

Nom : _____ Matricule : _____

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #3 – jeudi 3 décembre 2020

Durée: 1 heure.

Pondération: 0% (formatif).

1. Conditions du contrôle

Ce contrôle est calibré pour une durée de 45 minutes. Une période additionnelle de 15 minutes est ajoutée pour les étudiants qui souhaitent s'en prévaloir et comme tampon s'il devait y avoir des problèmes techniques.

Le contrôle est à livres ouverts, toutes les sources sont permises. Si vous utilisez du matériel trouvé en ligne ou ailleurs, citez bien vos sources dans votre réponse.

Le contrôle doit être fait individuellement. Toute forme de communication avec d'autres personnes que le surveillant est interdite pendant l'examen. Une attestation à cet effet vous sera demandée dans la dernière question.

2. Communication avec le surveillant pendant l'examen

Le surveillant est disponible via le serveur Discord du cours pour aider à résoudre les problèmes techniques. Si le serveur Discord du cours ne fonctionne pas, envoyez un courriel à pierre.langlois@polymtl.ca.

Ne posez pas de questions au surveillant concernant les questions de l'examen, aucune réponse ne sera donnée. En cas de doute sur le sens d'une question, faites une supposition raisonnable, énoncez-la clairement dans votre réponse et poursuivez.

Question 1. (0 point)

Acceptation des conditions du contrôle.

Question 2. (1 point)

Réponses brèves.

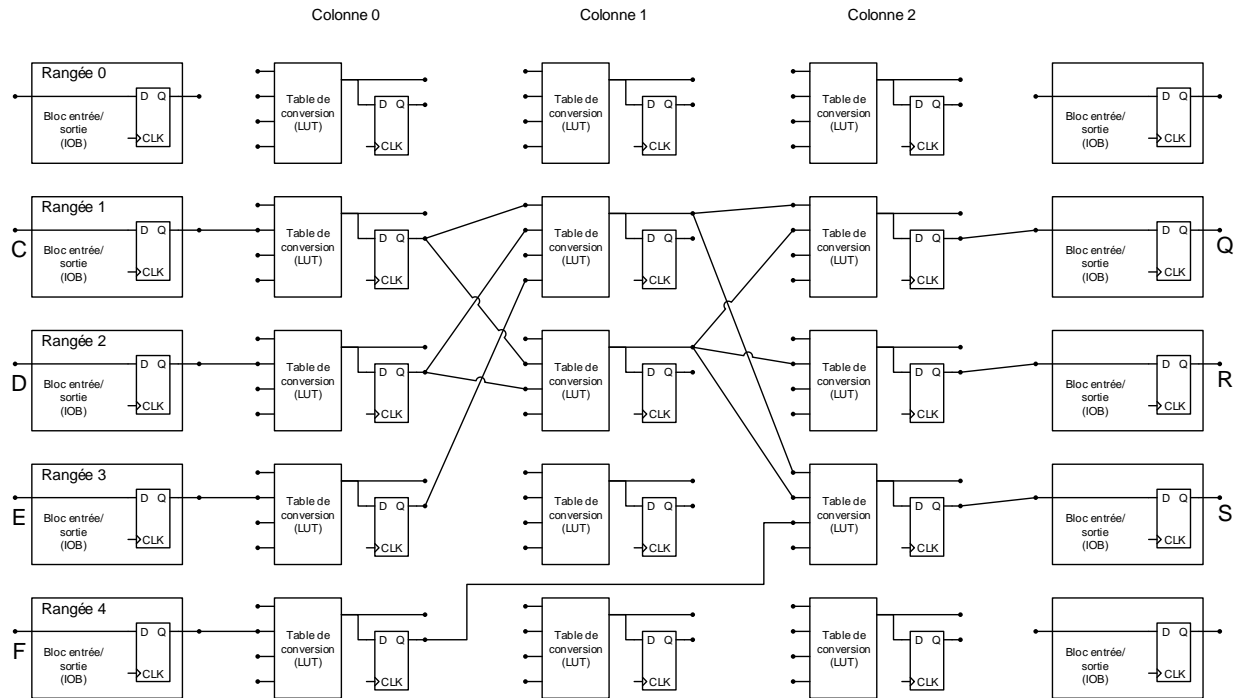
a. Expliquez dans quels cas la division et le calcul du modulo sont supportés par les outils de synthèse présentement sur le marché, et pourquoi ces opérations ne le sont pas dans les autres cas.

b. Qu'est-ce qui est produit par un outil de synthèse à partir d'un code VHDL valide?

