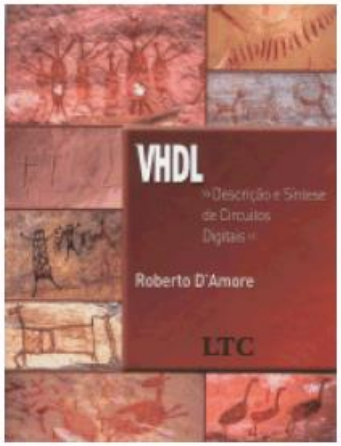
SEL5752/SEL0632 – Linguagens de Descrição de Hardware
Aula 13 – Circuitos Síncronos

Prof. Dr. Maximiliam Luppe

Livro adotado:

VHDL - Descrição e Síntese de Circuitos Digitais
Roberto d'Amore

ISBN 85-216-1452-7
Editora LTC www.ltceditora.com.br



Para informações adicionais consulte: **www.ele.ita.br/~damore/vhdl**

Estratégias de descrição de circuitos síncronos

Tópicos

- Registrador sensível a nível
- Registrador sensível a borda - inicialização síncrona
- Registrador sensível a borda - inicialização assíncrona
- Registrador sensível a borda com habilitação para sinal de relógio
- Máquinas de estado finito
- Contadores
- Cuidados na descrição

Estratégias de descrição

- **Descrição visando**: simulação ou teste
  - sem limitações:
    - pode empregar todos os recursos disponíveis na linguagem

- **Descrição visando**: síntese
  - limitada:
    - às estruturas reconhecidas pela ferramenta de síntese
  - procurar utilizar recursos disponíveis na tecnologia alvo
    - benefícios: circuitos com melhor desempenho
      circuitos com menor custo

Inferência de um elemento de memória em VHDL

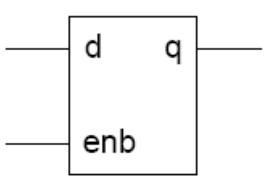
- **Inferência de um elemento de memória**:
  - um valor é atribuído a um sinal - variável em pelo menos uma condição e nenhum valor é atribuído a este objeto em pelo menos uma condição

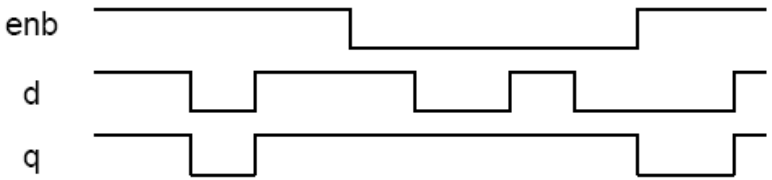
- **Em VHDL:**
  - a falta de atribuição de um valor:
    acarreta a necessidade de armazenamento do valor do sinal quando a condição não é atingida

- **Erro comum em descrições**:
  - condição não especificada:
    necessário *latch* para armazenar o valor

Registrador sensível a nível

- **Exemplo**:





- **Formato geral**: empregando processo
  - sinal **d**: deve estar na lista de sensibilidade

```
PROCESS (enb, d)
BEGIN
  IF (enb = '1') THEN  d <= q ;
  END IF ;
END PROCESS;
```

Registrador sensível a nível

- **Exemplo**: *reset* e *set* assíncronos (devem ser incluídos na lista de sensibilidade)

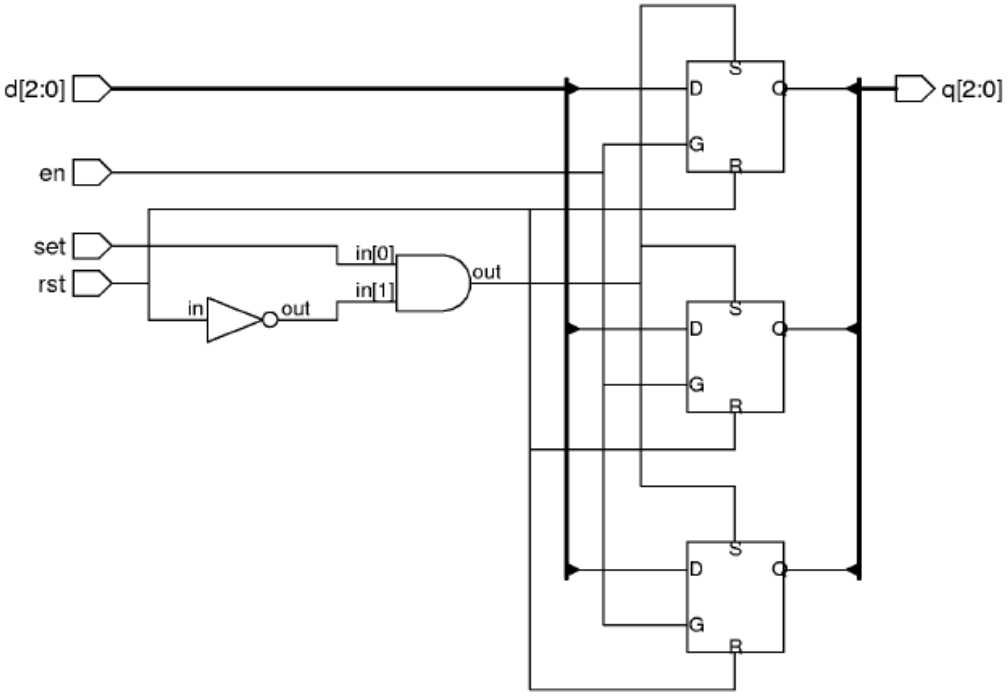
```
 1 ENTITY latch3_1 IS
 2   PORT (en   : IN  BIT;                -- habilita
 3         rst  : IN  BIT;                -- rst=1 leva q=000
 4         set  : IN  BIT;                -- set=1 leva q=111
 5         d    : IN  BIT_VECTOR(2 DOWNTO 0);
 6         q    : OUT BIT_VECTOR(2 DOWNTO 0));
 7 END latch3_1;
 8
 9 ARCHITECTURE teste OF latch3_1 IS
10 BEGIN
11   PROCESS (en, d, rst, set)
12   BEGIN
13     IF    (rst ='1') THEN q <="000";  -- q=000 independente de en
14     ELSIF (set ='1') THEN q <="111";  -- q=111 independente de en
15     ELSIF (en  ='1') THEN q <=d;      -- condicao do sinal para habilitar
16     END IF;
17   END PROCESS;
18 END teste;
```

