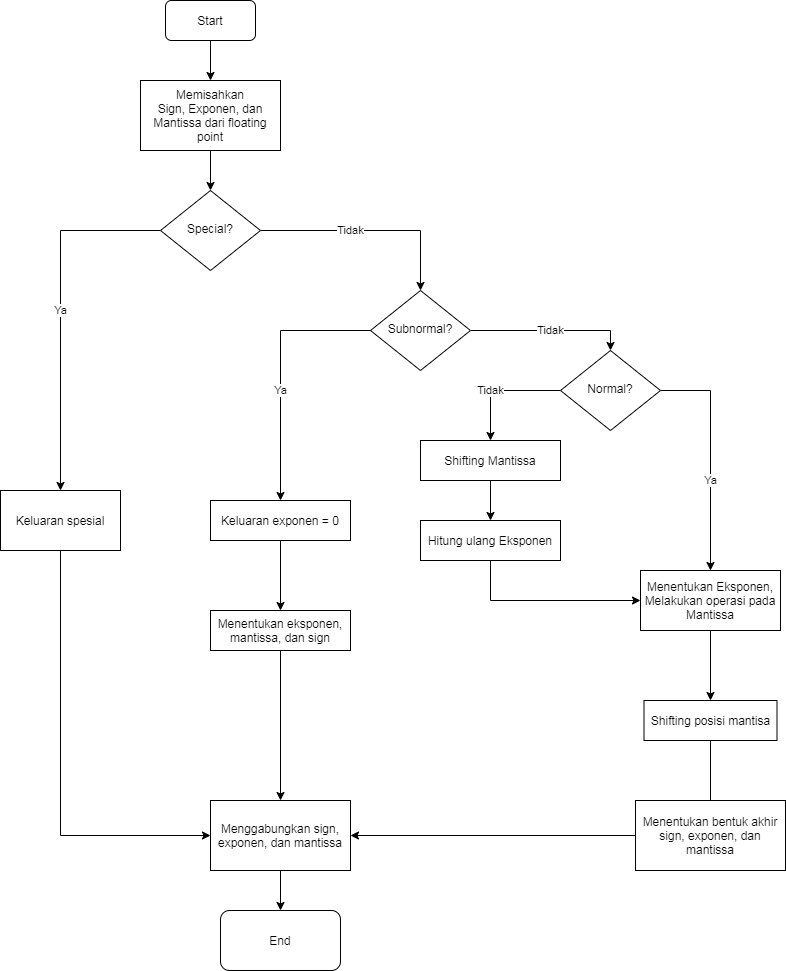
**Nama : Sidartha Prastya. P**

**NIM :13219033**

**Tugas Praktikum 3**

**EL2102 - Sistem Digital**

1. Pada praktikum ini ditugaskan untuk membuat penjumlah kombinasional half precision floating point sesuai dengan standar IEEE 754. Desain *(flowchart)* dari penjumlah ini dapat dilihat sebagai berikut.



Pola berpikir yang terdapat pada diagram alur tersebut adalah sebagai berikut.

- Pertama, floating point binary dipisahkan menjadi beberapa bagian, yaitu

1. Sign : polaritas dari floating point (bit 16).
2. Exponent : faktor pangkat pada floating point (bit 15 - 11).
3. Mantissa : angka-angka penting pada floating point (bit 10 - 1).

- Setelah mengelompokkan, program melihat apakah operasi floating point tersebut masuk ke dalam kasus spesial, yaitu:

1. Zero (nol) : sign, exponent, dan mantissa adalah nol.
2. Infinite : mantissa berjumlah 0 dan exponent berjumlah 31.
3. NaN : exponent berjumlah 31 dan mantissa memiliki nilai.

- Bila kasus spesial, maka program akan langsung mengeluarkan keluaran pasti, yaitu floating point itu sendiri untuk kasus Zero, infinite itu sendiri untuk kasus infinite, dan “1111110000000001” untuk kasus NaN.

