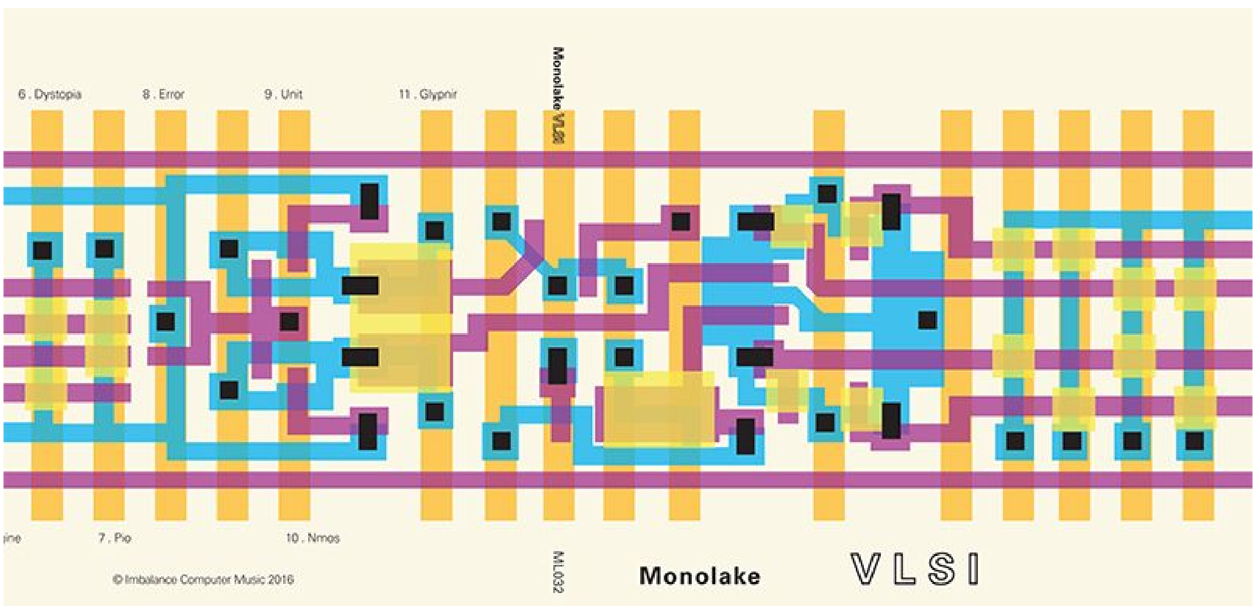
Harris Hawk Optimisation for Digital Circuits

Digital VLSI Project

Spring-19

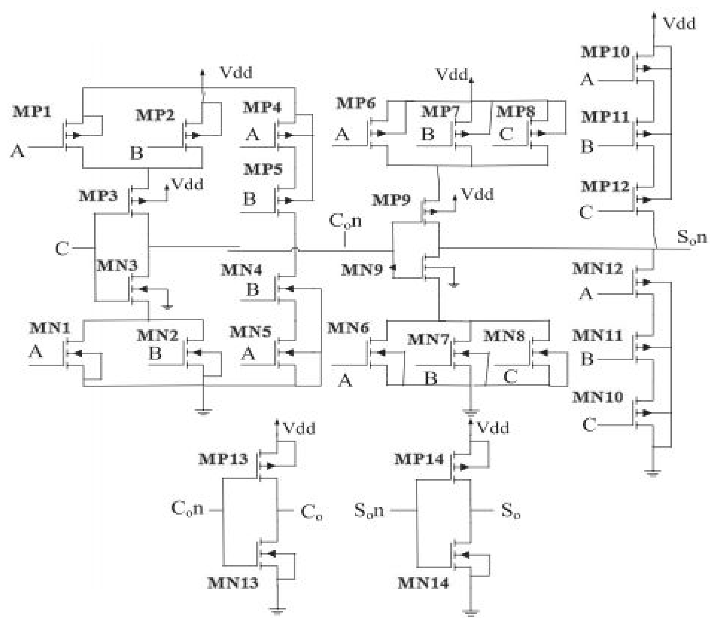


Sravan M
Sreyamsh B
Sreevatsava M
Shanmukh A

INTRODUCTION

Transistor scaling has consistently improved the performance of digital circuits for the past 50 years. Depending on transistor scaling for faster and more efficient integrated circuits is failing in recent years as we are approaching the physical limits of semiconductor. Hence the importance of circuit design and architecture to improve the performance of an IC have increased tremendously. In this project we aim to improve the performance of a CMOS full adder circuit by optimizing the sizing of transistors for static leakage and critical path propagation delay using the Harris Hawk optimization algorithm.

Harris Hawk Optimization is a single objective, population based, nature inspired genetic algorithm that can be used to optimize non differentiable functions. In our case since the dependence of transistor sizing with leakage and propagation delay is unknown or at least non differentiable we attempt use the HHO algorithm to optimize for the leakage and critical path propagation delay.