Registrador sensível a nível - circuito sintetizado: nível RTL

- **Descrição do *latch***

  - G: habilita memorização - S e R: inicialização assíncrona - ativos alto

- **rst =1** (independente set)
  - R =1 (direto) S =0 (via porta e)
- **rst =0 e set =1**
  - R =0 (direto) S =1 (via porta e)
- **rst =0 e set =0**
  - R =0 (direto) S =0 (via porta e)
  - possível memorizar se en =1



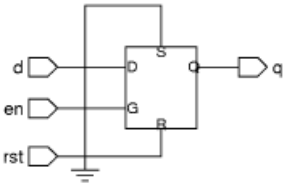
Obs: note a prioridade do comando IF na implementação

```
IF    (rst ='1') THEN  q <="000";
ELSIF (set ='1') THEN  q <="111";
```

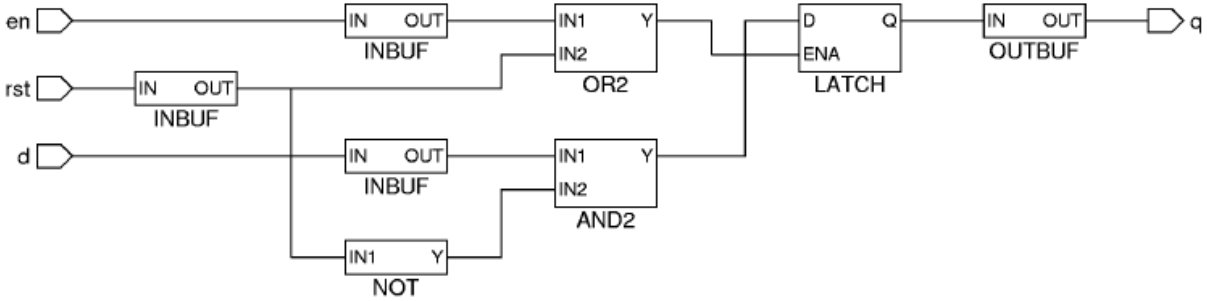
Registrador sensível a nível - exemplo de problema

- **Caso a tecnologia empregada não disponha dos elementos necessários**:
  - *latch, latch* com inicialização assíncrona, etc.
- **Ferramenta sintetiza com portas lógicas** (mensagens de aviso geradas)

- **Exemplo**: Descrição: *latch* contendo inicialização assíncrona
  Tecnologia alvo não é dispõe de *latch* com inicialização assíncrona



nível RTL:
reset assíncrono

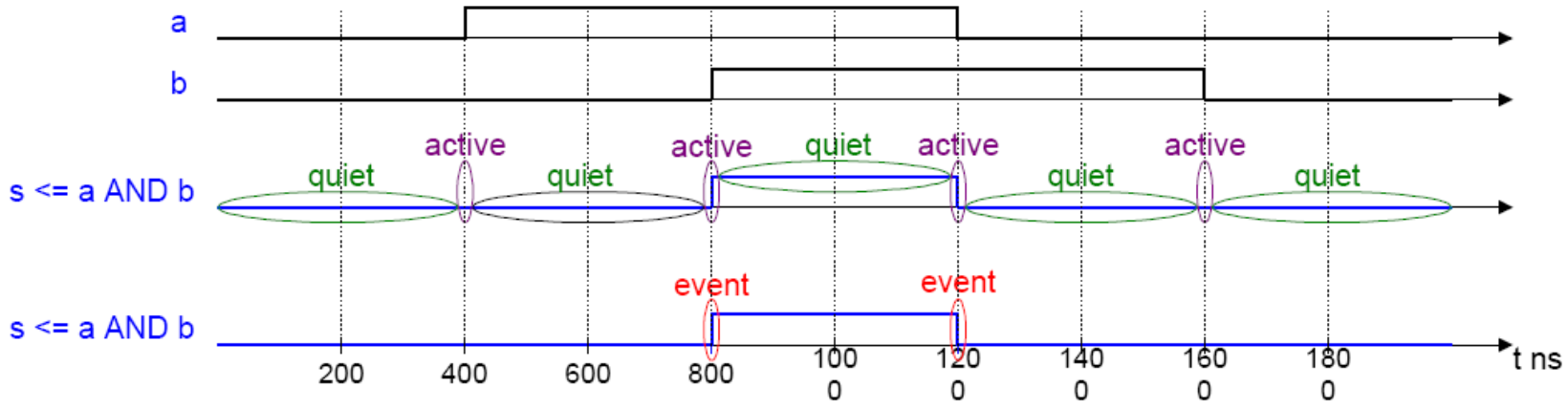


nível portas lógicas:
reset assíncrono implementado na forma síncrona

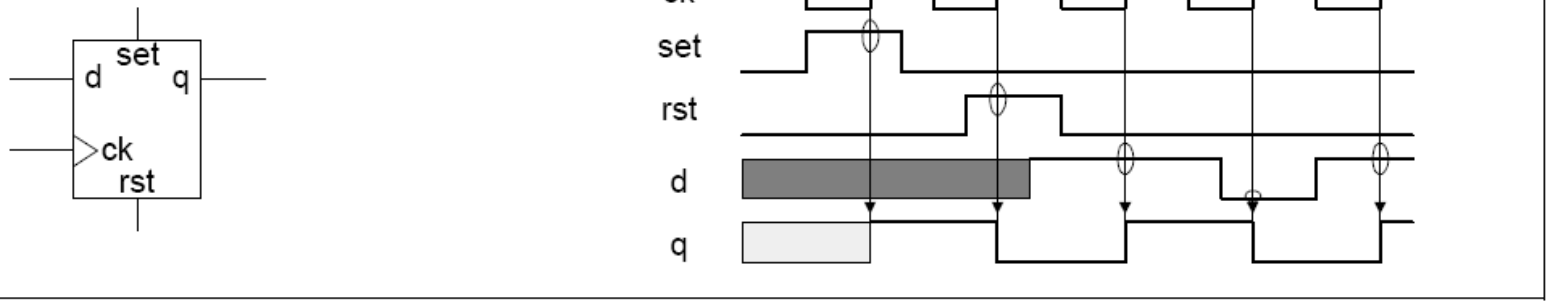
```
Warning, asynchronous reset on DLATCH q_EXMPLR replaced by combinational logic.
Warning, Transformations were required because of constraints in target technology max7.
Warning, Design may show simulation differences because of transformations.
```

