A close up of a logo

Description automatically generatedA close up of a sign

Description automatically generated

VLSI

Mini Project #1

Floating-Point Adder

# *Team Members Info:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Sec** | **BN** | **Email** |
| Sarah Mohamed Hossam | 1 | 30 | sarahelzayat@outlook.com |
| Ahmed Atta Abdallah | 1 | 6 | aatta1082001@gmail.com |
| Abdelrahman Hamdy |  |  |  |
| Abdelrahman Noaman | 2 | 2 | abdelrahmannoaman1@gmail.com |

**Adders Explanation:**

* **Floating Point Adder**

It sums the standard ieee format

First, I shift the mantissas if there is a difference in the exponents

Then I get the sum of the mantissas and Do normalization in case of the addition and subtraction.

I detect the overflow and underflow and round the results to ±INF or to Zero

* **Verilog (‘+’) version of adders**

Its just a simple adder with a + sign

* **Ripple Carry Adder**

It is constructed with full adders connected in sequence

* **Carry Save Adder**
* **Carry Look-Ahead Adder**
* **Carry Increment adder**
* Carry increment adder is divided into multiple sectors, the first is a regular ripple carry adder, the output sum of this adder is taken as it is.
* Any other sector consists of a ripple carry adder with a carry in of 0, half adders and the carry out of the previous block as an input.
* The first half adder adds the carry of the previous block and the first bit of the ripple carry adder’s sum, the sum of the half adder is takes as it is, and the carry out signal is taken as an input to the next half adder…etc.
* The final carry out signal of the block is the result of oring the ripple carry adder’s carry out and the carry out of the last half adder in the block.
* **Carry Skip adder**
* Carry skip adder consists of number of ripple carry adders with a special block which plays a role in speeding up the adding process.
* This block is called skip chain which makes Carry Skip Adder much faster than ripple carry adder especially in large number of bits, this block is repeated for every 4 bits.
* It aims to provide the propagation speed of the carry bit but how does it make that?
* First of all we calculate the group propagate signal  
  Pi = XOR(Ai, Bi) **(for i is from 0 to N/4-1)**  
  P = And(P0, P1, P2, P3, P4, P5, P6, P7) **(in case of 32 bit)**
* Then this signal is our selector so if it’s 1 then the carry in bit will propagate to the next ripple carry adder
* But if it’s 0 then the next ripple carry adders will need to wait for the carry out of the previous ripple adders.
* **Carry Bypass adder**
* Like the ripple carry adder each full adder waits for the carry of the previous one, Carry Bypass adder improves that in some conditions in which the next full adders won’t need to wait for the carry of the previous ones
* That cases are when the group propagate signal is set to one so the carry in will propagate to the next group of the full adders
* Pi = XOR(Ai, Bi) **(for i is from 0 to N/4-1)**  
  P = And(P0, P1, P2, P3, P4, P5, P6, P7) **(in case of 32 bit)**
* And this is repeated each 8 bits (for each 8 full adders)
* **Carry Select Adder**
* Carry select adder is divided into sectors, the adder calculates each sector’s sum and carry out for an input carry of 0 & 1 and selects the correct sum and carry out using a multiplexer with the selector as the actual input carry of the sector.
* As it does double the number of calculations for each sector, it consumes a large area and power but it’s very fast.

**Why choosing the carry lock ahead adder?**

There’s always a tradeoff between speed, power, and area.

In terms of speed, it has a moderate time performance (as shown in the excel sheet), not the fastest yet not the slowest.

In terms of power consumption, it has the 2nd best power consumption amongst all adders, the 1st is the Verilog’s + adder which is very slow.

In terms of area, it also has the 2nd best area next to the Verilog’s + adder.

So, we chose the excellent power consumption and area with a moderate time to implement the floating-point adder.

**Roles of each team member and estimated time they worked**

|  |  |  |
| --- | --- | --- |
| Name | Role | Estimated time |
| Sarah Mohamed Hossam | Carry increment adder and carry select adder | 2 days |
| Ahmed Atta Abdallah | Verilog’s + adder, carry ripple adder and floating-point adder | 2 days |
| Abdelrahman Hamdy | Carry lookahead adder and carry save adder |  |
| Abdelrahman Noaman | Carry skip adder and carry bypass adder | 1 day |