BAB III
METODE PENELITIAN

Penyusunan penelitian ini didasarkan pada masalah yang bersifat teoritis aplikatif, yaitu perancangan sistem FPU berdasarkan dari sistem-sistem yang telah ada dan merancangnya ke dalam bahasa hardware yang kemudian di aplikasikan ke dalam FPGA. Sistem-sistem FPU yang telah ada merupakan data utama yang kemudian akan dianalisis dan dirancang kembali sesuai dengan spesifikasi yang ditentukan.

Langkah-langkah kerja yang perlu dilakukan untuk merealisasikan sistem ini adalah sebagai berikut:

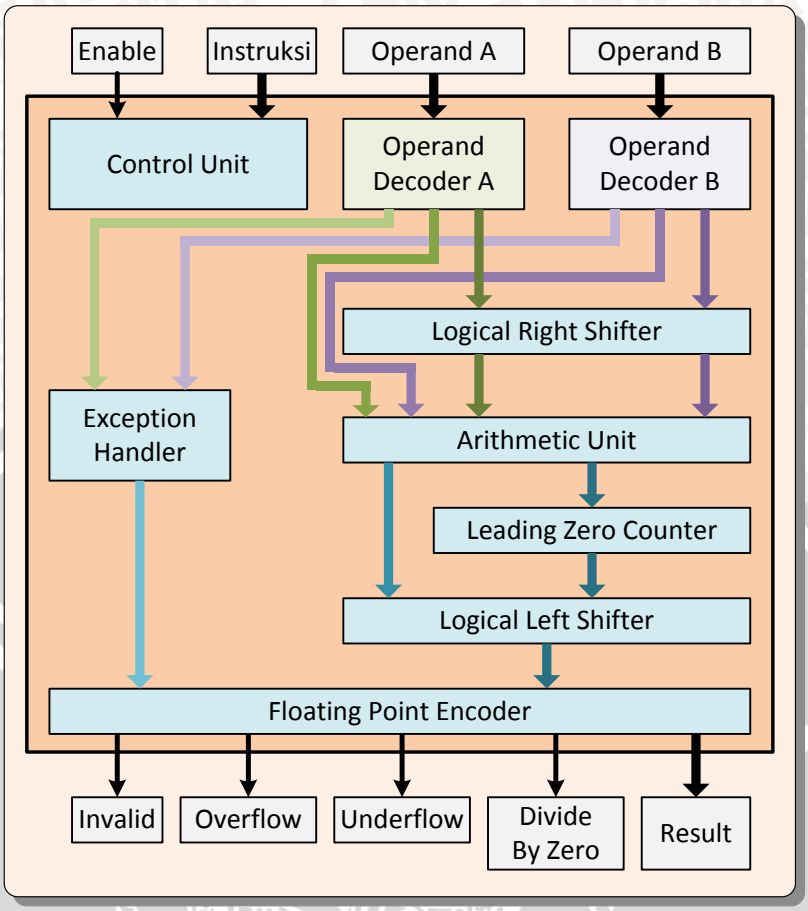
3.1 Penentuan Spesifikasi Rancangan

Sistem FPU yang akan dirancang memiliki spesifikasi sebagai berikut:

1) Memproses bilangan *floating point* biner standar IEEE 754 presisi tunggal (32 bit).
2) Mengerjakan operasi aritmatika penjumlahan, pengurangan, perkalian dan pembagian dua bilangan *floating point*.
3) Mengerjakan operasi aritmatika dari dua bilangan *floating point*.
4) Pembulatan bilangan floating *floating point* point yang digunakan adalah pembulatan terhadap 0 atau disebut juga dengan pemotongan.
5) Menangani bilangan khusus dan operasi-operasi yang membutuhkan perhitungan khusus.
6) Sistem ini dirancang dalam FPGA XILINX tipe SPARTAN XC3S500E.

3.2 Perancangan Sistem

Sistem *Floating Point Unit* (FPU) yang akan dirancang memiliki blok diagram yang ditunjukkan dalam Gambar 3.1.