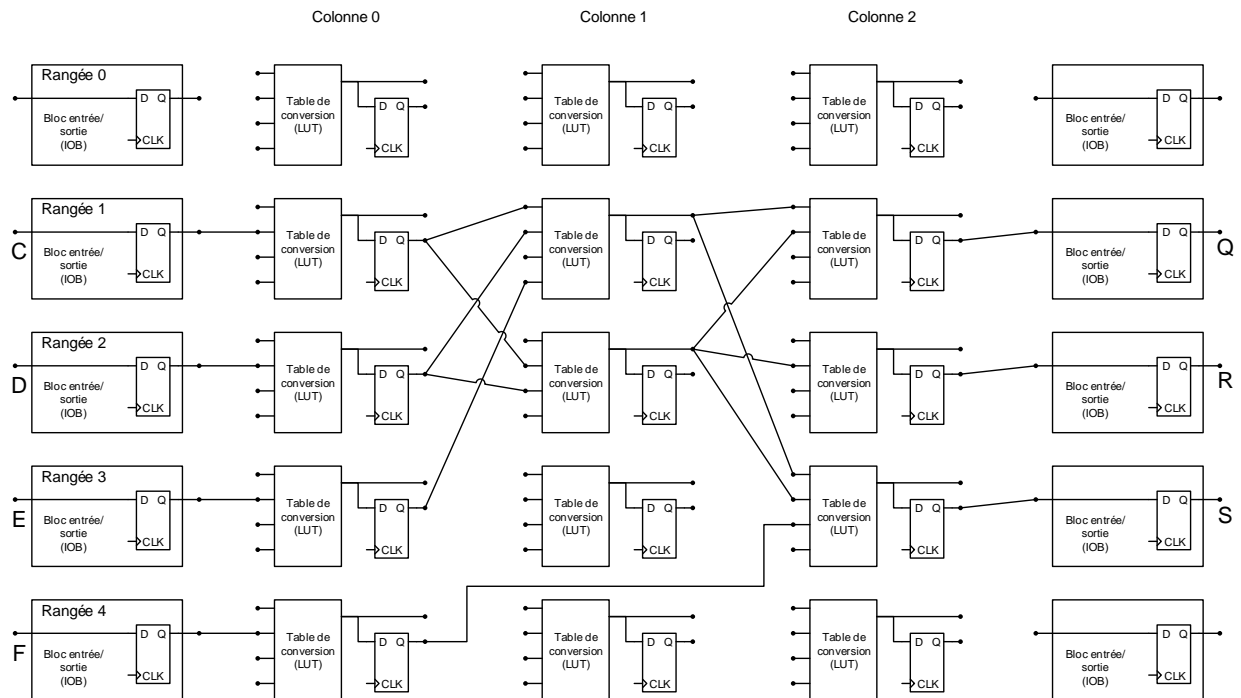
c. Dans quelles conditions une boucle en VHDL est-elle synthétisable ? Expliquez pourquoi.

d. Donnez deux métriques pour comparer différentes solutions de placement d'un circuit numérique sur un FPGA.

(suite de a., page précédente) Annotez le circuit pour indiquer le chemin critique.



(suite de c., page précédente) Annotez le circuit pour montrer comment le pipeliner.



