

## Projetos Digitais e Microprocessadores - BCC

Professor Daniel Oliveira

Aluno: Leonardo Moisés Pires GRR20232376

### Implementação ISA

O desafio principal foi decidir a quantidade de bits para opcode, registradores e imediato.

A divisão em 4 foi a mais intuitiva a se fazer, desde que rs2 fosse também o rd. Sem isso, alguns bits iriam sobrar ou faltar na instrução.

A ISA segue um padrão simples, mas mais simplificado por rs1 e imm terem a mesma posição. Após a instrução ser lida, separamos os fios 7-4 e já temos os rs1 e imm.

O banco de registradores possui 4 bits de posições, resultando em 15 registradores comuns + 1 com zero constante. A escrita primeiramente é controlada pelo sinal W e um demux, quando o mux da posição do registrador da instrução recebe o sinal para escrever a entrada B do banco.

A leitura é feita por MUX.

Caso tenhamos um imediato, a UC manda o sinal reg1 e o rs1 se torna 0. A decisão entre imm e rs1 para a ULA também é feita pela unidade de controle (tabela no final do arquivo).

Na ULA é feita a conversão para complemento de 2. A saída C e zero é definida pela UC. Há um circuito para implementar o bge e lese e o mux para definir qual dessas saídas irá para zero.

A unidade de controle foi definida com os 4 bits menos significativos do OP CODE e possui 7 sinais de controle:

- **pcSel** decide qual soma pc irá fazer.
- **reg1** decide se rs1 será acessado.
- **wbreg** decide se irá ter escrita no banco de registradores.
- **muxUla** decide se imm ou rs1 irá para ula
- **dadobReg** decide se o dado da ULA irá ser escrito no banco de registradores.
- **wm** decide se a memória será escrita.
- **opUla** decide qual operação a ULA irá realizar.

### Componentes

PC: Registrador de 12 bits

Banco de registradores: 16 registradores de 12 bits.

Ula: saída C com 12 bits e zero com 1 bit.

UC: ROOM com 11 bits de dados.

Memória de instruções: ROOM com 12 bits de dados.

Memória de dados: ROOM com 12 bits de dados.

c2: Circuito para transformar em complemento de 2 com inversores.

## UC

pcSel	reg1	wbreg	muxUla	dadobReg	wm	opUla
2	1	1	1	1	1	4
00	0	1	0	1	0	0000
00	0	0	0	0	1	0000
10	0	0	0	0	0	0000
01	0	0	0	0	0	0000
00	0	1	1	0	0	0100
00	0	1	1	0	0	0000
00	0	1	1	0	0	0001
00	0	0	1	0	0	0010
00	1	1	1	0	0	0011
00	1	1	1	0	0	0100
00	1	1	1	0	0	0101
00	1	1	1	0	0	0110
00	0	0	1	0	0	0111
00	0	0	1	0	0	1000
00	0	0	1	0	0	1001
00	0	0	1	0	0	1010