NSGA - II Implementation on Full Adder

Dr.Zia Abbas

Mentor : Prateek Gupta

Table of Contents

Objective

Reduce Leakages and Delays in Full Adder circuit based on transistor sizing

Motivation

- Increased downscaling of CMOS circuits with respect to feature size and threshold voltage has a result of dramatically increasing leakage current and decreasing delays.
- So, leakage power and delay reduction is an important design issue for active and standby modes as long as the technology scaling increased.
- Decreasing one parameter by transistor sizing increases the other. Therefore, we need to size the transistors in which the delay and leakage are both reduced or in a specific bound.

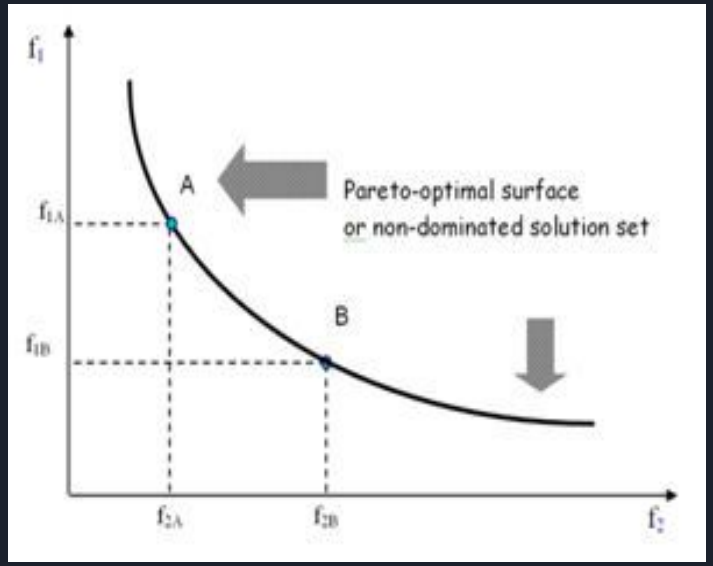
Approach

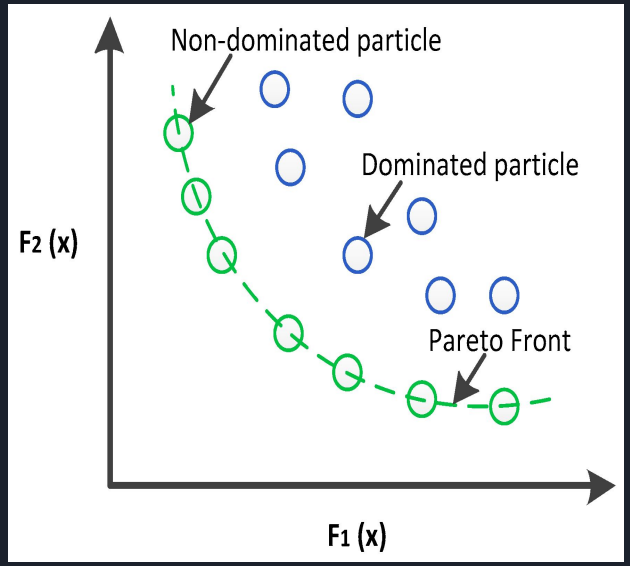
- As we see, decreasing one parameter by transistor sizing increases the other. Therefore, there is a need to control the sizing in such a way that we reduce both delays and leakages and get the best solution possible.
- For this, we need to use a multi-objective optimisation technique which reduces more than one parameters.
- The multi-objective function we adapted is Non-dominated Sorting Genetic Algorithm - II (NSGA - II)

Genetic Algorithm

- The concept of the GA is based on the evolutionary process of biological organisms in nature. During the course of evolution, populations evolve according to the principles of natural selection and “survival of the fittest”.
- Individuals that are more successful in adapting to their environment will have a better chance of surviving and reproducing, whereas individuals that are less fit will be eliminated.
- This means the genes from the highly fit individuals will spread to an increasing number of individuals in each successive generation.
- The combination of good characteristics from highly adaptive ancestors can produce even more fit offspring. In this way, the species evolve to become increasing more suitable to their environment.

