

Diseño e implementación de una CPU con el ISA básico de RISC-V: RV32I

Gamino del Río, Iván

González Escobar, Ángel

Jiménez Arribas, Miguel

Especificaciones

Antes de comenzar, cabe mencionar que estas especificaciones son provisionales y están sujetas a cambios propuestos por el tribunal, o por falta de tiempo durante la realización del proyecto.

- Implementación en VHDL de todas las instrucciones del ISA RV32I establecidas en el siguiente apartado. Específicamente:
 - Operaciones inmediatas.
 - Operaciones entre registros.
 - NOP.
 - Saltos incondicionales.
 - Saltos condicionales.
 - Instrucciones de lectura y escritura en memoria.
- Implementación de los 32 registros de propósito general y el contador de programa (PC).
- Implementación de una unidad de control que decodifique las instrucciones implementadas y se comuniquen con la ALU configurando su ejecución.
- Implementación de una pequeña memoria RAM a partir de la cual se puedan realizar las baterías de pruebas (testbench).

Descripción

Este proyecto consiste en la realización de una CPU con el ISA RISC-V más sencillo: RV32I[1]. Para mayor simplicidad, incluso se eliminarán algunas instrucciones más, tal y como se indica en la propia especificación (capítulo 2, página 9), quedando solamente las que pueden verse en la [figura inferior](#).

31	27	26	22	21	17	16	15	14	12	11	10	9	8	7	6	0	
jump target																opcode	J-type
rd	LUI-immediate															opcode	LUI-type
rd	rs1	imm[11:7]	imm[6:0]				funct3				opcode	I-type					
imm[11:7]	rs1	rs2	imm[6:0]				funct3				opcode	B-type					
rd	rs1	rs2	funct10								opcode	R-type					
rd	rs1	rs2	rs3				funct5				opcode	R4-type					

Control Transfer Instructions

imm25										1100111	J imm25	
imm25										1101111	JAL imm25	
imm12hi	rs1	rs2	imm12lo				000				1100011	BEQ rs1,rs2,imm12
imm12hi	rs1	rs2	imm12lo				001				1100011	BNE rs1,rs2,imm12
imm12hi	rs1	rs2	imm12lo				100				1100011	BLT rs1,rs2,imm12
imm12hi	rs1	rs2	imm12lo				101				1100011	BGE rs1,rs2,imm12
imm12hi	rs1	rs2	imm12lo				110				1100011	BLTU rs1,rs2,imm12
imm12hi	rs1	rs2	imm12lo				111				1100011	BGEU rs1,rs2,imm12
rd	rs1	imm12				000				1101011	JALR.C rd,rs1,imm12	
rd	rs1	imm12				001				1101011	JALR.R rd,rs1,imm12	
rd	rs1	imm12				010				1101011	JALR.J rd,rs1,imm12	

Memory Instructions

rd	rs1	imm12				010				0000011	LW rd,rs1,imm12	
imm12hi	rs1	rs2	imm12lo				010				0100011	SW rs1,rs2,imm12

Integer Compute Instructions

rd	rs1	imm12				000				0010011	ADDI rd,rs1,imm12	
rd	rs1	000000				shamt				001	0010011	SLLI rd,rs1,shamt
rd	rs1	imm12				010				0010011	SLTI rd,rs1,imm12	
rd	rs1	imm12				011				0010011	SLTIU rd,rs1,imm12	
rd	rs1	imm12				100				0010011	XORI rd,rs1,imm12	
rd	rs1	000000				shamt				101	0010011	SRLI rd,rs1,shamt
rd	rs1	imm12				110				0010011	ORI rd,rs1,imm12	
rd	rs1	imm12				111				0010011	ANDI rd,rs1,imm12	
rd	rs1	rs2	0000000				000				0110011	ADD rd,rs1,rs2
rd	rs1	rs2	1000000				000				0110011	SUB rd,rs1,rs2
rd	rs1	rs2	0000000				001				0110011	SLL rd,rs1,rs2
rd	rs1	rs2	0000000				010				0110011	SLT rd,rs1,rs2
rd	rs1	rs2	0000000				011				0110011	SLTU rd,rs1,rs2
rd	rs1	rs2	0000000				100				0110011	XOR rd,rs1,rs2
rd	rs1	rs2	0000000				101				0110011	SRL rd,rs1,rs2
rd	rs1	rs2	0000000				110				0110011	OR rd,rs1,rs2
rd	rs1	rs2	0000000				111				0110011	AND rd,rs1,rs2
rd	imm20										0110111	LUI rd,imm20