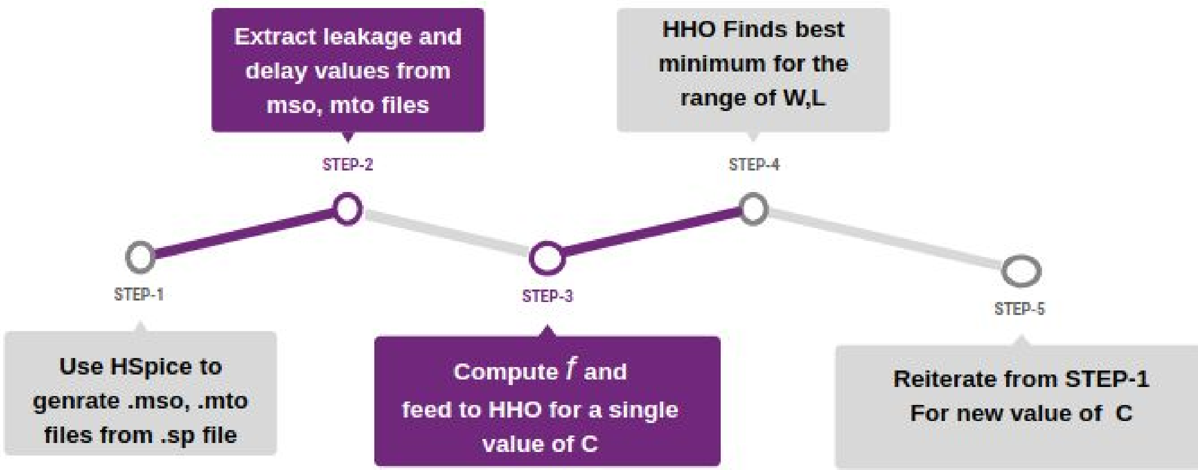


CMOS Full Adder Figure A

A CMOS full adder circuit has 28 transistors of which 14 are NMOS transistors and the rest PMOS transistors. The circuit can be seen in the Figure A. Each transistor has two sizing parameters the width and the length of the channel. This results in a total of 56 parameters to be found.

The circuit has 3 inputs and 2 outputs which lead to 6 paths from input to output which results in 6 delay objectives to optimize for and the circuit has 8 leakages to be optimized for this results in 14 objectives where as the optimization algorithm given accepts a single objective. To reduce the number of objectives we choose the max of all the delays to represent the delay of the critical path and choose the sum of all leakages to represent leakage. We then minimize the sum of the two objectives with varying weightage to obtain a pareto frontier from where we can choose a particular weightage based on the design constraints and then optimize the weighted sum to obtain the required parameters

Pipeline of the whole process



PROCEDURE

Harris Hawk Optimization

The Harris Hawk Optimisation(HHO) is inspired by how Harris Hawks use different exploration methods and pouncing strategies to ensure that they capture their prey. HHO is a population-based, gradient-free optimisation technique and hence it can be applied to any optimization problem subject to a proper formulation. We can use this nature inspired HHO for minimising leakage current and transmission delays, by varying the transistor widths and lengths.

The Algorithm assumes the following

- N points are chosen at random(referred to as Hawks)
- The optimum point obtained at each iteration(Motion of how a rabbit moves)
- Choosing how the N points(Hawks) move(Modelled based on Hawks motion)
- Global optimum obtained(The point at which the Hawk catches the prey)

The Algorithm is based on how HHO hunt its prey and there are 2 main phases to it:-

- Exploration Phase
- Exploitation Phase

The transition between these phases is done based on the equation below where E_0 is the initial energy of the rabbit.

$$E = 2E_0(1 - \frac{t}{T})$$

Exploration Phase

In HHO the hawks perch randomly on some location and wait to detect a prey based on two different strategies. The first strategy being a X_n Hawk takes a liking towards a certain X_m Hawk and moves toward it by some distance in the direction of the prey/rabbit. Another strategy being the Hawk X_n moves towards to mean position of the whole pack in the direction of the prey/rabbit. As we assume that both the strategies are equally likely we use a random number q in the function, and say that if q >= 0.5 we use the first strategy else we use the second strategy.