- Bila bukan termasuk kasus spesial, maka kita melihat apakah floating point tersebut termasuk subnormal atau normal, yaitu:

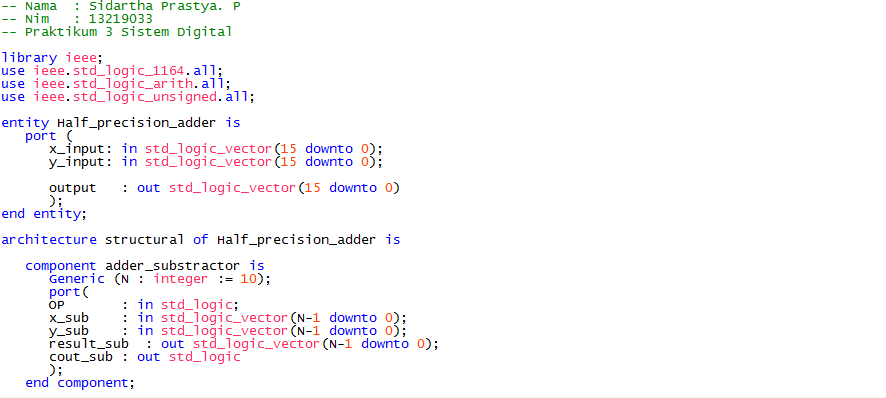
1. Subnormal : exponent bernilai 0 dan memiliki mantissa.
2. Normal : exponent di antara 0 dan 31 serta memiliki mantissa.

- Bila kasus tersebut subnormal untuk kedua input, exponent output dibuat menjadi 0 dan tinggal menghitung mantissa serta menentukan sign output. Namun, apabila masuk ke dalam kasus normal, maka perlu untuk membandingkan exponent lalu melakukan *shifting* pada mantissa sesuai dengan besar perbedaan kedua exponent input.

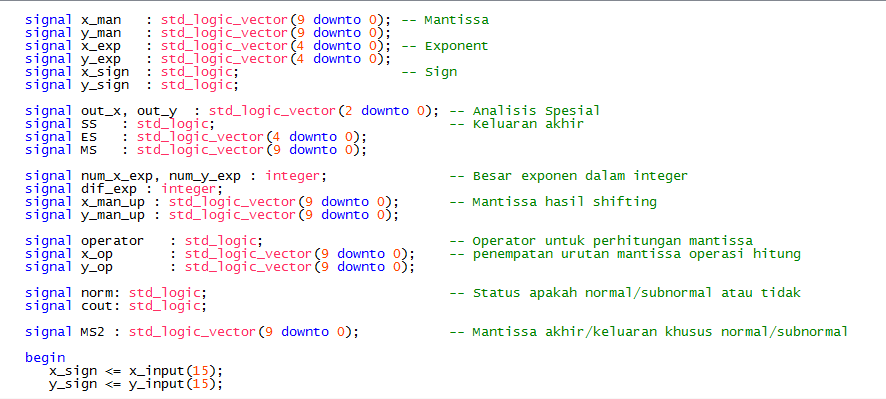
- Setelah melakukan shifting, lakukan operasi penjumlahan manitssa dengan adder. Kemudian, tentukan sign akhir apakah positif atau negatif.

- Semua nilai akhir yang didapat (sign, exponent, dan mantissa) digabungkan menjadi output akhir.

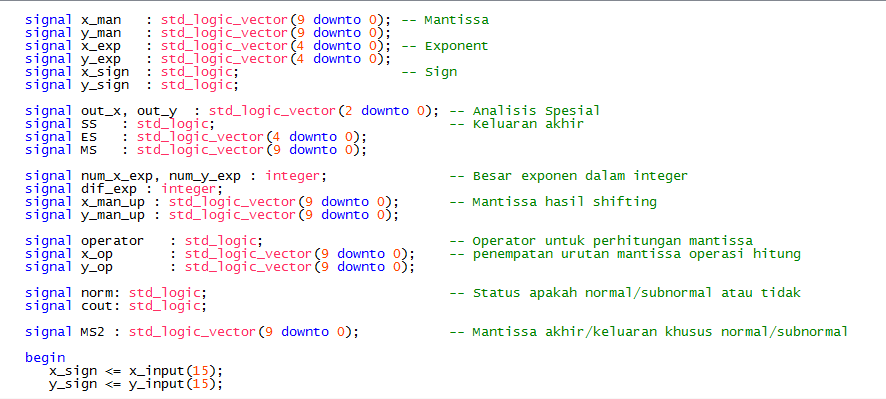
1. Berikut adalah desain VHDL dari penjumlah tersebut. Untuk VHDL code **Fulladder, Adder-nbit,** dan **Adder-substractor**, saya menggunakan program yang sama dengan praktikum 2 yang lalu.