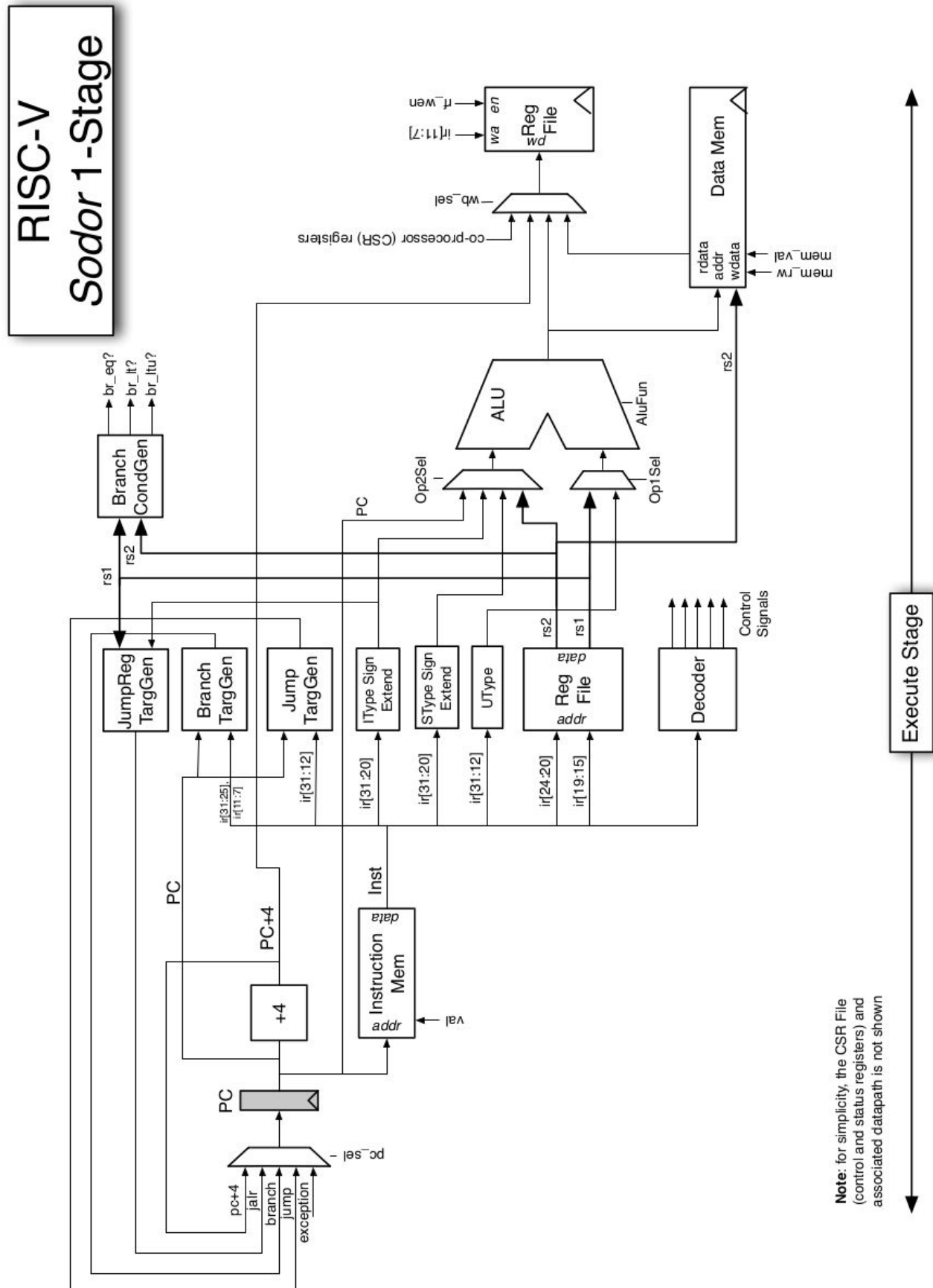
Para la realización de estas instrucciones nos basaremos en la ALU que hemos implementado durante las sesiones de laboratorio, completando el diseño con todas las

instrucciones que nos dé tiempo a implementar.

Para la realización de los registros nos basaremos en biestables tipo D, con registros de desplazamiento de entrada y salida paralela, basándonos en los diseños implementados durante las prácticas.

Para la realización de la unidad de control y de la RAM nos ayudaremos de las implementaciones de RISC-V ya realizadas en VHDL, dos de las cuáles pueden verse en las referencias.

En definitiva, salvo algunas simplificaciones, el diagrama (obtenido de [aquí](#)) de lo que queremos hacer básicamente sería:



Referencias documentales

1. [User-Level ISA Specification v2.1](#) - Capítulo 2.
2. <https://github.com/maikmerten/riscv-tomthumb> - Implementación de un core RV32I en VHDL para una placa Altera.
3. <https://github.com/inforichland/freezing-spice> - Implementación de un core RV32I con pipeline en VHDL.