

# P R O G R A M M E   D é T A I L L é   D E   P F S I

=====

*(le niveau important de détail en explique la longueur).*

## HISTOIRE DE L'INFORMATIQUE : (CMI)

Connaître les faits suivants :

- o codage binaire des caractères : Fou Hi - -3000 ;
- o notion d'algorithme : Al-Khowarzmi - 820
- o notation positionnelle des nombres avec base décimale en Europe : 1000 ;
- o lère machine à calculer : Pascaline - Blaise Pascal - 1642 ;
- o conception de l'architecture de base d'un ordinateur : Charles Babbage - 1833;
- o premières stuctures logicielles (boucle, conditionnelles ...) : Ada Lovelace (proche de Babbage) - 1840;
- o algèbre de Boole : Boole et Morgan - 1854
- o théorie de la complexité : Alan Turing - 1937 ;
- o 1er calculateur numérique électronique (à lampes) : ABC - John Atanazoff - 1941 ;
- o 1er ordinateur électronique spécialisé pour la cryptographie (décryptage de l'Enigma) : Colossus - 1943 ;
- o premier ordinateur programmable :
  - ENIAC 1 - 1946 - Eckert et Mauchly, à partir des travaux d'Atanazoff ;
- o premier transistor (bipolaire) : 1948
- o rapport sur l'EDVAC : John Von Neumann (à partir des travaux d'Eckert et Mauchly) ;
- o premier ordinateur électronique à programme enregistré en mémoire : EDSAC - Eckert et Mauchly - 1949;
- o premier ordinateur à transistors : MIT/Lincoln Laboratory - TX0 - 1957 ;
- o premier mini-ordinateur (CPU simplifié et petit) : DEC - PDP8 - 1964;
- o premier microprocesseur ( CPU intégré sur une puce monolithique i.e. monocristal de silicium ):
  - Intel i4004 - 1971
- o premiers micro-ordinateurs (utilisant un microprocesseur pour CPU) mi-70 :
  - Altair - 1975      avec microproc 8080, amélioration (pile en mémoire) du i8008 (Bill Gates & Paul Allen en ont écrit le moniteur et l'interpréteur du langage Basic)
  - IBM - PC - 1981    avec microprocesseur i8088, i8086 avec bus 8 bits = extension 16 bits du i8080.
- o premiers RISCs commerciaux: fin des années 80 .
- o pas au programme :
  - toute connaissance d'histoire en dehors des faits ci-dessus.

**CODAGE: (CM2 + TD1)** (cf. *polycopié distribué n°1 "Information"*)

- o système de numération positionnelle, base 10, 16, 8, 2.
- o conversions entre ces bases;
- o codage des nombres entiers non signés
- o codage des nombres entiers signés:
  - codage avec biais ;
  - code complément à 2 ;
- o arithmétique binaire : addition, soustraction, multiplication, division;
- o codage des nombres avec virgule fixe ;
- o codage du texte ASCII (la table ASCII n'est pas à connaître par coeur!)
  
- o pas au programme:
  - cette année, vous n'aurez pas le codage des nombres réels avec virgule flottante  
(vu en mathématique numérique);
  - le codage des images (vu en TNI), son (vu en TAN) etc. n'est pas à savoir pour PFSI;

**LOGIQUE COMBINATOIRE (CM3 + TD3)**

(cf. *polycopié distribué n°2 "logique combinatoire"*):

- o opérateurs NOT, OR, AND, XOR, NOR, NAND ;
- o algèbre de Boole et lois de Morgan ;
- o fonctions binaires de plusieurs entrées à plusieurs sorties ;
- o décomposition
  - tranches
  - couches
- o polynômes booléens (formes disjonctives) ;
- o réduction d'un polynôme booléen par table de Karnaugh
- o étude d'un opérateur :
  - spécification interface et comportement
  - analyse
  - synthèse
  
- o multiplexeurs ;
  
- o additionneur à propagation de retenue (cf. *polycopié et document sur le Net*);
  
- o unité arithmétique et logique (cf. *polycopié, TD 3 et corrigés examen sur le Net*).
  
- o décaleur (cf. *polycopié et TD3*)
  
- o pas au programme :
  - retenue anticipée ;
  - soustracteur ;
  - autres opérateurs autres que MUX, ADD, ALU, SHIFT, ALSU.

