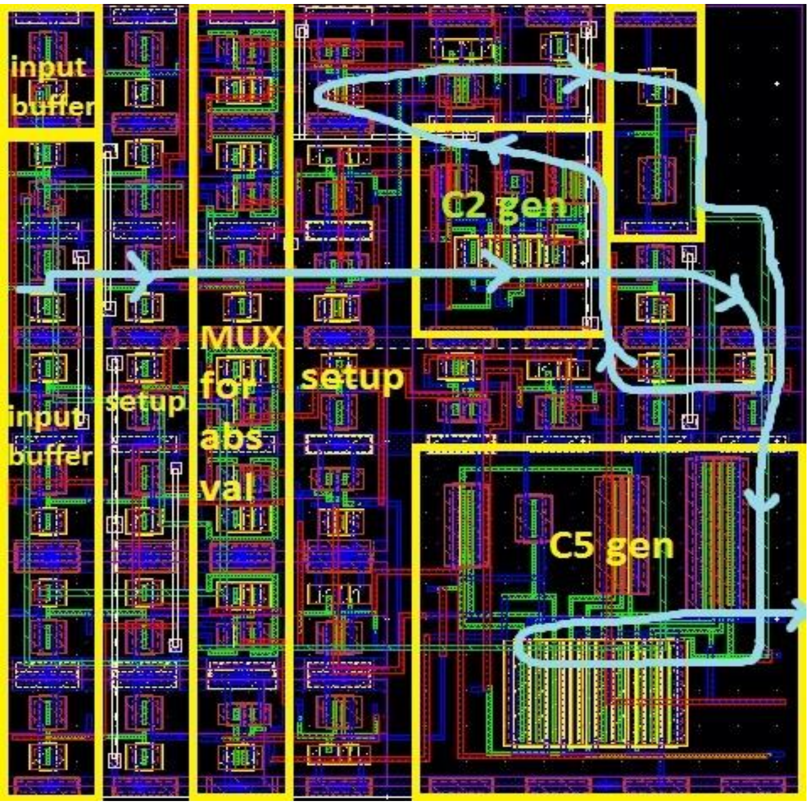
EE115C Winter 2016 Final Project:
A 1.25GHz, 312 fJ 6-bit Absolute-Value Detector

Matthew Quach (UID: 304159669)
Jeffrey Jiang (UID: 904255069)
Team 8

Layout

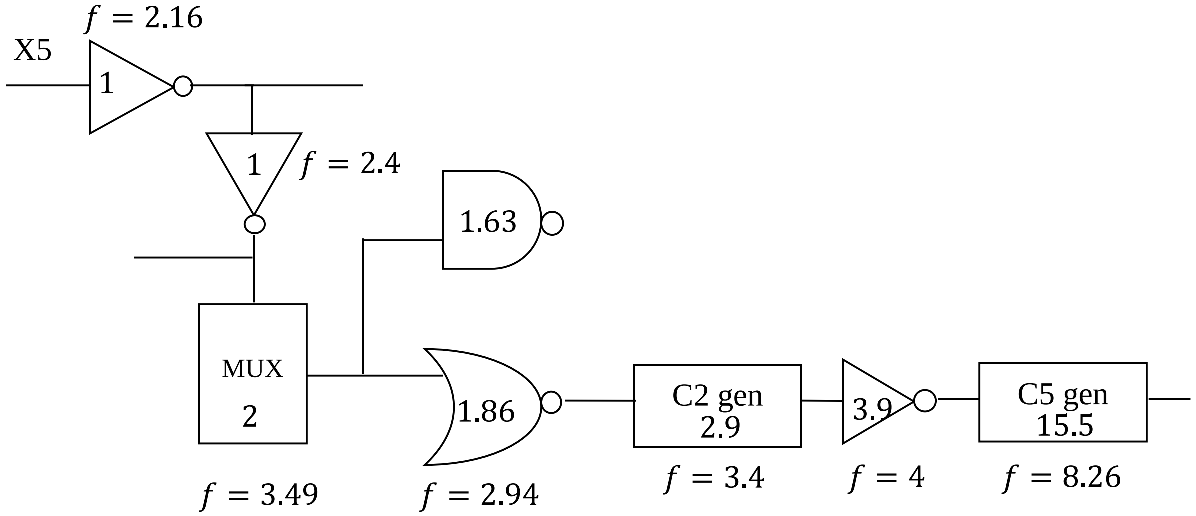


Specifications	
Critical Path (X3 to OUT) Delay, t_p, post layout	773.05 ps
$t_{p\_X0\text{-}OUT}$, post layout	644.15 ps
$t_{p\_X5\text{-}OUT}$, post layout	603.95 ps
X (horizontal dimension)	20.64 µm
Y (vertical dimension)	22.2 µm
Area (X×Y)	458.21 μm^2
Aspect Ratio	1.08
Supply Voltage	0.725 V
Schematic Energy	271.09 fJ
Layout Energy	312.17 fJ
Functionality	**Y**
DRC Clean	**Y**
LVS Clean	**Y**

We decided to approach this problem with a logic optimization effectively combining the absolute value generator and the comparator by creating a carry generator of an adder. Assume that both $X$and T are positive 2's complement numbers. Therefore, the sign bit of both X and T are 0. Then, we know that $\overline{T} = -T - 1$ will be a negative number with 1 as the sign bit, implying that $X + \overline{T} = X - (T + 1)$. If the sum is positive or zero (sign bit is 0), then we know that $X > T$. Otherwise, $X \leq T$ However, since the original MSB of both X and $\overline{T}$are known, the sum is definitely positive if the addition of the 5-th bit has a carry. Thus, we have reduced our problem to finding the carryout of the 5th-bit carry of the sum of a positive X and $\overline{T}$. We noticed that the block in Appendix C actually mirrors this idea.

