

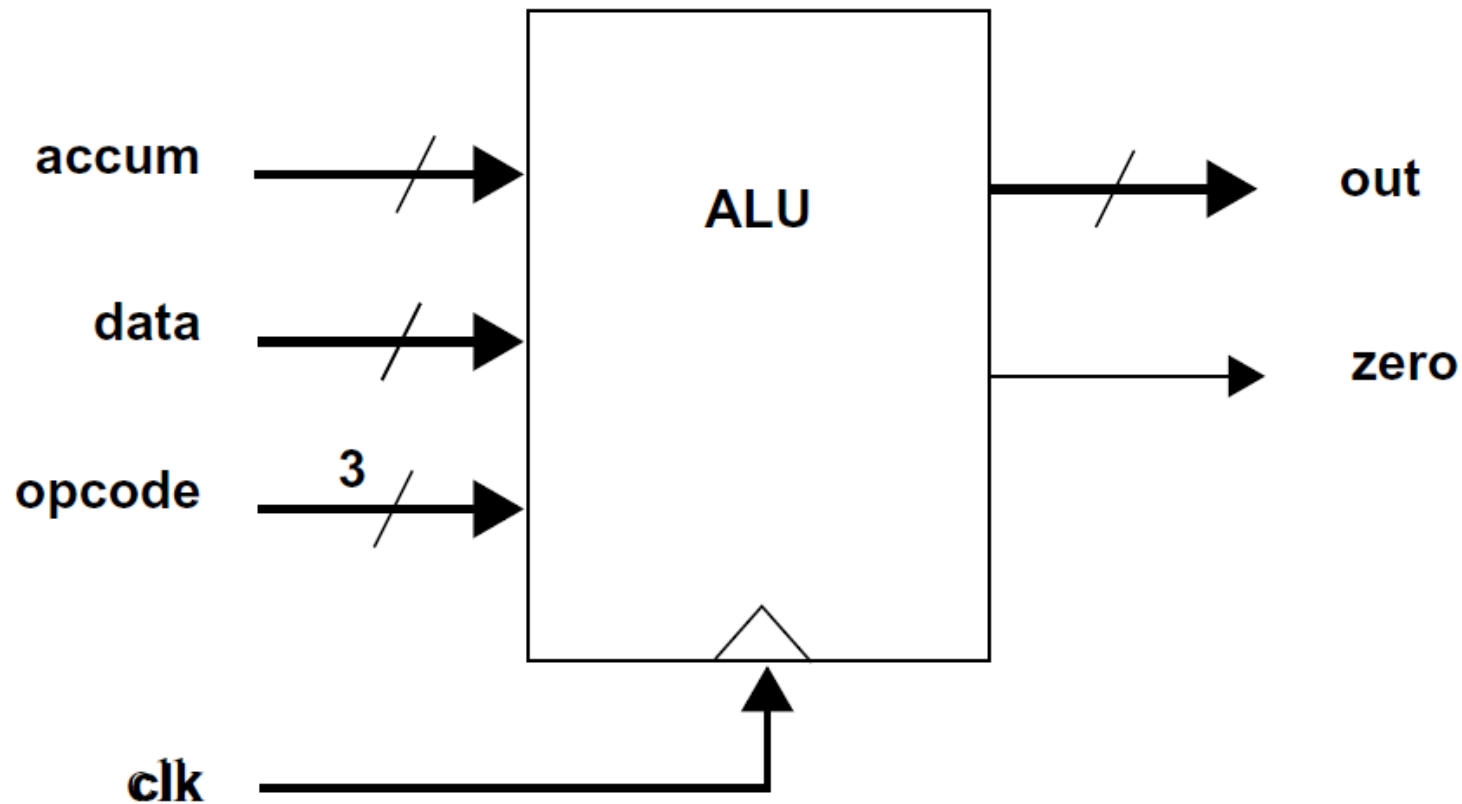
Design com SystemVerilog

Prof. Rômulo Calado Pantaleão Camara
CPU

Carga Horária: 2h/60h

Diagrama de Bloco

- ✓ Desenvolva a ULA para o CPU



Descrição ULA

- ✓ *accum, data, opcode e out são vetores logic;*
- ✓ O **zero** 1-bit é 1 quando o acumulador for 0. Senão será 0.
- ✓ As saídas receberão os valores da borda negativa do clk e estão descritos na tabela do próximo slide;

