capacitive load it presents and its effective resistance

$$d = 0.7CR_{\text{eff}}$$

where $R_{\text{eff}} = \frac{V_{\text{DD}}}{I_{\text{ON}}}$. V_{DD} and I_{ON} are known to be 1.2V and 100mA respectively and therefore $\text{R}_{\text{eff}} = 12\,\Omega$. In order for the delay of the combinational logic to increase by a maximum of 2%, the header switch must contribute at most 2% to the overall R_{eff} of the circuit. This corresponds to

$$R_{\text{eff}_{\text{header}}} \leq 2\% \frac{V_{\text{DD}}}{I_{\text{ON}}} = 2\% \cdot 12\Omega = 0.24\Omega$$

Since the pMOS used in the header switch has an effective resistance of $2.5\,\text{k}\Omega\,\mu\text{m}$, the width required to fulfill the above criteria is

$$W \geq \frac{R_{\text{eff}_{\text{pMOS}}}}{R_{\text{eff}_{\text{header}}}} = \frac{2.5\,\text{k}\Omega\,\mu\text{m}}{0.24\Omega} = 10416\mu m$$

b)

In *lab 1* it was determined that the gate capacitance C_G of a unit transistor $(W = 0.1\mu m)$ is 0.12 fF. The capacitance of the header switch can therefore be calculated

$$C = 1.2\,\text{fF}/\mu\text{m}^2 \cdot W\mu\text{m} = 12.5pF$$

c)

The energy required to use the switch is

$$E = \frac{1}{2} \cdot C \cdot V_{\text{DD}}{}^2 = 9pJ$$

Task #4: Ripple carry cell reconsidered

a)

$$
\begin{aligned}
G =& A \cdot B \\
P =& A \oplus B
\end{aligned}
$$

b)

$$
\begin{aligned}
C_{\text{out}} =& G + P \cdot C_{\text{in}} \\
S =& P \oplus C_{\text{in}}
\end{aligned}
$$

c)

In the same way as in prelab 2, only the inverted *carry function* can be realized in a single CMOS gate (using exclusively positive inputs). This gives us $f(P, G, C_{\text{in}}) = \overline{f_{\text{maj}}} = \overline{G + P \cdot C_{\text{in}}}$. Thus, for the n-net

$$f_n = C_{\text{out}} = G + P \cdot C_{\text{in}}$$

and just as before, the p-net then can be defined as the *dual* of the n-net

$$f_p = f_n^D = G \cdot (P + C_{\text{in}})$$

The final transistor schematic (with output inverter) is shown below in figure 3.

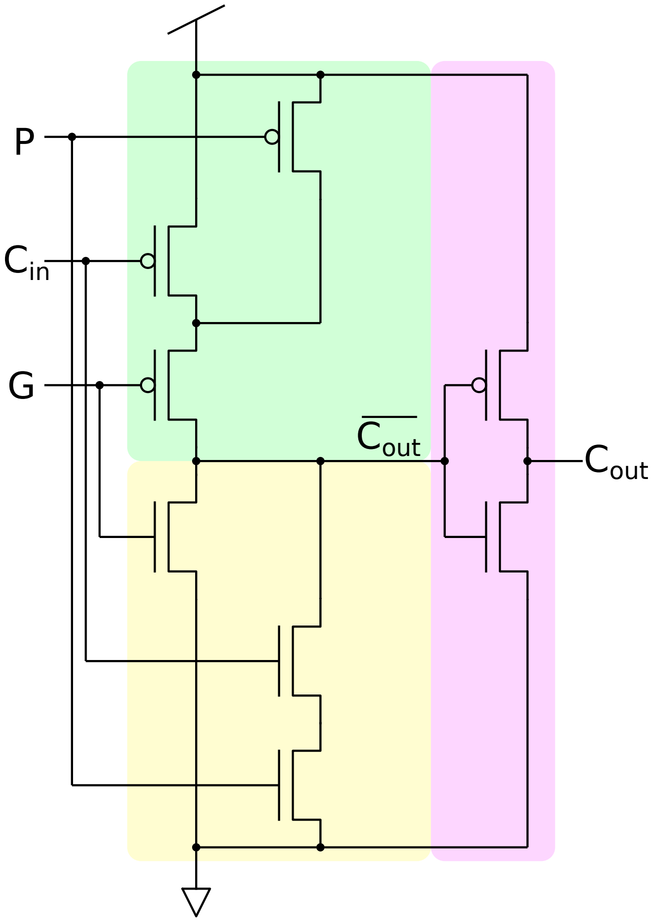

Figure 3: *Transistor-level carry cell using propagate and generate signals.*

d)

The *logical effort* g of the carry cell is the input capacitance of the gate relative to the input capacitance of an inverter that can drive the same amount of current.

$$g_{\mathrm{C_{in}}} = \frac{C_{\mathrm{in}}}{C_{\mathrm{inv}}} = \frac{2C + 4C}{3C} = 2$$

The *parasitic delay* p is the capacitance directly connected to the output of the gate relative to that of the unit inverter, that is 4C from the p-net and 3C from the n-net

$$p = \frac{4C + 1C + 2C}{3C} = \frac{7}{3}$$

It is worth noting that only p has changed in this reconsidered design. In other words, the *gate complexity* has stayed the same as far as $\mathrm{C_{in}}$ is concerned.

e)

In order to determine the relative size of the output inverter that will result in minimum delay, we have to find the optimal *stage effort* $\hat{f} = F^{\frac{1}{N}}$. The delay is then simply

$$D_{\min} = \sum d_i = D_F + P = \sum \hat{f}_i + P$$

where D_F is the *path effort delay* and P is the *path parasitic delay.*

The *logical efforts* of the carry cell and inverter are fixed (g only describes the relative complexity of the gate itself) as well as their respective parasitic delays. Thus it is the input capacitance of the inverter that must be changed.

