

Épreuve de VHDL L3 SPI, Mardi 21 Mai 2024, Durée 1h00 Les Documents sont autorisés et Les Calculatrices, téléphones, objets connectés, et ordinateurs sont interdits.
 Par question, la somme des réponses justes est de 1 point et la somme des réponses fausses est de -1 point.
 Plusieurs réponses possibles, Ne pas plier la copie, Bien noircir les cases justes, ou faire de belle croix, bien visible, en cas d'erreur blanchir la case sans refaire le cadre.

Nom :							Note :						
Prénom :													
1	1	2	3	4	5	6	1	2	3	4	5	6	
2	1	2	3	4	5	6	1	2	3	4	5	6	
3	1	2	3	4	5	6	1	2	3	4	5	6	
4	1	2	3	4	5	6	1	2	3	4	5	6	
5	1	2	3	4	5	6	1	2	3	4	5	6	
6	1	2	3	4	5	6	1	2	3	4	5	6	
7	1	2	3	4	5	6	1	2	3	4	5	6	
8	1	2	3	4	5	6	1	2	3	4	5	6	
9	1	2	3	4	5	6	1	2	3	4	5	6	
10	1	2	3	4	5	6	1	2	3	4	5	6	
11	1	2	3	4	5	6	1	2	3	4	5	6	
12	1	2	3	4	5	6	1	2	3	4	5	6	
13	1	2	3	4	5	6	1	2	3	4	5	6	
14	1	2	3	4	5	6	1	2	3	4	5	6	
15	1	2	3	4	5	6	1	2	3	4	5	6	
16	1	2	3	4	5	6	1	2	3	4	5	6	
17	1	2	3	4	5	6	1	2	3	4	5	6	
18	1	2	3	4	5	6	1	2	3	4	5	6	
19	1	2	3	4	5	6	1	2	3	4	5	6	
20	1	2	3	4	5	6	1	2	3	4	5	6	

- 1) Quel est le nom des FPGA développés par la société Xilinx ?
 1 Cyclone et Max 2 Agilex et Stratix 3 ICF
 4 Spartan et Artix 5 Virtex et Kintex 6 ZYNQ
- 2) Quel sont les logiciels développés par la société Xilinx, pour programmer les FPGA Xilinx ?
 1 Quartus 2 ISE 3 Vivado
 4 Diamond 5 Microsemi 6 Mplab
- 3) Quel sont les autres logiciels permettant le développement VHDL ?
 1 VHDLsoft 2 HDL Coder 3 CodeFPGA
 4 DevVHDLpro 5 Leonardo 6 Cadence
- 4) Un composant en VHDL synthétisable est au minimum composé :
 1 D'une architecture 2 D'une instance de composant 3 D'un process
 4 D'instructions concurrentes 5 D'une entité 6 D'un port
- 5) Un composant en VHDL est non synthétisable s'il possède :
 1 Une entité sans port 2 Un port sans out 3 Un port sans in
 4 L'instruction wait 5 L'instruction when 6 Un signal mis à l'état : 'U'
- 6) En VHDL, une entité peut contenir :
 1 La déclaration de library 2 La déclaration d'un generic 3 La déclaration de signal
 4 La déclaration d'un process 5 La déclaration des composants utilisés 6 La déclaration du port
- 7) En VHDL, un process, peut contenir :
 1 Une architecture 2 Un autre process 3 Des instructions séquentielles
 4 Des instructions concurrentes 5 Des instances de composants 6 un port
- 8) En VHDL entre Architecture et Begin, on peut trouver :
 1 Des déclarations de composants 2 Des instructions séquentielles 3 Des instances de composants
 4 Des instructions concurrentes 5 Des déclarations de signaux internes 6 La déclaration d'une entité
- 9) En VHDL, les instructions concurrentes sont :
 1 Une affectation permanente 2 Une affectation permanente avec when/else ou with/select
 3 Une affectation différée 4 Une affectation différée avec if/when/else, case/when
 5 Exécutées l'une après l'autre 6 Obligatoirement combinatoire
- 10) En VHDL, les instructions séquentielles sont :
 1 Une affectation permanente 2 Une affectation permanente avec when/else ou with/select
 3 Une affectation différée 4 Une affectation différée avec if/when/else, case/when
 5 Une affectation différée 6 Une affectation différée avec le même signal
- 11) En VHDL deux process différents peuvent modifier le même signal
 1 Non 2 Oui
- 12) Quel sont les différences entre use IEEE.STD_LOGIC_UNSIGNED.ALL; et use IEEE.STD_LOGIC_SIGNED.ALL; ?
 Donner les bonnes réponses :
 1 Les instructions d'affectation permanentes donnent des résultats différents
 2 Les instructions d'affectation différées donnent des résultats différents
 3 Les instructions de comparaison donnent des résultats différents
 4 Les instruction logique donnent des résultats différents
 5 L'instruction rising edge donne des résultats différents
 6 aucune de ces propositions n'est vrai, c'est autre chose, la différence est ailleurs
- 13) En VHDL peut-il y avoir plusieurs process dans la même architecture ?
 1 Non 2 Oui
- 14) En VHDL peut-il y avoir plusieurs instructions concurrentes qui modifient le même signal ?
 1 Non 2 Oui
- 15) En VHDL, l'instruction Wait est-elle synthétisable ?
 1 Non 2 Oui
- 16) Avec le logiciel ISE l'instruction := est-elle synthétisable ?
 1 Non 2 Oui

