

**Controle VHDL du jeudi 21 février 2019**

<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0
<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1
<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2
<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3
<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4
<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5
<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6
<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7
<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8
<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9

← Codez votre numéro d'étudiant ci-contre.

Nom et prénom :

Durée : 1 heure.

Aucun document n'est autorisé. L'usage de la calculatrice et des téléphones est interdit.

Il faut remplir complètement les cases avec un stylo noir pour les bonnes réponses aux questions.

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Réponse juste = 1 point. Réponse fausse : -0.5 point.

Si vous ne savez pas, il ne vaut mieux pas répondre !

Question 1 D'après la déclaration des signaux du circuit 9 (voir annexes) et des exemples fournis, choisissez l'instruction qui permet de réaliser la conversion INTEGER → STD_LOGIC_VECTOR

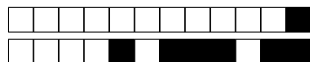
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N));
- ☐ V <= TO_SIGNED(N,8);
- ☐ N <= TO_SIGNED(V);
- ☐ V <= TO_SIGNED(STD_LOGIC_VECTOR(N));
- ☐ V <= STD_LOGIC_VECTOR(S);
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N,8));
- ☐ N <= SIGNED(TO_INTEGER(V));
- ☐ N <= TO_INTEGER(V);
- ☐ N <= TO_INTEGER(SIGNED(V));
- ☐ S <= TO_SIGNED(N,8);

Question 2 Les performances des ASIC (vitesse, low-power) sont meilleures que les FPGA ?

☐ Vrai ☐ Faux

Question 3 ♣ Quels sont les langages qui permettent de décrire un circuit logique (langages de description matérielle) ?

☐ JAVA ☐ Verilog ☐ Assembleur ☐ VHDL ☐ Langage C



Question 4 Les FPGA sont reconfigurables (programmable) ?

☐ Faux ☐ Vrai

Question 5 On considère un signal de type std_logic avec trois drivers. Donnez la valeur résolue de ce signal en fonction des valeurs des trois drivers. Rappel de la fonction de résolution du std_logic en annexe. Valeurs des drivers : '1','Z','L'. Quelle est la valeur résolue ?

☐ 'L' ☐ 'Z' ☐ 'W' ☐ '1' ☐ 'X'

Question 6 D'après la déclaration des signaux du circuit 9 (voir annexes) et des exemples fournis, choisissez l'instruction qui permet de réaliser la conversion INTEGER -> SIGNED

- ☐ V <= TO_SIGNED(N,8);
☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N));
☐ V <= STD_LOGIC_VECTOR(S);
☐ N <= TO_INTEGER(SIGNED(V));
☐ N <= TO_INTEGER(V);
☐ S <= TO_SIGNED(N,8);
☐ N <= SIGNED(TO_INTEGER(V));
☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N,8));
☐ V <= TO_SIGNED(STD_LOGIC_VECTOR(N));
☐ N <= TO_SIGNED(V);

Question 7 ♣ Choisissez les spécifications qui caractérisent le circuit 3 (voir annexes)

- | | |
|---|--|
| <input type="checkbox"/> Registre à décalage | <input type="checkbox"/> Bascule D Latch |
| <input type="checkbox"/> Multiplexeur 2 vers 1 | <input type="checkbox"/> Bascule D Flip-Flop avec reset asynchrone |
| <input type="checkbox"/> Circuit combinatoire | <input type="checkbox"/> Circuit synchrone |
| <input type="checkbox"/> Bascule D Flip-Flop avec reset synchrone | <input type="checkbox"/> Circuit asynchrone |
| <input type="checkbox"/> Circuit séquentiel | |

Question 8 Dans un process une variable prend sa nouvelle valeur immédiatement ?

☐ Faux ☐ Vrai

Question 9 On considère un signal de type std_logic avec trois drivers. Donnez la valeur résolue de ce signal en fonction des valeurs des trois drivers. Rappel de la fonction de résolution du std_logic en annexe. Valeurs des drivers : '0','W','1'. Quelle est la valeur résolue ?

☐ '0' ☐ 'X' ☐ 'Z' ☐ 'W' ☐ '1'

Question 10 Les ASIC sont plus adaptés pour de petite production ?

☐ Vrai ☐ Faux

Question 11 Si on considère le circuit 5 (voir annexes), si A change d'état, combien de temps plus tard F change-t-il d'état ?

☐ 4 ns ☐ 4 ns ☐ 5 ns ☐ 3 ns ☐ 2 ns



Question 12 L'ensemble du langage VHDL peut être utilisé pour la simulation ou la modélisation ?