Pareto Optimal Surface for 2-objective problem





The figures shown are for 2-objective problems. The Pareto optimal surface contains the best possible or non-dominated solutions

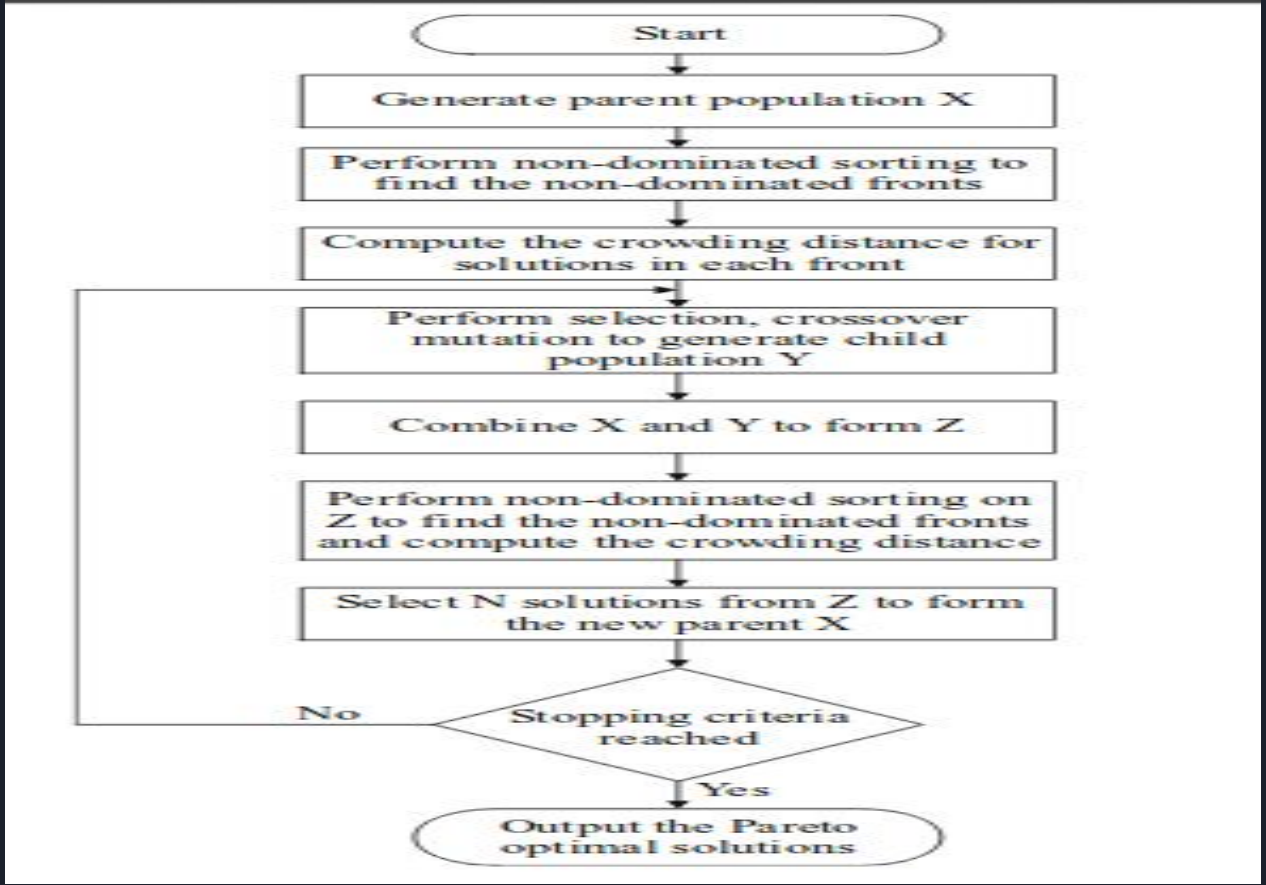
NSGA-II

- NSGA-II employs a fast non-dominated sorting approach with m number of objectives and N population to be sorted.
- The first is to classify the population into various non domination fronts using a non dominated sorting procedure. (ranking acc to heuristic function).
- The second is to estimate the density of solutions surrounding a particular point in the population by means of crowding distance computation.
- Non-dominated sorting is used to classify the population to identify the solutions for the next generation.
- The individuals are selected based on rank and crowding distance metric, every individual in the population is assigned a crowding distance value