#### **LOGIQUE "DYNAMIQUE" (CM4 + TD4):**

- o délai en logique combinatoire (cf. polycopié distribué n°2 "logique combinatoire")
  - délai de propagation;
  - composition des délais;
  - chronogrammes;
  - aléas (exemple: multiplexeur avec X0 X1 S = 1 1 v)
- o logique séquentielle asynchrone (cf. polycopié distribué n° 3 "logique séquentielle asynchrone"):
  - spécification et structure de la bascule verrou (LFF, Latch Flip-Flop);
  - spécification de la bascule D (DFF, Data Flip-Flop);
- o logique séquentielle synchrone (cf. polycopié distribué n°4 "logique séquentielle synchrone"):
  - machine de Moore synchrone ;
  - machine de Moore synchrone simplifiée (réseau logique de sortie = identité) ;
  - spécification et analyse du registre REG avec une machine de Moore synchrone simplifiée ;
  - idem compteurs CNT (avec remise à zéro, chargement);
- o pas au programme :
  - les bascules SR, T, JK ne sont pas au programme ;
  - la structure et l'analyse de la bascule D n'est pas au programme ;
  - le registre à décalage (SHIFT REG) n'est pas au programme.

#### **ELECTRONIQUE NUMERIQUE (CM5, TP MOS)**

- o interface et comportement qualitatif des transistors MOS:
  - N-MOSFET (G=1 => passant, sinon bloqué) ,
  - P-MOSFET (G=0 => passant, sinon bloqué) ;
- o porte NOT NMOS ;
- o porte NOR NMOS ;
- o porte NOT CMOS ;
- o porte NOR CMOS ;
- o états qualitatifs d'une borne électrique:
  - 0 (branché à la masse),
  - 1 (branché au + alimentation),
  - Z (débranché);
- o intégration: loi de Moore :
  - la taille minimale gravée divise par racine de 2 tous les deux ans  
(production débutée en 2007: 45 nm; 2005: 65nm; 2003: 90nm; 2001: 130nm; 1999: 180nm  
1997: 250 nm; 1995: 350nm; ...)
  - => le nombre de transistors intégrables sur une puce multiplie par deux tous les deux ans ;  
(ceci a été bien suivi, mais en fait, la loi de Moore originale prédisait "tous les ans" ...)
  - => le délai d'une porte divise par racine de 2 tous les deux ans ;
  - => la performance multiplie par 2,828 tous les deux ans.

## **CHEMIN DE DONNEES (CM6 + TD5)**

- o principes de l'architecture d'un ordinateur (CPU + CMU + I/O) (cf. document sur le Net);
- o architecture fonctionnelle d'une mémoire (SRAM) (cf. document sur le Net);
- o principes de l'architecture d'un CPU (cf. doc. sur Net);  
(unité ou chemin de données + unité ou chemin de contrôle)
- o chemin de données (cf. photocopié distribué):
  - unité de calcul (ALU + décaleur) cf. TD3 ;
  - registre accumulateur A ;
  - tampon ("buffer" BUF) 3 états: 0, 1, Z ;
  - bus de données, bus interne ;
  - registre d'index X ;
  - bus d'adresse ;
  - générateur de constantes ;
  - bloc de registres avec port bidirectionnel ;
  - gestion des bytes (big endian, small endian);
  - structure, fonctionnement du chemin de données
- o espace d'adressage (cf. TD5);
- o interfaçage sur le bus (cf. TD5)
  - SRAM, EPROM ;
  - port de sortie (registre) ;
  - port d'entrée (tampon) ;

## **CHEMIN DE CONTROLE (CM7 + TD6 + TD7), (cf. photocopié distribué)**

- o compteur ordinal PC ;
- o saut (ou branchement ou déroutement) conditionnel  
(cf. photocopié et TD6) ;
- o indicateurs
  - CF (retenue),
  - NF (signe),
  - ZF (zéro),
  - VF (débordement arithmétique),
  - registre d'état SR ;
- o conditions de saut :
  - générales (NV, AL, EQ, NE) ;
  - inégalités opérandes signés (LE, GE, LW, GT) ;
  - inégalités opérandes non signés (BE, AE, BL, AB) ;
- o contrôleur de branchement ;
- o registre d'instruction IR ;
- o "fetch" du code d'instruction ;
- o contrôleur microprogrammé :
  - séquenceur (ici micro-PC ou uPC) ,
  - décodeur d'instruction ;
- o codage de la micro-instruction (pas par coeur !) ;
- o microprogrammation.
- o Note :
  - le format du code de microinstruction sera fourni avec le sujet ;
  - l'architecture du CPU sera fourni avec le sujet .