Definição de termos relativos a sinais (útil para entender alguns atributos)

- **active**: novo valor atribuído ao sinal (o valor pode ser o mesmo)
- **quiet**: condição oposta a active
- **event**: mudança do valor de um sinal
- **Exemplo**:



Registrador sensível a borda - inicialização síncrona - (exemplo)



- **Formato geral**: empregando processo

  - apenas sinal ck: necessita estar na lista de sensibilidade

```
PROCESS (ck)
BEGIN
  IF (ck'EVENT and ck = '1') THEN      -- detecta borda de subida do sinal ck
    IF    rst = '1' THEN  q <= '0';    -- sequencia de eventos sincronos com ck
    ELSIF set = '1' THEN  q <= '1';    --    .
                                       --    .
                                       --    .
    ELSE                  q <= d;      -- finaliza com a memorizacao
    END IF;
  END IF;
END PROCESS;
```

Registrador sensível a borda - inicialização síncrona - (descrição)

- **Exemplo**:
  - 3 bits
  - *reset* e *set* síncronos (não é necessário inclusão na lista de sensibilidade)

```
 1 ENTITY flip3_2 IS
 2   PORT (ck   : IN  BIT;                -- relogio
 3         rst  : IN  BIT;                -- rst=1 leva q=000 sincrono
 4         set  : IN  BIT;                -- set=1 leva q=111 sincrono
 5         d    : IN  BIT_VECTOR(2 DOWNTO 0);
 6         q    : OUT BIT_VECTOR(2 DOWNTO 0));
 7 END flip3_2;
 8
 9 ARCHITECTURE teste OF flip3_2 IS
10 BEGIN
11   PROCESS (ck)
12   BEGIN
13     IF (ck'EVENT AND ck = '1') THEN        -- condicao do sinal relogio
14       IF    (rst = '1') THEN  q <= "000";  -- teste para levar q=000
15       ELSIF (set = '1') THEN  q <= "111";  -- teste para levar q=111
16       ELSE                    q <= d;      -- armazena dado
17       END IF;
18     END IF;
19   END PROCESS;
20 END teste;
```

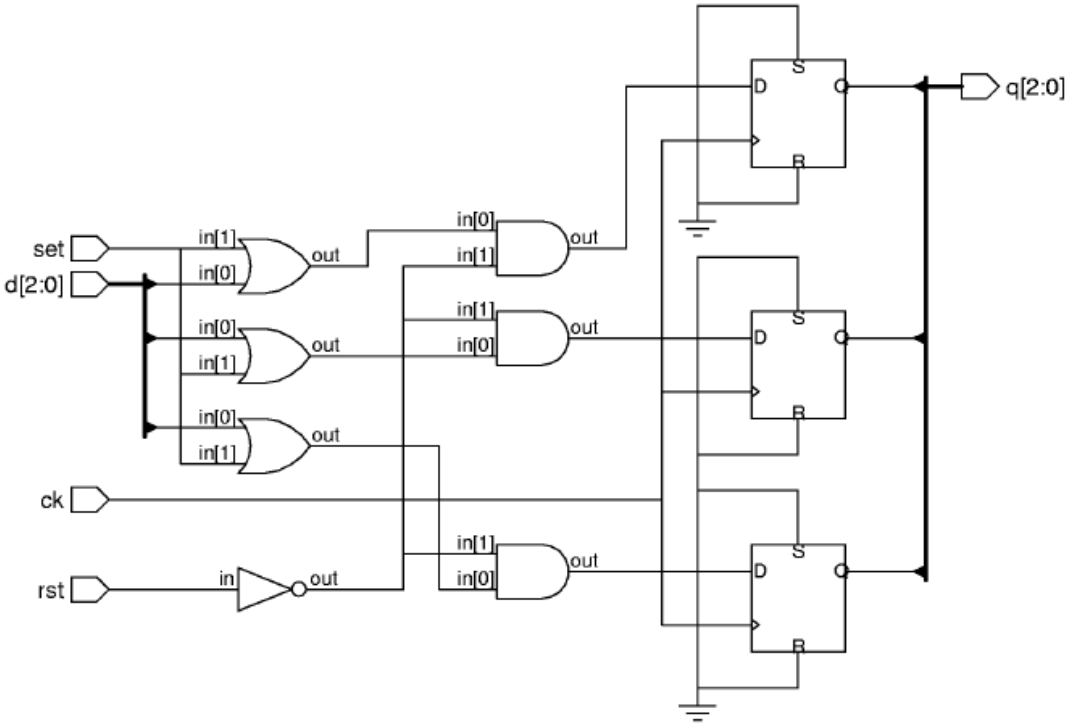
Registrador sensível a borda - inicialização síncrona - (circuito nível RTL)

- **Descrição do *flip flop* do esquema**:
  - sinal de relógio: entrada com o símbolo > (borda de subida)
  - terminais S e R inicialização assíncrona: *set* *reset* (ativo baixo)
- **Sinais S e R desabilidados** (fixos em nível baixo)

- **rst =1:**
  - D_flip_flop =0 (independente de set e d)
  - operação síncrona de rst

- **rst =0:**
  - dados presentes na saída das portas ou transferidos
    - set =1: D _flip_flop =1
    - set =0: D _flip_flop = D

(continua prox. imagem)