NSGA-II

- The normal Genetic Algorithm is a single objective function which only reduces a single parameter but to reduce multiple parameters, we use NSGA-II which is a multi-objective function.
- It employs non-dominated sorting techniques to provide the solution as close to the Pareto-optimal solution as possible in a single run.
- It uses crowding distance technique to provide diversity in solution.
- Uses elitist techniques to preserve the best solution of current population in next generation. It uses explicitly diversity preserving mechanism

Flowchart of NSGA-II



NSGA-II Generation and Combination

Initial population Generation ->Non-domination sorting -> Binary tournament selection -> Recombination -> Mutation -> .... -> Pareto solution set (crowding distance assignment is used after reproduction steps)

Fast non-domination sorting has to be performed twice, i.e.

1. after population initialisation

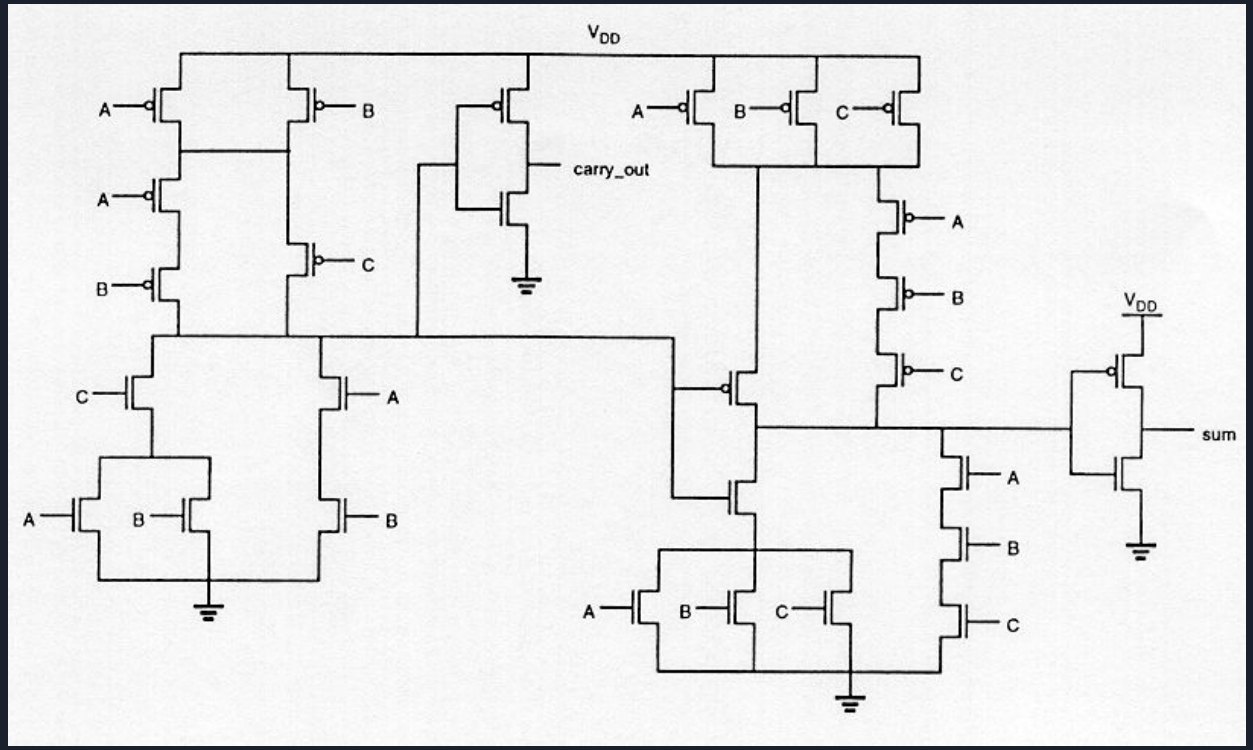
2. after parent and offspring population combination.

