

Nome: [REDACTED]

Notas Importantes!

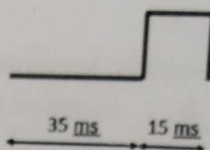
1. Verifique, para todas as questões, qual a resposta correta e assinale com um "X" a sua escolha na tabela ao lado. Por cada resposta incorreta será descontada, à cotação global, 1/3 da cotação da respetiva pergunta.
2. Durante a realização do teste não é permitida a permanência junto do aluno, mesmo que desligado, de qualquer dispositivo eletrónico não expressamente autorizado (nesta lista incluem-se calculadoras, telemóveis, smartwatches e qualquer outro dispositivo de captura de imagem e/ou comunicação). A sua deteção durante a realização do exame implica a imediata anulação do mesmo.
3. Não é permitido escrever na área branca em torno da matriz de respostas.
4. Cotações: Grupo I (24 perguntas) – cada 0.5 valores; Grupo II (8 perguntas) – cada 1 valor.

Grupo I

1. A figura representa um período de um sinal. A sua frequência é:

- ☒ a) 50 MHz $\rightarrow 20 \text{ ns}$ ☒ c) 20 Hz
☒ b) 20 kHz $\rightarrow 20 \text{ 000 Hz}$ ☒ d) nenhuma das anteriores

$50 \text{ ms} = 0.05 \text{ s}$



$f = \frac{1}{T} = \frac{1}{0.05} = 20 \text{ Hz}$

2. O duty-cycle do sinal da figura anterior é:

- ☐ a) 15% ☒ b) 30% ☐ c) 35% ☐ d) 50%

3. A lookup table (tabela de verdade) da figura ao lado implementa a seguinte função lógica:

- ☒ a) $z \leq a \text{ and } b$;
☒ b) $z \leq a \text{ or } b$;
☐ c) $z \leq a \text{ nor } b$;
☒ d) $z \leq a \text{ xor } b$;

Ligação da LUT		Posição (I2 I1 I0)	Out = f(I2, I1, I0)
a	I0	000	0
b	I1	001	1
		010	1
		011	1
'0'	I2	100	0
		101	1
		110	1
		111	0

4. Uma LUT 4:1 pode ser programada para implementar:

- ☒ a) um decodificador binário 1:2. ☒ c) um codificador binário 4:2.
☒ b) um multiplexer 2:1 de 1 bit. ☐ d) nenhum dos componentes das alíneas anteriores.

5. Em VHDL, qual das opções seguintes permite converter um inteiro (integer) para std_logic_vector?

- ☒ a) função to_integer ☐ c) macro de conversão std_logic_vector
☒ b) função to_unsigned ☒ d) nenhuma das opções anteriores

6. Um processo (process) em VHDL:

- ☐ a) é um comando sequencial. ☒ c) pode ser usado dentro de um outro processo.
☒ b) é um comando concorrente. ☒ d) precisa de incluir na sua lista de sensibilidade todos os sinais que são modificados dentro do processo.

7. Deve-se passar todos os sinais de entrada da entidade top-level por registos (flip-flops), com o objetivo de:

- ☒ a) sincronizá-los e reduzir problemas relacionados com meta-estabilidade e inconsistência de valores.
☐ b) fazer o debouncing desses sinais.
☐ c) garantir um duty-cycle de 50% em todos os sinais de entrada.
☐ d) garantir o reset de todos os componentes do sistema.

8. Qual dos identificadores seguintes de VHDL é um identificador válido:

- ☐ a) Program# $\rightarrow \#$ não é permitido ☐ c) attribute \rightarrow palavra reservada
☐ b) 2:4dec \rightarrow ':' não é permitido ☒ d) xYz_12

9. Considerando a definição do tipo: `type my_type is ('A', 'B', 'C', 'D', 'E');` o resultado da expressão `my_type'val(2)` é:

- a) 3
b) 'D'
c) 'C'
d) indefinido
10. Se um circuito liga diretamente a entrada do botão KEY(1) com o LEDG(0), então
- a) LEDG(0) pisca com a frequência de relógio da placa
b) LEDG(0) ascende quando o botão está pressionado
c) LEDG(0) não muda com o estado do botão porque falta um circuito de *debouncing*
d) LEDG(0) ascende quando o botão está solto

11. Uma memória RAM de dois portos foi parametrizada com duas constantes genéricas: X - número de bits nos barramentos de endereço e Y - número de bits nos barramentos de dados. Para instanciar uma RAM 64kx32 os parâmetros X e Y devem ter valores seguintes:

- a) X=16 e Y=32.
b) X=32 e Y=32.
c) X=32*2¹⁰ e Y=32.
d) nenhum dos pares de valores anteriores.
- 64k palavras

12. A partir do extrato de código seguinte correto, determine o tipo do sinal `sig`.

```
signal s_a, s_b : signed(3 downto 0);
...
s_a <= "1110"; s_b <= "0110"; sig <= s_a + s_b;
```

