ULA unidade lógica aritmética

Marcos Monteiro Junior

ULA

- Como a ULA funciona e manipula dados e instruções.

Relembrando

- Representação de números em complemento de dois.
  - É a forma de representar os números negativos em binários
  - Número negativo é o espelho do número positivo
- Pode-se converter os números de várias formas.

Relembrando

- Uma das maneiras é inverter todos os números positivo (binários) e somar 1.
- Outra forma é seguir a equação:

$$(-2^{n-1})\ 2^{n-2}\ \ldots\ 2^2\ \ 2^1\ \ 2^0$$

$$100101 = 1*(-2^5) + 1*2^2 + 1*2^0 = -32 + 4 + 1 = -27$$

Relembrando

- O MIPS usa 32 bits
- Se não fosse complemento de 2 teriamos:

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{dois} = 0_{dez}$

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_{dois} = 1_{dez}$

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_{dois} = 2_{dez}$

$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_{dois} = 4.294.967.293_{dez}$

$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_{dois} = 4.294.967.294_{dez}$

$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{dois} = 4.294.967.295_{dez}$

Relembrando

- Como há complemento de 2 temos:

$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_{dois} = 2.147.483.645_{dez}$

$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_{dois} = 2.147.483.646_{dez}$

$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{dois} = 2.147.483.647_{dez}$

$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{dois} = -2.147.483.648_{dez}$

$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_{dois} = -2.147.483.647_{dez}$

$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_{dois} = -2.147.483.646_{dez}$

Complemento de 2

- Afeta tanto operações matemáticas como operações de endereçamento de memória.
- A função de um load com sinal é de copiar o sinal repetidamente até preencher o registrador. Sua intenção é colocar no registrador uma representação correta do número. Instrução load byte (lb) replica o sinal
- Um load sem sinal simplesmente preenche o registrador com zeros à esquerda dos dados. Instrução load byte unsigned (lbu) não replica o sinal.

Acessando bits

- Algumas vezes interessa acessar campos de uma palavra, e não apenas a palavra inteira

- Algumas instruções auxiliam o acesso a bits em uma palavra

Acessando bits

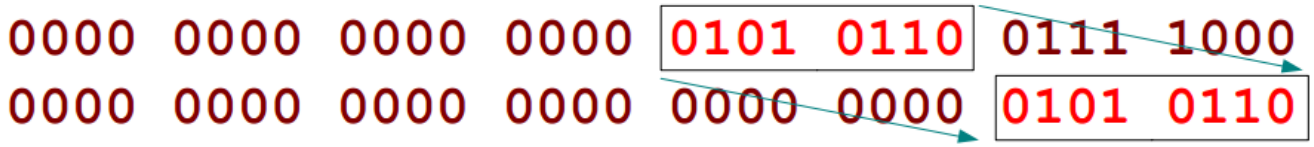
- Operações de deslocamento (shift) movem os bits a esquerda ou a direita, introduzindo zeros na extremidade
- Operações lógicas bit a bit, como AND e OR

O campo shamt, é usado

Op	Rs	Rt	Rd	Endereço/ shamt	funct

Deslocando bits

- Aplicações:
- Selecionando 1 byte
  - srl $t0, $t0, 8



- Elevando 2^n
  - sll $t0, $t0, 2 # multiplica por $2^2 = 4$

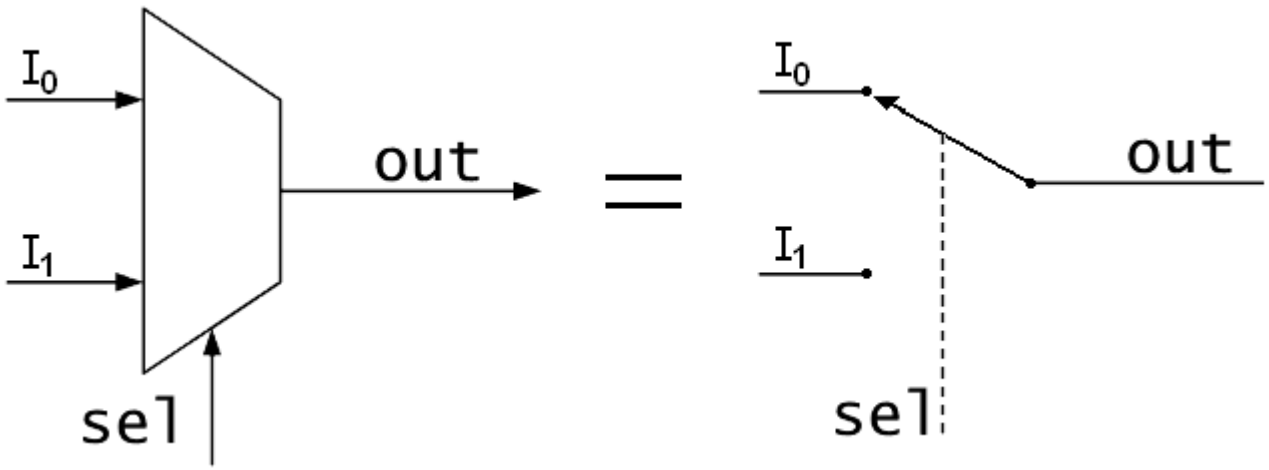
0000 0000 0000 0000 0000 0000 0101 0110
0000 0000 0000 0000 0000 0001 0101 1000