17)	Quel est la syntaxe pour l'opération qui réalise la puissance dans la norme IEEE? use IEEE.math_reallall; ?	1 a 4	2 5	3 \$	4 pow(x,y)	5 \$	6 //
18)	Examiner le morceau de code VHDL 18 et cocher les bonnes réponses : Il y a une erreur sur : "when others" il manque quelque chose, par exemple S<= "ZZZZ"; Il y a juste un "warning" sans erreur, car il y a deux signaux asynchrones dans ce process Il y a une erreur ici : "when '10' '11' =>" cela n'existe pas cette barre Il manque un ";" ou il y en a un en trop quelque part Le signal A doit être un std_logic_vector de 4 bits Le signal Rst_En_signal doit être un std_logic_vector de 4 bits	1 2 3 4 5 6	2 5	3 \$	4 pow(x,y)	5 \$	6 //
19)	Examiner le process VHDL 19 et cocher les bonnes réponses : Ce code est synthétisable, il n'y a pas d'erreur, il s'agit d'une porte AND Le composant créé est 100% combinatoire Le composant créé est analogique	1 3 5	2 5	3 \$	4 pow(x,y)	5 \$	6 //
20)	Examiner le morceau de code VHDL 20 et cocher les bonnes réponses : Le process est actif uniquement sur front montant Il s'agit d'une bascule D avec un multiplexeur Le signal Rst est asynchrone	1 3 5	2 5	3 \$	4 pow(x,y)	5 \$	6 //
21)	Examiner le code VHDL 21, s'agit-il d'une multiplication ?	1	2	3	4	5	6
22)	Examiner le code VHDL 21, s'agit-il d'un additionneur ?	1	2	3	4	5	6
23)	Examiner le code VHDL 21, addition, ou multiplication, ou autre : les nombres négatifs ne sont pas gérés Effectivement, binaire naturel uniquement	1	2	3	4	5	6
24)	Examiner le code VHDL 21, il y a-t-il une erreur.	1	2	3	4	5	6
25)	Examiner le code VHDL 21, les débordements sont-ils gérés	1	2	3	4	5	6
18)	<pre> Rst_En_signal <= Rst & En; Process(Rst_En_signal, clk) begin case Rst_En_signal is when '10' '11' => S <= "0000"; when '01' => S <= "1111"; when '00' => S <= A; if (clk='1' and clk'event) then S <= A; end if; when others => end case; end process; End process; </pre>	19)	<pre> Process(Rst, clk) begin if (A='1') then Q <= B; else Q <= '0'; end if; S <= A; else S <= B; end if; End if; End process; </pre>	20)	<pre> Library IEEE; Use IEEE.Std_Logic_1164.all; Entity TOTO is port(A,B : in std_logic; Q : out std_logic); end TOTO; Architecture TITI of TOTO is Begin Q <= A when B='1' else not A; end; </pre>	34)	<pre> Library IEEE; Use IEEE.Std_Logic_1164.all; Entity TOTO is port(A,B : in std_logic; Q : out std_logic); end TOTO; Architecture TITI of TOTO is Begin Q <= A when B='1' else not A; end; </pre>

