

Atraso, variável e atributos

Tópicos

- **Atraso**
- **Variável**
- **Atributos**
 - Atributos pré-definidos relativos a sinais
 - Atributos pré-definidos relativos a vetores
 - Atributos pré-definidos relativos a tipos



Atraso

- **Modelagem do atraso:** 2 tipos

- com inércia (*inertial*)
- com transporte (*transport*)

- **Modelagem com inércia:** cláusula **AFTER**

- similar à operação de uma porta lógica
- pulso menor que o atraso: não é propagado

```
signal_a <= valor_x AFTER tempo_1;
```

- **Modelagem com transporte:** cláusulas **TRANSPORT** e **AFTER**

- o pulso é sempre propagado

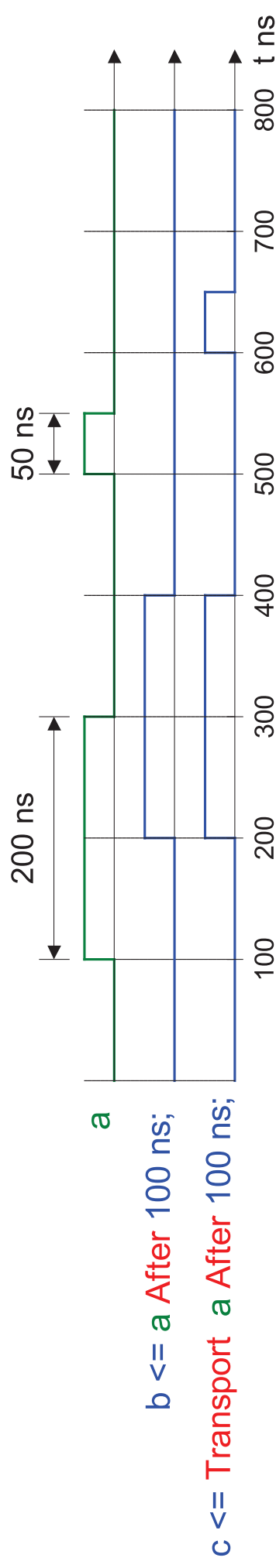
```
signal_d <= TRANSPORT valor_x AFTER tempo_1;
```



Exemplo – Comparação da modelagem com inércia e transporte

- **b**: pulso com duração menor que 100 ns não é propagado
- **c**: pulso sempre propagado

```
1 ENTITY c04_atr1 IS
2   PORT (a : IN BIT;
3         b, c : OUT BIT);
4 END c04_atr1;
5
6 ARCHITECTURE teste OF c04_atr1 IS
7   BEGIN
8     b <= a AFTER 100 ns;
9     c <= TRANSPORT a AFTER 100 ns;
10  END teste;
```



Exemplo – cláusulas com inércia e transporte

- Formato geral das cláusulas

```
senal_a <= valor_1 AFTER tempo_1;  
senal_b <= valor_1, valor_2 AFTER tempo_2, valor_3 AFTER tempo_3;  
senal_c <= expressao_1 AFTER tempo_1, expressao_2 AFTER tempo_2;  
  
senal_d <= TRANSPORT valor_1 AFTER tempo_1;  
senal_e <= TRANSPORT valor_1, valor_2 AFTER tempo_2, valor_3 AFTER tempo_3;  
senal_f <= TRANSPORT expressao_1 AFTER tempo_1;
```

- Ferramentas de síntese

- O atraso não é sintetizado:
 - inexistência de circuitos para implementar o atraso com precisão

- As cláusulas **AFTER** e **TRANSPORT** são ignoradas



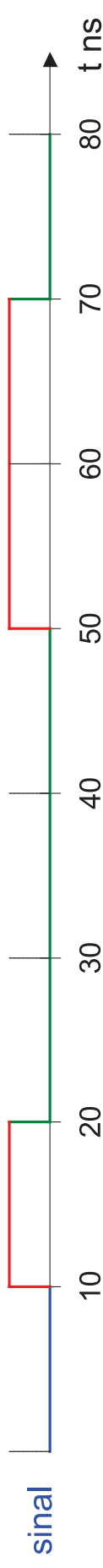
Outra aplicação da cláusula AFTER

- Gerar formas de onda

- estímulos para teste interno (própria entidade)
- entidade especial para teste de outra entidade

