Processador MAGIC

Alunos:

Miller Raycell Rodrigo de Andrade



Introdução

O seguinte projeto final trata da criação de um processador uniciclo de 16 bits, capaz de realizar operações lógicas e aritméticas, de decisão, e saltos na memória.



Conjunto de Instruções

 Tipo R: Operações aritméticas.

4 bits	3 bits	3 bits	3 bits	3 bits
15-12	11-9	8 - 6	5 - 3	2 - 0
Opcode	RD	RS	RT	Funct

Tipo I: Load, Store, Beq e Bne.

4 bits	3 bits	3 bits	6 bits
15-12	11-9	8 - 6	5 - 0
Opcode	RS	RT	Valor

Tipo J: Desvios incondicionais.

4 bits	12 bits
15-12	11 - 0
Opcode	Valor do salto

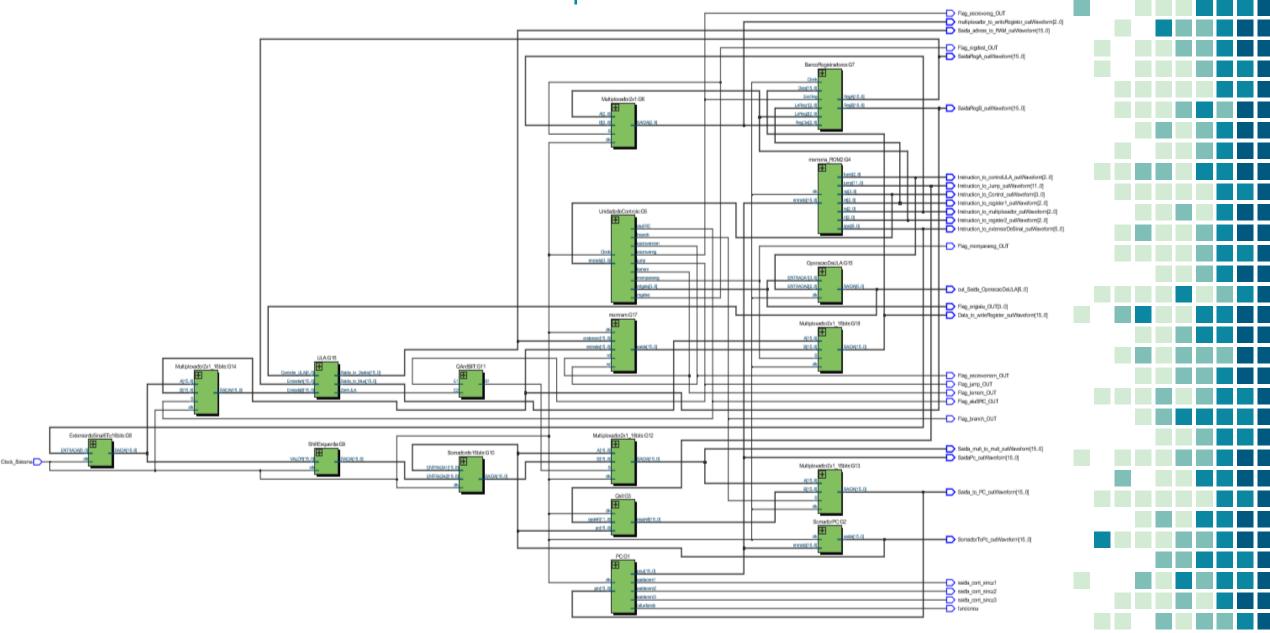
OPCODE: Código de Operação.

RS: Registrador contendo o primeiro operando.

RT: Segundo registrador.

RD: Registrador de destino.

