Verilog CheatSheet

Oficina Introdução ao Verilog Comportamental

XXXI SIECOMP

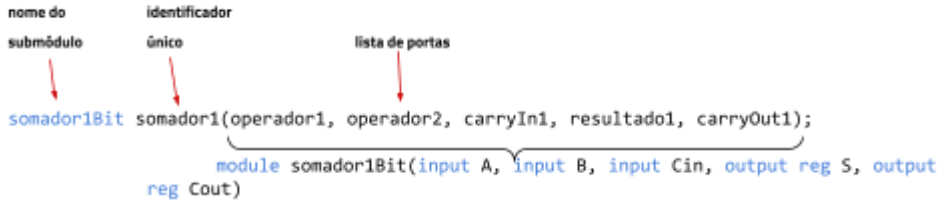
OPERADORES

+	a = b + c	Soma binária	=	a = b	a recebe o valor de b
-	a = b - c	Subtração binária	==	a == b	a é igual a b?
|	a = |b	OR para todos os bits de b	===	a === b	a é igual a b? (inclui X e Z)
&	a = &b	AND para todos o bits de b	!=	a != b	a é diferente de b?
^	a = ^b	XOR para todos o bits de b	!==	a !== b	a é diferente de b? (inclui X e Z)
~|	a = ~|b	NOR para todos o bits de b	<<	a << n	desloca a à esquerda n vezes e completa com 0
~&	a = ~&b	NAND para todos o bits de b	>>	a >> n	desloca a à direita n vezes e completa com 0
~^	a = ~^	XNOR para todos o bits de b	<<<	a <<< n	desloca a à esquerda n vezes e completa com 0 (mantém o sinal)
{}	a = {b,c}	concatena 'b' e 'c' e atribui a 'a'	>>>	a >>> n	desloca a à direita n vezes e completa com 0 (mantém o sinal)

DEFINIÇÃO DE MÓDULOS

```
module <nome_do_modulo> (<lista_de_portas>);

        <declaração_dos_tipos_das_portas> // portas INPUT/OUTPUT/INOUT

        <declaração_de_tipo> // WIRE,REG, INTEGER, etc

        <funcionalidade_do_circuito> // always/assign

        <especificação_de_tempo> // usado em simulação

endmodule
```

INSTANCIAÇÃO DE MÓDULO



ASSIGN

assign <variável> = <expressão>;

ATRIBUIÇÕES

- Bloqueante: usada em circuitos combinacionais

a = <expressão>

- Não bloqueante: usada em circuitos sequenciais

a <= <expressão>

BLOCOS

- Always

always @ (<lista_de_triggers>) begin

end

Exemplos de triggers: *, posedge <sinal>, negedge <sinal>, sinal1 <operador> sinal2 <operador> sinal3 ...

- initial

```
initial
        begin
                <declaracoes_multiplas>
        end
```

- begin ... end: agrupa múltiplas declarações para serem executadas sequencialmente
- fork ... join: agrupa múltiplas declarações para serem executadas paralelamente

---

ESTRUTURAS DE DECISÃO

- IF-ELSE-IF

```
if (<condicao>)
        <declaracao_unica>
else if (<condicao_2>)
        begin
                <declaracoes_multiplas>
        end
else
        begin
                <declaracoes_multiplas>
        end
```

- CASE

```
case (<expressao>)
        case_item1 :
                <declaracao_unica>
        case_item2 :
                begin
                        <declaracoes_multiplas>
                end
        default:
                <declaracao_unica>
endcase
```
