

---

# Revue générale



Pierre Langlois

<http://creativecommons.org/licenses/by-nc-sa/2.5/ca/>

# Organisation de la revue

---

- 0 Rappels (pas matière à examen)
- 1 Introduction
- 2 Technologies de logique programmable
- 3 Modélisation et vérification de circuits combinatoires
- 4 Modélisation et vérification de circuits séquentiels
- 5 Conception de chemins des données
- 6 Conception et implémentation de fonctions arithmétiques sur FPGA
- 7 Vérification de systèmes numériques
- 8 Simulation, synthèse et implémentation
- 9 Processeurs à usage général
- 10 Performance partie 1
- 11 Performance partie 2

# Niveaux de connaissance attendus

---

- La taxonomie de Bloom est un modèle pédagogique pour classer six niveaux d'acquisition des connaissances. Dans le cours nous avons couvert les 5 premiers niveaux. Le niveau B6 est l'évaluation.
- Les exercices et exemples des diapos et les questions des devoirs étaient principalement aux niveaux B2 et B3.
- Les exercices en classes étaient principalement aux niveaux B4 et B5.
- Les questions des contrôles et de l'examen final sont principalement aux niveaux B4 et B5.

Code	Niveau ( <a href="http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom">http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom</a> )
B1	Connaissance - mémoriser de l'information.
B2	Compréhension – interpréter l'information.
B3	Application – confronter les connaissances à des cas pratiques simples.
B4	Analyse – décomposer un problème, cas pratiques plus complexes.
B5	Synthèse – expression personnelle, cas pratiques plus complexes.

# Mise en garde concernant cette revue du cours

---

- Cette vidéo énumère les objectifs d'apprentissage spécifiques globaux de chaque semaine de cours.
- Chaque vidéo comportait aussi des objectifs spécifiques, sur lesquels on ne revient pas ici.
- La vidéo présente un exemple de question pour chaque semaine.
- Les laboratoires font partie intégrante de la matière du cours.
- Il est essentiel de revoir les devoirs, exercices et laboratoires de chaque semaine, puis les diapositives et vidéos correspondantes pour les sujets qui sont moins bien compris.
- Cette vidéo peut servir de guide général pour la préparation à l'examen, mais ne saurait être suffisante par elle-même.

# 1 Introduction

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Présenter une vue d'ensemble des systèmes numériques : leurs applications, niveaux d'abstraction et considérations d'implémentation. (B3)
  - Analyser et évaluer différentes options d'implémentation et les comparer en termes de coût, effort de design et performance. (B4)
  - Étant donné des requis de performance, et des spécifications de débit et de parallélisme de dispositifs, choisir la meilleure solution. (B3)
  - Appliquer le flot de conception de circuits numériques. (B3)
  - Écrire le code VHDL pour un module combinatoire simple. (B3)

