Universidade Tecnológica Federal do Paraná – Toledo
Engenharia da Computação – COENC

Lógica Reconfigurável

Código Sequencial

FOR-LOOP
WHILE-LOOP
WAIT

Tiago Piovesan Vendruscolo

Comando FOR-LOOP

- Estrutura

```
optional_label: for parameter in range loop
   sequential statements
end loop label;
```

Exemplo:

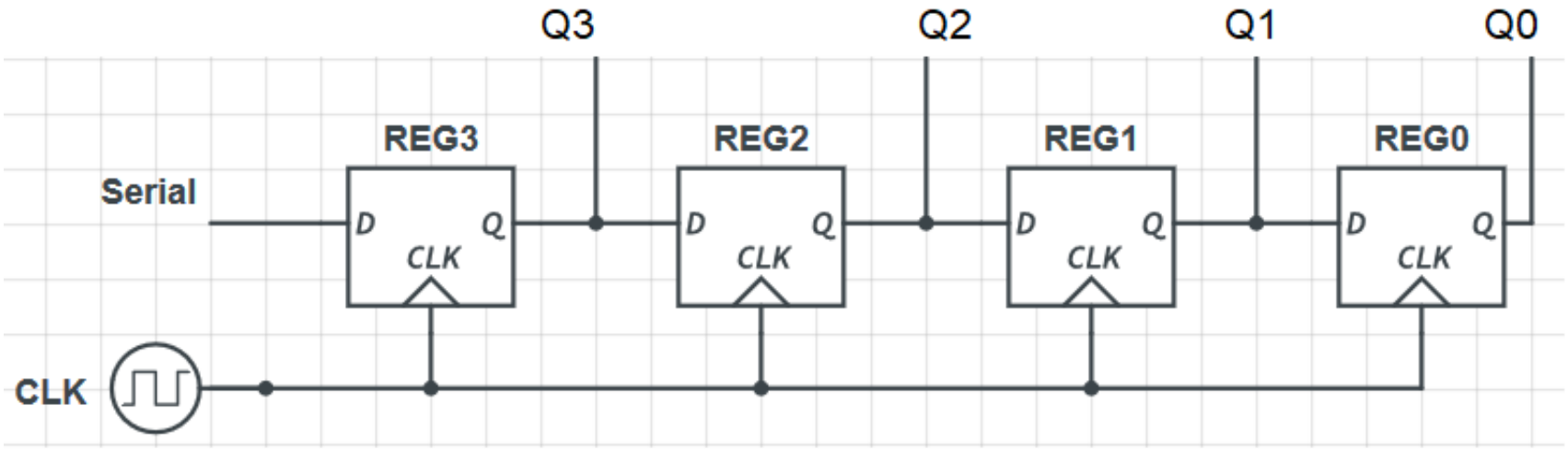
```
Exemplo_FOR: FOR i IN 0 TO 10 LOOP
             Q(i)  <=  Q(i+1);
END LOOP;
```

Comando EXIT para sair do loop durante a execução:

```
FOR i IN dado'RANGE LOOP
   CASE dado(i) IS
      WHEN '0' => faça algo;
      WHEN OTHERS => EXIT;
   END CASE;
END LOOP;
```

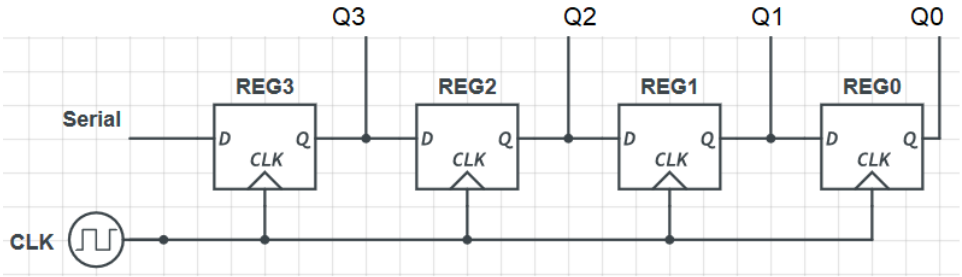
Comando FOR-LOOP

- Exercício 1: Projete e simule o registrador-deslocador abaixo (parametrizado), com entrada serial.



Comando FOR-LOOP

Aula sobre IF-THEN-ELSE



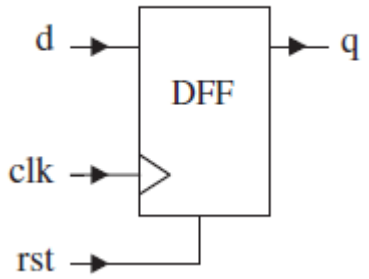

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY shiftreg1 IS
      PORT ( serial, clk :IN STD_LOGIC;
             Q           :BUFFER STD_LOGIC_VECTOR(3 DOWNTO 0));
END shiftreg1;

ARCHITECTURE funcao OF shiftreg1 IS
BEGIN
    PROCESS (clk)
    BEGIN
      IF clk'EVENT AND clk = '1' THEN
                  Q(0)  <=  Q(1);
                  Q(1)  <=  Q(2);
                  Q(2)  <=  Q(3);
                  Q(3)  <=  serial;
      END IF;
    END PROCESS;
END funcao;
```

Comando FOR-LOOP

Exercício 2: Com base no código de um Flip-Flop abaixo, faça um conjunto com 3 Flip-flops que funcionem de forma totalmente independente (incluindo clk e rst) utilizando o comando FOR-LOOP. Faça a simulação.

```
-------------------------------------
ENTITY flipflop IS
PORT ( d, clk, rst: IN STD_LOGIC;
q: OUT STD_LOGIC);
END flipflop;
-------------------------------------
ARCHITECTURE funcao OF flipflop IS
BEGIN
PROCESS (rst, clk)
BEGIN
IF (rst='1') THEN
q <= '0';
ELSIF (clk'EVENT AND clk='1') THEN
q <= d;
END IF;
END PROCESS;
END funcao;
```



FF do tipo D, com clock na borda de subida e sinal de reset assíncrono (quando rst = ‘1’, a saída vai a zero).

Comando WHILE-LOOP

- Estrutura

```
WHILE condition LOOP
   sequential statements
END LOOP;
```

```
while (I <= 3) loop
      Z(I) <= '1';
      I := I + 1;
end loop;
```

Comando WHILE-LOOP

- Exercício 3: Faça um programa que tenha uma entrada (PORT) que varie de 0 a 7 e uma saída de 4 bits. O programa deve seguir a seguinte relação entre a entrada e a saída:
  - A = 0; Z = 0001;
  - A = 1; Z = 0010;
  - A = 2; Z = 0100;
  - A = 3; Z = 1000;
  - A = 4 - 7; Z = 0000;

Comando WAIT

- Estrutura:

```
wait until CLK'event and CLK='1';

wait until condition;

wait on signal_list;

wait for time;

wait;
```

Não funciona no QUARTUS

Não é sintetizável, utilizado apenas para test benches.

- O comando WAIT UNTIL só pode ser usado:
  - Em PROCESS sem lista de sensibilidade;
    - Não pode ser usado em uma PROCEDURE chamado por um PROCESS com lista de sensibilidade.
  - Em uma FUNCTION;
    - Não pode ser usado em uma PROCEDURE chamado por uma FUNCTION.
  - Apenas um WAIT pode ser usado dentro de cada PROCESS.

Comando WAIT

- Exercício 4: Faça um programa que coloque na saída Z o valor “1111” quando a entrada A for >=2.
  - * O comando WAIT UNTIL irá “travar” o PROCESS até que a condição seja atendida.

Próxima aula

- Exemplos de aplicações