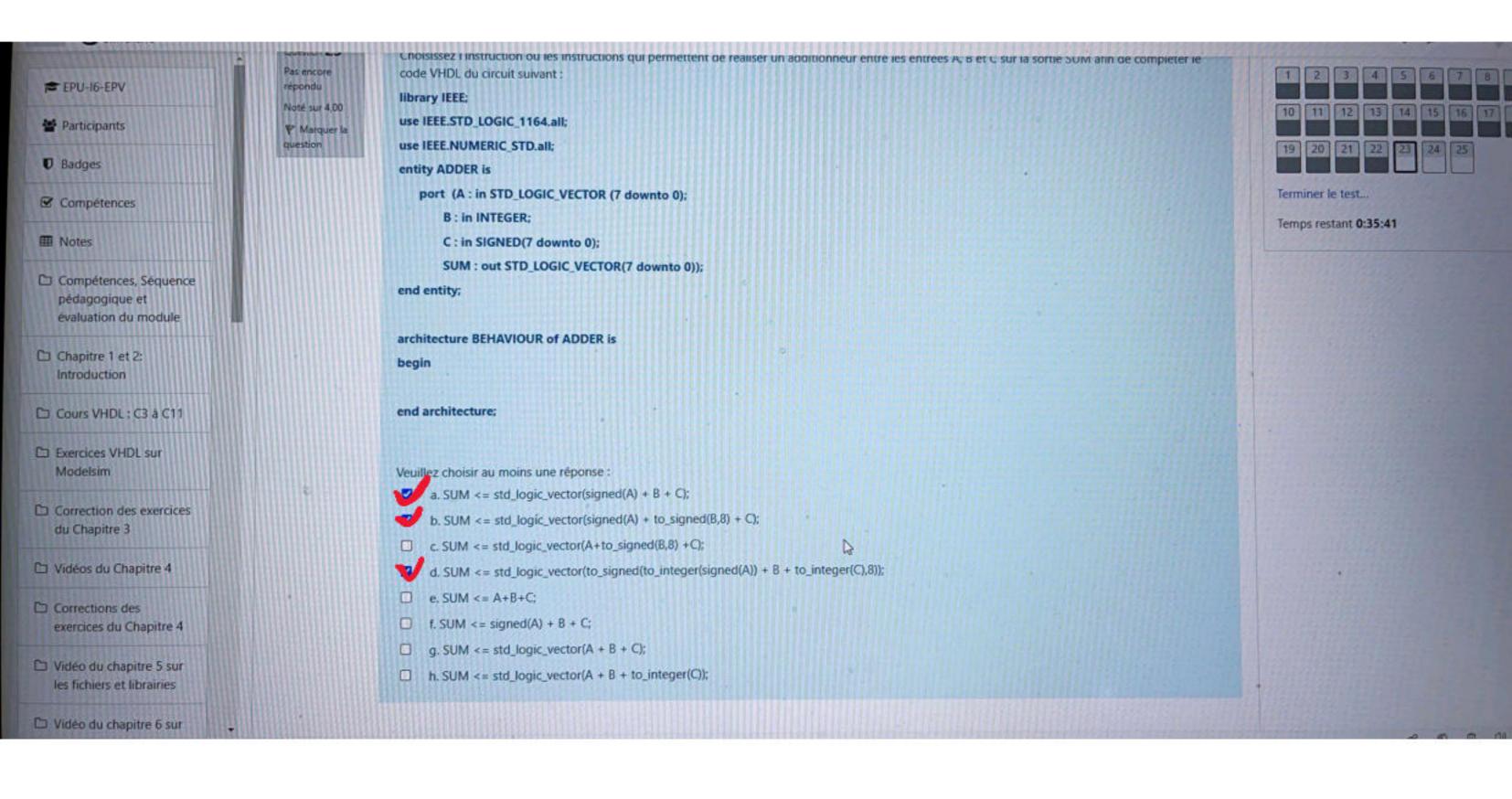
Les performances des ASIC (vitesse, low-power) sont meilleures que les FPGA ?

