

Sistemas Digitais

Avaliação 3 – VHDL e PLA

Nome: Mariana da Costa Zatta Matrícula: 2079950

Nome: Kevin Azevedo Lopes Matrícula: 2079909

a) Questionário VHDL

1. O que descreve um código VHDL e o que a partir dele pode ser inferido?

O código descreve o comportamento ou estrutura de um circuito eletrônico, a partir do qual um circuito físico compatível pode ser inferido por um compilador.

2. É possível diferenciar-se a VHDL de outras linguagens de descrição de hardware anteriores como a Verilog ou a AHDL devido a que fator e o que faz tal fator ser importante?

O VHDL permite realizar a síntese do circuito, assim como a sua simulação, fator importante para verificar o funcionamento mesmo em processo de projeto e após a síntese.

3. O que é um código VHDL para síntese?

O código VHDL para síntese é a tradução de um código-fonte em uma estrutura de hardware que implementa a funcionalidade desejada.

4. O que é um código VHDL para simulação?

O código VHDL para simulação é um procedimento de teste para assegurar que a funcionalidade implementada anteriormente pela síntese é de fato alcançada pelo circuito sintetizado.

5. Citar as seis etapas existentes em uma ferramenta de EDA para a síntese de um circuito usando-se VHDL.

1ª etapa: Especificações;

2ª etapa: Código VHDL;

3ª etapa: Análise e síntese;

4ª etapa: Local e rota (montador);

5ª etapa: Gerar arquivo de programação;

6ª etapa: Baixar para dispositivo físico.

6. Por que a simulação temporal só é possível depois da etapa denominada "Place and Routing (Fitter)"?

Porque é nesta etapa (local e rota (montador)) que acontece a simulação de tempo pós-ajuste e é gerada uma netlist funcional.

7. Um circuito físico gerado por uma ferramenta de EDA a partir de um mesmo código VHDL pode, para um mesmo dispositivo de destino, ser diferente caso a caso? Justifique.

Sim, na simulação tudo ocorre como programado, porém no circuito físico, podem ocorrer atrasos de propagação afetando o sistema.

8. Um inteiro em VHDL tem quantos bits?

No VHDL, um inteiro possui 32 bits.

9. Expressarmos um número inteiro utilizando a notação 3_250 é permitido?

Sim, essa notação é permitida, podendo ser usado em qualquer casa do número, sem mudar o valor sintetizado.

10. Podemos expressar em VHDL um inteiro em bases diferentes da decimal, por exemplo, base 2, base 7, etc?

Sim, podemos expressar em bases que variam entre 2 e 16, sendo assim o exemplo acima pode ser expresso em VHDL.

11. Qual a forma de representação de número binário com sinal é usada em VHDL para números negativos?

A representação mais usual para números negativos é o complemento de 2. Se o MSB (por convenção o bit mais significativo é o mais à esquerda) é 0, então o número é positivo, se for 1, é negativo.

12. Escreva em quatro bits, os números 6 e -6 utilizando esta representação. Pode-se buscar uma calculadora online.

O número +6 é igual a 0110, e o número -6 é igual a 1010.

b) Questionário PLA

1. Fazer uma tabela relacionando cada modo e configuração para a OLMC de uma GAL juntamente com seus bits de controle (SYN, ACO, AC1). Identifique quais OLMCs podem ser usadas em cada configuração.