Solutions

Q2.

a. La division et le modulo sont supportés quand le deuxième argument est une puissance de deux. Le résultat s'obtient alors par un décalage ou une sélection de bits, respectivement. La division générale et le modulo général ne sont pas présentement synthétisables parce qu'il n'y a pas de bloc dédié à cette opération dans la plupart des technologies, dont les FPGA. Ces opérations peuvent être réalisées de différentes façons, par exemple par la multiplication par la réciproque du diviseur et par des circuits itératifs. Dans les deux cas, il y a beaucoup de choix de design à faire, incluant la complexité matérielle et la latence du circuit résultant. Les concepteurs de synthétiseurs préfèrent laisser ces choix à celles et ceux qui écrivent du code VHDL.

b. Le produit de la synthèse est une liste de composantes de base et des interconnexions entre elles qui correspond à l'interprétation matérielle du code VHDL. Dans leur plus simple expression, les composantes peuvent être des portes logiques. Les composantes peuvent correspondre à des blocs plus complexes disponibles sur la technologie visée. Par exemple, pour un FPGA, une composante peut être un multiplieur ou un bloc de mémoire RAM.

c. Pour être synthétisable, une boucle logicielle doit pouvoir être déroulée complètement, et à chaque itération correspond un bloc matériel distinct. Cela correspond à échanger du temps pour de l'espace. Il faut donc que les bornes de la boucle soient connues de façon statique au moment de la synthèse, par exemple en étant des constantes ou des valeurs spécifiées par des énoncés `generic`.

d. Longueur totale des interconnexions, chemin le plus long, congestion des interconnexions.

Q3.

Opération ou groupe d'opérations simultanées	Possible ?	chargeBR	choixCharge	choixA	choixB	valeur	ChoixB_UAL	Op_UAL	Charge_MD	choixDon-nee_BR
$R4 := M[R0 + R2] ;$	oui	1	4	0	2	-	0	2	0	1
$R7 := R1 + R2 ;$ $M[R1 + R2] := R1 ;$	oui	1	7	2	1	-	0	2	1	0
$R0 := 534 ;$ $R4 := M[R0 + 534] ;$	non	L'architecture ne permet pas de charger deux registres simultanément. L'UAL n'a qu'une sortie et ne peut donc pas faire l'addition et le routage de 534 en même temps.								
$R3 := R4 + R5 ;$ $R4 := M[R4 + R5] ;$	non	L'architecture ne permet pas de charger deux registres simultanément.								
$R2 := R1 \text{ ET } 85 ;$ $M[R1 + 85] := R3 ;$	non	L'UAL n'a qu'une sortie et ne peut donc pas faire l'addition et l'opération logique ET en même temps.								

Pour les opérations #3, il faudrait un bloc de registres à deux entrées, deux chemins de rétroaction vers l'entrée du bloc de registres, et deux signaux distincts choixCharge1 et choixCharge2. Il faudrait de plus pouvoir aiguiller la valeur provenant de l'unité de contrôle (ici 534) vers le bloc des registres.

Pour les opérations #4, il faudrait un bloc de registres à deux entrées, deux chemins de rétroaction vers l'entrée du bloc de registres (un de la mémoire et un de la sortie de l'UAL), et deux signaux distincts choixCharge1 et choixCharge2.

Pour les opérations #5, il faudrait avoir deux sorties à l'UAL, par exemple une pour les fonctions logiques et une pour les fonctions arithmétiques. Les additions devraient pouvoir être envoyées à l'adresse de la mémoire. Toutes les opérations devraient pouvoir être retournées vers le bloc des registres en rétroaction. [En pratique, on utilise un 2^e additionneur pour calculer les adresses de la mémoire.]

Q4.

Il faut $4 \times 16 = 64$ bascules.

Les deux multiplexeurs de sortie ont chacun 6 entrées : un bit par registre et deux bits pour le choix de la sortie. Donc il faut 1 LUT par bit pour chacun des multiplexeurs, donc 16 LUT chacun, donc 32 LUT.