-8 + 4 = -4 *0* *-2 + 6 = 4* *0100*

- a) signed(4 downto 0)
b) signed(3 downto 0)
c) std_logic_vector(7 downto 0)
d) é impossível determinar inequivocamente o tipo de `sig`

13. Continuando a análise do código da pergunta 12, determine o resultado de execução da linha seguinte:

```
to_integer(s_a * s_b);
```

-2 x 6 = -12

- a) 84
b) -60
c) -36
d) -12

14. A gama de representação de um sinal declarado como `unsigned(4 downto 0)` é:

- a) [0₁₀, 15₁₀]
b) [0₁₀, 31₁₀]
c) [-32₁₀, 31₁₀]
d) [-15₁₀, 16₁₀]

15. Considerando o extrato de código seguinte, cuja intenção é descrever uma *latch* tipo D, pode-se afirmar que:

```
process(clk)
begin
    if (clk = '1') then    dataOut <= dataIn;    end if;
end process;
```

- a) simula, sintetiza e funciona corretamente em *hardware*
b) não simula corretamente, apesar de sintetizar e funcionar corretamente em *hardware*
c) simula corretamente, mas sintetiza e funciona incorretamente em *hardware*
d) não simula corretamente e sintetiza e funciona incorretamente em *hardware*

16. O comando VHDL seguinte descreve um:

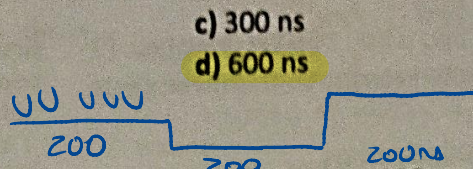
```
dataOut <= std_logic_vector(shift_left(unsigned(dataIn), s_shAmount));
```

- a) registo de deslocamento à esquerda de `dataIn` 'length bits
b) registo de deslocamento à esquerda de `s_shAmount` bits
c) barrel shifter de `s_shAmount` 'length bits
d) barrel shifter de `dataIn` 'length bits

17. Numa memória RAM de dois portos, em que barramentos de endereço são de 4 bits e os de dados são de 8 bits, o número total de bits de dados armazenados é:

18. O processo VHDL ao lado gera um sinal periódico do tipo `std_logic`, cujo período é:

- a) 100 ns
- b) 200 ns

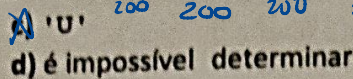


```
stim_proc : process
begin
    wait for 200 ns;
    s_X <= '0';
    wait for 200 ns;
    s_X <= '1';
    wait for 200 ns;
end process;
```

19. Continuando a análise do código da pergunta 18, nos 1100 ns do tempo de simulação o sinal `s_X` vai ser igual a:

- a) '0'
- c) '1'

d) é impossível determinar



20. Continuando a análise do código da pergunta 18, a síntese do mesmo resulta:

- a) num divisor de frequência
- b) em erro pois o código não é sintetizável
- c) numa linha de atraso
- d) num gerador de pulsos

21. A síntese do código VHDL ao lado resulta:

- a) num somador de N bits
- b) num codificador binário $2^N:N$
- c) num decodificador binário $N:2^N$
- d) num multiplexer de $2^N:1$

```
entity xyz is
    generic(N : positive := 3);
    port(x : in std_logic_vector(N-1 downto 0);
          y : out std_logic_vector(2**N-1 downto 0));
end xyz;
architecture Behavioral of xyz is
begin
    out_loop: for i in 0 to 2**N-1 generate
        y(i) <= '1' when (i=to_integer(unsigned(x))) else '0';
    end generate;
end Behavioral;
```

22. Relembre o estilo de codificação de uma máquina de estados finitos baseado em dois processos: um sequencial e um combinatório. O trecho de código à direita corresponde a um desses processos. Pode afirmar-se que:

- a) qq2 representa o estado atual da MEF.
- b) qq1 representa o estado atual da MEF.
- c) MMM representa o estado atual da MEF.
- d) qq2 representa o estado seguinte da MEF.

```
sync_proc: process(clk)
begin
    if (rising_edge(clk)) then
        if (reset = '1') then
            qq2 <= MMM;
        else
            qq2 <= qq1;
        end if;
    end if;
end process;
```

23. Continuando a análise do código da pergunta 22, o segundo processo deve obrigatoriamente incluir na sua lista de sensibilidade:

- a) o sinal `clk`
- b) o sinal `qq1`
- c) o sinal `reset`
- d) o sinal `qq2`

Slide 6 (p.6)

24. A síntese do processo VHDL ao lado:

- a) corre sem problemas
- b) produz um erro de "multiple-source drive"
- c) infere uma latch
- d) infere duas latches

```
process(decodIn)
begin
    validOut <= '1';
    if (decodIn(1) = '1') then
        encodOut <= '1';
    elsif (decodIn(0) = '1') then
        encodOut <= '0';
    else validOut <= '0';
        encodOut <= '-';
    end if;
end process;
```