	OLMC – modo simples	OLMC – modo complexo	OLMC – modo registrador
SYN	1	1	0
AC0	0	1	1
AC1	0 – Saída combinacional dedicada 1 – Entrada dedicada	1	0 – Configuração síncrona 1 – Configuração Combinacional
XOR (pinos de saída)	0 – Ativo Baixo 1 – Ativo Alto	0 – Ativo Baixo 1 – Ativo Alto	0 – Ativo Baixo 1 – Ativo Alto
Configuração	Saída combinacional dedicada: Possui seu inversor tristate sempre habilitado. Todos os pinos podem ser usados Entrada dedicada: Conecta-se um pino de I/O de uma OLMC na matriz de entrada através do redirecionamento do sinal por uma OLMC adjacente. Pinos 15 e 16 não podem ser usados	Pinos 13 a 18 são realimentados para a matriz de entrada. Pinos 12 e 19 não podem ser realimentados nem ser usados como entrada. Pinos 11 e 1 podem ser usados como entrada.	Combinacional: A habilitação da saída é controlada por uma expressão produto da matriz de entrada, fornecendo habilidade de usar sinais bidirecionais em aplicações de barramentos. Síncrona: Utiliza um flip-flop D em cada OLMC para sincronizar todas as saídas dos flip-flops à transição de um sinal de clock comum

2. Escrever todos os modos que podem ser usados para se implementar os recursos a seguir:

a) Um latch D com saídas tristate

Modo complexo, simples e registrador.

b) Um flip-flop D

Modo registrador

c) Um contador síncrono

Modo registrador

d) Um circuito decodificador

Modo simples, complexo e registrador

3. Para cada modo e configuração, dizer qual é o número máximo de termos em uma expressão do tipo soma de produtos.

Modo simples saída dedicada: 8 termos;

Modo complexo: 7 termos;

Modo registrador combinacional: 7 termos

4. Identificar a fonte do sinal de realimentação (FMUX) para cada modo e configuração.

Modo simples: saída realimentada, realimenta uma matriz de entrada por meio de uma célula adjacente;

Modo complexo: saída com realimentação, é realimentado um estado existente para a matriz de entrada;

Modo registrador: com registrador de deslocamento, flip-flop D, realimentado para a matriz de entrada.

5. Identificar a fonte do sinal de habilitação das saídas tristate (ISMUX) para cada modo e configuração.

Modo	Habilitação	Configuração
Simplex	GND Vcc	Entrada Saída
Complexo	8º termo produto da matriz de entrada	-
Registrador	Pino 11 (\overline{OE}) 8º termo produto	Com registrador Combinacional

6. Mostrar que se for programado o bit XOR em nível ALTO, o nível do pino de saída será o mesmo que o da saída da porta OR na OLMC para os Exemplos dos itens 12-1, 12-2 e 12-3.

Por meio dos compiladores lógicos, conseguimos com que o nosso projeto seja expresso em forma de tabela-verdade, realizamos a redução lógica, a geração das expressões lógicas e a produção do tipo de saída que pretendemos. Ao levarmos em conta que o XOR é equivalente ao OR, temos que sua saída também será a mesma.

7. Relacionar os equipamentos necessários para desenvolver circuitos usando PLDs.

São necessários para construir e projetar circuitos usando PLDs, um computador, um software para acionar o programador, dispositivo de lógica programável, o programador, software de desenvolvimento para PLDs.

Com esses elementos é possível construir o circuito. O ponto de partida é o projeto inicial fornecido como entrada ao software de desenvolvimento, esse software traduz o projeto inicial em um arquivo chamado “mapa de fusíveis”, o mapa é transformado para outro arquivo de saída cujo formato é mais adequado para a sua transmissão para o programador. O software de programação se comunica com o programador é então chamado no computador, para que seja informado que tipo de dispositivo será programado, o arquivo de saída será enviado através de um cabo para o

programador, para finalmente o PLD ser colocado no soquete do programador e um comando é enviado do computador para programar o PLD.

8. Escrever um arquivo de entrada que programe uma GAL 16V8A para atuar como um decodificador 74LS138 de 3x8.

/* Entradas */

pin 1 = C;

pin 2 = B;

pin 3 = A;

/* Saídas */

pin 19 = Z;

pin 18 = Y;

pin 17 = X;

pin 16 = W;

pin 15 = V;

pin 14 = U;

pin 13 = T;

pin 12 = S;

/* Cálculos */

Z = !A & !B & !C;

Y = !B & !C;

X = !A & !C;

W = !C;

V = !B & !A;

U = !B;

T = !A;

S = 1;