



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA

LABORATÓRIO DE CIRCUITOS DIGITAIS

Verilog – Abordagem por Fluxo de Dados (Parte 2)

Prof.: Pedro Thiago Valério de Souza
UFERSA – Campus Pau dos Ferros
pedro.souza@ufersa.edu.br

Descrição por Fluxo de Dados

■ Operações Aritméticas:

Operador	Operação Realizada	Exemplo: $a = 11$ $b = 5$
+	Soma	$a + b = 16$
-	Subtração	$a - b = 6$
*	Multiplicação	$a * b = 55$
/	Divisão	$a / b = 2$
%	Resto da divisão	$a \% b = 1$
**	Exponenciação	$b ** 2 = 25$

- O *bit* de *Carry-out* é gerado automaticamente;
 - Se somar dois números de 4 *bits*, o 5ª *bit* resultante da soma é o *carry-out*;

■ Exemplos:

```
assign x = a + b;  
assign y = a - b;  
assign w = a * b;  
assign z = a / b;
```

Descrição por Fluxo de Dados

■ Operações de Concatenação e Replicação:

- Servem para juntar os operandos e formar um novo vetor;

Operador	Operação Realizada	Exemplo: $a = 3'b011$ $b = 2'b01$
{ }	Concatenação	$\{a,b\} = 5'b01101$
{ { }	Replicação	$\{3\{b\}\} = 6'b010101$

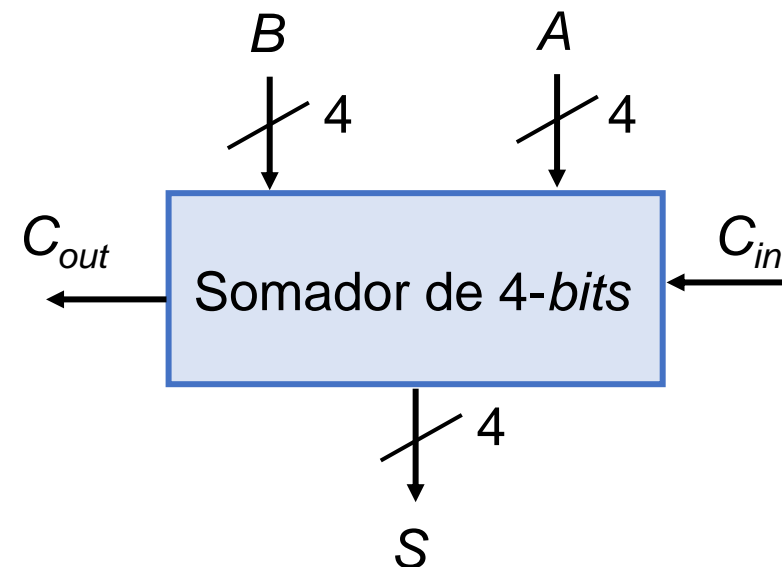
■ Exemplos:

```
assign x = {a, b};  
assign y = {2{a}};
```

Descrição por Fluxo de Dados

Projeto 07 – Somador de 4-bits (revisto)

Projete um somador de 4-bits em Verilog. Desta vez, implemente o somador utilizando fluxo de dados juntamente com as operações aritméticas.



Descrição por Fluxo de Dados

■ Operações de Deslocamento:

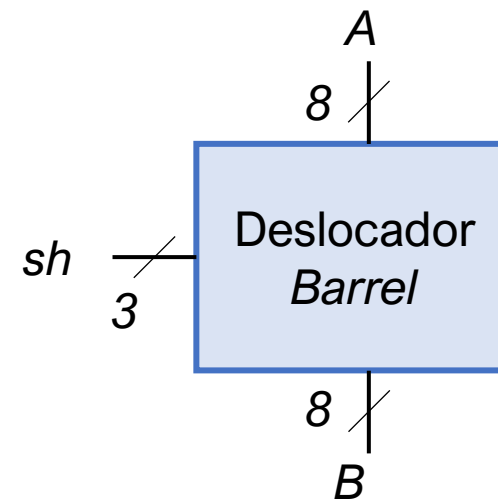
- Deslocamentos à esquerda (lógico ou aritméticos): Posições vagas são preenchidas com zero;
- Deslocamentos à direita:
 - Lógico: Posições vagas são preenchidas com zero;
 - Aritmético (*unsigned*): Posições vagas são preenchidas com zero;
 - Aritmético (*signed*): Posições vagas são preenchidas com o MSB.
- *Bits* deslocados são perdidos.