Sélectionnez une réponse :



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 suivant :		
library IEEE;		
use IEEE.STD_LOGIC_1164.all;		
use IEEE.NUMERIC_STD.all;		
entity ADDER is		
port (A: in STD_LOGIC_VECTOR (7 downto 0);		
B: in STD_LOGIC_VECTOR (7 downto 0);		
C: in STD_LOGIC_VECTOR (7 downto 0);		
SUM : out STD_LOGIC_VECTOR(7 downto 0));		
end entity;		
architecture BEHAVIOUR of ADDER is		
begin		
end architecture;		
Veuillez choisir au moins une réponse :		
Veuillez choisir au moins une réponse : a. SUM <= signed(A) + signed(B) + signed(C);		
a. SUM <= signed(A) + signed(B) + signed(C);		
a. SUM <= signed(A) + signed(B) + signed(C);		
□ a. SUM <= signed(A) + signed(B) + signed(C); □ b. SUM <= A+B+C; 7 X		
 a. SUM <= signed(A) + signed(B) + signed(C); b. SUM <= A+B+C; 7 X c. SUM <= std_logic_vector(A + B + C); 		
 a. SUM <= signed(A) + signed(B) + signed(C); b. SUM <= A+B+C; 7 X c. SUM <= std_logic_vector(A + B + C); d. SUM <= std_logic_vector(unsigned(A) + unsigned(B) + unsigned(C)); 		
 a. SUM <= signed(A) + signed(B) + signed(C); b. SUM <= A+B+C; 7 X c. SUM <= std_logic_vector(A + B + C); d. SUM <= std_logic_vector(unsigned(A) + unsigned(B) + unsigned(C)); e. SUM <= unsigned(A) + unsigned(B) + unsigned(C); 		



```
port( i0, i1: in std_logic;
   out0 : out std_logic);
end entity;
architecture circuit2 of cell2 is
  if (i0= '1') then

out0 <= i1;

lif; \( \) else \( \) ess;
begin
process (i0, i1)
begin
end process;
end architecture;
Veuillez choisir au moins une réponse :
      a. Multiplexeur 2 vers 1
     b. Circuit combinatoire
     c. Registre à décalage
      d. Circuit asynchrone
      e. Bascule D Flip-Flop avec un reset synchrone
f. Circuit séquentiel
      g. Circuit synchrone
      h. Bascule D Latch
```

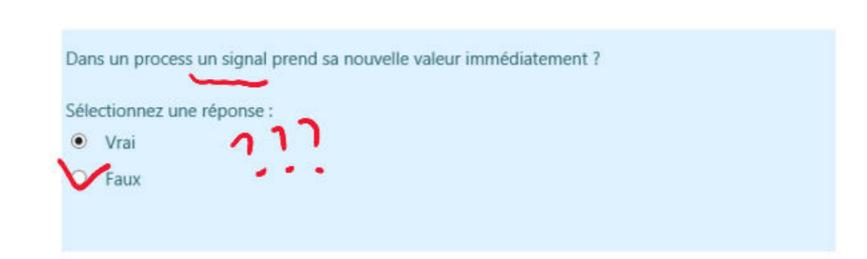
Que	elles conditions doit-on respecter pour qu'un process en VHDL mêne à la synthèse d'un circuit combinatoire?	
Veuillez choisir au moins une réponse :		
W	a. Les sorties doivent être assignés dans tout les cas afin d'éviter les bascule D Latch (verrou) et les boucles de retour.	
	b. Liste de sensibilité doit inclure le clock et le reset (si il existe) et rien d'autre.	
	c. Les signaux doivent prendre leurs nouvelles valeurs qu'à la fin du process.	
	d. Respectez la structure if resest = '1' then elsif rising_edge(clk) then	
A	Je. La liste de sensibilité doit être complète : tous les signaux lus dans le process doivent apparaître dans la liste de sensibilité.	

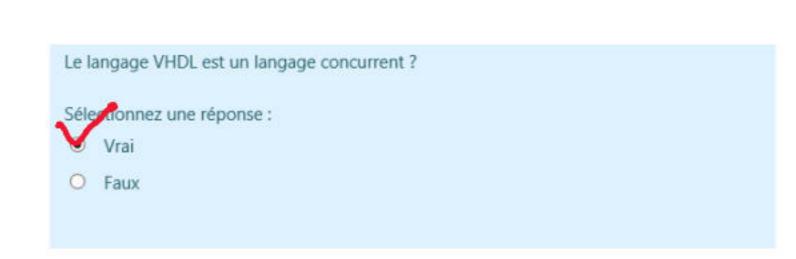
Un FPGA peut comporter un ou plusieurs microprocesseur?