Examen intra #1

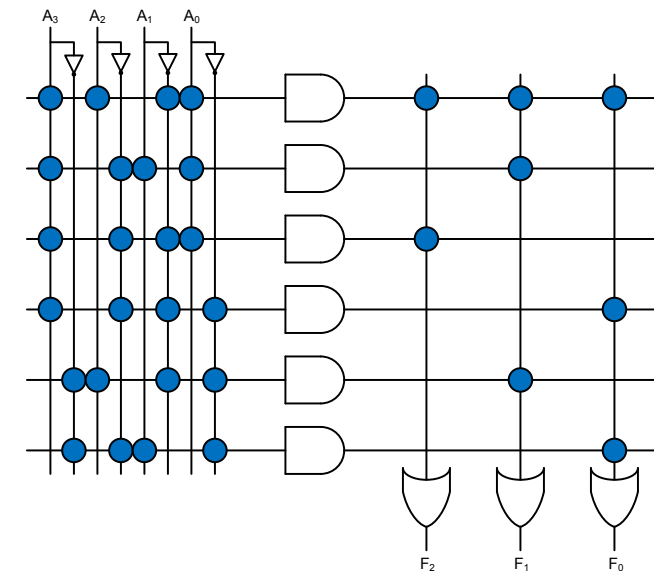
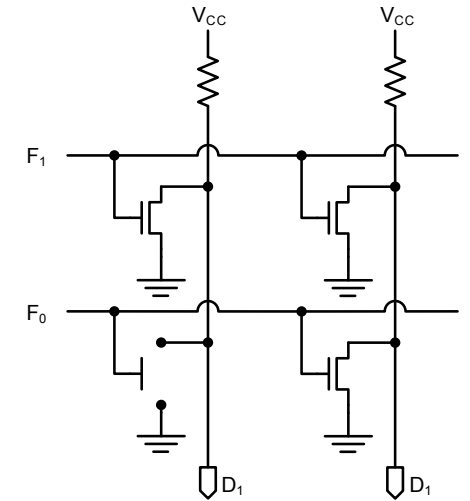
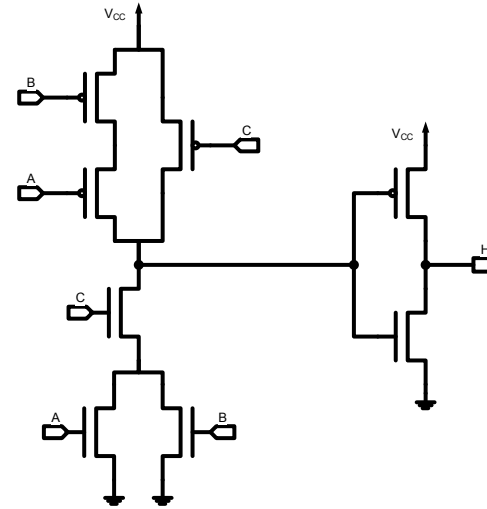
## Une question type

Vous travaillez pour une compagnie montréalaise dans le domaine du traitement d'images en temps réel. Votre compagnie a obtenu un contrat en sous-traitance d'un important intégrateur de systèmes avioniques pour la segmentation automatique d'images saisies par une caméra infra-rouge installée dans le nez d'un appareil CSeries de Bombardier. À ce jour, Bombardier a des commandes fermes pour 203 appareils. Votre compagnie entrevoit la possibilité de vendre son produit à d'autres avionneurs, avec un potentiel de plus de 10K (dix mille) unités.

Votre patron vous demande de l'aider à déterminer s'il faut viser une implémentation en logique fixe ou sur FPGA. Pour la solution FPGA, vous estimez le temps de conception à 4 mois pour 3 ingénieurs, et à 6 mois pour 3 ingénieurs pour la logique fixe. Un ingénieur coûte \$100K/an. Les licences d'outils de design pour toute l'équipe coûtent \$5K pour le FPGA et \$20K pour la logique fixe. Les frais fixes de la fonderie s'élèvent à \$250K puis les puces coûtent \$10 l'unité, et les premières puces seraient livrées 4 mois après la commande. Le FPGA qui accommoderait votre design coûte \$250 et peut être livré dans 4 semaines. Les frais de production et de distribution sont les mêmes pour les deux technologies. Que dites-vous à votre patron? Montrez tous vos calculs et énoncez toutes vos suppositions.