Descrição ULA

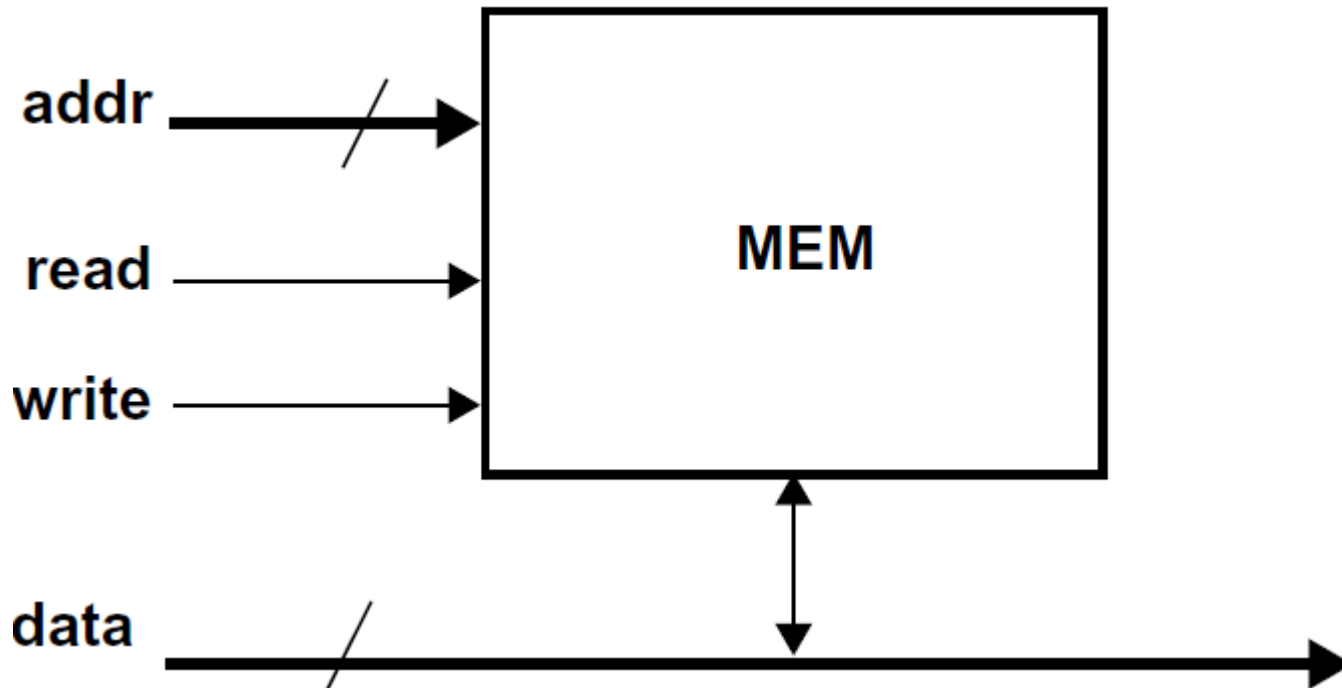
Opcode	Mnemonic	ALU Operation
000	HLT	out = accum
001	SKZ	out = accum
010	ADD	out = data + accum
011	AND	out = data & accum
100	XOR	out = data ^ accum
101	LDA	out = data
110	STO	out = accum
111	JMP	out = accum

Descrição ULA

- ✓ Crie um tipo **enumerated** opcode_e;
- ✓ O nome do arquivo deve ser alu_m.sv e o nome do módulo alu_m;
- ✓ Parametrize a largura da ULA em 1.
- ✓ Use a declaração de time de SystemVerilog;
- ✓ Use always_comb para gerar o **zero**.
- ✓ Use always_ff e unique case para gerar os resultados da ULA;

Diagrama de Bloco

- ✓ Desenvolva uma memória assíncrona para o CPU

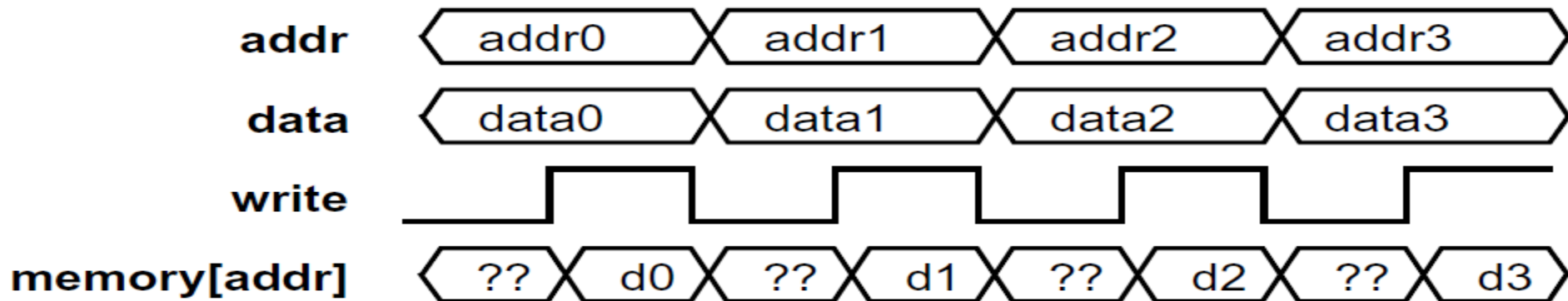


Descrição Memória

- ✓ *Addr e data* são vetores logic;
- ✓ Os bits *write* e *read* nunca poderão ser ativos no mesmo instante de tempo.
- ✓ O *write* guardará o dato no endereço enviado na borda de subida;
- ✓ Quando *read* for ativo em alto, será lido o dado existente no endereço informado para a memória;
- ✓ A figura do próximo slide mostra o funcionamento da memória;

Descrição Memória

✓ Escrita



✓ Leitura