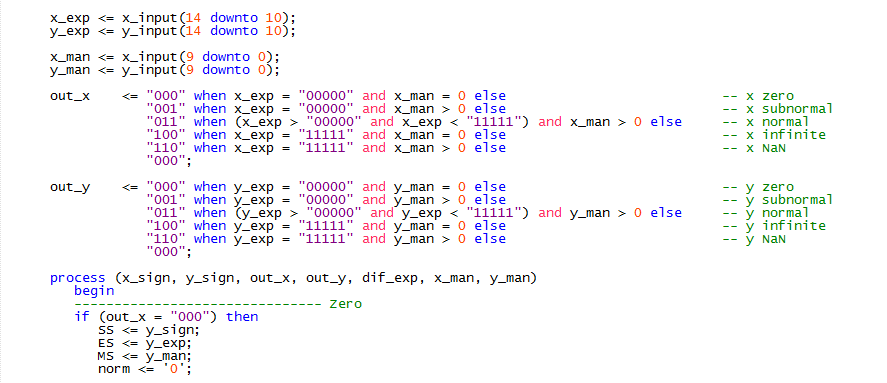


Di sini saya menggambarkan Half precision adder yang tentunya memiliki 16 bit terdiri dari dua input (x dan y) serta menghasilkan sebuah output. Saya memanggil komponen adder substractor untuk melakukan perhitungan pada mantissa.

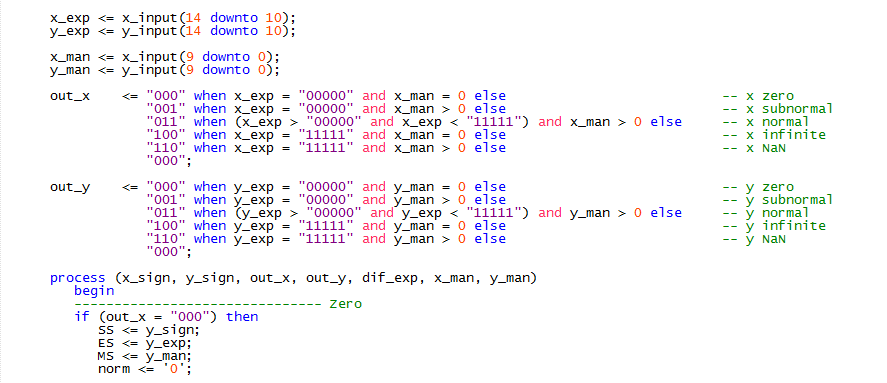


Lalu, ada beberapa signal yang dibutuhkan pada program ini, yaitu sign, exponent, dan sign untuk masing-masing input; SS, ES, MS untuk output akhir; norm, out\_x dan out\_y untuk mendeteksi kasus spesial; x\_man\_up, y\_man\_up untuk mantissa hasil shifting; operator, x\_op, y\_op, dan MS2 untuk operasi hitung floating point subnormal/normal.