# 2 Technologies de logique programmable

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Expliquer l'historique du développement de la logique programmable, des portes logiques aux FPGA. (B2)
  - Expliquer la structure des types de dispositifs de logique programmable. (B2)
  - Décrire les technologies de programmation des dispositifs de logique programmable. (B2)
  - Montrer comment programmer les ROM, PLA, PAL pour implémenter un circuit combinatoire. (B3)



Examen intra #1

# 3 Modélisation et vérification de circuits combinatoires

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Donner le modèle VHDL d'un circuit combinatoire complexe à partir de sa spécification, en choisissant le style de description, les types et les structures du langage appropriés. (B5)
  - Donner le code VHDL d'un banc d'essai pour stimuler et vérifier un circuit combinatoire de façon exhaustive ou avec des vecteurs de test spécifiés. (B5)
  - Analyser le modèle VHDL d'un circuit combinatoire complexe, en donner le circuit correspondant, et en effectuer la synthèse pour l'implémenter dans un FPGA. (B4, B5)

Examen intra #1

Un système de contrôle de la température ambiante d'une pièce reçoit une entrée d'un thermomètre numérique et a deux sorties, une pour activer la climatisation et une pour activer le chauffage. Le thermomètre numérique exprime les températures selon un code Gray sur 4 bits, selon le tableau suivant. On vise à maintenir une température entre 20°C et 24°C. Le système de contrôle est très simple : le climatiseur doit être activé quand il fait trop chaud, et la fournaise doit être activée quand il fait trop froid.

Donnez une architecture en VHDL synthétisable d'un module combinatoire pour implémenter le contrôleur de température qui correspond à la définition d'entité suivante. ...

Considérez le code VHDL suivant.

```
library ieee;
use ieee.std_logic_1164.all;

entity module8 is
port (
  A, B, C, D, E: in std_logic;
  X, Y, Z : out std_logic
);
end module8;
Etc. ...
```

Montrez, sur le diagramme qui suit, un résultat possible de la synthèse et de l'implémentation de ce code sur un FPGA. Indiquez directement sur le dessin où chaque signal se situe ainsi que les interconnexions entre les blocs.

Indiquez dans les tables de vérité le contenu de chacune des tables de conversion que vous utilisez.





# 5 Conception de chemins des données

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Concevoir et modéliser en VHDL un chemin de données qui réalise des fonctions arithmétiques et logiques complexes au niveau de transfert entre registres (*Register Transfer Level* – RTL). (B5)

La méthode de division par multiplications successives, ou méthode de Goldschmidt, consiste à multiplier le dividende  $N$  et le diviseur  $D$  par une suite de nombres  $X_k$ . Les nombres  $X_k$  sont choisis afin que le produit  $D \times X_1 \times X_2 \times X_3 \dots$  converge vers 1. Le produit  $N \times X_1 \times X_2 \times X_3 \dots$  converge alors vers  $Q$ .

À chaque étape on calcule une nouvelle valeur du numérateur  $N$  et du dénominateur  $D$ . Avec  $D_0$  dans l'intervalle  $[0.5, 1]$  (suite à une normalisation préalable de  $N$  et  $D$ ).

On peut décrire l'algorithme avec les micro-opérations suivantes où  $N_0$ ,  $D_0$  et  $init$  sont des ports d'entrée du circuit. On suppose que la valeur  $D_0$  est déjà dans l'intervalle  $[0.5, 1]$ .

init :  $k \leftarrow 0$ ; init' ET  $k \neq 6$ :  $k \leftarrow k + 1$ ;  
init :  $N \leftarrow N_0$ ; init' ET  $k \neq 6$ :  $N \leftarrow N \times (2 - D)$   
init :  $D \leftarrow D_0$ ; init' ET  $k \neq 6$ :  $D \leftarrow D \times (2 - D)$

Donnez le diagramme d'un chemin des données correspondant à ces micro-opérations.

