Definição de requisitos do projeto

Alunos:

- Guilherme Serra Camargo Gomes
- Felipe Aragão Nogueira de Freitas
- Melissa Lenskaia Monni

Quais instruções sua CPU poderá processar? (Ou seja, o início da definição da ISA.)

A CPU poderá processar um sub-conjunto das instruções do RISCV, conforme a tabela abaixo, incluindo-se também a operação NOP:

Inst	Name	FMT	Opcode	funct3	funct7	Description (C)	Note
add	ADD	R	0110011	0x0	0x00	rd = rs1 + rs2	
sub	SUB	R	0110011	0x0	0x20	rd = rs1 - rs2	
xor	XOR	R	0110011	0x4	0x00	rd = rs1 ^ rs2	
or	OR	R	0110011	0x6	0x00	rd = rs1 | rs2	
and	AND	R	0110011	0x7	0x00	rd = rs1 & rs2	
sll	Shift Left Logical	R	0110011	0x1	0x00	rd = rs1 << rs2	
srl	Shift Right Logical	R	0110011	0x5	0x00	rd = rs1 >> rs2	
sra	Shift Right Arith*	R	0110011	0x5	0x20	rd = rs1 >> rs2	msb-extends
slt	Set Less Than	R	0110011	0x2	0x00	rd = (rs1 < rs2)?1:0	
addi	ADD Immediate	I	0010011	0x0		rd = rs1 + imm	
xori	XOR Immediate	I	0010011	0x4		rd = rs1 ^ imm	
ori	OR Immediate	I	0010011	0x6		rd = rs1 | imm	
andi	AND Immediate	I	0010011	0x7		rd = rs1 & imm	
slli	Shift Left Logical Imm	I	0010011	0x1	imm[5:11]=0x00	rd = rs1 << imm[0:4]	
srli	Shift Right Logical Imm	I	0010011	0x5	imm[5:11]=0x00	rd = rs1 >> imm[0:4]	
srai	Shift Right Arith Imm	I	0010011	0x5	imm[5:11]=0x20	rd = rs1 >> imm[0:4]	msb-extends
slti	Set Less Than Imm	I	0010011	0x2		rd = (rs1 < imm)?1:0	
lw	Load Word	I	0000011	0x2		rd = M[rs1+imm][0:31]	
sw	Store Word	S	0100011	0x2		M[rs1+imm][0:31] = rs2[0:31]	
beq	Branch ==	B	1100011	0x0		if(rs1 == rs2) PC += imm	
bne	Branch !=	B	1100011	0x1		if(rs1 != rs2) PC += imm	
blt	Branch <	B	1100011	0x4		if(rs1 < rs2) PC += imm	
bge	Branch ≤	B	1100011	0x5		if(rs1 >= rs2) PC += imm	

Tamanho(s) da instrução.

32 bits

Tamanho do(s) dado(s) que a sua CPU será capaz de processar.

32 bits

Capacidade de memória que a sua CPU será capaz de endereçar (tem a ver com o *PC*)

4 KB

Formas de endereçamento que a sua CPU será capaz de tratar.

Register-only, Immediate, Base addressing, PC-relative addressing

Formas de E/S que a sua CPU será capaz de tratar.

Via interrupções.

Priorizará ou não o uso de banco de registradores no processamento dos dados?

Priorizará.

Modelo RISC ou Modelo CISC.

RISC

Modelo Von Neumann ou Modelo Harvard.

O modelo utilizado será um misto entre modelo Von Neumann e modelo Harvard.

Modelo de CPU: ciclo único, multiciclo ou *pipeline* simples?

Pipeline simples

Endianess – ordenamento de *bytes* dentro da palavra (*little endian* ou *big endian*)

Little endian

Faixa de endereços (hex)	Tamanho da mem. (bytes)	Utilidade da mem	Nome do componente vhd de memória	Tecnologia da mem

0x0000 0FFF – 0x0000 0880	1919 B	Memória de dados dinâmica (*stack e heap*)	mem	RAM
0x0000 087F – 0x0000 0480	1024 B	Memória de dados estática	mem	RAM
0x0000 047F – 0x0000 0080	1024 B	Memória de programa	mem	RAM
0x0000 007F – 0x0000 0000	128 B	Periféricos mapeados	mem	RAM

Observação: o exemplo acima considera que as palavras (e também os registradores) são todos de 32 bits de tamanho (4 bytes) e que o endereçamento dessa CPU é por byte, ou seja, cada byte tem o seu próprio endereço. Portanto, na região de memória reservada para o mapeamento dos registradores dos periféricos, cabe até 64 registradores de 32 bits. Preencha os detalhes da tabela 2, somente quando for implementar os periféricos do seu MCU.