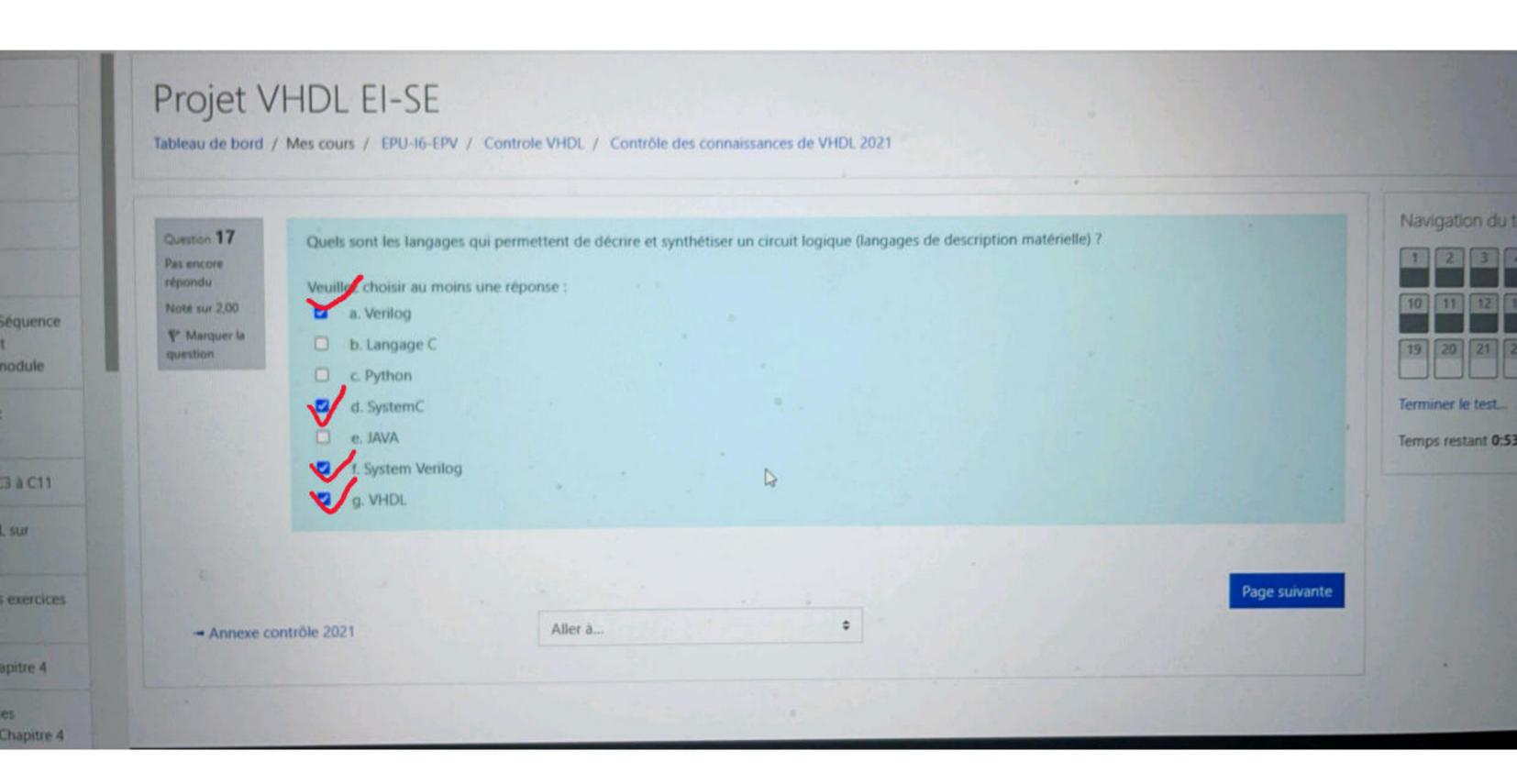
Sélectionnez une réponse :