Examen intra #2

# 6 Conception et implémentation de fonctions arithmétiques sur FPGA

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Analyser un modèle VHDL de fonctions arithmétiques afin d'estimer le nombre de ressources nécessaires pour l'implémenter. (B4)
  - Concevoir et modéliser en VHDL, en vue de son implémentation, un circuit qui effectue des calculs arithmétiques avec des nombres binaires et décimaux à virgule fixe. (B5)

Considérez le code VHDL suivant. Pour chacun des signaux en sortie, estimez le nombre de ressources nécessaires en termes de tables de conversion (LUT), bascules (FF) et tranches DSP48 pour l'implémenter sur un FPGA de la famille Virtex 5 de Xilinx. Justifiez complètement votre réponse.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity ual_ressources is
    generic (
        W : positive := 8
    );
    port (
        clk : in std_logic;
        A, B : in signed(W - 1 downto 0);
        F, H, J : out signed(W - 1 downto 0);
        K : out signed(W + 2 downto 0);
        L : out signed(2 * W - 1 downto 0)
    );
end ual_ressources;

architecture arch of
    ual_ressources is
begin
    process (clk)
    begin
        if rising_edge (clk) then
            F <= A + B;
            H <= abs(A);
            J <= A / 4;
        end if;
    end process;

    K <= A * 5;
    L <= A * B;
end arch;
```

Examen intra #2

# 7 Vérification de systèmes numériques

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Énumérer et décrire les principes généraux de la vérification d'un système numérique ainsi que les qualités d'un ensemble de vecteurs de test. (B1, B2)
  - Appliquer les techniques de test de boîte noire et de boîte blanche à la vérification de circuits combinatoires et séquentiels. (B3)
  - Analyser la spécification d'un système numérique, concevoir un ensemble de vecteurs de tests pour en vérifier le fonctionnement, et coder un banc d'essai correspondant en VHDL. (B4, B5)

Considérez le code VHDL suivant. ...

Combien de vecteurs de test seraient nécessaires pour effectuer un test exhaustif ? Justifiez complètement votre réponse et montrez tous vos calculs.

Proposez un ensemble de vecteurs de test à appliquer au port x qui maximise le nombre d'énoncés couverts. L'ensemble devrait être le plus petit possible. Justifiez votre réponse.

Examen intra #2

# 8 Simulation, synthèse et implémentation

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Appliquer les principes de la simulation d'évènements concurrents à un modèle VHDL. (B3)
  - Expliquer le processus de synthèse d'un modèle VHDL, expliquer pourquoi certaines structures de code ne sont pas synthétisables, et écrire du code synthétisable. (B2, B3)
  - Proposer un placement et un routage pour un module VHDL maximisant des métriques de performance. (B5)

Examen intra #3

Considérez l'extrait de code VHDL suivant et les valeurs des signaux CLK, reset et A montrées sur le chronogramme. Complétez le chronogramme pour les signaux et variables T, U, V et F

```
library IEEE;
use IEEE.std_logic_1164.all;

entity VHDLEstMonAmi2 is
  port (
    clk, reset : in std_logic;
    A : in integer;
    F : out integer
  );
end VHDLEstMonAmi2;

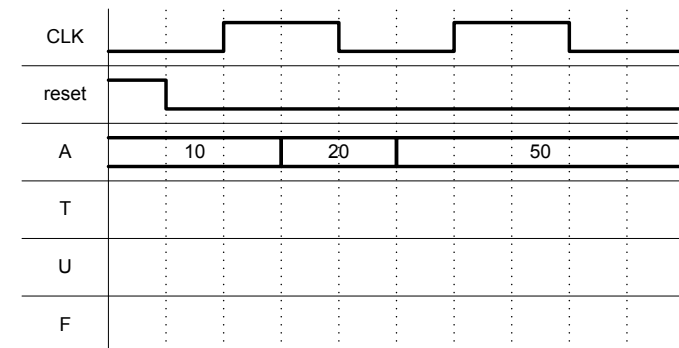
architecture jaiameVHDL of VHDLEstMonAmi2
is
  signal T, U : integer;

begin

  process (clk, reset)
  variable V : integer := 10;
  begin
    if reset = '1' then
      T <= 0;
      U <= 0;
    elsif rising_edge(clk) then
      T <= T + V;
      V := V + 5;
      U <= T + V;
    end if;
  end process;

  process (A)
  begin
    F <= A + T + U;
  end process;

