

Índex

1	Introducció	6
1.1	Descripció del projecte	6
1.1.1	Risc-V (que és)	6
1.1.2	Estat del Art	6
1.1.3	Finalitat del projecte	6
2	Core Risc-V	6
2.1	Instruction set	6
2.1.1	RV32I	6
2.2	Arquitectura	6
2.2.1	Unitats i blocs arquitecturals	6
2.3	Senyals de Control	6
2.4	Unitat de Control	6
2.5	Unitat Aritmetico-Lògica	6
2.5.1	Funcionament	6
2.5.2	Implementació	6
2.6	Load	6
2.6.1	Funcionament	6
2.6.2	Implementació	6
2.7	Store	6
2.7.1	Funcionament	6
2.7.2	Implementació	6
2.8	Fetch and Jumps	6
2.8.1	Funcionament	6
2.8.2	Implementació	6

2.9	CSR	6
2.9.1	Funcionament	6
2.9.2	Implementació	6
2.10	Nucli RISC-V complet	6
2.10.1	Arquitectura	6
2.10.2	Memòria de Programa	6
2.10.3	Memòria de Dades	6
3	Core Risc-V adaptat a memòries arbitrades	6
3.1	Plataforma Pulpino	6
3.2	Modificacions necessàries	6
3.2.1	Senyals	6
3.2.2	Estats	6
3.2.3	Stall Core	6
4	Eines Utilitzades	6
4.1	Eines de Programació	6
4.1.1	Interpret de Verilog	6
4.1.2	Visual Studio Code	6
4.1.3	Docker	6
4.1.4	RiscV Gnu Toolchain	6
4.1.5	Git	6
4.1.6	Continus integration	6
4.2	Eines de síntesi per a FPGA	6
4.2.1	Quartus II	6
4.2.2	Mega Wizard	6
4.3	Eines de validació	6

4.3.1	Tests amb Verilog	6
5	Resultats dels Tests	6
5.1	Aritmeticològic	6
5.2	Branch	6
5.3	CSR	6
5.4	Load-Store	6
6	Sintetitzat en FPGA	6
6.1	Embolcall FPGA	6
6.1.1	Memòries	6
6.1.2	PLL	6
6.1.3	Debounce	6
6.1.4	Display 7 segments	6
6.2	Programació FPGA	6
6.2.1	Síntesi	6
6.2.2	Flash	6
6.3	Funcionament	6
7	Resultats, conclusions i treball de futur	6
7.1	Resum de resultats (desenvolupament i test)	6
7.2	Conclusions	6
7.3	Treball futur	6
	Referències	6