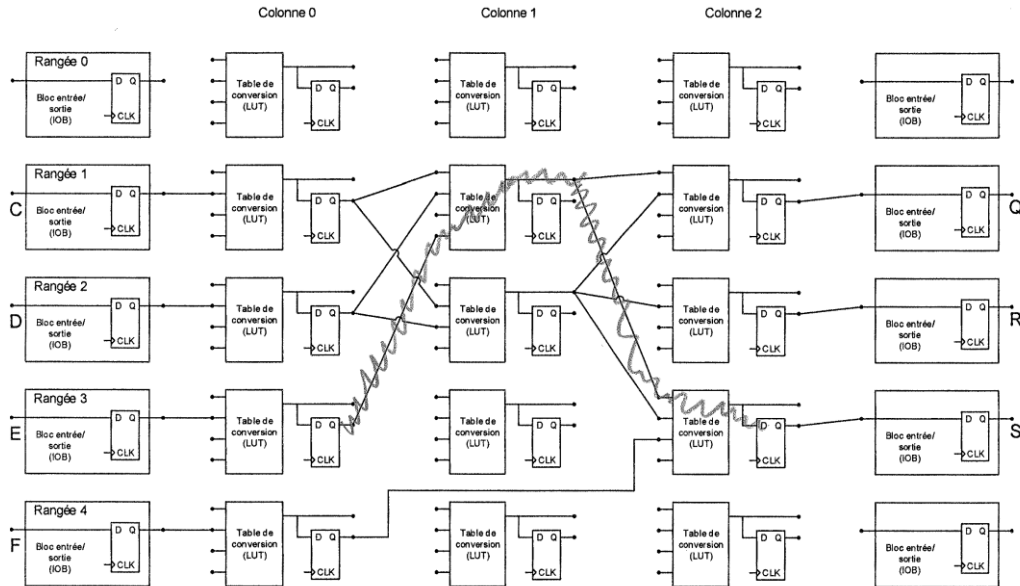
Il faut 4 LUT pour les 4 portes ET.

Le décodeur a deux entrées et 4 sorties, il faut 4 LUT.

Total BR : 64 bascules, 40 LUT environ.

Q5.

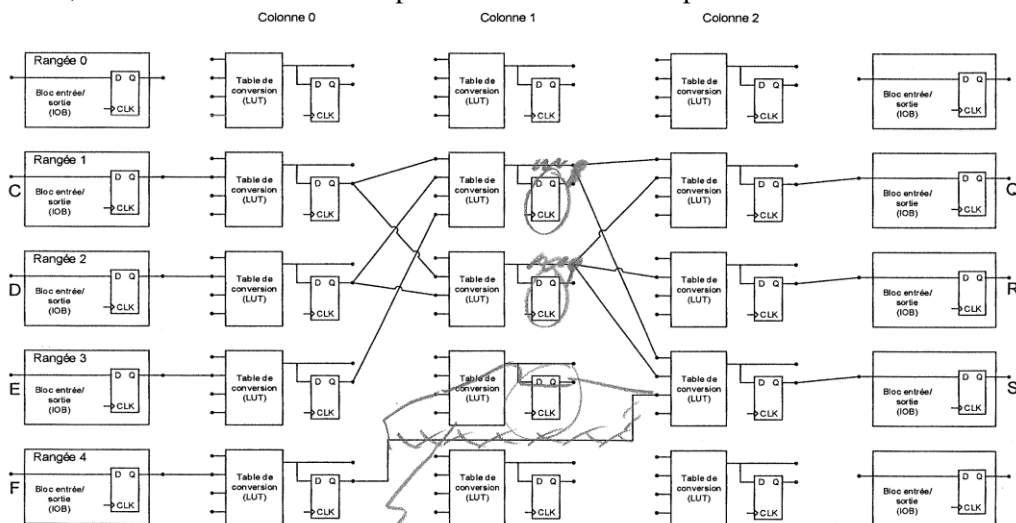
a. Le chemin critique a un délai de $0.75 + 0.6 + 2.25 + 0.6 + 2.25 + 0.25 = 6.7$ ns, pour une fréquence maximale d'horloge de 149 MHz.



b. Sur le chemin critique, on doit ajouter 0.4 ns à la période minimale d'horloge, donc $T_{min} = 7.1$ ns, $f_{max} = 141$ MHz.

Sur les chemins courts, p.ex. de IOB C à C0R1, on a un délai de $0.75 + 0.2 + 2.25 + 0.25 = 3.2$ ns $> t_h$ ok. Donc $f_{max} = 141$ MHz.

c. On insère des registres de pipeline dans la colonne 1. Attention, il faut utiliser une tranche supplémentaire pour le chemin de C0R4 à C2R3. Le chemin critique a maintenant un délai de $0.75 + 0.6 + 2.25 + 0.25 = 3.85$ ns, donc $f_{max} = 259$ MHz ok pour 250×10^6 résultats par seconde.



Cette LUT effectue l'opération double.