In selection process, crowded comparison operator is used based on non domination rank and crowding distance properties. So, crowding distance assignment need to be done before selection process.

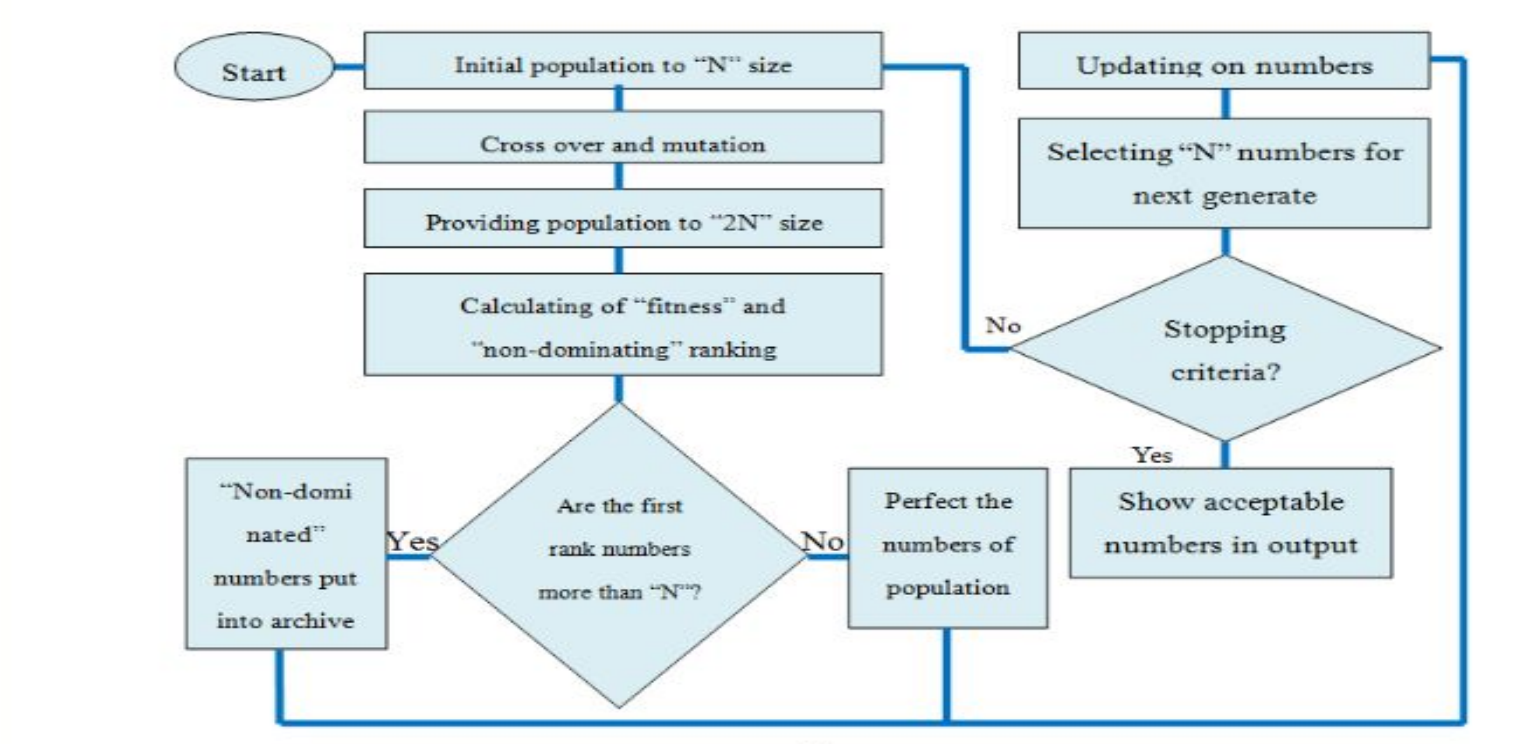
NSGA-II Crowding Distance Computation

- With respect to each objective crowding distance is computed , to represent how best the solution is wrt that particular parameter.
- The best solution at the end will be known by considering Crowded Tournament Selection.
- Within the first front, the one with highest crowding distance wrt leakage objective is taken as the best solution.(after running for 100 generations.