First, we derive the various *path efforts* (under the assumption that the output drives an identical carry cell)

$$\begin{aligned} G =& g_{\mathrm{carry}} \cdot g_{\mathrm{inv}} = 2 \cdot 1 = 2 \\ B =& 1 \\ H =& \frac{C_{\mathrm{out}}(path)}{C_{\mathrm{in}}(path)} = \frac{6C}{6C} = 1 \end{aligned}$$

The *path effort* F is therefore

$$F = GBH = 2$$

giving an optimal *stage delay* of $\hat{f} = F^{\frac{1}{N}} = \sqrt{2}$, where N is the number of stages, two in this case.

Now that $\hat{f}$ is known, the input capacitance of the inverter can be calculated using the *capacitance transformation formula*

$$C_{\mathrm{in_i}} = \frac{C_{\mathrm{out_i}} \cdot g_i}{\hat{f}_i} \Rightarrow \frac{6C \cdot 1}{\sqrt{2}} = 3\sqrt{2}C$$

again, assuming the output is driving an identical carry cell.

The inverter size relative to the carry gate C_{in} (or equivalently, the *electrical effort* h_{carry}) is thus $\frac{3\sqrt{2}C}{6C} = 0.707$

f)

The resulting normalized delay of a 16-bit ripple-carry chain can be computed directly

$$\begin{aligned} D = N\hat{f} + P =& 16 \cdot 2 \cdot \sqrt{2} + 16 \cdot (\frac{7}{3} + 1)p_{\mathrm{inv}} \\ =& 32\sqrt{2} + \frac{160}{3}p_{\mathrm{inv}} \xrightarrow{p_{\mathrm{inv}} \approx 1} 98.6\tau \\ =& 492.9ps \end{aligned}$$

with each link having 6.2τ vs. the 9.5τ of prelab 2; a 35% improvement!

Task #5: The Dot operator or PG logic

a)
The generic block propagate can be expressed as

$$P_{\text{i:j}} = P_{\text{i:k}} P_{\text{k-1:j}}$$

In this case we have two consecutive bits and hence the expression becomes reduced to

$$P_{\text{i+1:i}} = P_{\text{i+1}} P_{\text{i}}$$

b)
Likewise, the generic block generate is

$$G_{\text{i:j}} = G_{\text{i:k}} + P_{\text{i:k}} G_{\text{k-1:j}}$$

which is in this case reduced to

$$G_{\text{i:j}} = G_{\text{i+1}} + P_{\text{i+1}} G_{\text{i}}$$

c)
The expression for the *generate* in task **5b)** is the same as the *carry function* in task **4c)**. This is no coincidence, as one can redefine C_{in} as the initial *generate*, $G_{0:0}$, to derive the same expression. Thus the schematic will remain unchanged, but with slightly different inputs, as shown in figure 4.

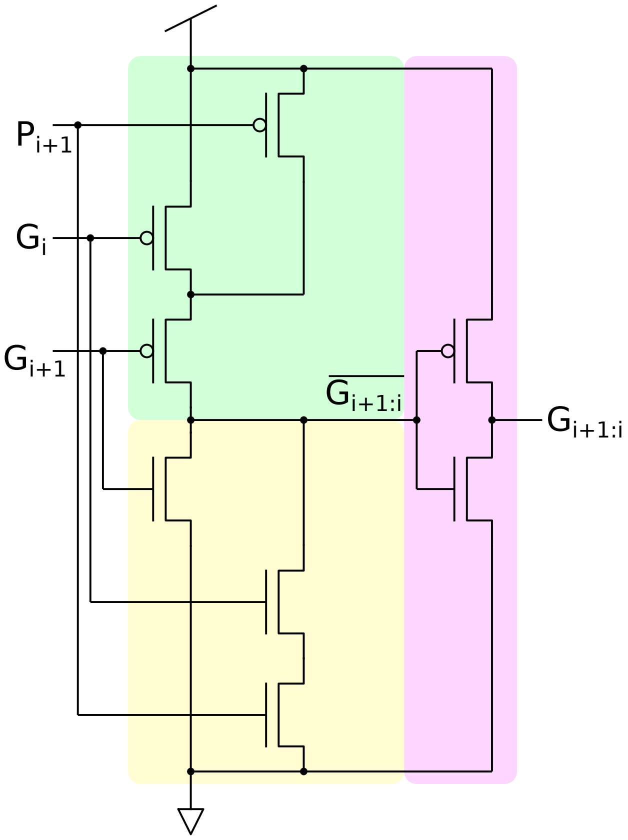

Figure 4: *Transistor-level block generate gate. p-net, n-net and output inverter highlighted.*

d)
Using the generic expressions from **4a)** and **b)**, the *generate* and *propagate* for the pairwise signals are

$$P_{\text{i+3:i}} = P_{\text{i+3:i+2}} \cdot P_{\text{i+1:i}}$$
$$G_{\text{i+3:i}} = G_{\text{i+3:i+2}} + P_{\text{i+3:i+2}} \cdot G_{\text{i+1:i}}$$