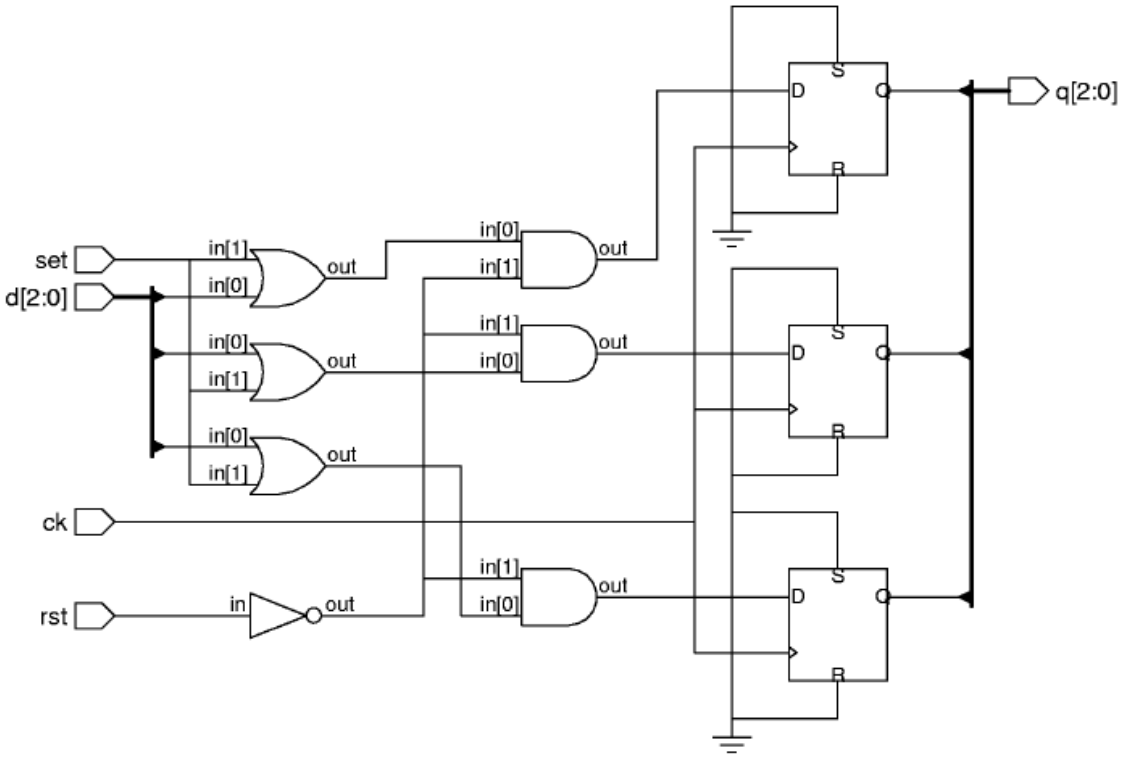
Registrador sensível a borda - inicialização síncrona - (circuito nível RTL)

- **Resumo**: as portas lógicas operam como um circuito de seleção:
  - implementa a construção IF ELSE - linhas 14 a 17
- **Observar**: circuito de seleção contém a prioridade prevista na seqüência de decisões tomadas na construção IF ELSE

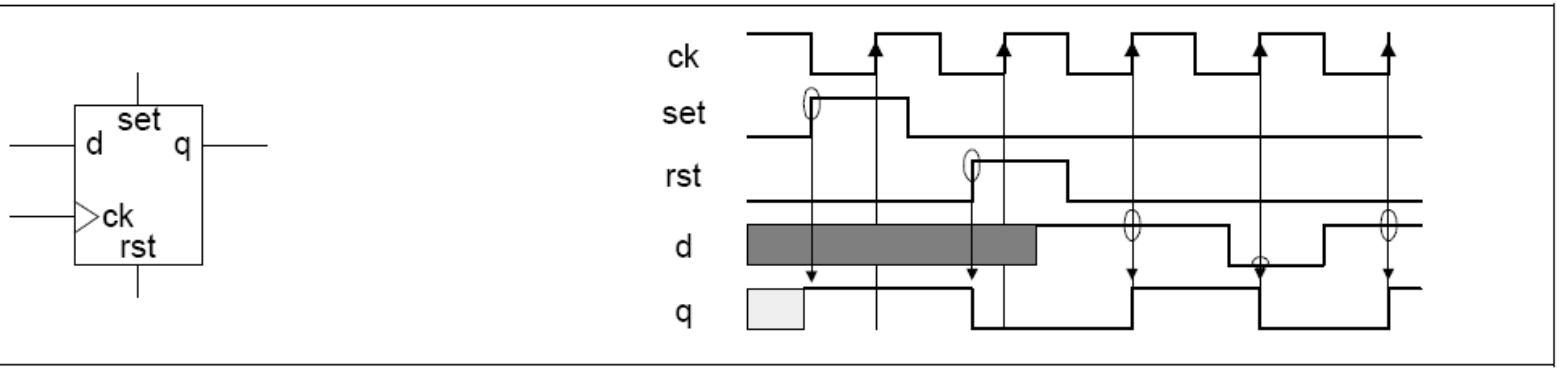
  - rst =1 possível *reset* independente de set
  - set =1 possível *set* independente de d
  - set = rst =0 possível armazenar d

```
IF (rst = '1') THEN
     q <= "000";

ELSIF (set = '1') THEN
     q <= "111";
ELSE
     q <= d
```



Registrador sensível a borda - inicialização assíncrona - (exemplo)



- **Formato geral**: empregando processo
  - sinais ck, rst e set necessitam estar na lista de sensibilidade

```
PROCESS (ck, rst, set)
  BEGIN
    IF       (rst = '1')          THEN q <= '0'; -- eventos assincronos
    ELSIF    (set = '1')          THEN q <= '1'; --    .
                                                 --    .
                                                 --    .
    ELSIF (ck'EVENT AND ck ='1') THEN q <=d;     -- detecta borda de subida de ck
    END IF;
 END PROCESS;
```

Registrador sensível a borda - inicialização assíncrona

- **Exemplo**:
  - 3 bits
  - reset set assíncronos (é necessário inclusão na lista de sensibilidade)

```
 1 ENTITY flip3_3 IS
 2   PORT (ck   : IN  BIT;                -- relogio
 3         rst  : IN  BIT;                -- rst=1 leva q=000 assincrono
 4         set  : IN  BIT;                -- set=1 leva q=111 assincrono
 5         d    : IN  BIT_VECTOR(2 DOWNTO 0);
 6         q    : OUT BIT_VECTOR(2 DOWNTO 0));
 7 END flip3_3;
 8
 9 ARCHITECTURE teste OF flip3_3 IS
10 BEGIN
11   PROCESS (ck, rst, set)
12   BEGIN
13     IF       (rst = '1')          THEN q <="000"; -- q=000 independente de ck
14     ELSIF    (set = '1')          THEN q <="111"; -- q=111 independente de ck
15     ELSIF (ck'EVENT AND ck ='1') THEN q <=d;     -- condicao do sinal relogio
16     END IF;
17   END PROCESS;
18 END teste;
```