Instead of explicitly finding $|X|$, we know that since $-X = \overline{T} + 1$, we can instead find the inverse first and allow for a carry-in of 1 to signal that we need to add 1, selecting the inverse input of a MUX rather than the input and carrying in 1 when the sign bit is 1. This adds a bit of complexity to our "comparator" in order to remove the extra adder stage. The MUX is our closest form of a absolute value generator, and is implemented with transmission gates.

Using the general logic from carry-lookahead logic, we have that the carry is: $OUT[C5] = G_4 + G_3P_4 + G_2P_4P_3 + G_1P_4P_3P_2 + G_0P_4P_3P_2P_1 + C_iP_4P_3P_2P_1P_0$, where $C_i = X_5$ and $G_i = X_i\overline{T_i}$ and$P_i = X_i + \overline{T_i}$. Instead of implementing a carry-lookahead adder, however, we just intend to make a CMOS gate with this function. Of course, since we do not want to have a single gate to have 6 different inputs, we split the logic into two stages to implement: $OUT[C5] = G_4 + G_3P_4 + G_2P_4P_3 + C_2P_4P_3P_2$, $C_2 = G_1 + G_0P_1 + C_iP_1P_0$.



The schematic for the critical path is shown below. The schematic of our design is relatively similar, except with 5 MUXes for each of the input bits all connecting into the C2_gen which continues to C5_gen for the output. The C2_gen takes in all the generate and propagate signals produced by the first two input bits into it as well as a buffered sign-bit. Similarly, the C5_gen takes in the carry from C2_gen and all the generates and propagates and generates an output.

The critical path that we proposed on the schematic was the path that passed through the most stages. This was the path from X_5 to the output, which passed through a total of 7 stages. As a result, we have the critical path delay equation of $t_{critical} = 2t_{inv} + t_{mux} + t_{NOR} +$

$t_{C2Gen} + t_{inv'} + t_{C5Gen}$.

For our analysis, we assumed a 1.5:1 reference inverter, due to the size of the unit inverter. All the stage efforts and sizes we chose are shown in the diagram. Since we have a relatively large gate of “C5_gen,” we took advantage of a large stage effort as the last stage in order to minimize our area in hopes of lowering the energy. Using the fact that our design has relatively few stages, we were able to lower the energy significantly by making our sizing somewhat suboptimal for speed. After we minimized our two complex gates as much as we could, we just wanted to optimize the remaining stage efforts by letting their stage efforts to be about the same, increasing slightly if possible. This allowed for a good trade-off of low energy consumption with a decent delay.

However, after implementation and layout we found that the slowest path did not actually take place through the switching of X_5, which surprised us. We believe that in thinking that we had already chosen the critical path, we believe we did not choose optimal gate sizing for buffers off of the critical path because we believed that they would not impact the delay, opting to choose lower area for higher energy performance, as well as ease in layout due to fewer inverter types. These considerations ended up costing us in terms of delay performance. One point that does confuse us, however, is the large difference in the performance of each input bit. While we expect all the input bits to operate in parallel, in both the schematic and the layout, there is a large difference in the performances of $X_3$and X_4. We cannot find a good explanation for why this occurs, but we still indicate the fact that X_3constitutes our critical path.

In the end, we were able to decrease voltage to 710 mV with the given testbench and we increased it to 725 mV just to add a buffer from unknown test cases creating a bad situation. However, based on our worst case input patterns, this definitely enough. The one we tested on was when the threshold was 15, but the input was switching between -16 and -15. This forced both of our long stacks in our carry generators to be on, which we expected to be the worst case input pattern, which led to a delay of around 780ps.

In terms of layout, we generally opted to make a very standardized set of sizings for dimensions in order to make it easy to combine at the higher level stages. Next, we tried to place the modules in a way such that the average wiring length as short as possible, while maintaining as few metal layers as we could. We ended up having a relatively compact, highly dense, chip that was wired with only 3 metal layers (other than the power rails). Finally, we also paid attention to the critical path and made sure that it appeared to be relatively short.

Given another chance, the main things we would do differently is to spend more time understanding the critical path and sizing. The main drawback of our analysis is that we were too focused on a single critical path and we forgot to take into consideration new critical paths that could emerge as a result of poor sizing on “non-essential” routes. While this is true, I feel like we could have been more lenient during this process to save a tremendous amount of time, since we iterated through this process over several days. Taking more time to consider topologies could also have benefitted us, since having the 4-stack of PMOS transistors really pushed at our sizing and energy constraints. Possibly splitting our gates into more, but smaller, gates may have been a better choice that we never attempted to try.