- **Exemplo:** geração de uma forma de onda aleatória

```
signal <= '0', '1' AFTER 10 ns, '0' AFTER 20 ns, '1' AFTER 50 ns, '0' AFTER 70 ns ;
```



- **Exemplo:** geração de um sinal periódico

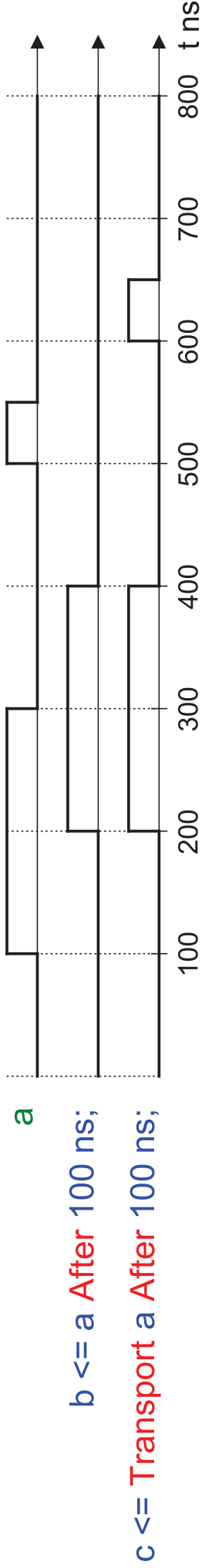
```
x0 <= NOT x0 AFTER 100 ns;
```



Exemplo: geração de formas de onda

- Descrição com estímulo interno para teste

```
1 ENTITY c04_atr2 IS
2   PORT (a : BUFFER BIT;
3         b, c : OUT BIT);
4 END c04_atr2;
5
6 ARCHITECTURE teste OF c04_atr2 IS
7 BEGIN
8   a <='0','1' AFTER 100 ns,'0' AFTER 300 ns,'1' AFTER 500 ns,'0' AFTER 550 ns;
9   b <= a AFTER 100 ns;
10  c <= TRANSPORT a AFTER 100 ns;
11 END teste;
```



Sinal e Variável

- **Sinal e Variável:** objetos (objetos: elementos que armazenam um valor)

- **Sinal:**

- regiões de código concorrente ou sequencial
- atribuição de valor: agendada 1Δ após a avaliação

- **Exemplos:**

```
sinal_2 <= sinal_1;      -- transferencia entre sinais
sinal_3 <= variavel_1;    -- transferencia de variavel para sinal
sinal_4 <= constante_1;  -- transferencia de constante para sinal
```

- **Variável:**

- unicamente regiões de código sequencial
- atribuição de valor: imediata

- **Exemplos:**

```
variavel_2 := variavel_1;  -- transferencia entre variaveis
variavel_3 := constante_1; -- transferencia de constante para variavel
```