Operador	Operação Realizada	Exemplo: $a = 4'b0110$ $b = 4'b1011$
<<	Deslocamento à esquerda lógico	$a << 1 = 4'b1100$
>>	Deslocamento à direita lógico	$a >> 1 = 4'b0011$
<<<	Deslocamento à esquerda aritmético	$a <<< 1 = 4'b0100$
>>>	Deslocamento à direita aritmético	$b >>> 1 = 4'b1101$

Descrição por Fluxo de Dados

Projeto 08 – Deslocadores barrel em Verilog

Projete um deslocador *barrel* em Verilog. O deslocador em questão possui uma entrada *A*, de 8 *bits*, que indica o dado a ser deslocador e uma entrada *sh*, de 3 *bits*, que indica quantos deslocamentos devem ser dados ao número *A*. A saída é apresentada em *B*, também de 8 *bits*.



Descrição por Fluxo de Dados

■ Operações Relacionais:

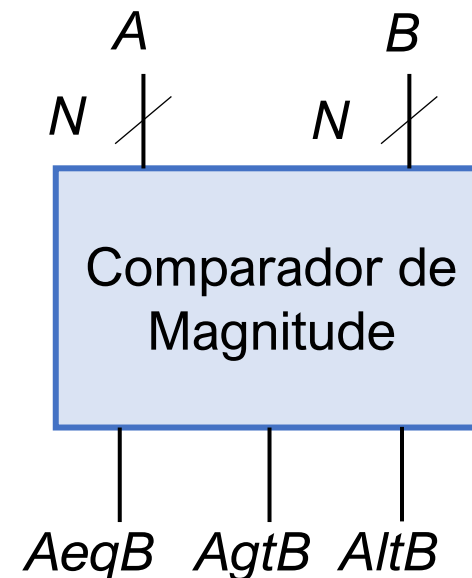
- São utilizados para comparar dois operandos;
- Como resultado tem-se 0, quando a relação é falsa ou 1, quando a relação é verdadeira;

Operador	Operação Realizada	Exemplo: $a = 4'b0110$ $b = 4'b0011$ $c = 4'b0011$
$>$	Maior que	$a > b = 1'b1$
$<$	Menor que	$a < b = 1'b0$
$==$	Igual a	$a == b = 1'b0$
$>=$	Maior ou igual que	$a >= b = 1'b1$ $a >= c = 1'b1$
$<=$	Menor ou igual que	$a <= b = 1'b0$
$!=$	Diferente	$a != b = 1'b1$

Descrição por Fluxo de Dados

Projeto 09 – Comparadores de magnitude (revisitado)

Projete, utilizando o Verilog, um circuito que se comporte como um comparador de magnitude de 4 *bits*. O circuito possui duas entradas, denominadas de *A* e *B*, ambas de 4 *bits*. O circuito possui três saídas, denominadas de *AeqB*, *AgtB* e *AltB*, que indicam quando, respectivamente, $A=B$, $A>B$ e $A<B$. No seu projeto, utilize os operadores relacionais.



Descrição por Fluxo de Dados

- **Atribuição Condicional:**

- Corresponde a um ternário em C;

```
assign x = (condição) ? (verdadeiro) : (falso);
```

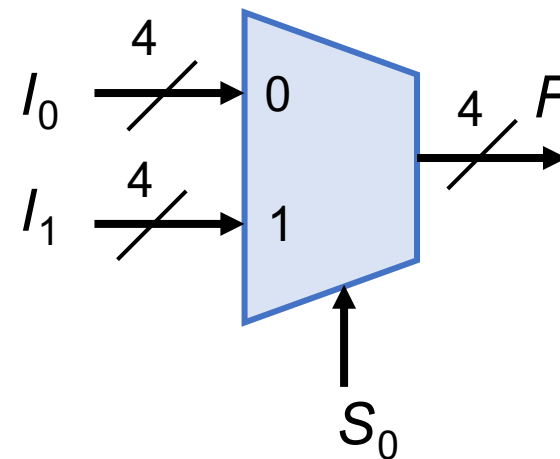
- Exemplo:

```
assign x = sel ? 4'b0010 : 4'b1100;  
//Verifica sel.  
//Se sel = 1 (verdadeiro), então x = 4'b0010.  
//Se sel = 0 (falso), então x = 4'b1100.
```

Descrição por Fluxo de Dados

Projeto 10 – Multiplexador 2x1 de 4-bits

Descreva o circuito correspondente a um multiplexador 2x1 em Verilog utilizando a abordagem por fluxo de dados e utilizando atribuição condicional.