ULA

- O MIPS tem uma ULA de 32 bits que realiza as operações lógicas e aritméticas básicas.
- Parte fundamental do processador
- Composta de vários elementos eletrônicos

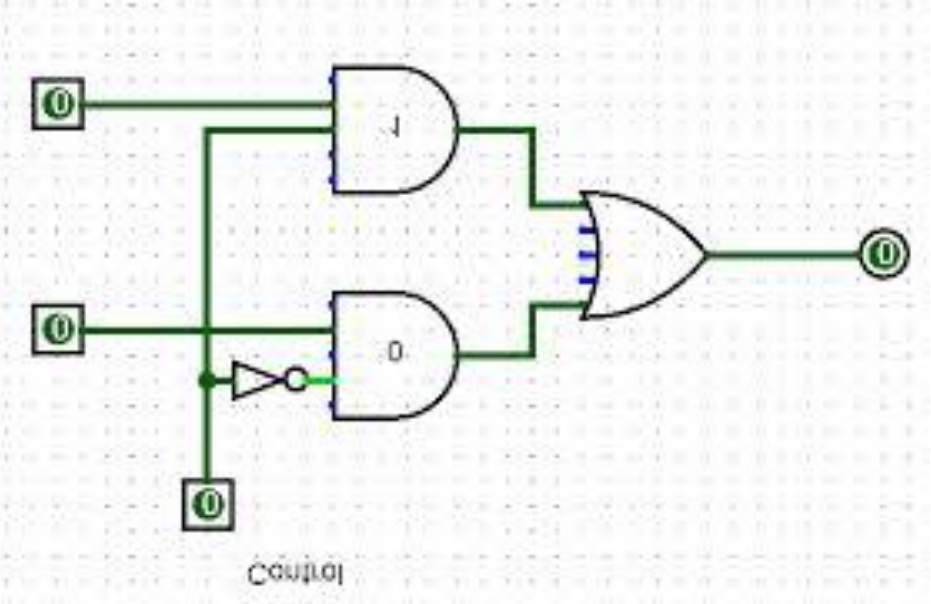
Multiplexador

- Funciona como um seletor
- Multiplexador de 2 bits
- Ao carregar o sel, com valor, 0 ou 1 irá selecionar a saída de acordo com a entrada I0 ou ou I1



