Elaboration on Reduced Logic

Magnitude- General logic for 2’s complement of n+1 bits (n+1)th bit is sign bit

+)+]

We then take the complement of the function above/ simplify, and set Y=

=; an XOR function between and Y which we will call our Carry unit, its equal to the summation of the previous bits \* sign bit

We could’ve further simplified the implementation of this function using transmission gates, which would of greatly reduced the energy consumed by our circuit.

Comparator- We first realize that there are three results when comparing two bits

Win-

Tie-

Loss-

We then Analyze each bit starting from the MSB and carry the Tie bit which we will call C to the next bit to indicate whether its value matters or not

C=

Output of MSB=

We carry C=, which will =0 if A is greater or less than B

We then and this C to the next bit (telling it whether its output/ carry matters)

Output of next bit =

Carry; C=

We then take the outputs and OR them to reach a verdict.

Elaboration on problems

Our biggest problem was an unwanted spike in our output. This is due to the fact that the delay of each path is different and because of that our output reads a transient value for a brief moment that is undesired. To try and solve this problem we initially sized our gates so that their respective path delay would be equal, in doing so our gates consumed more power than they would have unsized/ with the spike. This would of also effected the size of the layout, so we chose the easiest solution and ignored it. If we had more time, we would of built custom sized logic/ layouts to solve this problem or implemented a better topology that computed our carry values simultaneously; equalizing any delay that would have been associated with our carry gen stages. There was also another unexplored potential solution to this, which was rearranging/ restructuring our layout in such a way where we equalize our delay there rather than in the schematic, but this involves much more time and knowledge about layout design than what we have now.