Descrição por Fluxo de Dados

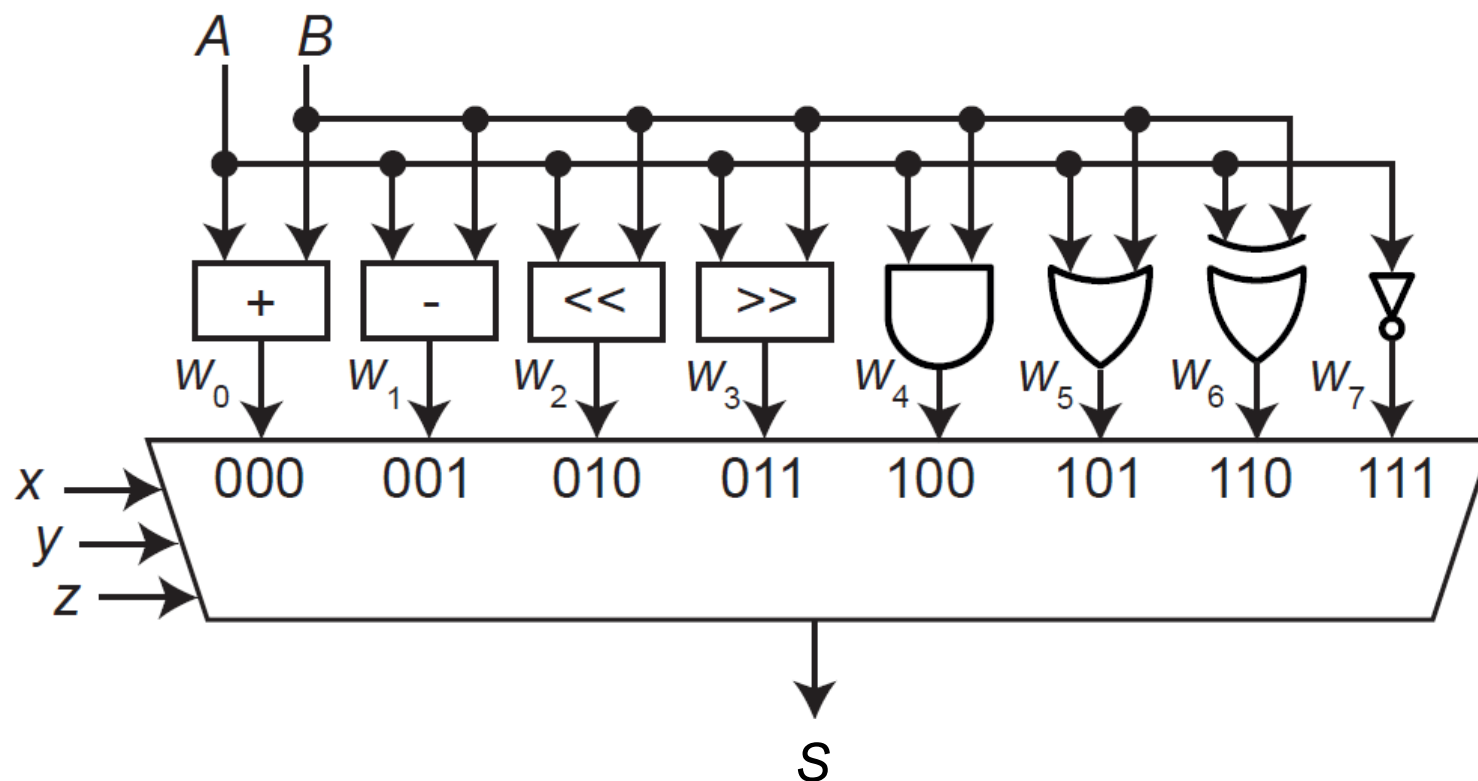
Projeto 11 – Unidade Lógico-Aritmética em Verilog

Implemente, em FPGA e utilizando o Verilog, uma unidade lógico-aritmética de 4-*bits* que possua a tabela de operação apresentada ao lado.

Seleção			Operação
<i>x</i>	<i>y</i>	<i>z</i>	
0	0	0	$S = A + B$
0	0	1	$S = A - B$
0	1	0	$S = A \ll B$
0	1	1	$S = A \gg B$
1	0	0	$S = A \text{ and } B$
1	0	1	$S = A \text{ or } B$
1	1	0	$S = A \text{ xor } B$
1	1	1	$S = \text{not } A$


Descrição por Fluxo de Dados

Projeto 11 – Unidade Lógico-Aritmética em Verilog



Descrição por Fluxo de Dados

■ Prioridade dos Operadores:

Operador	Prioridade
Operadores unários	
**	
* / %	
+ -	
<< >> <<< >>>	
< > <= >=	
== ~= === ~==	
& (operador binário)	
^ ~^ ^~ (operador binário)	
(operador binário)	
&&	
?:	
{ } { }	

Referências

- **PIMENTA, TALES CLEBER – Circuitos Digitais – Análise e Síntese Lógica: Aplicações em FPGA. Elsevier, 2017.**
- Altera's Verilog HDL Basics (<https://www.youtube.com/watch?v=PJGvZSIsLKs>)
- VAHID, FRANK - Sistemas Digitais - Projeto, Otimização e HDLs. Bookman, 2008.