

NOM:	Prénom:	N° Groupe:
------	---------	------------

**ESIAL 1^{ÈRE} ANNÉE
MODULE PFSI**

EXAMEN

Jeudi 20 Mars 2008

Les quatre parties sont indépendantes.

Aucune calculette n'est autorisée;

*Aucun document autre que
la documentation fournie
n'est autorisé;*

L'épreuve est individuelle;

Durée: 2 heures

1. HISTOIRE DE L'INFORMATIQUE (RÉPONDRE SUR CETTE FEUILLE)

1.1. Durant quelle décennie le premier calculateur électronique a été inventé ?

1.2. Qui était l'inventeur ?

2. CODAGE DE L'INFORMATION (RÉPONDRE SUR CETTE FEUILLE)

2.1. On suppose que la mémoire d'un système informatique "big endian" contient les octets suivants (ici indiqués en hexadécimal) à des adresses consécutives :

34 45 F8 2A

Ces octets sont maintenant interprétés comme des entiers codés selon les formats ci-dessous, donner les valeurs représentées :

a/ **Non signés** codés sur 16 bits :

b/ **Signés** codés sur 16 bits en **code complément à deux** :

2.2. On considère des valeurs réelles dans $[-M ; M]$. On cherche à les coder au plus près comme des valeurs entières en les arrondissant, donc avec une erreur.

a/ quelle erreur maximum peut-on commettre ?

b/ quel nombre N de bits doit-on avoir pour coder ces valeurs avec une précision relative $= |\text{erreur}| / M < 1\%$ pour $M = 200$?

3. ANALYSE ET SYNTHÈSE D'OPÉRATEUR LOGIQUE (RÉPONDRE SUR LA COPIE)

On considère l'opérateur AU (Arithmetic Unit) dont les spécifications suivent :

Interface:

No m	Sens	Taille en bits	Commentaire
P	entrée	1	commande de passage de <u>B</u>
I	entrée	1	retenue d'entrée
<u>A</u>	entrée	4	opérande A en code binaire
<u>B</u>	entrée	4	opérande B en code binaire
<u>R</u>	sortie	4	résultat en code binaire
C	sortie	1	retenue de sortie

Comportement :

P	R
0	<u>A</u> # I
1	<u>B</u>

- 3.1. Dessiner la **représentation symbolique** (interface) de cet opérateur.
- 3.2. Dessiner le **schéma synoptique de la décomposition** (structure) de cet opérateur en tranches de 1 bit de large.
- 3.3. Dessiner la **représentation symbolique d'une tranche**.
- 3.4. Spécifier le **comportement d'une tranche** à l'aide d'un tableau donnant pour chaque valeur de la commande P l'expression algébrique des sorties en fonction des autres entrées.
- 3.5. En déduire la table de Karnaugh de chacune des sorties d'une tranche.
- 3.6. En déduire les **polynômes booléens réduits** (i.e. forme normale disjonctive réduite) de chacune des sorties d'une tranche.
- 3.7. Dessiner la **structure interne d'un opérateur MYST** de spécifications suivantes en utilisant l'opérateur AU et éventuellement **un seul** opérateur classique parmi : NOT, AND, OR, MUX, ADD, LFF, DFF, REG. On utilisera une machine de Moore simplifiée.
- 3.8. Cet opérateur MYST est-il combinatoire, séquentiel asynchrone ou séquentiel synchrone ?
- 3.9. Quel est cet opérateur MYST ?

Interface :

Nom	Sens	Taille en bits	Commentaire
k	entrée	1	Horloge (« clocK »)
T	entrée	1	Commande d'incrémentation ("Toggle")
L	entrée	1	Commande de chargement ("Load")
D	entrée	4	Entrée de donnée ("Data")
Q	sortie	4	Donnée contenue dans l'opérateur

Comportement:

Événement	Action	Commentaires
L=1 et k↑	<u>Q</u> ← <u>D</u>	Changement
L=0 et T=1 et k↑	<u>Q</u> ← <u>Q</u> # 1	Incrémentation

(Lorsque k ne monte pas et **pour les autres événements**, Q garde sa valeur).

NOM: **Prénom:** **N° Groupe:**

4. MICROPROGRAMMATION (RÉPONDRE SUR CETTE FEUILLE)

On considère la machine décrite dans la documentation (identique à celle vue en cours)

4.1. Donner toutes les **actions** concernant le chemin de donnée de la machine décrite dans la documentation (i.e. qui changent l'état de la machine, c'est à dire affectations des cases mémoire, des registres etc.) nécessaires pour effectuer *l'instruction* suivante :

NEG R2, R3

Globalement, cette instruction "NEGATE" charge le registre R3 avec l'opposé du contenu du registre R2.

(*On ne considérera pas ici le chemin de contrôle*, micro-PC, PC, registre d'état SR, d'instruction IR etc.).

4.2. Déterminer la valeur numérique (en binaire) de chacun des champs du code de micro-instruction (MIC, « Micro-Instruction Code ») (répondre sur la feuille du sujet).

Les champs dont la valeur est sans importance pourront être marqués .

NOM:	Prénom:	N° Groupe:
------	---------	------------

5. PROGRAMMATION EN LANGAGE D'ASSEMBLAGE (*RÉPONDRE SUR LA FEUILLE*):

5.1. **Écrire une fonction** qui prend le paramètre x dans le registre R1, et retourne le résultat f(x) dans R0 :

$$f(x) = 2|x| - 5$$

Note: Une fonction est un sous-programme qui retourne une valeur.

5.2. **Écrire *ci-dessous* un programme en assembleur qui empile la valeur 0323h sur la pile** (dont le pointeur est R15).

5.3. **Écrire *ci-dessous* un programme en assembleur qui additionne le contenu de la case mémoire 2000h au contenu de la case mémoire 2002h** avec aussi peu d'instructions que possible.