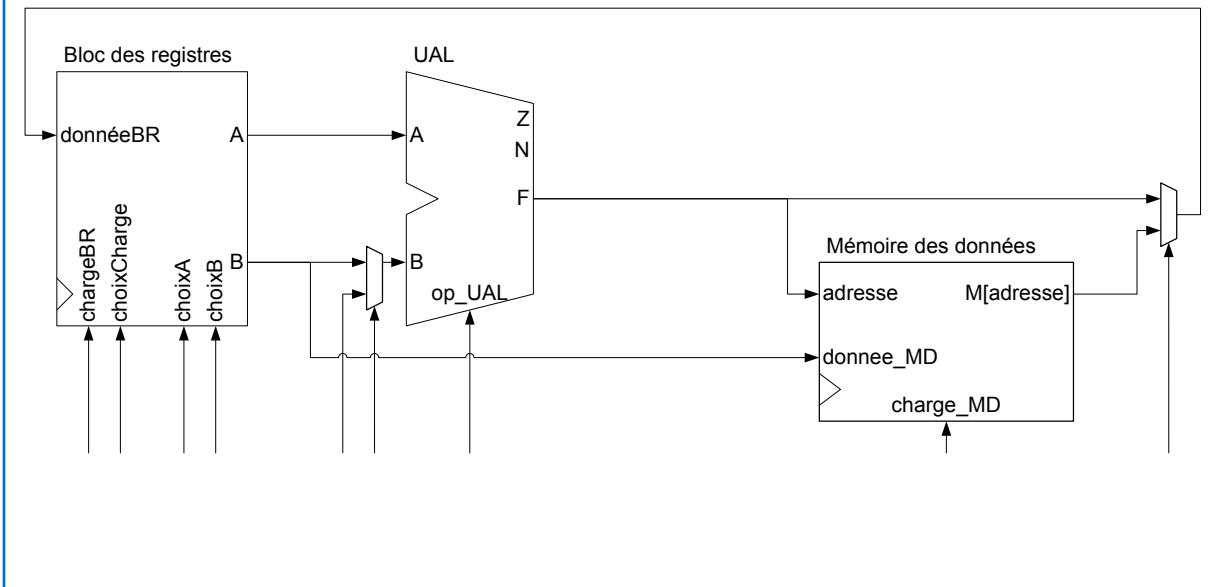
end jaiameVHDL;
```



# 9 Processeurs à usage général

- Après avoir couvert ce chapitre, l'étudiant(e) devrait être capable de:
  - Analyser le chemin des données d'un processeur à usage général afin d'estimer les ressources requises, déterminer les limites opérationnelles du processeur, proposer des changements architecturaux pour implémenter de nouvelles opérations, et donner le code VHDL modifié. (B5)
  - Proposer des modifications à l'unité de contrôle et à l'encodage des instructions pour ajouter de nouvelles instructions à un processeur à usage général, et donner le code VHDL modifié. (B5)

Estimez combien de ressources seraient nécessaires pour implémenter le chemin des données du processeur PolyRISC sur un FPGA de la famille Virtex-5. Supposez que le bloc des registres comporte 4 registres de 16 bits, que l'UAL peut effectuer les 8 opérations suivantes : A, B, A + B, A - B, A ET B, A OU B, NON A, A OUX B, et que la mémoire des données comporte 256 mots de 16 bits. Donnez votre réponse en termes de LUT, de bascules et de bits de mémoire Block RAM. Montrez tous vos calculs et justifiez complètement votre réponse. Utilisez le verso si nécessaire.

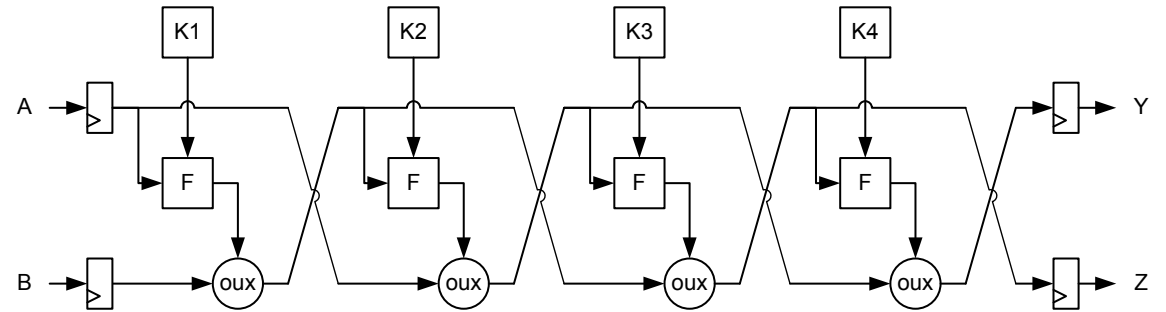


Examen intra #3

# 10 et 11: Performance parties 1 et 2

- Après avoir couvert ces chapitres, l'étudiant(e) devrait être capable de:
  - Analyser un système numérique pour identifier le chemin critique et en donner la fréquence maximale d'horloge. (B4)
  - Dans le processus de conception d'un circuit, adopter de bonnes pratiques pour atteindre les spécifications temporelles et éviter des problèmes de synchronisation. (B5)
  - Analyser un circuit numérique pour en mesurer ou estimer la latence, le débit et la surface. (B4)
  - Modifier un circuit existant, ou en concevoir un nouveau, en optimisant la latence, le débit et/ou la surface. (B5)

Le réseau de Feistel est utilisé dans les algorithmes de chiffrement par bloc.