31)	En VHDL, les nombres en virgule flottante sont de -1.0E38 a 1.0E38 :	1 oui	2 non	3 \$	4 pow(x,y)	5 \$	6 //
32)	Quel est en VHDL la bonne syntaxe pour une comparaison de type « inférieur ou égal » if (a<=b) if (a<=b) if (a<b or a=b) if (a<b or a=b)	1 1 4	2 5	3 \$	4 pow(x,y)	5 \$	6 //
33)	Soit le code VHDL 33, ce code est synthétisable il n'y a pas d'erreur et le test bench est défini comme suivant : Arrive à la valeur 100ns, A prend la valeur de 105 soit "01101001" et ne change plus : Quel est la valeur de Result à 150 ns "01101001" "00000000" "00000000"	1	2	3	4	5	6
34)	Examiner le composant VHDL 34 :	1	2	3	4	5	6
1	Ce code est synthétisable, il n'y a pas d'erreur, il s'agit d'une porte non xor	1	2	3	4	5	6
2	Ce code n'est pas synthétisable, il y a une erreur	1	2	3	4	5	6
3	Le composant créé est 100% combinatoire	1	2	3	4	5	6
4	Le composant créé est analogique	1	2	3	4	5	6
5	Le composant créé est séquentiel synchrone	1	2	3	4	5	6
6	Il s'agit d'une bascule D	1	2	3	4	5	6
35)	En VHDL, le type : Integer est-il synthétisable, même si on l'utilise dans un port ?	1	2	3	4	5	6
36)	En VHDL, le type : char est-il synthétisable ?	1	2	3	4	5	6
37)	En VHDL, deux instructions concurrentes peuvent modifier le même signal en utilisant des conditions différentes	1	2	3	4	5	6
38)	Quel est l'opérateur en VHDL qui permet de réaliser un décalage à droite ou à gauche, c'est-à-dire une multiplication ou une division par une puissance de 2. << et >> gapr et gapl	1 4	2 5	3 \$	4 pow(x,y)	5 \$	6 //
39)	Quel est l'extension du fichier, utilisé pour la simulation, c'est-à-dire le test bench ? tb vhdl	1 4	2 5	3 \$	4 pow(x,y)	5 \$	6 //
40)	Lequel des composants suivants ne peut pas être modélisé avec des instructions concurrentes uniquement ?	1	2	3	4	5	6
1	Un Multiplexeur	1	2	3	4	5	6
2	Un Décodeur	1	2	3	4	5	6
3	Un Additionneur	1	2	3	4	5	6
4	Un Compteur	1	2	3	4	5	6
21)	<pre> Library IEEE; use IEEE.STD_LOGIC_1164.ALL; use IEEE.STD_LOGIC_ARITH.ALL; use IEEE.STD_LOGIC_UNSIGNED.ALL; entity Inconnu is port (A : in std_logic_vector(7 downto 0); B : in std_logic_vector(7 downto 0); S : out std_logic_vector(15 downto 0)); end Inconnu; architecture Behavioral of Inconnu is signal B_integer : integer range 0 to 255; type C_mem is array (integer range 0 to 255) of integer range 0 to 65535; signal C : C_mem; begin B_integer <= conv_integer(B); C(0) <= 0; adder_gen : for i in 0 to 254 generate C(i+1) <= C(i) + conv_integer(A); end generate; S <= conv_std_logic_vector(C(B_integer), 16); end Behavioral; </pre>	33)	<pre> Library IEEE; use IEEE.STD_LOGIC_1164.ALL; entity test_V1 is port (A : in std_logic_vector(7 downto 0); Result : out std_logic_vector(7 downto 0)); end test_V1; architecture Behavioral of test_V1 is begin process(A) begin Result <= Result tmp; end process; end Behavioral; </pre>				