2D Group 26

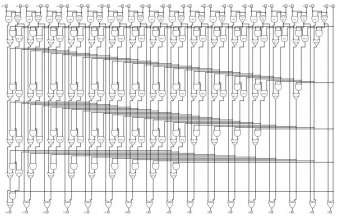
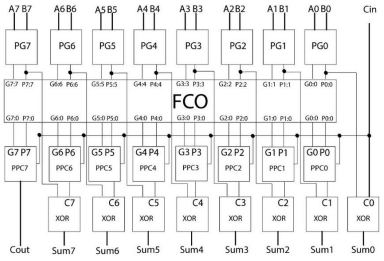
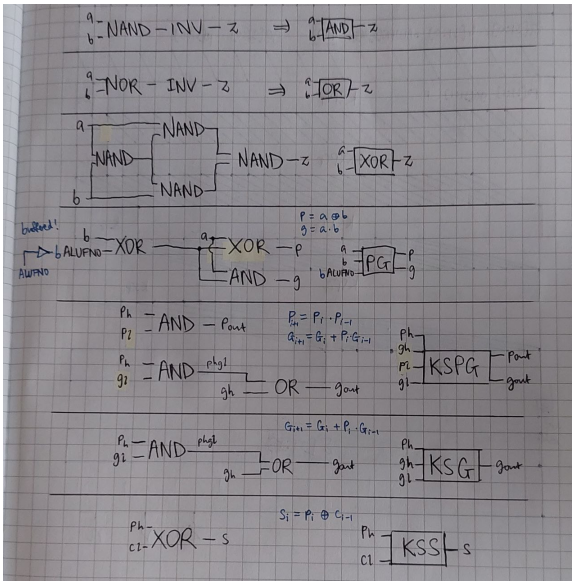
- Anshu Arun Ghate
- Cassie Chong Kenci
- Christy Lau Jin Yun (50.002 leader)
- Li Si Hui Brenda
- Tan Lay Lin
- Win Tun Kyaw

50.002 - Optimising an Adder

Part 1

Structure

Our goal is to optimise a 32 bit adder with delay being under 3ns.
Inspired by Kogge-Stone adder, our adder architecture comprises of gates arranged in a tree-like structure to allow parallel computation of addition, be using propagated and generated signals to calculate carry instead of relying on the previous outpit bit's carry out as done in a ripple carry adder.

16bit with all gates	8bit with abstracted components	Components
		

32 bit adder consists of 1 `PG` layer, 5 `KSPG` and `KSG` layers, one last layer for one `KSG` for the carry out, and one `sum` layer to compute the sum bits.
Lastly, `z`, `v` and `n` are computed from the results of the adder.

What was done

All `AND`, `XOR` and `OR` gates were remade with inverting logic.
The most loaded signal `ALUFN0` is buffered.
Kogge-Stone architecture was chosen to allow for parallel prefix adding to speed up computation as compared to ripple carry adders.

Stats

Fastest timing of adder: 2.814ns
Circuit size: 1337 gates
Minimum observed setup =829.52ps

Part 2

We chose the 8th bit for easier computation.
A BC file is created from the circuit design, which is then converted into a CNF file by using the telegram submission bot.

The CNF file is fed into `findsolssat.jar` and UNSAT result is true.

```
C:\Users\chris\Downloads>java -jar findsolssat.jar Compareadders.cnf
Unsat : true

C:\Users\chris\Downloads>_
```