☐ Vrai ☐ Faux

Question 13 Un FPGA peut comporter un ou plusieurs microprocesseurs ?

☐ Faux ☐ Vrai

Question 14 Le langage VHDL est un langage concurrent ?

☐ Faux ☐ Vrai

Question 15 ♣ Quelles conditions doit-on respecter pour qu'un process en VHDL mène à la synthèse d'un circuit combinatoire?

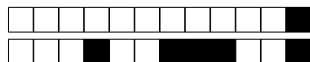
- ☐ La liste de sensibilité doit être complète : tous les signaux lus dans le process doivent apparaître dans la liste de sensibilité.
- ☐ Les signaux doivent prendre leurs nouvelles valeurs qu'à la fin du process.
- ☐ Respectez la structure `if reseat = '1' then ... elsif rising_edge(clk) then ...`
- ☐ Les sorties doivent être assignés dans tout les cas afin d'éviter les bascule D Latch (verrou) et les boucles de retour.
- ☐ Liste de sensibilité doit inclure le clock et le reset (si il existe) et rien d'autre.

Question 16 ♣ Quelles conditions doit-on respecter pour qu'un process en VHDL mène à la synthèse d'un circuit synchrone ?

- ☐ Les sorties doivent être assignés dans tout les cas afin d'éviter les bascule D Latch (verrou) et les boucles de retour.
- ☐ Les signaux doivent prendre leurs nouvelles valeurs qu'à la fin du process.
- ☐ Liste de sensibilité doit inclure le clock et le reset (si il existe) et rien d'autre.
- ☐ Respectez la structure `if reseat = '1' then ... elsif rising_edge(clk) then ...`
- ☐ La liste de sensibilité doit être complète : tous les signaux lus dans le process doivent apparaître dans la liste de sensibilité.

Question 17 On considère un signal de type `std_logic` avec trois drivers. Donnez la valeur résolue de ce signal en fonction des valeurs des trois drivers. Rappel de la fonction de résolution du `std_logic` en annexe. Valeurs des drivers : 'H', 'W', '0'. Quelle est la valeur résolue ?

☐ 'Z' ☐ 'L' ☐ '0' ☐ 'W' ☐ 'X'



Question 18 D'après la déclaration des signaux du circuit 9 (voir annexes) et des exemples fournis, choisissez l'instruction qui permet de réaliser la conversion SIGNED → STD_LOGIC_VECTOR

- ☐ N <= SIGNED(TO_INTEGER(V));
- ☐ V <= TO_SIGNED(STD_LOGIC_VECTOR(N));
- ☐ V <= TO_SIGNED(N,8);
- ☐ S <= TO_SIGNED(N,8);
- ☐ N <= TO_INTEGER(SIGNED(V));
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N));
- ☐ N <= TO_SIGNED(V);
- ☐ V <= STD_LOGIC_VECTOR(S);
- ☐ N <= TO_INTEGER(V);
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N,8));

Question 19 ♣ Choisissez l'instruction ou les instructions qui permettent de réaliser en une ligne de code l'architecture d'un démultiplexeur (ou décodeur) 1 vers 32 afin de compléter le code VHDL du circuit 8 (voir annexes). D étant l'entrée à démultiplexer, Y la sortie et SEL l'entrée de sélection. Rappel de la librairie Numeric_STD en annexes.

- ☐ Y(SEL) <= D;
- ☐ D <= Y(SEL);
- ☐ Y(to_integer(signed(SEL))) <= D;
- ☐ D <= Y(to_integer(signed(SEL)));
- ☐ D <= Y(to_integer(SEL));
- ☐ Y <= SEL(D);
- ☐ Y(to_integer(unsigned(SEL))) <= D;
- ☐ D <= Y(to_integer(unsigned(SEL)));
- ☐ Y(to_integer(SEL)) <= D;

Question 20 Les ASIC (Full Custom) sont reconfigurables (programmable) ?

- ☐ Vrai ☐ Faux

Question 21 ♣ Choisissez l'instruction ou les instructions qui permettent de réaliser un additionneur entre les entrées A, B et C sur la sortie SUM afin de compléter le code VHDL du circuit 6 (voir annexes). Rappel de la librairie Numeric_STD en annexe.