Registrador sensível a borda - inic. assíncrona - (circuito nível RTL)

- **Observar**: circuito de seleção contém a prioridade prevista na seqüência de decisões tomadas na construção IF ELSE

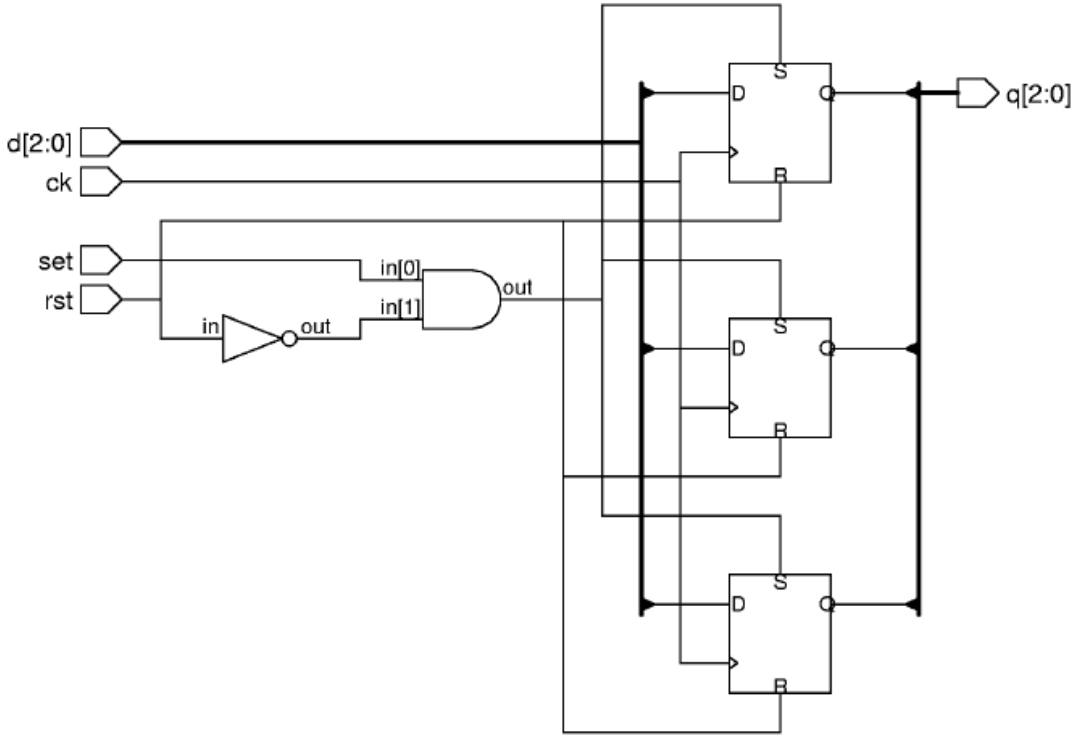
  - rst =1 possível *reset* independente de set

  - set =1 possível *set* independente de d

  - set = rst =0 possível armazenar d

- Estrutura de inicialização assíncrona idêntica:

  - *latch* 3 bits
  (visto anteriormente)



Registrador sensível a borda - inic. assíncrona - exemplo de problema

- **Caso a tecnologia empregada não disponha dos elementos necessários**:
  - *flip-flop* com inicialização assíncrona

- **Exemplo**: Descrição: *flip-flop* contendo inicialização assíncrona

  Tecnologia alvo não é dispõe de *flip-flop* com inicialização assíncrona

  Ferramenta substituiu por *set* síncrono - mensagem de alerta:

```
Warning, asynchronous set on DFF q(2)_EXMPLR replaced by synchronous set.
Warning, asynchronous set on DFF q(1)_EXMPLR replaced by synchronous set.
Warning, asynchronous set on DFF q(0)_EXMPLR replaced by synchronous set.
Warning, Transformations were required because of constraints in target
technology flex6.
Warning, Design may show simulation differences because of transformations.
```

Registrador sensível a borda - inic. assíncrona - exemplo de problema

- **Circuito gerado**:

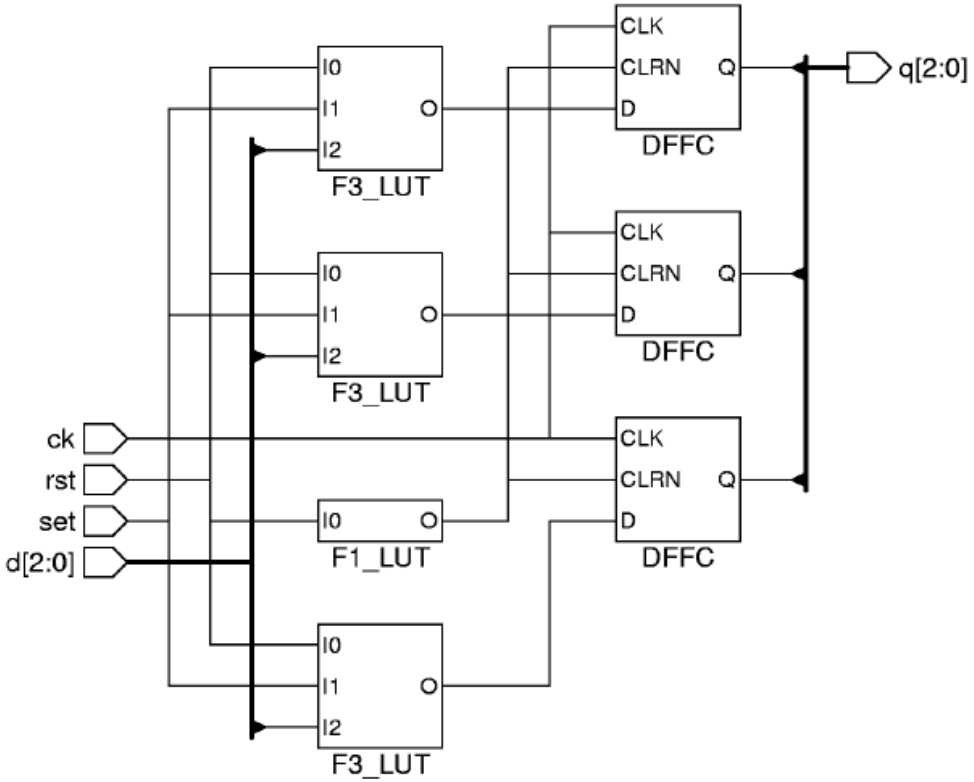
  - caixas F3_LUT

    implementam a função:

