

Laços de Repetição em Verilog

Laços de Repetição

Assim como em outras linguagens, **Verilog** também possui estruturas de repetição, que são úteis para executar blocos de código várias vezes. As principais estruturas de repetição em Verilog são:

- `for`
- `while`
- `repeat`
- `forever`

For

A estrutura `for` é utilizada para repetir um bloco de código enquanto uma condição for satisfeita. A sintaxe básica é:

```
for (inicializa o; condi o; incremento) begin
    // código a ser executado
end
```

A inicialização é executada uma vez antes do loop, a condição é verificada antes de cada iteração e o incremento é executado após cada iteração. Aqui está um exemplo:

```
module exemplo(
    input [3:0] entrada,
    output reg [3:0] saida
);
    integer i;
    always @(entrada) begin
        saida = 4'b0000; // valor inicial da saída
        for (i = 0; i < 4; i = i + 1) begin
            if (entrada[i] == 1'b1) begin
                saida[i] = 1'b1; // se o bit i da entrada
                                // for 1, seta o bit i da saída para 1
            end
        end
    end
end
```

```

        end
    end
endmodule

```

Neste exemplo, o loop **for** percorre cada bit da entrada e, se o bit for 1, seta o bit correspondente na saída para 1.

While

A estrutura **while** executa um bloco de código enquanto uma condição for verdadeira. A sintaxe básica é:

```

while (condição) begin
    // código a ser executado
end

```

Aqui está um exemplo:

```

module exemplo(
    input [3:0] entrada,
    output reg [3:0] saida
);
    integer i = 0;
    always @(entrada) begin
        saida = 4'b0000; // valor inicial da saída
        while (i < 4) begin
            if (entrada[i] == 1'b1) begin
                saida[i] = 1'b1; // se o bit i da entrada
                                // for 1, seta o bit i da saída para 1
            end
            i = i + 1; // incrementa i
        end
        i = 0; // reseta i para uso futuro
    end
endmodule

```

Neste caso, o código tem a mesma função do exemplo anterior, mas utiliza a estrutura **while** para iterar sobre os bits da entrada. O circuito sintetizado será o mesmo, mas o uso de **for** é preferível em hardware descritivo, pois **while** pode gerar loops infinitos, além de ser mais verboso.

Repeat

A estrutura **repeat** é utilizada para repetir um bloco de código um número específico de vezes. A sintaxe básica é:

```

repeat (número_de_repetições) begin
    // código a ser executado
end

```

Exemplo:

```
repeat(10) begin
    clk = ~clk; // inverte o sinal do clock 10 vezes
end
```

O exemplo acima é um uso clássico de **repeat**, que é encorajado para testbenches, mas não é recomendado para circuitos sintetizáveis, pois o número de repetições deve ser conhecido em tempo de compilação. Se o valor passado para o **repeat** for uma variável, o sintetizador não conseguirá gerar o hardware corretamente.

Forever

A estrutura **forever** é utilizada para criar um loop infinito, que continuará executando até que seja interrompido por uma condição externa. A sintaxe básica é:

```
forever begin
    // c digo a ser executado
end
```

Forever é utilizado **APENAS** em testbenches, pois em circuitos sintetizáveis, um loop infinito não faz sentido. Exemplo:

```
forever begin
    clk = ~clk; // inverte o sinal do clock indefinidamente
    #5;         // espera 5 unidades de tempo
end
```