Multiplexador

- O interior do Multiplexador de 2 bits

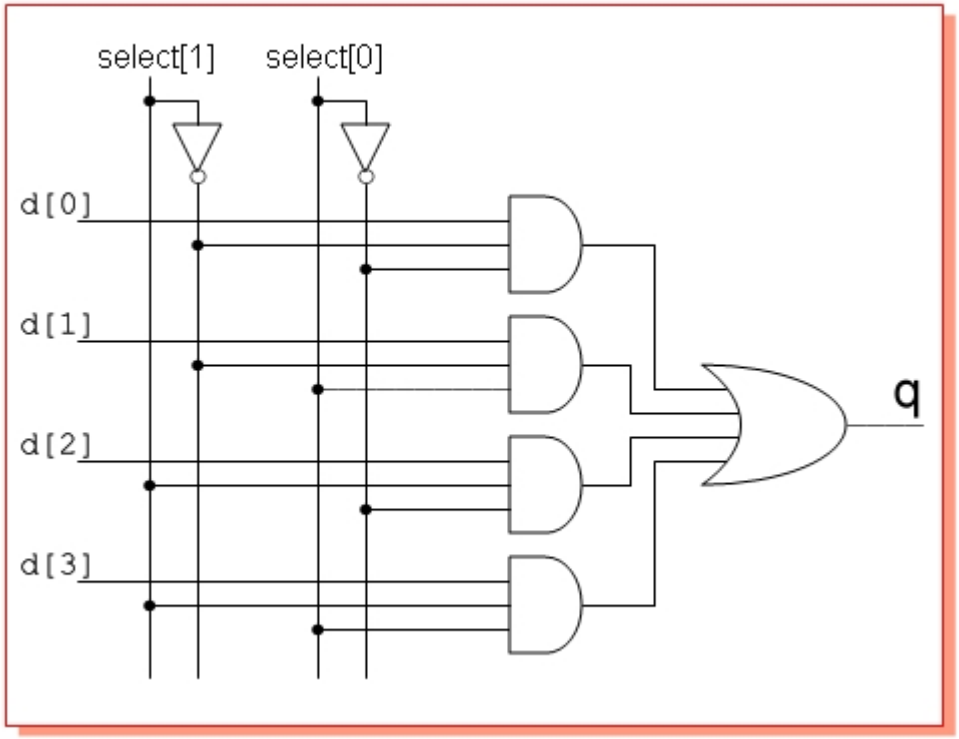


Multiplexador

- Ao seta o seletor com 1 temos:
  - Porta E0 com valor 0 (já que tem uma negação na entrada)
  - Sua saída sempre será 0.
  - Porta E1 teremos o valor 1
  - Sua saída dependera da entrada externa em E1
- Por fim temos a saída ou que permite que o resultado de E1 sempre se repita.
- Caso se inverta o seletor a porta E0 será selecionada

Multiplexador

- De 4 bits, funciona exatamente com o de 2 bits



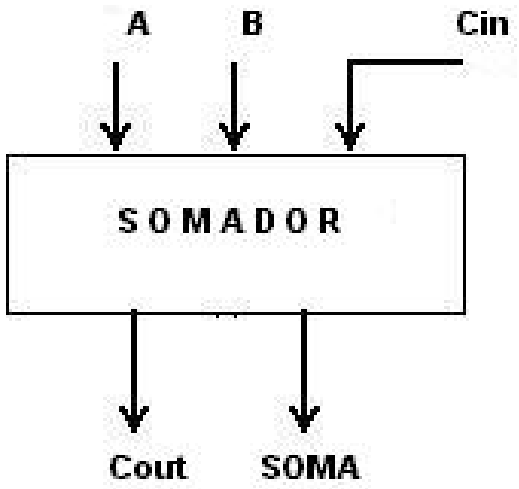
Somador

- Circuito responsável por executar a soma bit a bit.
- Existem várias versões de somador

- O MIPS utiliza o lookahead

Somador

- A e B são os bits a serem somados
- Cin entrada do bit carry
- Cout saída do bit carry
- Soma saída da soma