Exemplo com sinal e variável

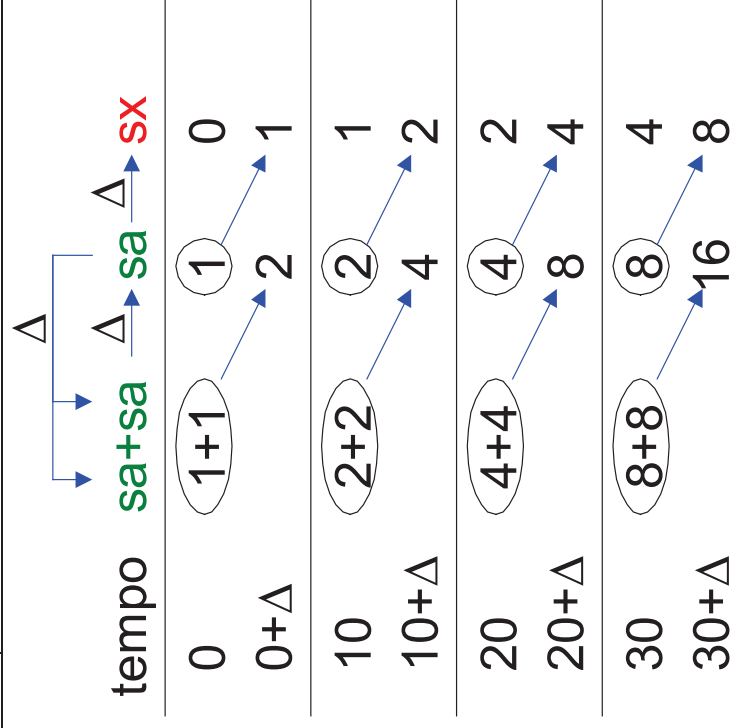
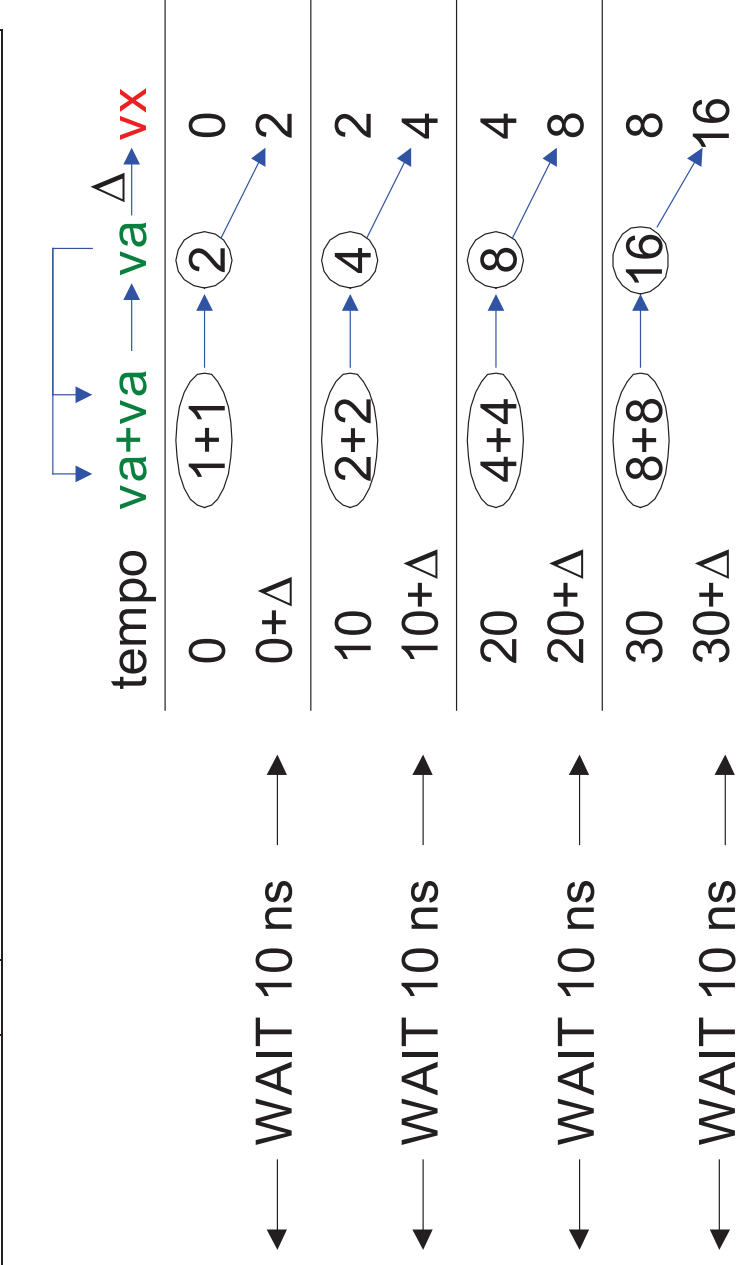
- Dois processos semelhantes: controle → comando **WAIT** (linhas 13 e 21)
- Processos executados a cada **10 ns**
- Mesmas operações: sinais **sa** **sx**
 variáveis **va** **vx**
- Valores iniciais: **sa = 1** **va = 1**

