


Info0204

Éléments d'architecture des ordinateurs

Chapitre 3 : *Notions d'architecture*

1. Les différents niveaux de langage
2. Architecture de von Neumann
 - a. principes
 - b. éléments caractéristiques
3. Un exemple simple de processeur

Ch Jaillet
Janv. 2019



Ch Jaillet

- URCA > UFR Sciences > Dept Maths, Méca, Info
- christophe.jaillet@univ-reims.fr
- <http://cosy.univ-reims.fr/~cjaillet>

1. Les différents niveaux de langage

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Langage objet	<div style="background-color: #f4a460; padding: 5px; border: 1px solid black;">class Voiture { ... } données, méthodes</div>	compi- lateur
Langage procédural	<div style="background-color: #ffff00; padding: 5px; border: 1px solid black;">for, if, read, write variables, instructions</div>	
Langage assembleur	<div style="background-color: #90c080; padding: 5px; border: 1px solid black;">MOV, CMP, ADD mnémoniques</div>	compi- lateur
Langage machine	<div style="background-color: #a0a0a0; padding: 5px; border: 1px solid black;">011010111... code machine</div>	

□ **Langage de haut niveau**

- abstraction
- indépendant de la machine

□ **Langage de bas niveau**

- concret, réalisation pratique
- dépend de la machine
 - jeu d'instructions du processeur
 - câblage du processeur

Info0201

Ch Jaillet

Introduction à la programmation orientée objet

Chapitre 3 : *Notions d'architecture*

1. Les différents niveaux de langage
2. **Architecture de von Neumann**
 - a. principes
 - b. éléments caractéristiques :
mémoire, UC et UAL, instructions
3. Un exemple simple de processeur

2. Architecture de von Neumann a. principes

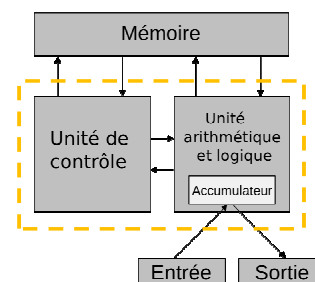
Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

extrait **Chapitre 1 : Introduction** (p. 20)

- ❑ 1944, Princeton
 - J. von Neumann + A. Buks + H. Goldstine
 - ⇒ théorisent les caractéristiques des futurs ordinateurs

- ❑ *utilisation du binaire*
- ❑ *usage multiple ; programmable*
- ❑ programmes stockés en mémoire

- ❑ structure :
 - Processeur + Mémoire + Bus
 - instructions et données stockées en mémoire
 - processeur = unité centrale (CPU : *Central Processing Unit*)
 - Unité de commande (UC)
 - Unité de calcul (UAL = Unité Arithmétique et Logique)
 - + Unité d'entrées/sorties (E/S)



2. Architecture de von Neumann

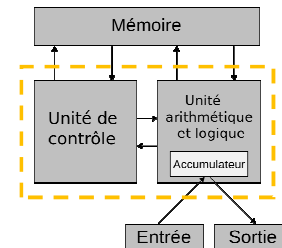
b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Mémoire

- informations :
 - données : les informations demandées ou produites par le calcul
 - instructions : les actions élémentaires à réaliser (éléments du programme)
- programmes stockés en mémoire : 2 modèles
 - *modèle de Harvard* :
 - mémoire d'instruction
 - mémoire de donnée
 - => utilisé surtout dans les MCU (micro-contrôleurs)
 - *modèle de von Neumann* = mémoire unifiée
=> se généralise pour les MPU (micro-processeurs)
- 2 types de mémoire
 - mémoire *volatile* :
 - programme et données en cours d'utilisation
 - mémoire «vive»
 - mémoire *permanente* :
 - programme et données de base de la machine
 - mémoire «de masse»

selon qu'elle risque, ou non, d'être altérée par un défaut d'alimentation électrique