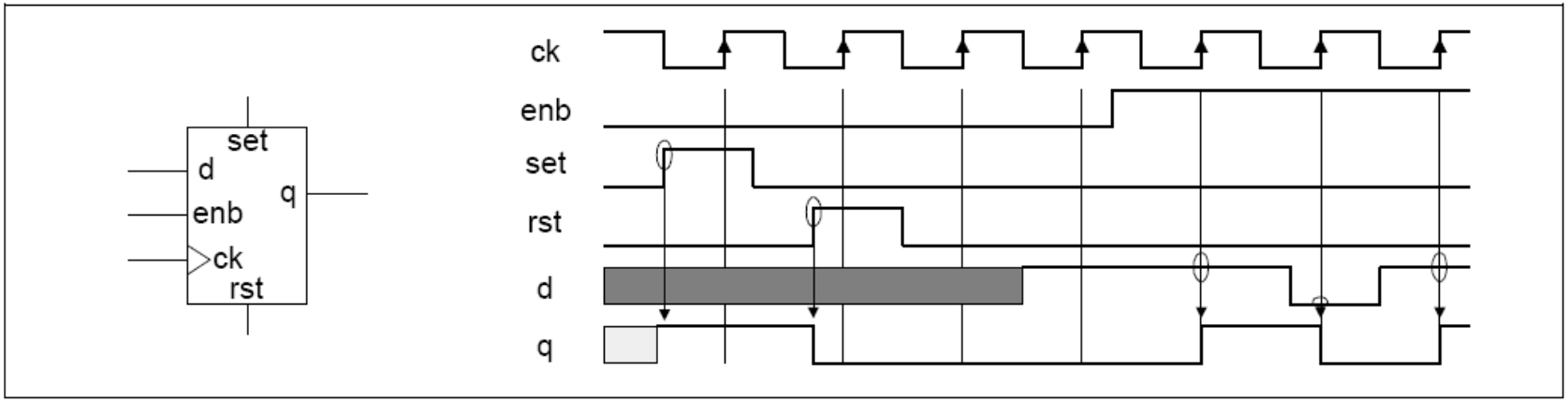
    I2 + (NOT I0 . I1)

  - operação *set* implementada:

    síncrona



Registrador sensível a borda com habilitação para sinal de relógio



- **Formato geral**: empregando processo
  - sinais rst e set necessitam estar na lista de sensibilidade (para inic. assíncrona)
  - sinal enb não é necessário na lista de sensibilidade

```
PROCESS (ck, rst, set)
  BEGIN
    IF       (rst = '1')          THEN q <= '0'; -- eventos assincronos
                                                 --    .
    ELSIF (ck'EVENT AND ck ='1')                 -- detecta borda de subida de ck
      IF (enb = '1') THEN   q <=d;               -- armazena se habilitado
      END IF;
    END IF;
 END PROCESS;
```

Registrador sensível a borda com habilitação para sinal de relógio

- **Exemplo**:
  - *reset* *set* assíncronos (é necessário inclusão na lista de sensibilidade)
  - sinal de habilitação: enb

```
 1 ENTITY flip3_4 IS
 2   PORT (ck        : IN  BIT;      -- relogio
 3         enb       : IN  BIT;      -- habilita sinal de relogio
 4         rst, set  : IN  BIT;      -- rst=1 q=000, set=1 q=111 assincronos
 5         d     : IN  BIT_VECTOR(2 DOWNTO 0);
 6         q     : OUT BIT_VECTOR(2 DOWNTO 0));
 7 END flip3_4;
 8
 9 ARCHITECTURE teste OF flip3_4 IS
10 BEGIN
11   PROCESS (ck, rst, set)
12   BEGIN
13     IF    (rst = '1') THEN q <= "000";    -- q=000 independente de ck
14     ELSIF (set = '1') THEN q <= "111";    -- q=111 independente de ck
15     ELSIF (ck'EVENT AND ck = '1') THEN    -- detecta borda de ck
16       IF  (enb = '1') THEN q <= d;        -- verifica habilitacao
17       END IF;
18     END IF;
19   END PROCESS;
20 END teste;
```

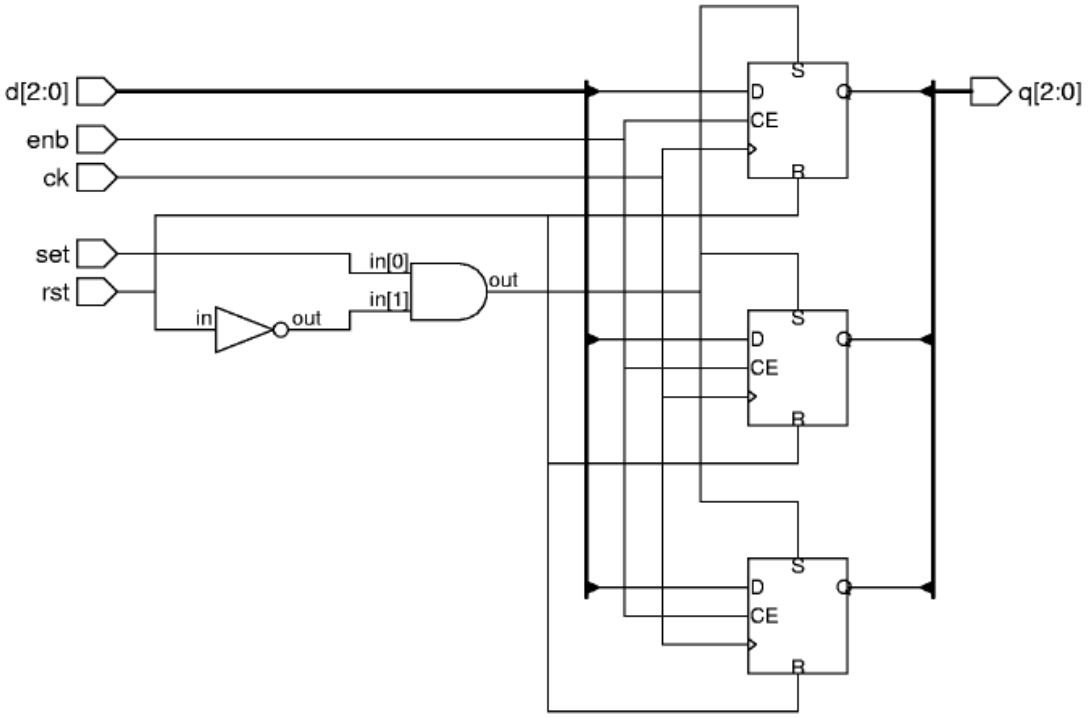
Registrador sensível a borda - habilitação para sinal de relógio - RTL