The equations for the above are modelled as follows:-

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases}$$

Exploitation Phase

The exploitation phase has 4 types which are as follows

- Soft Besiege
- Hard Besiege
- Soft Besiege with progressive dives

- Hard Besiege with progressive dives

Soft and Hard Besiege

When the rabbit still has energy left to move i.e we still haven't found the global optimum yet, then it tries to get away by moving in some random direction. During these attempts the Harris Hawks encircle it slowly to make the rabbit exhausted and then perform the surprise pounce. This is know as Soft Besiege and is modelled as follows.

$$X(t+1) = \Delta X(t) - E\left|JX_{rabbit}(t) - X(t)\right|$$
$$\Delta X(t) = X_{rabbit}(t) - X(t)$$

When the rabbit/prey is exhausted to point that it is not able to make significant jumps and moves only by a little then the Harris Hawks encircle the pray and perform a direct pounce on the rabbit instead of waiting for it to move. This is know as Hard besiege and can be modelled as follows.

$$X(t+1) = X_{rabbit}(t) - E\left|\Delta X(t)\right|$$

Soft besiege and hard besiege are chosen at based on the energy of the prey i.e if E_0 >= 0.5 then we perform soft besiege else we perform hard besiege.

Soft and Hard Besiege with progressive dives

Levy flight function

The progressive dives used by the Hawks in HHO is said to be modelled by the given equation given below

$$Z = Y + S \times LF(D)$$

Where ***LF(D)*** is known as the Levy flight function which can be calculated as follows

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}}$$

Now the main assumption here is that Hawks may or maynot wait for the entire time they encircle the prey hence they encircle for sometime and then perform a dive to a point which the Hawks thinks the prey will be at in the future. Therefore the only difference between this and the previous methods is the progressive dive which has been modelled, They are modelled as follows

SOFT Besiege with progressive dives

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X(t)\right|$$

$$Z = Y + S \times LF(D)$$

$$X(t+1) = \begin{cases} Y & ifF(Y) < F(X(t)) \\ Z & ifF(Z) < F(X(t)) \end{cases}$$

Hard besiege with progressive dives

$$X(t+1) = \begin{cases} Y & ifF(Y) < F(X(t)) \\ Z & ifF(Z) < F(X(t)) \end{cases}$$

where Y and Z are obtained using:

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X_m(t)\right|$$

$$Z = Y + S \times LF(D)$$

Therefore in total there are 4 stages for exploitation, when the energy of the rabbit is more then we perform Soft besiege or soft besiege with progressive dives else if the energy of the rabbit is less then we use either Hard besiege or Hard besiege with progressive dives. As there are two options for Hard and soft besiege we use a random variable **r** to select whether to perform the dive or not.

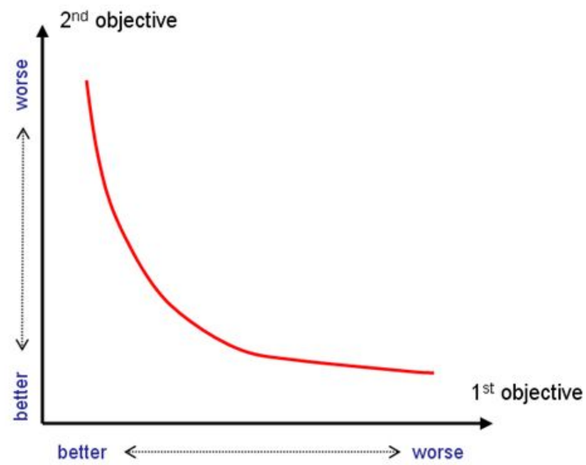
Objective Function Conversion

The HHO defined above has only single objective function and our problem requires dual objective problem. Hence we use the approach of Pareto optimality curve to create a combination of two functions into one.

Pareto Optimality Curve

Pareto optimality curve is similar to that shown in figure below. Extremes represent single objectives while the function is multi objective. Points on the curve are obtained by varying weightage for the individual objectives(c). Points above the curve are feasible and below are infeasible In our case,

1st Objective : Leakage
2nd Objective : Delay



Since the optimization is randomized, pareto curve cannot be reproduced for multiple simulations.In the graph, the lower Skyline is used as pareto curve by taking points from multiple simulations. In our case, delays are represented by max(6-delays). Leakages are represented by sum(8-leakages). Hence the two functions are to be combined into one.

Objective function to convert : **f** = **c*max(delays)+(1-C)*sum(leakages)**

When C=0, delays are ignored and while C=1, leakages are ignored. Trade off for both occurs at some C which the algorithm helps to find.

Issues and modifications of the Objective functions

We take logarithm of objective function as the values of the objectives is very small. Leakages are in the order of 10^{-6} while delays are in the range of 10^{-11}. Taking a logarithm normalizes these values. Due to this, the linear combination tends to increase weightage for one objective creating bias. Hence we use exponentially varying weightage values to get even distribution of points to counter this effect. That is, instead of C varying as 0.1,0.2,0.3,……, incrementation is done by:

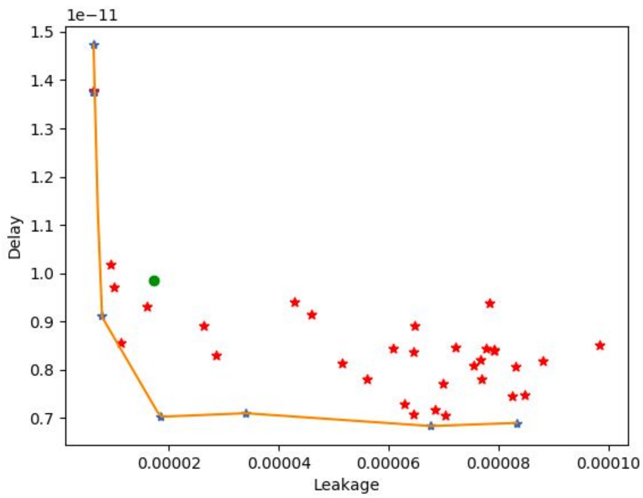
$$C_i = 1 - 1/k^i$$

The value of k is change from 3-6 to get various points of optimality. Finally we plot them and take the skyline points as the best values.

RESULTS

Results from both the optimizations and application of it on the adder circuit are given below.

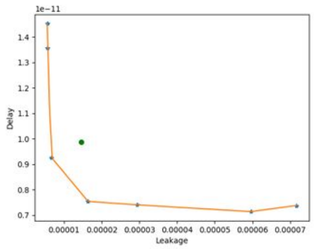
Complete simulation Results



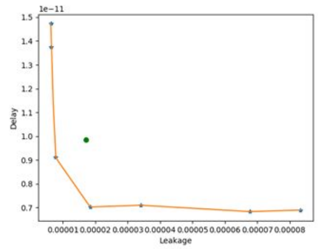
This is the pareto-optimality curve for different ‘c’ values in single graph. The red points are the optimum values on a range of ‘c’ values, green point is the default parameters point. Line indicates the skyline points connected. Skyline means the lowest possible points such that there is no point to the left, once they are joined.

Temperature Scaling

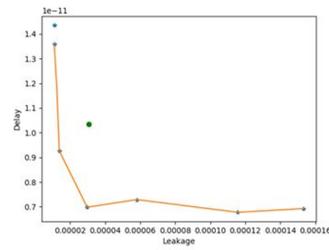
Pareto optimality curve at different temperatures.



-55°C



25°C



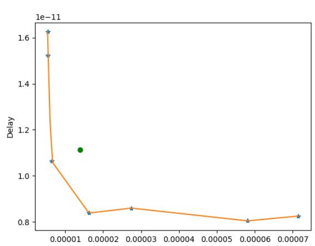
125°C

The change in temperature does not affect our optimization as we can see here. The green point(default parameter) lies always above and hence our optimization is surely better.

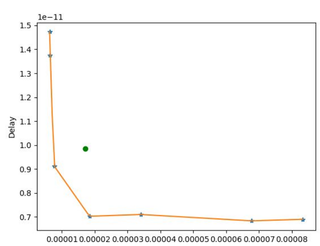
Since our algorithm is nature inspired and not a deterministic algorithm ie; does not give the same output for the same input parameters everytime. Hence we use different values of c and obtain the best points which are the skew lines above.

Supply Voltage Scaling

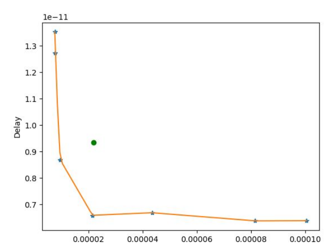
Pareto optimality curve for +/-10% in supply voltage.



0.72 V



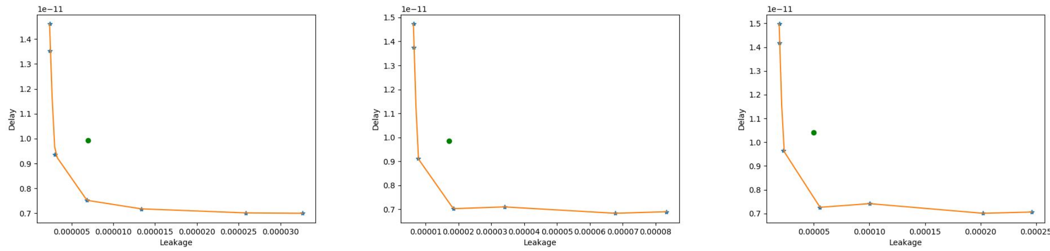
0.8 V



0.88 V

Process Variation Scaling

Pareto optimality curve for +/-10% process variation.



+10% 0 -10%

It can be observed that the optimal values found on the pareto frontier are better than the default parameter even at various temperature, process and supply voltage variations.

Results for optimizing single objective with no loss in the other.

Optimizing Objective	Leakage(reduction)	Delay(reduction)
Leakage	41.79%	Default
Delay	Default	28.73%

RESOURCES

1. HSpice
2. Python
3. Linux

REFERENCES

1. HHO optimization