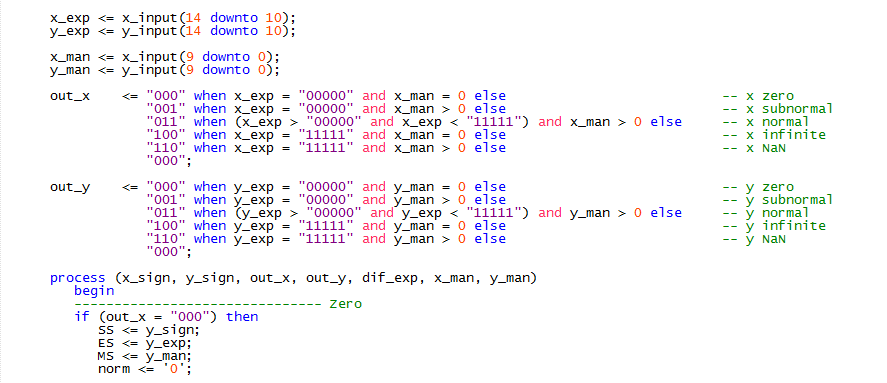


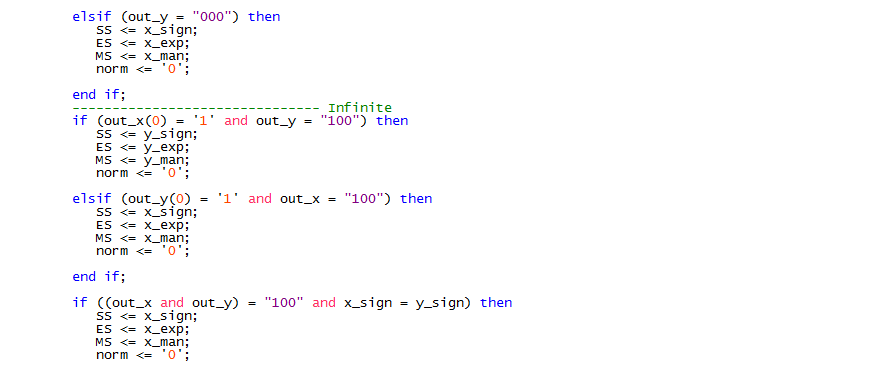


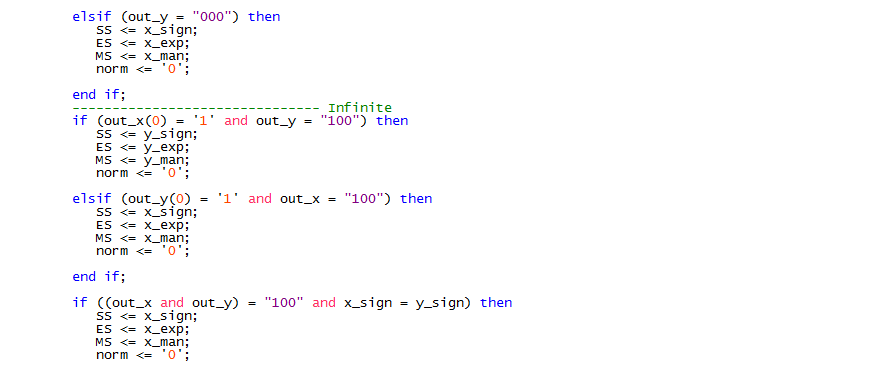
Setelah itu, dilakukan pemisahan floating number ke dalam sign, exponent, dan mantissa.

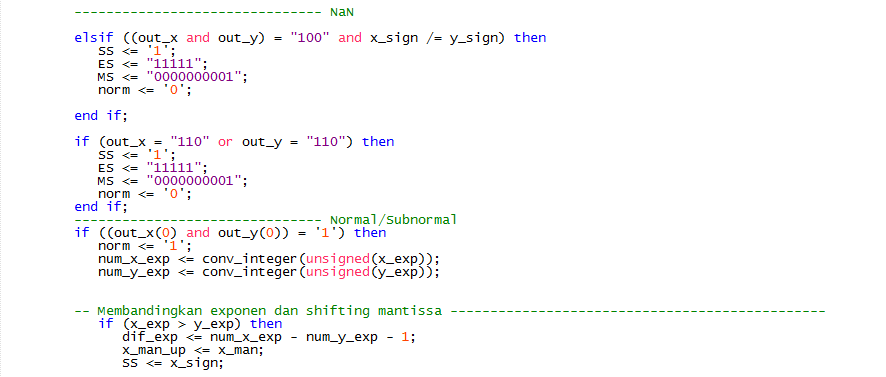


Lalu dilakukan inisiasi sifat/kasus spesial.