Datapath



Exemplos

Evernolo 1, Somo de deic voleres

Exemplo 1: Soma de dois valores												
		0 ps	U ps									
<u>in</u>	Clock_Sistema	B 1										
**	> SaidaPc_outWaveform	в 00000000000000					00000000000000					
**	> SomadorToPc_outWaveform	в 00000000000000					00000000000000					
**	> Saida_to_PC_outWaveform	в 00000000000000					000000000000000					
out	Flag_aluSRC_OUT	B 0										
out	Flag_branch_OUT	B 0										
out	Flag_escrevemem_OUT	B 0										
out	Flag_escrevereg_OUT	B 1									4	
out	Flag_jump_OUT	B 0									4-4	
out	Flag_lemem_OUT	B 0										
out	Flag_memparareg_OUT	B 1										
**	> Flag_origialu_OUT	B 0000					0000					
out	Flag_regdest_OUT	B 1										
**	> Instruction_to_Control_outWaveform	B 0000					0000				illi	ĦΤ
**	> Instruction_to_controlULA_outWaveform	B 000					000					
**	> Instruction_to_extensorDeSinal_outWaveform	B 000000					000000					
**	> Instruction_to_Jump_outWaveform	B 00000000000					00000000000					
**	> Instruction_to_multiplexador_outWaveform	B 100	(OCX				000					
**	> Instruction_to_register1_outWaveform	B 000					000					
**	> Instruction_to_register2_outWaveform		21:X				000					
**	> multiplexador_to_writeRegister_outWaveform	B 100	(OCX				000					
**	out_Saida_OperacaoDaULA	B 0000000					0000000					
**	> Saida_mult_to_mult_outWaveform	В 00000000000000					000000000000000					
**	> SaidaRegA_outWaveform	U 13824					13824					
**	> SaidaRegB_outWaveform	U 11992	(95)				13824					
**	> Saida_ula	U 25816					25816					
**	dado_register_destino	U 25816	KIIIIIIII				25816					
**	Data_to_writeRegister_outWaveform	U 25816	KIIIIII				25816					
**	> Saida_adress_to_RAM_outWaveform	U 25816	KIIIIIIII				25816					

Exemplos

Exemplo 2: subtração de dois valores

			-	-	-			1 1		-					
-	_	B 1											\rightarrow	\rightarrow	_
>	SaidaPc_outWaveform	B 00000000													_
>	SomadorToPc_outWaveform	B 00000000													_
>	Saida_to_PC_outWaveform	B 00000000													
ut	Flag_aluSRC_OUT	B 0								-	-				
ut	Flag_branch_OUT	B 0													
ut	Flag_escrevemem_OUT	B 0													_
ut	Flag_escrevereg_OUT	B 1													$\overline{}$
ut	Flag_jump_OUT	B 0													_
ut	Flag_lemem_OUT	B 0													
ut	Flag_memparareg_OUT	B 1													_
*	Flag_origialu_OUT	B 0000													=
ut	Flag_regdest_OUT	B 1													_
3	Instruction_to_Control_outWaveform	B 0000													_
5	Instruction_to_controlULA_outWaveform	B 001	001	X											
= >	Instruction_to_extensorDeSinal_outWaveform	B 000000													
>	Instruction_to_Jump_outWaveform	B 00000000													=
5	Instruction_to_multiplexador_outWaveform	B 101	101	X											=
>	Instruction_to_register1_outWaveform	B 010	010	X											=
= >	Instruction_to_register2_outWaveform	B 011	011	X											_
= >	multiplexador_to_writeRegister_outWaveform	B 101	101	X											
3	out_Saida_OperacaoDaULA	B 0000001	0000001	X											
<u></u>	Saida_adress_to_RAM_outWaveform	U 43706	43706	X											
<u>"</u>		UO													
- ×	SaidaRegA_outWaveform	U 43705	43705	X											
5		U 65535	65535	X											
-		S -21830	-21830	X											
-	-	U 43706													\equiv
<u></u>		U 43706													_

Limitações

- Infelizmente encontramos alguns desafios que não conseguimos transpassar como: O não funcionamento ao utilizar tipos diferentes de instrução, a aceitação de apenas uma entrada partir da ROM;
- Para saltos infelizmente não conseguimos fazer o endereçamento adequado para o salto;

Conclusão

O processador MAGIC comprovou a necessidade de se ter um bom conhecimento de hardware, pois com um mal-uso, o componente não será utilizado com eficiência, o que não é interessante para nenhum arquiteto de hardware.