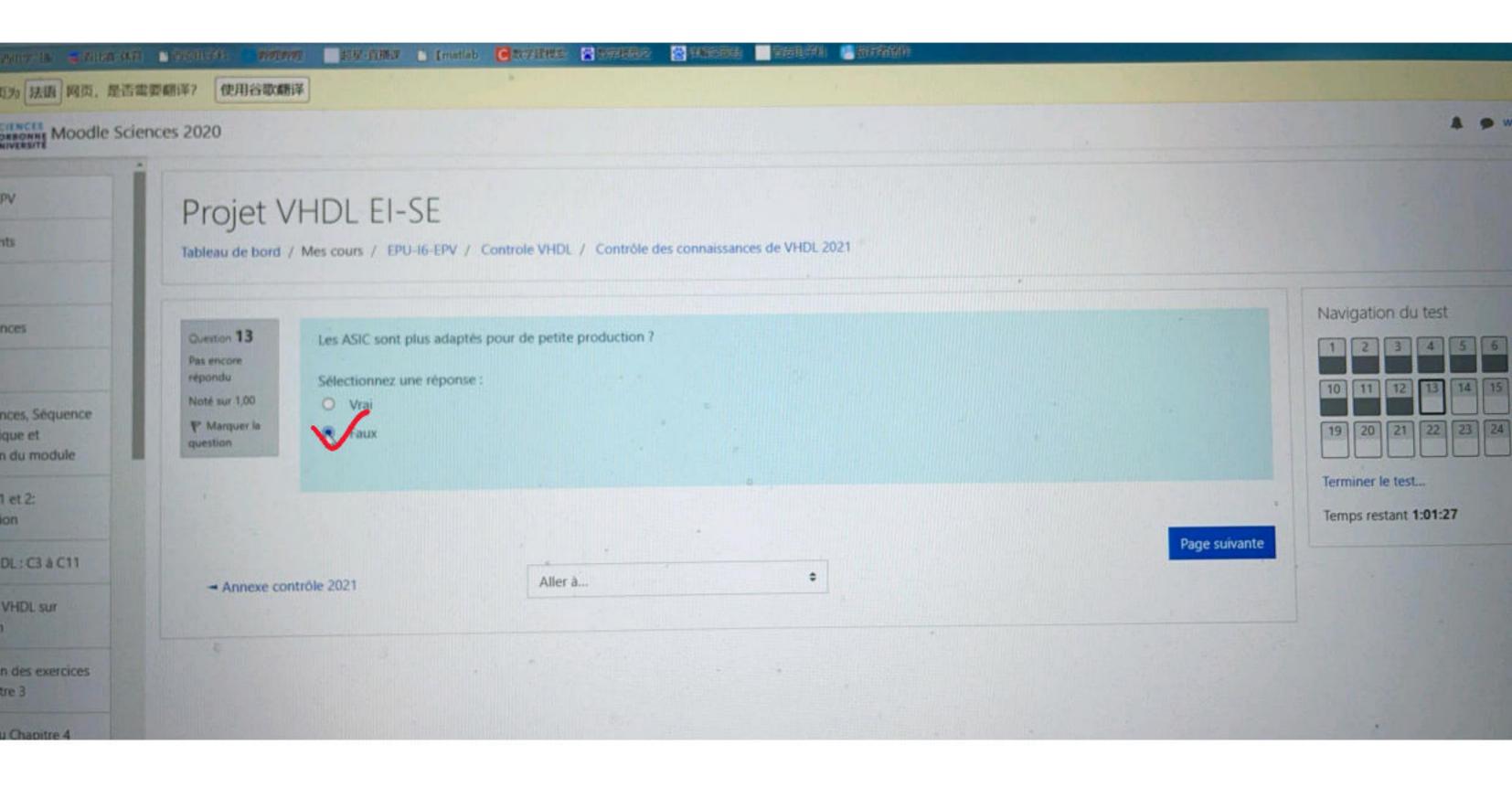




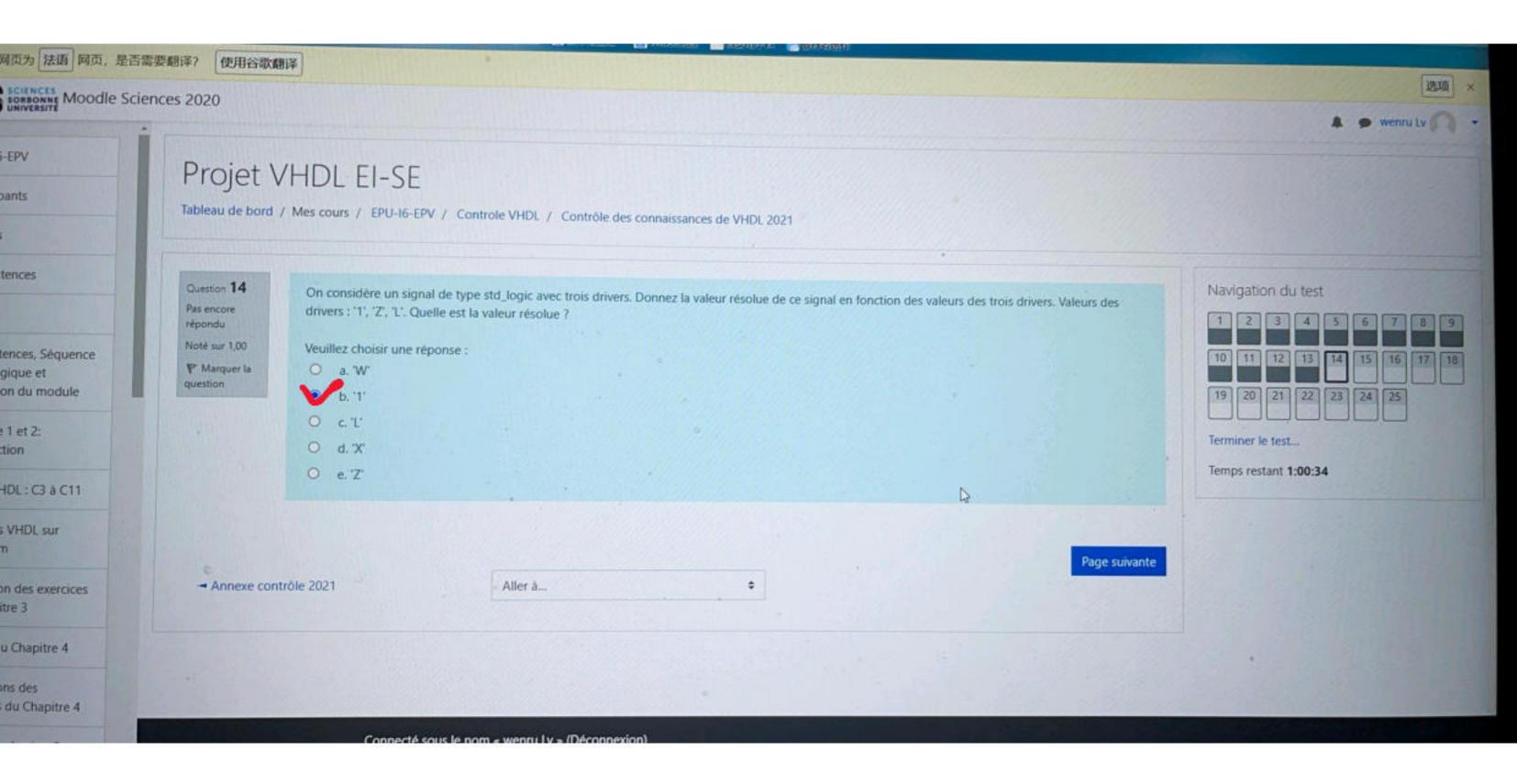


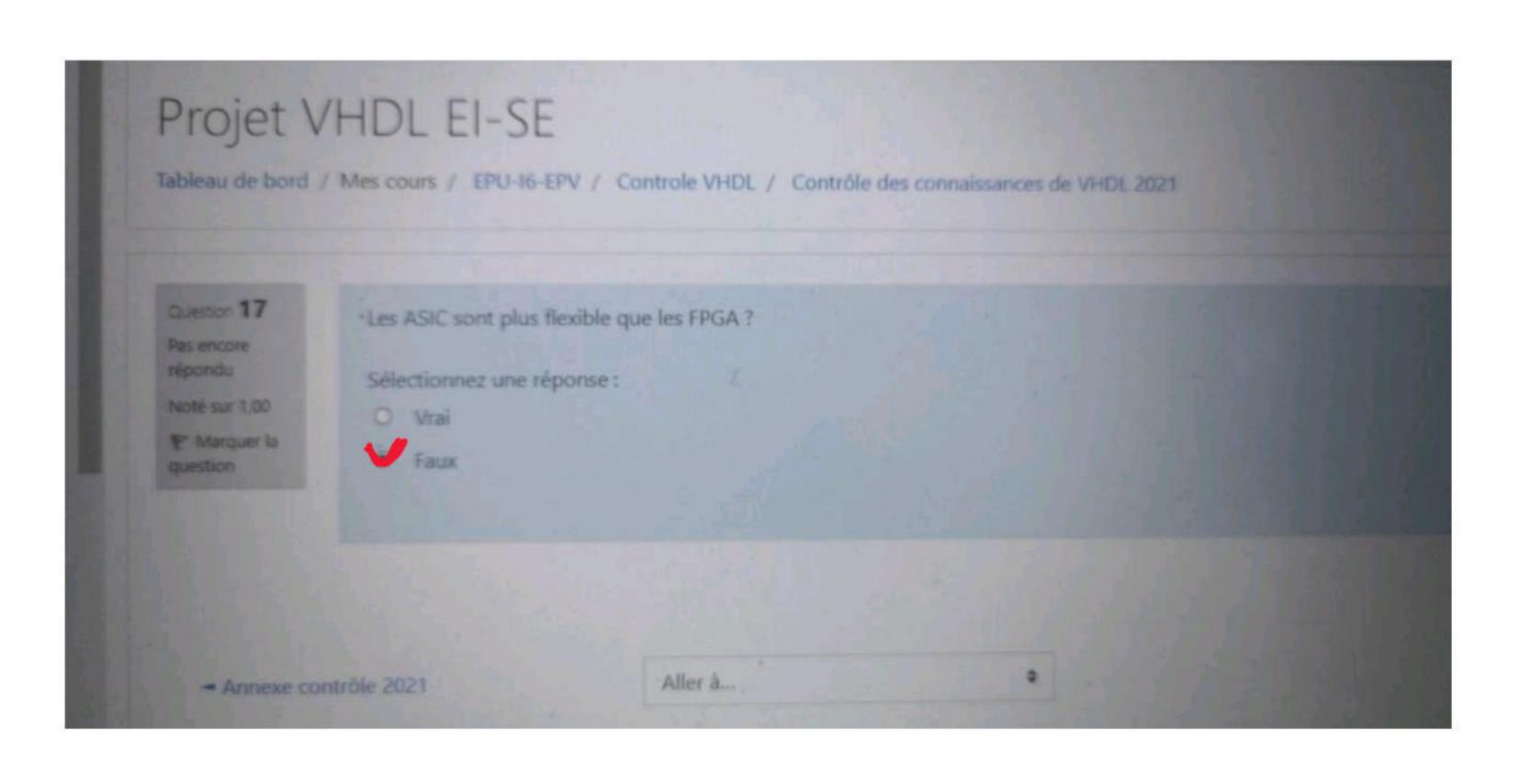


D'après la déclaration des signaux ci-dessous et des exemples fournis, choisissez l'instruction qui permet de réaliser la conversion INTEGER --> SIGNED Signal U: UNSIGNED(7 downto 0); Signal S: SIGNED (7 downto 0); Signal V: STD_LOGIC_VECTOR(7 downto 0); Signal N: INTEGER; Exemples: U <= UNSIGNED(V); -- conversion STD_LOGIC_VECTOR à UNSIGNED N <= TO_INTEGER(S); -- conversion SIGNED à INTEGER Veuillez choisir une