## **JEU D'INSTRUCTIONS (CM8 + TD7 + TD8) cf. "carte de programmation" distribuée en TD8**

- o codage des instructions (cf. "carte de programmation" distribuée);
  - groupes et formats d'instructions ;
  - champ et format .
- o instructions d'opération de traitement (groupes I et II) :
  - opérations arithmétiques ADD, SUB, NEG ,
  - opérations logiques AND, OR, XOR, NOT ,
  - opérations de décalage logique SRL, SLL et arithmétique SRA ;
- o instructions de transfert Load et Store (groupe III);
- o modes d'adressage (cf. polycopié distribué)
  - registre ,
  - basé (sur un registre) dit aussi "deffered" ,
  - immédiat ;
- o branchements (Branch) ou saut (Jump)  
inconditionnels et conditionnels relatifs :
  - long (Jcc : JMP ou JAL, JEQ, ...) ;
  - court (Bcc: BMP ou BAL, BEQ, ...) ;
- o saut inconditionnel absolu (JEA) ;
- o instructions de contrôle diverses:
  - NOP (No Oparation) ,
- o pas au programme :
  - les modes indirect, indexé et indirect-indexé ne sont pas au programme (vus en trad);
  - les instructions d'extension ADC, SBB, RLC, RRC ne sont pas au programme.
  - la carte de programmation n'est pas à apprendre par coeur:  
elle sera fournie avec le sujet d'examen.
  - les instructions :  
ADI, ANI, SWP, EXT, RLB, TST, TSR, JPA  
ne sont pas au programme ;

## **PROGRAMMATION EN CODE MACHINE (TD 8)**

- o codage des instructions en hexadécimal ;
- o adresse de chargement ;
- o adresse de démarrage .

## **PROGRAMMATION EN LANGAGE D'ASSEMBLAGE** (cf. TD9 + résumé distribué)

- o syntaxe
  - étiquette ,
  - mnémonique instruction ,
  - opérandes (avec indication du mode d'adressage),
  - commentaire ;
- o instructions complémentaires "rapides" (groupe VIII)
  - ADQ ,
  - LDQ ;
- o mode d'adressage complémentaires :
  - direct ;
- o directives d'assemblage :
  - org (adresse de chargement),
  - start (adresse de démarrage),
  - equ (remplacé par)
  - rsb (réserve des octets);
- o expressions et \$
- o pas au programme :
  - les syntaxes assembleur ne sont pas à connaître par coeur:  
le résumé sera fourni avec le sujet d'examen.

## **PILE** (CM9 + animation sur le Net)

- o empilage, dépilage ;
- o modes d'adressage complémentaires :
  - basé pré-décrémenté ,
  - basé post-incrémenté ;
- o initialisation de pile ;
- o débordement de pile .

## **SOUS-PROGRAMME** (CM10 + animation sur le Net, TD 10)

- o programme ré-entrant ;
- o empilage de l'adresse de retour ;
- o instructions complémentaires:
  - JSR ,
  - RTS ;
- o passage de paramètre par registre ;
- o pas au programme de PFSI (sera vu en trad):
  - passage de paramètre par la pile ;
  - passage de paramètre par descripteur ;
  - variable locale ;
  - environnement pile .

**EXCEPTIONS** (CM 11, TD 11, TP exceptions)

o taxonomie des exceptions :

- exception non vectorisée: (adresse du programme de service fixe)
  - \* INITIALisation (ligne ReSeT) adresse = FFFA;
- exceptions vectorisées (adresse du prog. de service dans un vecteur en table):
  - \* matérielles :
    - . déclenchée par un périphérique (ligne IRQi) : interruption (matérielle) ;
    - . déclenchée par le CPU : trappe matérielle ;
  - \* logicielle :
    - . instruction TRP: trappe logicielle.

o déroulement de la réponse du CPU à une requête d'exception matérielle (i.e. interruption);

o décodeur d'exception matérielle ;

o indicateurs complémentaires :

- IF (Interrupt enable Flag),
- WF (Wait Flag);

o ligne de requête d'interruption décodée IRQi

o n° de ligne de requête d'interruption ;

o n° d'exception (ou d'interruption) INT ;

o vecteur d'exception (ou d'interruption) ;

o table des vecteurs d'exception (ou d'interruption) ;

o programme de service d'une exception (ou d'interruption);

o quelques trappes matérielles :

- instruction illégale ...
- débordement pile ,

o instructions complémentaires:

- CMP (CoMPare) ;
- RTI (Return from exception service routine),
- ENI (Enable exception),
- DSI (Disable exception),
- HLT (HaLT program execution and wait for interrupt),
- TRP (software TRaP ) ,

o pas au programme :

- distinction entre trappe matérielle, faute et erreur (vue en réseau-système).