Gambar 3.1. Diagram blok perancangan FPU.

Operand A dan *operand B* adalah bilangan *floating point* presisi tunggal (32 bit) dalam bentuk standar IEEE 754-2008 yang akan dihitung. Instruksi adalah masukan FPU yang menentukan operasi apa yang akan dilakukan untuk kedua masukan *operand*. Dan *Enable* adalah sinyal trigger untuk mengaktifkan FPU agar memulai proses perhitungan.

Operand decoder A dan *operand decoder B* adalah blok yang berfungsi untuk medekoderkan *floating point* presisi tunggal (32 bit) dalam bentuk standar IEEE 754-2008 menjadi pecahan informasi berupa eksponen (8 bit), mantisa (24 bit) dan *flag* (*Sign, Not a Number, Infinity,* dan *Zero*). *Logical right shifter* adalah blok yang berfungsi menggeser deret bit ke kanan atau ke arah bit yang bernilai lebih kecil, blok ini diperlukan ketika FPU memproses perhitungan penjumlahan dan pengurangan. *Arithmetic Unit* adalah blok yang berfungsi menangani operasi penjumlahan, pengurangan, pembagian, dan perkalian mantissa maupun eksponen. *Leading zero counter* adalah blok yang berfungsi untuk menghitung

jumlah nol yang mendahuli angka 1 pertama dalam suatu bilangan biner, blok ini diperlukan untuk proses normalisasi. *Logical left shifter* adalah blok yang berfungsi untuk menggeser deret bit ke kiri atau kearah bit yang bernilai lebih besar. *Exception Handler* adalah blok yang berfungsi untuk menangani perhitungan-perhitungan khusus, blok ini bekerja berdasarkan masukan *flag status* kedua *operand*. *Floating point encoder* berfungsi untuk mengemas atan mengenkoderkan hasil proses FPU sesuai standar bilangan *floating point* presisi tunggal (32 bit) IEEE 754-2008

Keluaran FPU yang terdiri dari *Invalid, overflow, underflow,* dan *divide by zero* adalah keluaran yang akan aktif ketika terdapat operasi-operasi khusus (*exception*), sinyal ini diberikan guna membantu unit luar dalam mendeteksi hasil perhitungan khusus. *Result* adalah keluaran FPU hasil perhitungan yang diinginkan, keluaran ini berupa standar bilangan *floating point* presisi tunggal (32 bit) dalam bentuk standar IEEE 754-2008. *Status* adalah keluaran FPU untuk menandai apakah proses perhitungan yang dilakukan FPU telah selesai atau belum.

3.3 Pengujian

Pengujian dilakukan bersamaan dengan penulisan bahasa *hardware* ke dalam modul FPGA, hal ini dilakukan untuk memberikan hasil rancangan mendekati dengan sistem yang telah direncanakan dan menanggulangi keterbatasan modul FPGA dalam penerapan rancangan yang telah dibuat.

3.3.1 Pengujian Perblok

Pengujian per-blok dilakukan untuk meminimalisir error yang mungkin terjadi, yang pada umumnya disebabkan oleh kesalahan penulisan bahasa *hardware* atau kesalahan logika pemikiran. Pengujian perblok yang dilakukan antara lain:

1) Pengujian *Arithmetic Unit*

Pengujian ini dilakukan guna mengetahui apakah *Arithmetic Unit* dapat menerima masukan yang diberikan berupa eksponen A, eksponen B, mantissa A dan mantissa B. dan mampu memberikan keluaran sesuai dengan fungsinya. Keluaran *Arithmetic Unit* antara lain eksponen terbesar,

selisih eksponen, hasil jumlah atau kurang kedua eksponen, hasil jumlah atau kurang kedua mantisa, perkalian mantisa, pembagian mantisa dan eksponen tambahan hasil pembagian mantisa. Dan keluarannya akan diproses lebih lanjut oleh *Control Unit.*

2) Pengujian *Operand Decoder*

Pengujian ini dilakukan untuk mengetahui apakah *operand decoder* dapat menghasilkan keluaran berupa eksponen, mantissa dan *flag* (*sign, zero, Not a number,* dan *infinity*) sesuai dengan yang diharapkan ketika diberi masukan berupa operand.