- **Observar**: circuito de seleção contém a prioridade prevista na seqüência de decisões tomadas na construção IF ELSE

  - rst =1 possível *reset* independente de set
  - set =1 possível *set* independente de d
  - set = rst =0 possível armazenar d caso enb =1

- **Estrutura de inicialização assíncrona idêntica**:

  - *latch* com 3 bits
  - *flip flop* 3 bits



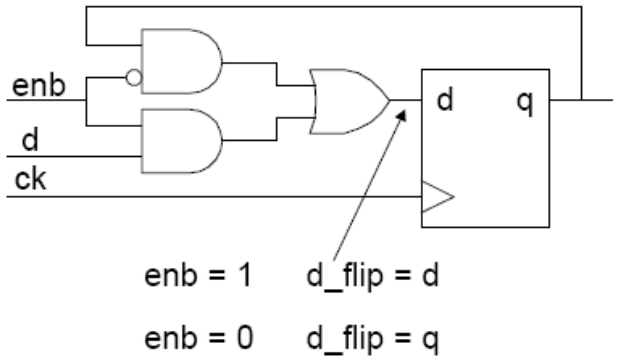
Registrador sensível a borda - habilitação para sinal de relógio

- **Comparação entre códigos** (correto e incorreto)

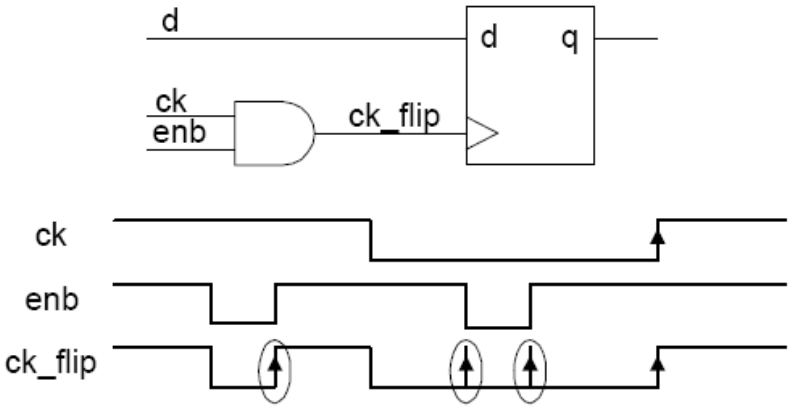
```
-- esquema a
IF (ck'EVENT AND ck ='1') THEN
   IF  (enb ='1') THEN q <= d;
   END IF;
END IF;
```

```
-- esquema b
IF (ck'EVENT AND ck ='1' AND enb ='1')
   THEN q <= d;
END IF;
```

esquema "a"



esquema "b"



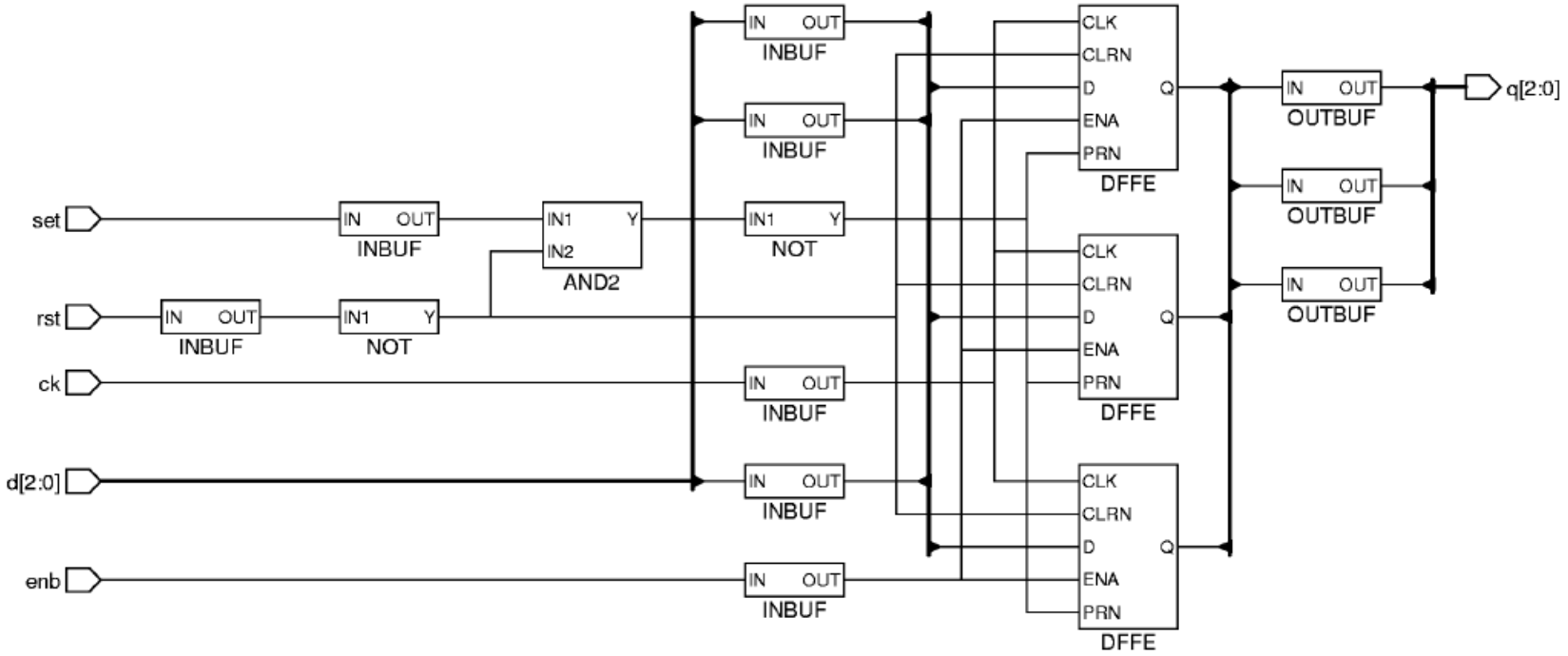
- **Mensagem de erro gerada para o esquema 'b'**

```
/vhdl/flip3_1.vhd",line 15: Error, clock expression should contain only one signal.
```

