
Révision mi-session

GIF-1001

Ordinateurs : Structure et Applications

Hiver 2018

Marc-André Gardner

Jean-François Lalonde





Logistique

- L'examen aura lieu le 27 février de **14h30 à 17h20**
- PLT-1112
- **Une** feuille recto-verso **écrite à la main** permise
 - Préparer la feuille est souvent plus utile que la feuille elle-même!
 - Pas besoin de copier la spécification ARM au complet : un annexe vous est fourni
- Calculatrice autorisée (modèle approuvé par la FSG)
- Vous **devez** avoir votre carte étudiante en votre possession
- Vos réponses **doivent** être écrites **uniquement** dans le cahier bleu fourni
 - Vous assurer que c'est bien votre nom qui y est écrit pourrait être utile

Avant l'examen

- Parmi les éléments utiles à l'étude :
 - Travaux pratiques
 - Exercices facultatifs
 - Examens des années antérieures
 - Notes de cours
- Mais aussi :
 - Pensez à dormir
 - Pensez à manger
 - Arrivez avec un esprit positif



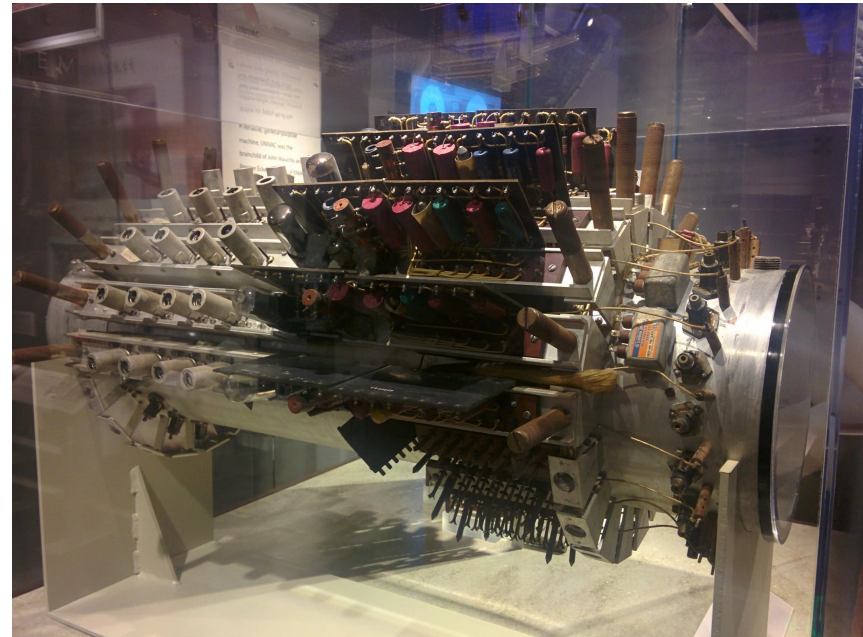
Pendant l'examen

- **Avant de commencer :**
 - Compter les pages
 - Détacher les annexes
 - Survoler *toutes* les questions
 - Rappelez-vous du concept de *verso*
- **Par la suite :**
 - Commencez par les questions faciles
 - Gérez votre temps (vous pouvez passer une question!)
 - Demandez-vous si une de vos réponses contredit une (ou mieux, plusieurs) loi de la physique : si oui, révisez cette réponse ou songez à aller réclamer un prix Nobel
 - Prenez le temps de respirer



Contenu de l'examen

- Tout ce qui a été vu jusqu'ici :
 - ~20% représentation des nombres
 - ~20% structure d'un ordinateur
 - ~50% assembleur
 - ~10% questions générales
 - Par exemple : qu'est-ce que ceci?
- Vous n'avez **pas** vu :
 - Les interruptions
 - Les micro-instructions





Représentation des données

- Si vous ne devez retenir qu'une seule chose :

**La seule chose qui existe
réellement dans
l'ordinateur est le binaire**

Représentation des données



- Tout, absolument tout, est stocké en format binaire
- L'hexadécimal est une autre manière d'écrire le binaire (4 bits = 1 carac.)
- Une chaîne binaire peut représenter *n'importe quoi*:
 - Du texte (ASCII)
 - Des nombres entiers (signés ou non) ou à virgule
 - Une instruction
 - Une adresse
 - ~~La déception pour un chien de ne pas pouvoir posséder toutes les balles de tennis du monde~~



Représentation des données

- Si vous acceptez de pousser le vice à retenir *deux* choses :

**Sans contexte, il est
impossible de savoir ce qu'une
chaîne de bits signifie**



Représentation des données : à savoir

- Conversion décimal/binaire (et hexadécimal)
- Complément 2
- Notions de *retenue* et de *débordement*
- Convertir des nombres au format IEEE-754 (nombres à virgule)