4) Pengujian *Logical Right Shifter*

Pengujian ini dilakukan untuk mengetahui apakah sistem *Logical Right Shifter* yang berfungsi untuk perhitungan penjumlahan dan pengurangan bilangan *floating point*, dapat bekerja sesuai dengan spesifikasi dan fungsi yang dirancang. *Logical Right Shifter* harus dapat melakukan pergeseran bit sebanyak yang diinginkan kearah bit yang bernilai lebih kecil (LSB).

3) Pengujian *Logical Left Shifter*

Pengujian ini dilakukan untuk mengetahu apakah sistem *Logical left shifter* yang berfungsi untuk proses normalisasi hasil perhitungan, dapat bekerja sesuai dengan spesifikasi dan fungsi yang dirancang. *Logical Left Shifter* harus dapat melakukan pergeseran bit sebanyak yang diinginkan kearah bit yang bernilai lebih besar (MSB).

4) Pengujian *Leading Zero Counter*

Pengujian ini dilakukan untuk mengetahu apakah sistem *leading zero counter* yang berfungsi untuk proses normalisasi hasil perhitungan, dapat bekerja sesuai dengan spesifikasi dan fungsi yang dirancang. *Leading zero counter* harus dapat melakukan perhitungan jumlah 0 yang mendahului angka 1 pertama dari nilai masukan.

5) Pengujian *Exception Handler*

Pengujian ini dilakukan untuk mengetahui apakah *exception handler* berjalan sesuai dengan spesifikasi dan fungsi yang telah ditentukan, yaitu harus dapat memberikan keluaran jika terdapat perhitungan-perhitungan yang membutuhkan penanganan khusus. *Exception handler* bekerja berdasarkan masukan sinyal *flag* (*sign, Not a Number, Zero, Infinity*) dan mode operasi (perkalian, pembagian, penjumlahan, dan pengurangan).

6) Pengujian *Floating Point Encoder*

Pengujian ini dilakukan untuk mengetahui apakah *floating point encoder* berjalan sesuai dengan spesifikasi dan fungsi yang telah ditentukan, yaitu harus dapat mengemas hasil perhitungan FPU menjadi format standar IEEE 754 untuk bilangan *floating point* presisi tunggal, dan jika *exception handler* memberi sinyal peringatan perhitungan khusus maka *floating point encoder* harus dapat memberikan keluaran sesuai dengan perintah *exception handler*.

7) Pengujian *Unit Input Output*

Unit input output berfungsi untuk mengirim data dari pengguna ke FPGA dan mengeluarkan data dari FPGA. Karena itu meskipun unit input output bukan bagian dari FPU yang dirancang, pengujian terhadap unit input output penting dilakukan untuk mengetahui apakah sistem input output dapat memberikan masukan dan menampilkan keluaran sesuai dengan spesifikasi dan fungsi yang dirancang.

3.3.1 Pengujian Keseluruhan Sistem

Apabila pengujian tiap blok telah terlaksana, maka pengujian keseluruhan sistem akan dilakukan guna menguji apakah sistem yang dirancang telah sesuai dengan spesifikasi yang telah ditentukan.

Untuk menguji keseluruhan sistem, dibutuhkan penggabungan blok-blok yang telah dirancang kemudian membuat sistem *control* yang mengendalikan masukan dan keluaran data semua blok yang membentuk

FPU. Karena itu pengujian keseluruhan identik dengan pengujian *control unit*.

Pengujian keseluruhan dilakukan dengan memberikan masukan-masukan berupa dua buah bilangan *floating point* dan jenis operasi yang akan dilakukan kepada sistem FPU dan melihat apakah hasil perhitungan yang telah dilakukan oleh sistem FPU sesuai dengan perhitungan secara manual.