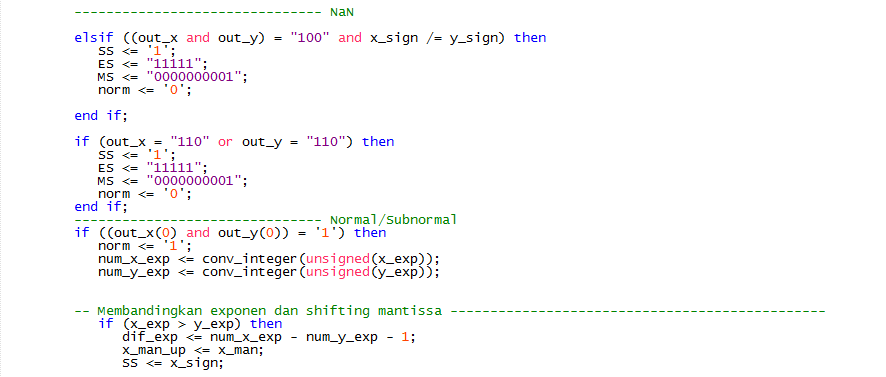


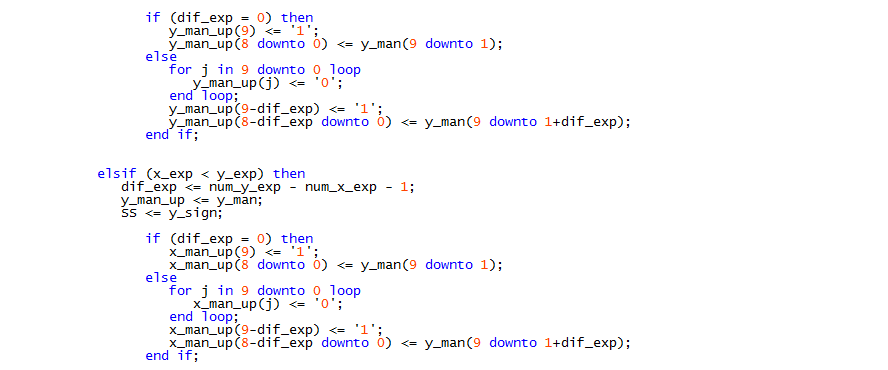


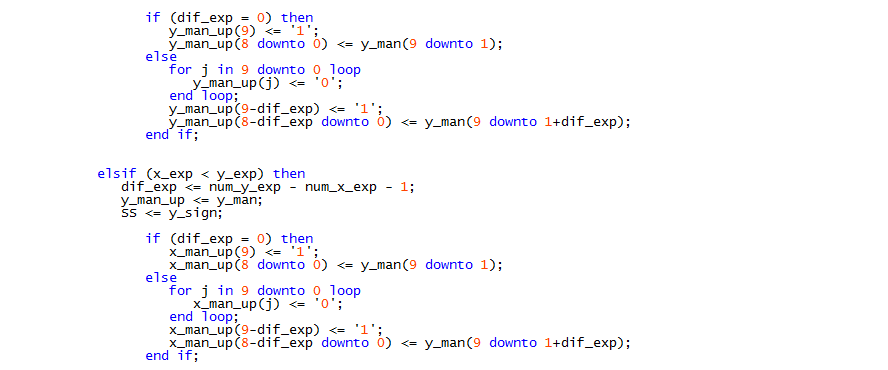


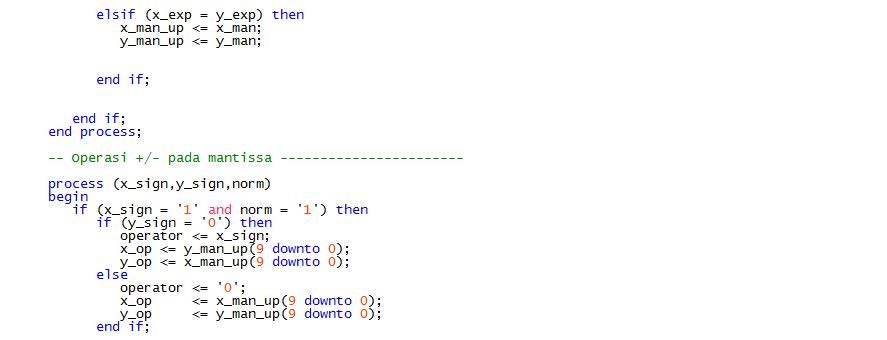


Dilakukan klasifikasi kasus spesial beserta keluarannya.