<pre>1 ENTITY sg_var8 IS 2 PORT (sx : OUT INTEGER; 3 vx : OUT INTEGER); 4 END sg_var8; 5 6 ARCHITECTURE teste OF sg_var8 IS 7 SIGNAL sa : INTEGER := 1; 8 BEGIN 9 sig: PROCESS 10 BEGIN 11 sa <= sa +sa; 12 sx <= sa; 13 WAIT FOR 10 ns; 14 END PROCESS sig;</pre>	<pre>16 var: PROCESS 17 VARIABLE va : INTEGER :=1; 18 BEGIN 19 va := va +va; 20 vx <= va; 21 WAIT FOR 10 ns; 22 END PROCESS var; 23 END teste;</pre>
---	---



Exemplo com sinal e variável – Operações executadas

7 8 9 10 11 12 13 14 15	<pre> SIGNAL sa : INTEGER := 1; BEGIN sig: PROCESS BEGIN sa <= sa +sa; sx <= sa; WAIT FOR 10 ns; END PROCESS sig; </pre>	<pre> 16 var: PROCESS 17 VARIABLE va : INTEGER :=1; 18 BEGIN 19 va := va +va; 20 vx <= va; 21 WAIT FOR 10 ns; 22 END PROCESS var; 23 END teste; </pre>
---	---	--



Atributos

- **Informações adicionais associadas a:**

- tipos, objetos, unidades de projeto (entre outros)

- **Exemplo:**

- um objeto: (em um dado instante de tempo)
 - pode conter um único valor
 - pode possuir vários atributos
- o atributo:
 - pode ou não estar relacionado com o valor do objeto

- **Exemplos de atributos:**

- um valor
- um tipo
- uma constante
- um sinal
- os limites de um vetor: (0 TO 3) (7 DOWNT0 0)



Atributos

- Um atributo pode ser:
 - pré-definido
 - definido pelo usuário
- **Formato para referenciar um atributo de um item:**

```
prefixo' nome_atributo
```

- **prefixo:** corresponde ao item
- **apóstrofo:** separa o prefixo do nome_atributo
- **nome_atributo:** o atributo desejado do item
- **Exemplo**
 - item referenciado: sinal **s**

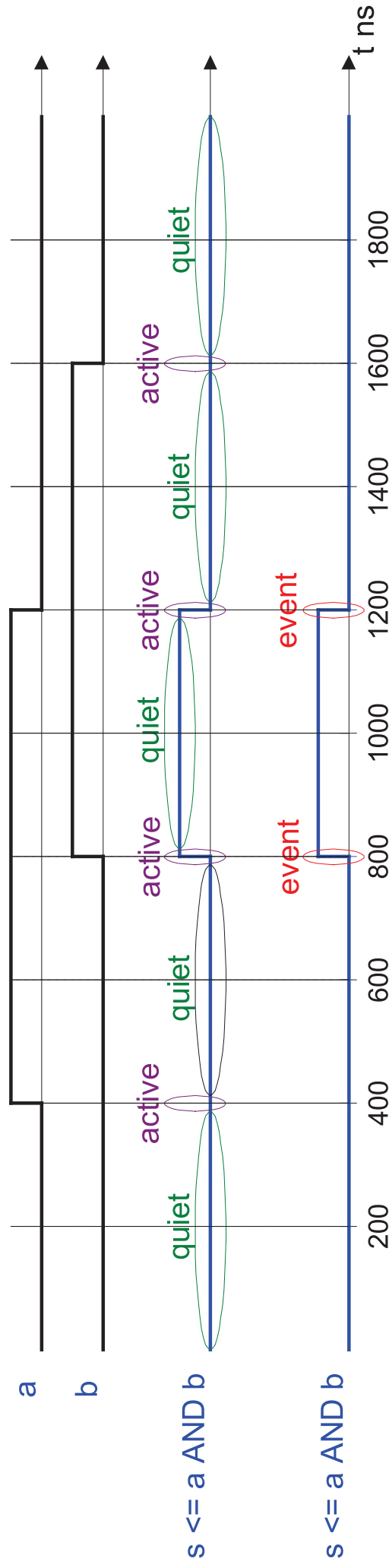
```
s'QUIET    -- referenciado o atributo QUIET do sinal s  
s'STABLE   -- referenciado o atributo STABLE do sinal s
```



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:



Atributos pré-definidos relativos a sinais → não resultam em novos sinais

s'EVENT - verdadeiro:

ocorreu uma troca de valor no ciclo corrente da simulação

- falso:

caso contrário