Full Adder Circuit



NSGA-II on Full Adder Circuit



Approach

Inputs: sizes of generative Full Adder transistors w,l values of 14 pmos & 14 nmos.

Outputs: average Leakage power and max delay optimisation.

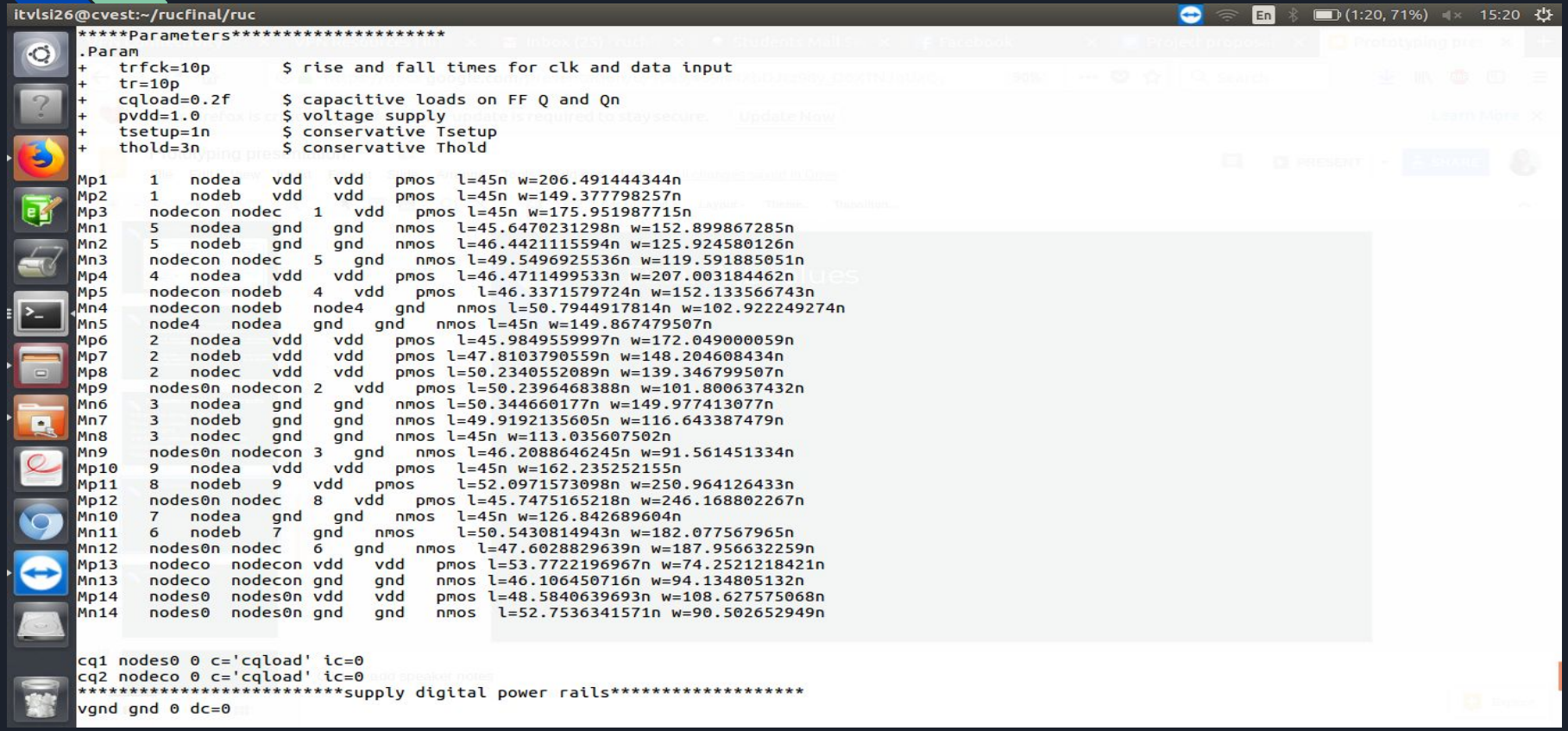
Objective: To reduce the average leakage power by 50% while keeping the maximum delay in bound(10% of initial value) for various temperatures and Vdd's.

