

## TIẾN ĐỘ HÀNG TUẦN

Tuần	Buổi	Nội dung công việc	Tiến độ
1	1	Viết RTL bộ giải mã lệnh RV32I. Đọc lệnh 32bit và giải mã ra thành nhiều trường như funct3, funct7, immediate, rs1, rs2, rd, opcode.	Đã hoàn thành code. Đã mô phỏng thử nghiệm đầy đủ chức năng
		Viết RTL bộ tính toán số học ALU. Thực hiện việc tính toán các lệnh số học ADD, SUB, AND, OR, XOR, Shift,... Đồng thời có chức năng so sánh đưa ra kết quả cho nhóm lệnh Branch.	Đã hoàn thành cơ bản. Chưa mô phỏng thử nghiệm các chức năng.
	2	Hoàn thiện chức năng đơn vị ALU. Test chức năng.	Đã hoàn thành. Đã mô phỏng thử nghiệm đầy đủ chức năng.
		Viết mã RTL bộ nhớ lệnh và bộ nhớ dữ liệu.	Đã hoàn thành. Chưa mô phỏng thử nghiệm các chức năng.
2	1	Hoàn thiện mã bộ nhớ lệnh và bộ nhớ dữ liệu. Viết mã RTL cho khối 32 thanh ghi x0 – x31	Đã hoàn thành. Đã mô phỏng thử nghiệm đầy đủ chức năng.
		Viết mã thực thi lệnh cho CPU (5 state) (chưa có pipeline). Tạo firmware đơn giản để có được file hex nạp vào bộ nhớ lệnh.	Đã hoàn thành cơ bản. Đang mô phỏng và sửa lỗi.
	2	Sửa lỗi hoạt động của CPU thực hiện chưa chính xác. Sửa lỗi các trường hợp mã chương trình liên quan tới Stack Pointer do Firmwave chưa cấu hình sections.lds(linker script) phân rõ cho bộ nhớ lệnh và dữ liệu.	Đã hoàn thành. Đã mô phỏng thử nghiệm. Chưa sửa được lỗi Stack Pointer.
		Ghép ngoại vi UART vào thử nghiệm. Chỉnh sửa sections.lds, start.S, Makefile.	Đã hoàn thành ghép ngoại vi Còn lỗi chưa thành công.
3	1	Sửa lại linker script, MakeFile để tạo ra 2 mã hex cho bộ nhớ lệnh và bộ nhớ dữ liệu riêng. Sửa lại các lỗi trong CPU	Đã hoàn thành. Đã mô phỏng đầy đủ chức năng.
		Nạp code vào FPGA để chạy thử. Quan sát Uart bằng hercules.	Đã chạy thử và đúng hoàn toàn.

	2	Thực hiện viết RTL chế độ hoạt động đơn giản thứ hai nhưng nhanh hơn chế độ thứ nhất. Chế độ thứ nhất sử dụng 4-5 chu kì cho 1 lệnh. Chế độ thứ hai chỉ sử dụng 3 chu kì.	Đã hoàn thành. Đã mô phỏng chạy thử. Đã nạp vào kit FPGA chạy thử và thành công.
4	1	Thực hiện viết RTL chế độ hoạt động pipeline 5 chu kì. Tạo bộ dự đoán rẽ nhánh động(bộ dự đoán 2-bit)...	Hoàn thành module dự đoán rẽ nhánh.
	2	Tìm cách xử lý xung đột dữ liệu (Data Hazard).	Chưa hoàn thành.
5	1	Nghỉ	
	2	Xử lý hazard cho dữ liệu sử dụng của các lệnh ngoại trừ LOAD.	Đã hoàn thành
6	1	Xử lý tất cả Hazard còn lại liên quan tới LOAD và STORE.	Đã hoàn thành.
	2	Xử lý lỗi CPU không đồng bộ với ngoại vi do đọc ghi quá nhanh. Xử lý lỗi khi dự đoán rẽ nhánh sai, lỗi hazard đặc biệt với lệnh JALR. Hoàn thành CPU pipeline.	Đã hoàn thành
7	1	Bổ sung các thanh ghi CSR, xử lý lệnh system.	Đã hoàn thành
	2	Triển khai Bus AHB và APB kết nối các slave.	Chưa hoàn thành
8	1	Hoàn thành bus AHB và APB	Đã hoàn thành

	2	Tối ưu bộ dự đoán rẽ nhánh, tiết kiệm 500LUT, 400reg	Đã hoàn thành
--	---	---	---------------