- ☐ SUM <= std_logic_vector(std_logic_vector(A) + B + to_integer(C));
- ☐ SUM <= std_logic_vector(A + B + signed(C));
- ☐ SUM <= A + B + C;
- ☐ SUM <= std_logic_vector(A + B + C);
- ☐ SUM <= std_logic_vector(to_signed(to_integer(A) + B + to_integer(signed(C))),9));



Question 22 ♣ Choisissez les specifications qui caracterisent le circuit 4 (voir annexes)

- | | |
|---|--|
| <input type="checkbox"/> Bascule D Latch | <input type="checkbox"/> Circuit séquentiel |
| <input type="checkbox"/> Bascule D Flip-Flop avec reset synchrone | <input type="checkbox"/> Registre à décalage |
| <input type="checkbox"/> Circuit combinatoire | <input type="checkbox"/> Circuit synchrone |
| <input type="checkbox"/> Multiplexeur 2 vers 1 | <input type="checkbox"/> Bascule D Flip-Flop avec reset asynchrone |
| <input type="checkbox"/> Circuit asynchrone | |

Question 23 L'ensemble du langage VHDL peut être utilisé pour la synthèse de circuit numérique ?

- ☐ Faux ☐ Vrai

Question 24 Les ASIC sont plus flexible que les FPGA ?

- ☐ Faux ☐ Vrai

Question 25 ♣ Choisissez les specifications qui caracterisent le circuit 1 (voir annexes)

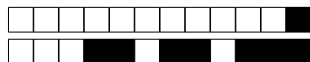
- | | |
|---|--|
| <input type="checkbox"/> Circuit combinatoire | <input type="checkbox"/> Bascule D Flip-Flop avec reset synchrone |
| <input type="checkbox"/> Registre à décalage | <input type="checkbox"/> Multiplexeur 2 vers 1 |
| <input type="checkbox"/> Circuit séquentiel | <input type="checkbox"/> Bascule D Latch |
| <input type="checkbox"/> Circuit synchrone | <input type="checkbox"/> Bascule D Flip-Flop avec reset asynchrone |
| <input type="checkbox"/> Circuit asynchrone | |

Question 26 D'après la déclaration des signaux du circuit 9 (voir annexes) et des exemples fournis, choisissez l'instruction qui permet de réaliser la conversion STD_LOGIC_VECTOR -> INTEGER

- ☐ V <= TO_SIGNED(N,8);
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N,8));
- ☐ N <= TO_SIGNED(V);
- ☐ S <= TO_SIGNED(N,8);
- ☐ V <= TO_SIGNED(STD_LOGIC_VECTOR(N));
- ☐ N <= TO_INTEGER(SIGNED(V));
- ☐ N <= SIGNED(TO_INTEGER(V));
- ☐ V <= STD_LOGIC_VECTOR(TO_SIGNED(N));
- ☐ N <= TO_INTEGER(V);
- ☐ V <= STD_LOGIC_VECTOR(S);

Question 27 ♣ Choisissez les specifications qui caracterisent le circuit 2 (voir annexes)

- | | |
|---|--|
| <input type="checkbox"/> Circuit combinatoire | <input type="checkbox"/> Bascule D Latch |
| <input type="checkbox"/> Bascule D Flip-Flop avec reset synchrone | <input type="checkbox"/> Circuit asynchrone |
| <input type="checkbox"/> Registre à décalage | <input type="checkbox"/> Circuit synchrone |
| <input type="checkbox"/> Circuit séquentiel | <input type="checkbox"/> Bascule D Flip-Flop avec reset asynchrone |
| <input type="checkbox"/> Multiplexeur 2 vers 1 | |



Question 28 ♣ Choisissez l'instruction ou les instructions qui permettent de réaliser en une ligne de code l'architecture d'un multiplexeur 64 vers 1 afin de compléter le code VHDL du circuit 7 (voir annexes). Y étant l'entrée à multiplexer, D la sortie et SEL l'entrée de sélection. Rappel de la librairie Numeric_STD en annexes.

- ☐ D <= Y(to_integer(SEL));
- ☐ D <= Y(to_integer(signed(SEL)));
- ☐ Y(to_integer(signed(SEL))) <= D;
- ☐ D <= Y(SEL);
- ☐ D <= Y(to_integer(unsigned(SEL)));
- ☐ Y(to_integer(unsigned(SEL))) <= D;
- ☐ Y(to_integer(SEL)) <= D;
- ☐ Y <= SEL(D);
- ☐ Y(SEL) <= D;

Question 29 Dans un process un signal prend sa nouvelle valeur immédiatement ?

- ☐ Faux ☐ Vrai

Question 30 On considère un signal de type std_logic avec trois drivers. Donnez la valeur résolue de ce signal en fonction des valeurs des trois drivers. Rappel de la fonction de résolution du std_logic en annexe. Valeurs des drivers : 'Z', 'L', 'H'. Quelle est la valeur résolue ?

- ☐ 'Z' ☐ 'X' ☐ 'L' ☐ 'H' ☐ 'W'