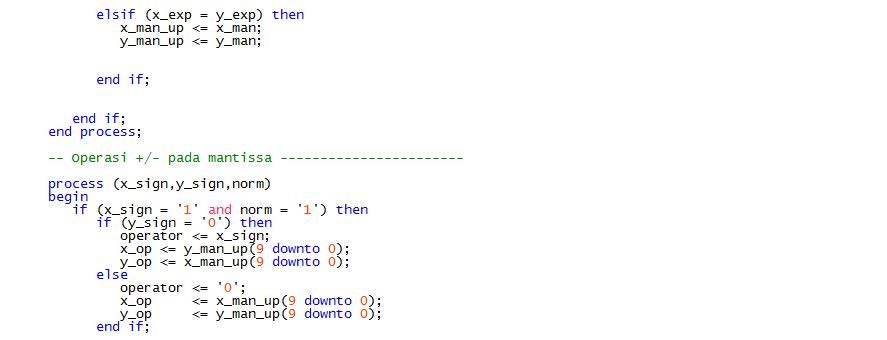


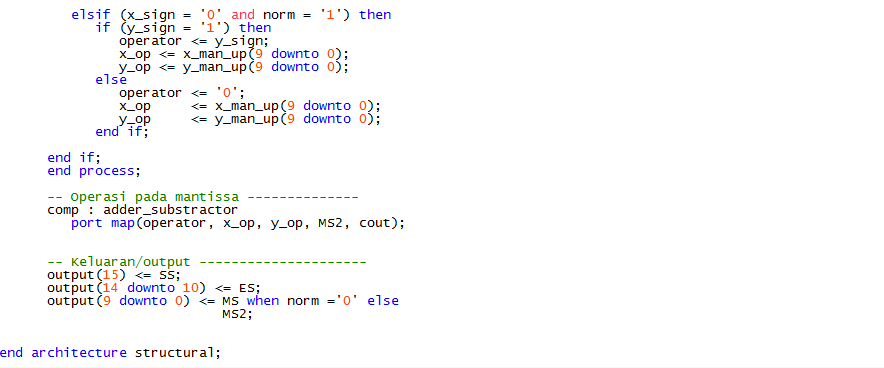




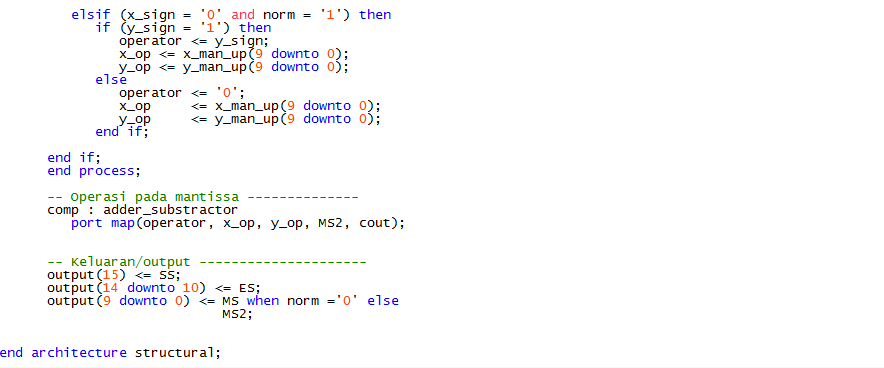


Apabila kasus normal/subnormal, dilakukan pembandingan besar exponent, lalu dilakukan shifting apabila nilai exponent berbeda.





Karena menggunakan adder substractor, mantissa yang akan dikurangi (dijadikan negatif) diletakkan pada urutan operasi kedua. Mantissa yang diubah menjadi negatif adalah mantissa pada floating point dengan sign 1 (negatif).



Terakhir, dilakukan operasi pada mantissa apabila normal, kemudian keluaran akhir ditaruh pada port output. Untuk output mantissa menggunakan MS apabila kasus spesial dan menggunakan MS 2 apabila kasus subnormal/normal.

Hasil *Run RTL* didapat sebagai berikut.