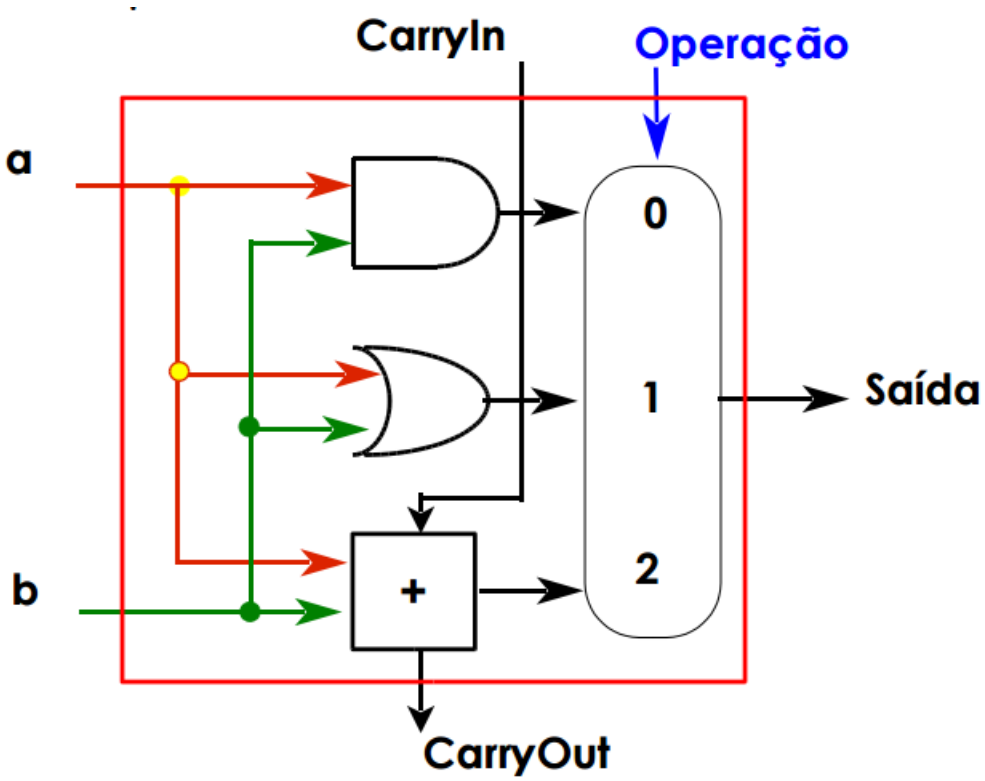


lookahead

- O lookahead segue a tabela verdade a seguir:

Entradas			Saídas	
a	b	CarryIn	CarryOut	Soma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ULA de 1 bit

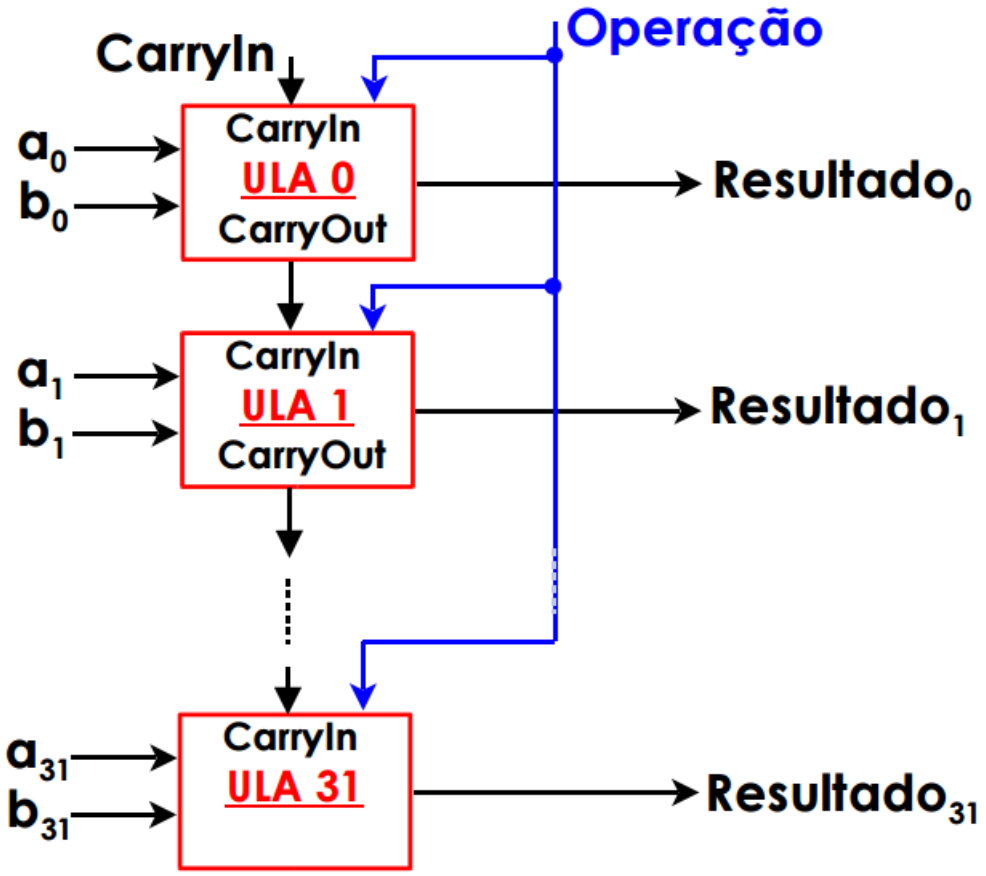


ULA

- O somador anterior operações, and, or, soma de apenas 1 bit

- É utilizado um multiplexador de 3 bits para a escolha da operação.

ULA 32 bit


Operação
CarryIn
CarryIn
ULA 0
CarryOut
a0
b0
Resultado0
CarryIn
ULA 1
CarryOut
a1
b1
Resultado1
CarryIn
ULA 31
a31
b31
Resultado31

Referências

- Universidade de Brasília Departamento de Ciência da Computação (Pdf utilizado em aula)