Registrador sensível a borda - habilitação para sinal de relógio

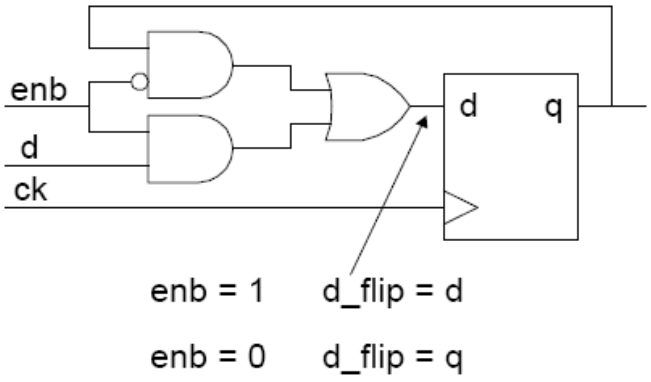
- **Problemas de implementação**: ausência de flip flops com sinal de habilitação

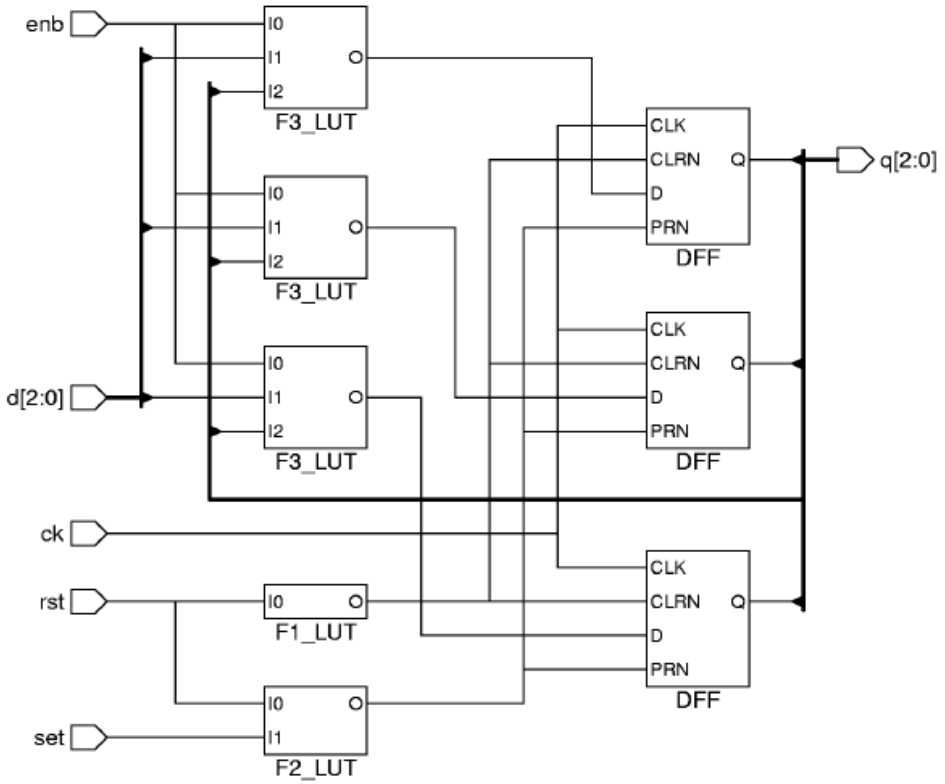
- **Caso 1**:
  - arquitetura alvo dispõe de *flip flops* com sinal de habilitação
  - implementação sem problemas:



Registrador sensível a borda - habilitação para sinal de relógio

- **Problemas de implementação**: ausência de flip flops com sinal de habilitação
- **Caso 2**:
  - arquitetura alvo não dispõe de *flip flops* com sinal de habilitação
  - necessário implementar com lógica adicional
- Caixas F3_LUT realizam a função: $(I0 \cdot I1) + (\overline{I0} \cdot I2)$
- Solução adotada segue o esquema:





Cuidados na descrição - Ciclo de inicialização

- **Projeto de um circuito síncrono**:
  deve sempre prever um ciclo de inicialização

- **O valor dos registradores após a energização do circuito**:
  desconhecido

- **Falta de um ciclo de inicialização**:
  - comportamento não previsto pela simulação: exemplo
    - iniciar estado diferente
    - iniciar estado não previsto na descrição

- **Portanto**:
  - deve-se incluir de operações de inicialização
    - assíncronas ou síncronas

Cuidados na descrição - Inserção desnecessária de um *latch*

- **Na linguagem VHDL**:
  - a falta de atribuição de um valor em um objeto:
    - implica na manutenção do valor neste objeto
- **Problemas**:
  - transição de estados especificada incompletamente
  - falta de atribuição de um valor a uma saída
    - leva a inserção *latch* desnecessário.

- **Construção CASE WHEN na especificação da transição de estados**
  - todas as condições devem ser obrigatoriamente cobertas

- **Evitar cláusula OTHERS**
  - lógica combinacional extra pode ser gerada
    - necessário detectar os estados restantes

- **Opção**:
  - criar tipo enumerado definindo todos os estados previstos
  - não é necessário uso da clausula OTHERS