

BAB VI

KESIMPULAN DAN SARAN

6.1 KESIMPULAN

Berdasarkan pengujian dan analisis keseluruhan yang telah dilakukan pada Bab V, dapat ditarik kesimpulan sebagai berikut:

1. Sistem FPU dirancang dari susunan blok-blok pemrosesan logika yang didesign khusus untuk setiap sifat dan fungsinya. Untuk mengoptimasi penggunaan *slice* pada FPGA, setiap blok pemrosesan logika memiliki *bus data* yang telah disesuaikan dengan kebutuhan FPU. Total penggunaan *slice* FPGA XILINX tipe SPARTAN XC3S500E adalah sebanyak 40% dari total 4,656 *slice* yang tersedia.
2. Sistem FPU dirancang dengan memiliki sinyal masukan *enable* sebagai tanda agar FPU memulai proses perhitungan. Sistem FPU ini juga memiliki sinyal keluaran *status* sebagai tanda proses perhitungan yang dilakukan telah selesai. Adanya kedua sinyal ini bertujuan untuk membantu unit luar yang memanfaatkan sistem FPU ini sebagai pendukung kinerja (*co-processor*) dalam melakukan perhitungan bilangan *floating point*.
3. Blok-blok pemrosesan logika yang menyusun FPU, dirancang khusus bekerja secara paralel dan tidak membutuhkan *clock*. Blok-blok tersebut diatur sedemikian rupa agar bekerja sekali untuk setiap pemrosesan yang dilakukan FPU. Sehingga *Control unit* "hanya" mengatur aliran data yang masuk dan keluar pada setiap blok. Dan kinerja FPU bergantung pada *propagation delay* sistem.
4. Sistem FPU yang diimplementasikan pada FPGA dapat melakukan empat perhitungan dasar dari dua buah *operand* berupa bilangan *floating point* presisi tunggal standar IEEE 754. Perhitungan yang dapat dilakukan yaitu penjumlahan, pengurangan, perkalian dan pembagian. Sistem FPU juga dapat menangani perhitungan-perhitungan khusus yang tidak bisa diselesaikan oleh hardware perhitungan standar.
5. Dari proses *synthesize* yang dilakukan menggunakan *software* XILINX ISE Project Navigator 11.1, besar *delay* total adalah 233.689ns, artinya FPU dapat bekerja pada frekuensi maksimal 4279191Hz.

6.2 SARAN

Saran yang dapat diberikan dari penelitian perancangan sistem FPU ini antara lain:

1. Penambahan proses aritmatika yang dapat dikerjakan oleh FPU, misalnya akar kuadrat, Logaritma, dll.
2. Penambahan converter bilangan integer ke bilangan *floating point* dan sebaliknya.
3. Perbaiki sistem aritmatika agar kecepatan perhitungan FPU dapat lebih tinggi.

