

Guia 5

Aluno: Jourdan Henrique Martins Pereira

Matricula: 2018019850

1. Qual a diferença entre os tipos bit e std_logic?

No tipo Bit, temos uma representação idealizada de um sistema digital, onde o mesmo só pode assumir como valor 0 ou 1.

Em contra partida, temos o tipo std_logic, que fornece uma representação mais fiel da realidade no que se diz de um sistema digital. O mesmo pode assumir os seguintes valores:

- 'U': Não inicializado. Este sinal ainda não foi definido.
- 'X': Desconhecido. Impossível determinar este valor / resultado.
- '0': Lógico 0
- '1': Lógica 1
- 'Z': Alta Impedância
- 'W': Sinal fraco, não posso dizer se deve ser 0 ou 1.
- 'L': Sinal fraco que provavelmente deve ir para 0
- 'H': Sinal fraco que provavelmente deve ir para 1
- '-': Tanto faz.

2. Quais pacotes da library “ieee” são extensões definidas por desenvolvedores de ferramentas e devem ser evitadas por não serem pacotes padrão definidos pelo IEEE?

São os pacotes std_logic_arith, std_logic_unsigned e std_logic_signed. Essas bibliotecas foram desenvolvidas para eliminar a necessidade da conversão de dados. Mesmo com essas bibliotecas trazendo uma grande facilidade ao desenvolvedor, o uso delas não é uma boa prática, uma vez que, pode causar problemas de compatibilidade durante o processo de síntese.

3. Como realizar operações aritméticas (soma, subtração,...) usando std_logic_vector?

O tipo std_logic_vector (definido em ieee.std_logic_1164) não permite operações numéricas. Assim é inválida a operação: $q \leq q + 1$.

Os operadores infixos + (adição), - (subtração) e * (multiplicação) são definidos para dois operandos do tipo unsigned. O tamanho do resultado para adição e subtração é o máximo de os tamanhos de seus operandos. Para multiplicação, o tamanho do resultado é a soma dos tamanhos de seus operandos. Qualquer um dos operandos também pode ser do tipo inteiro. O operador infix / (divisão) geralmente é

suportado. Contudo, ele irá gerar hardware caro, a menos que o segundo operando tenha um valor constante que é uma potência de 2.

Sendo assim, para realizar operações com `std_logic_vector`, devemos realizar transformações dos tipos de dados, usando dados do tipo `integer`, `unsigned` e `signed` para fazermos a operação aritmética desejada e posteriormente, converter para o `std_logic_vector`.

4. Onde um signal deve ser declarado?

Signal representa ligações entre elementos. Sua declaração é `signal <identificador>: <tipo> [:= valor];` podem ser declaradas globalmente(`package`) ou internamente (`entity`, `architecture` – é o mais utilizado).

5. Onde uma variable deve ser declarada?

Variable é utilizado para armazenar valores intermediários entre expressões. Sua declaração é `variable <identificador>: <tipo> [:= valor];` podem ser declaradas apenas em processos (variáveis locais).

6. Qual a diferença entre um signal e uma variable?

- As variáveis só podem ser usadas dentro dos processos, os sinais podem ser usados dentro ou fora dos processos.
- Qualquer variável criada em um processo não pode ser usada em outro processo, os sinais podem ser usados em vários processos, embora só possam ser atribuídos em um único processo.
- Variáveis precisam ser definidas após a palavra-chave `process` mas antes da palavra-chave `begin`. Os sinais são definidos na arquitetura antes da instrução `begin`.
- As variáveis são atribuídas usando o símbolo de atribuição `:` `=` . Os sinais são atribuídos usando o símbolo de atribuição `<=` .
- As variáveis atribuídas imediatamente assumem o valor da atribuição. Os sinais dependem se é um código combinacional ou sequencial para saber quando o sinal assume o valor da atribuição.

7. Qual a diferença entre um signal e uma variável de uma linguagem de programação, como C, por exemplo?

O signal é como declarar um fio , sinal de dados, definido a partir dos parâmetros de entrada, pode ser combinacional ou sequencial. Já a variável é sempre sequencial.

8. Qual a diferença entre uma `variable` e uma `variável` de uma linguagem de programação, como C?

A `variable` será computada a partir da lista de sensibilidade do processo, ela é dependente do processo e das suas mudanças (exemplo clock). Já uma `variável` depende apenas do clock do processador.