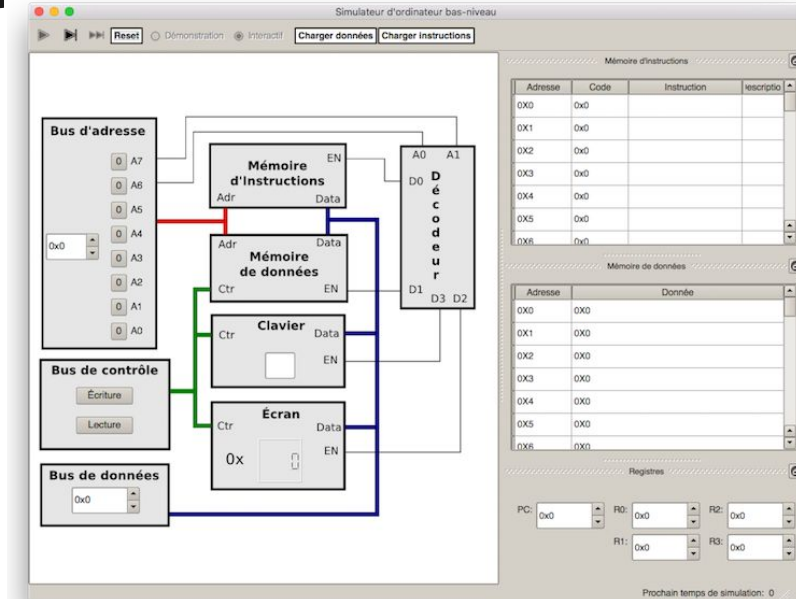


Bus, mémoires et périphériques

- Une mémoire se caractérise par 2 nombres
 - (A) La taille des *mots* qu'elle peut stocker (en bits)
 - (B) Le *nombre* de mots qu'elle peut stocker
- La taille du bus de *données* est *au moins* (A) (mais peut être un multiple)
- La taille du bus *d'adresse* est liée à (B) (mais **pas** égale)
- La taille totale de la mémoire est le produit de (A) et (B)
- La taille du bus de contrôle... ne dépend de rien de tout ça

Bus, mémoires et périphériques

- Un *décodeur* d'adresse peut être présent sur le bus d'adresse
- Permet d'adresser *plusieurs* mémoires et/ou périphériques de manière transparente pour le processeur
 - Mais réduit le nombre d'adresses disponibles pour chacun des composants du bus
- De là dérive le concept de *carte mémoire*



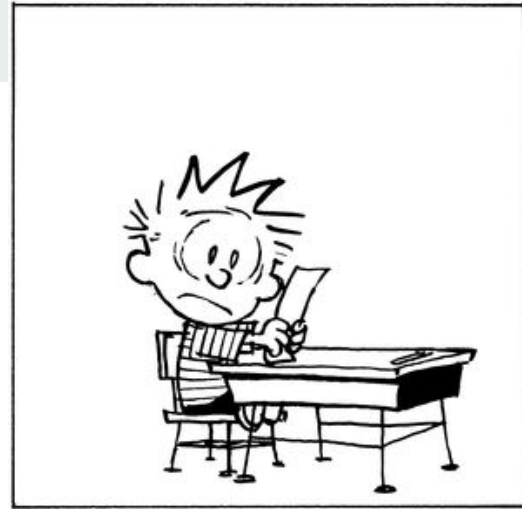


Assembleur ARM (TP2 et suivants)

- Connaître certains éléments fondamentaux de l'architecture ARM
 - Quelle taille font les instructions?
 - Combien de registres un processeur ARM possède-t-il?
 - RISC ou CISC?
 - Comment la mémoire est-elle adressée?
 - $PC = PC + 4$
 - Fetch, decode, execute
 - PC est toujours en *avance* sur l'instruction présentement exécutée! Il est déjà en train de lire 2 instructions plus loin!
 - $PC = \text{adresse de l'instruction exécutée présentement} + 8$

Assembleur ARM

- Éléments de programmation devant être maîtrisés :
 - Variables et étiquettes
 - Distinction entre la *valeur* d'une variable et son *adresse*
 - Adressage relatif à PC
 - Décalages
 - Drapeaux et codes de condition
 - Sauvegarde et restauration de valeurs à partir de la pile
- N'oubliez pas que vous n'avez pas droit au simulateur!





Assembleur ARM

- Vous n'avez pas à connaître les instructions par coeur
- Mais vous devez savoir les utiliser et les comprendre
- Vous allez devoir *analyser et expliquer* des programmes simples
 - Exemples : n'importe quel exercice ou démonstration, examens des années antérieures
- Vous allez devoir *écrire* des programmes simples :
 - Exemple de programme : *écrivez une fonction qui compte le nombre de '1' dans le contenu binaire de R4 et stocke ce nombre dans R5*

Bonne chance!

(et oui, il y aura un cours vendredi prochain)