- Identifiez le chemin critique du circuit sur le diagramme et donnez la fréquence maximale d'opération.
- Pipeliner le circuit pour atteindre un débit de  $50 \times 10^6$  résultats par seconde, où un résultat est une paire (Y, Z). Minimisez la latence.
  - Montrez sur le diagramme où vous insérez des registres de pipeline
  - Indiquez le nouveau chemin critique du circuit et donnez la fréquence maximale d'opération
  - Donnez la nouvelle latence.

Examen intra #3

# Se préparer à un examen

---

1. (*principe maître*) À l'examen, le but pour l'étudiant(e) est d'amasser le plus de points possible.
2. Comme on dit dans l'infanterie, "le temps passé en reconnaissance est rarement perdu": donc, avant d'écrire quoi que ce soit, lire rapidement l'examen au complet 2 fois.
3. Répondre aux questions dans un ordre croissant de difficulté.
4. Il vaut mieux donner une réponse imparfaite à toutes les questions qu'une réponse parfaite à une seule question.
5. Diviser le temps alloué selon les points accordés à chaque question.
6. Amasser des points:
  - Aucun point n'est accordé pour ce que l'étudiant(e) veut dire: seuls des points sont accordés pour ce qui est écrit;
  - Aucun point n'est accordé pour une réponse à une question qui n'est pas posée;
  - Aucun point n'est accordé à une question pour laquelle une réponse n'est pas soumise.
7. Effacer plus de quelques mots est du temps perdu - il vaut mieux raturer, surtout quand on réalise plus tard que ce qu'on avait écrit était correct.
8. Il vaut mieux étudier une matière peu et souvent plutôt que d'essayer de tout revoir en une seule période.
9. Le cerveau travaille mieux quand il est reposé.
10. La meilleure pause pendant l'étude est l'exercice physique.

# En conclusion

---

- Les systèmes numériques sont fascinants. Ils sont à la base de tous les équipements informatiques de la vie de tous les jours.
- J'espère que vous êtes satisfait de vos apprentissages dans ce cours.
- Bonne chance dans la suite de votre programme d'études.
- Bonne chance dans votre future carrière comme ingénieur ou ingénieure.