Method :Full Adder is made of two separate circuits each part is optimized separately.The number of population for both circuits are 100 and algorithm will continue to 20 generations.

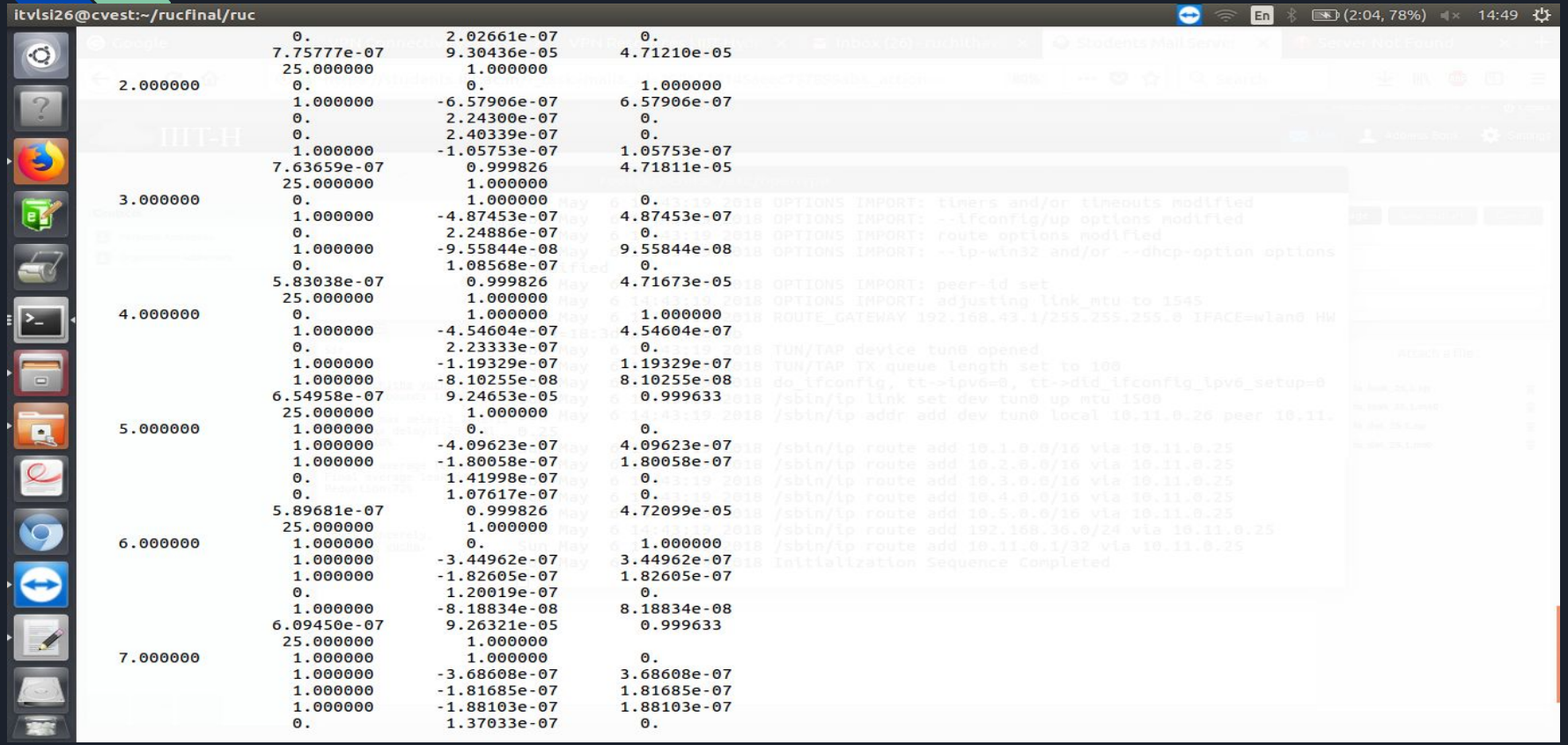
Implementation and Results

Initial max delay:1.18 e-11
Final max delay:1.29 e-11
Bounds:10%
Initial average leakage:3.09 e-06
Final average leakage:6.738 e-07
Reduction:78%

Final W,L values



Final Delay Leakage values



```
    4.000000          0.                1.000000          1.000000
                      1.000000         -4.54604e-07       4.54604e-07
                      0.                2.23333e-07       0.
                      1.000000         -1.19329e-07       1.19329e-07
                      1.000000         -8.10255e-08       8.10255e-08
                     6.54958e-07       9.24653e-05        0.999633
                     25.000000          1.000000
    5.000000          1.000000          0.                0.
                      1.000000         -4.09623e-07       4.09623e-07
                      1.000000         -1.80058e-07       1.80058e-07
                      0.                1.41998e-07       0.
                      0.                1.07617e-07       0.
                     5.89681e-07        0.999826          4.72099e-05
                     25.000000          1.000000
    6.000000          1.000000          0.                1.000000
                      1.000000         -3.44962e-07       3.44962e-07
                      1.000000         -1.82605e-07       1.82605e-07
                      0.                1.20019e-07       0.
                      1.000000         -8.18834e-08       8.18834e-08
                     6.09450e-07       9.26321e-05        0.999633
                     25.000000          1.000000
    7.000000          1.000000          1.000000          0.
                      1.000000         -3.68608e-07       3.68608e-07