réponse : a. V <= STD_LOGIC_VECTOR(TO_SIGNED(N)); b. V <= STD_LOGIC_VECTOR(TO_SIGNED(N,8));</p> C, N <= TO_SIGNED(V);</p> d. S <= TO_SIGNED(N,8); e. N <= TO_INTEGER(V);</p> f. V <= STD_LOGIC_VECTOR(S);</p> O g. V <= TO_SIGNED(STD_LOGIC_VECTOR(N));</p> h. N <= TO_INTEGER(SIGNED(V));</p> i. N <= SIGNED(TO_INTEGER(V));</p> j. V <= TO_SIGNED(N,8);</p>



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 ci-dessou sélection. Rappel de la librairie NUMERIC_STD en annexe.		
entity MUX64V1 is		
port (D : out STD_LOGIC;		
SEL: in STD_LOGIC_VECTOR(5 downto 0);		
Y: in STD_LOGIC_VECTOR(63 downto 0));		
end entity;		
architecture RTL of MUX64V1 is begin		
end architecture;		
uillez choisir au moins une réponse :		
a. D <= Y(SEL);		
b. Y(SEL) <= D;		
c, D <= Y(to_integer(unsigned(SEL)));		
d. D <= Y(to_integer(SEL));		
e. D <= Y(to_integer(signed(SEL)));		
f. D <= Y(to_signed(SEL));		
g. Y(to_integer(signed(SEL))) <= D;		
h. Y(to_integer(unsigned(SEL))) <= D;		
i. Y <= SEL(D);		





Dans un process une variable prend sa nouvelle à la fin du process ?

Sélectionnez une réponse :

O Vrai

Fair