                      1.000000         -1.81685e-07       1.81685e-07
                      1.000000         -1.88103e-07       1.88103e-07
                      0.                1.37033e-07       0.
                     7.38396e-07       9.33960e-05        0.999633
                     25.000000          1.000000
    8.000000          1.000000          1.000000          1.000000
                      1.000000         -1.52799e-07       1.52799e-07
                      1.000000         -1.79906e-07       1.79906e-07
                      1.000000         -1.87686e-07       1.87686e-07
                      1.000000         -1.59938e-07       1.59938e-07
                     6.80329e-07        0.999826          0.999634
                     25.000000          1.000000
[itvlsi26@cvest ruc]$ cat gen1208/mutation/delay/84_delay.mt0
$DATA1 SOURCE='HSPICE' VERSION='B-2008.09 64-BIT'
.TITLE '*************************************************'
 delay_lh_nodeaco delay_hl_nodeaco delay_lh_nodebco delay_hl_nodebco
 delay_lh_nodecco delay_hl_nodecco temper           alter#
  1.28864e-11      1.00405e-11      1.20291e-11      1.16464e-11
  1.29130e-11      1.01722e-11       25.000000         1.000000
[itvlsi26@cvest ruc]$
```

Conclusion

After performing the algorithm ,several results will be obtained as they have no priority to each other ,so designer can select whichever according to his need. Simulation results show that final structure uses less average power. Algorithm program is written in Python and the circuit simulated by Hspice.

References

- Revisiting the NSGA-II Crowding-Distance Computation.

  -University level, Canada

- A high-speed hybrid Full Adder with low power consumption

  -ICICE Electronics Express

- Application of Evolutionary Algorithms for Multi-objective Optimization in VLSI and Embedded Systems

Our Team

UnniKrishnan R

Ruchita V

Sathya Sravya V

Raja V

Thank you!

Group-1