5

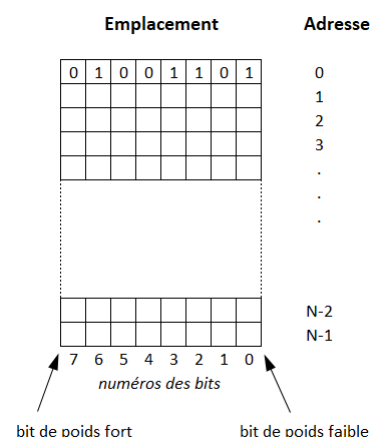
2. Architecture de von Neumann

b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Mémoire (suite)

- organisation :
 - un ensemble de cellules (=cases), contenant les informations
 - une information sur une case
 - ... ou sur plusieurs
 - => ensemble d'octets (8 bits) ou de mots (32 bits / 64 bits)
 - chacun identifié par son adresse
- utilisation :
 - écriture : enregistrer des données en mémoire
 - lecture : consulter une donnée stockée en mémoire ... par mots
- performance :
 - débit = vitesse moyenne de transfert
 - en octets par seconde Ko/s ...
 - ou bits / sec Mbps ...
 - latence (US : *lag*) = délai
 - temps de démarrage des transferts
 - en millisecondes / microsec. (10 µs)
 - dépend du bus



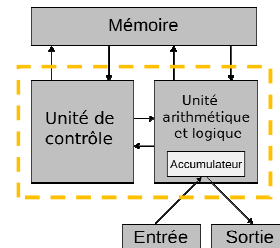
6

2. Architecture de von Neumann b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Bus

- = squelette des ordinateurs
 - communication entre proc., mémoire, cartes vidéo/son/réseau, accélérateurs
 - éléments reliés au bus par des *connecteurs*
 - bus = « *fond de panier* »
- différents types de signaux circulent sur le bus
 - alimentation
 - adresses, données
 - contrôles (instructions)
- plusieurs canaux d'interconnexion
 - transferts en parallèle
 - performance :
 - fréquence MHz
 - bande passante (largeur de bande) Kbps ou Ko/s
 - dépend de la performance de la mémoire / des éléments connectés



7

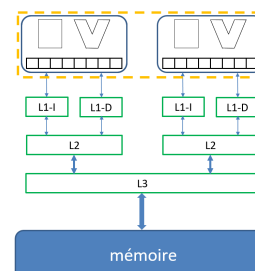
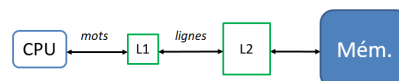
2. Architecture de von Neumann b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Processeur :

connecté à la mémoire

- relié à la mémoire par le bus
- **registres** (dans le proc.)
 - mémoires très rapides, petite quantité
 - stockage temporaire de données / résultats
 - registres spécialisés
 - pour les calculs / le déroulement des instructions
 - accumulateur : stockage du résultat d'un calcul
 - compteur ordinal (CO) [PC = *Program Counter*]
 - pointe sur la prochaine instruction à exécuter
 - ...
- **mémoire cache** (dans le proc.)
 - = *antémémoire* : anticiper les transferts (principe de localité)
 - souvent plusieurs *niveaux*
 - cache L1 (proche UC / UAL, qui peut être spécifique (L1-data vs L1-instr.))
 - cache L2 plus grand (mémoire unifiée)
 - éventuellement cache L3
 - proc. multi-cœur : partage possible entre plusieurs cœurs de calcul



8

2. Architecture de von Neumann

b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Processeur :
unités fonctionnelles

- ❑ **UC** : Unité de commande [*Control Unit*]
 - contient un dispositif de décodage des instructions (*décodeur*)
 - et un *séquenceur* (contrôle les circuits nécessaires à l'exécution de l'instruction en cours)
- ❑ **UAL** : Unité Arithmétique et Logique [*ALU = Arithmetic & Logical Unit*]
 - réalise les calculs +, -, *, / AND, NOT, ... [et combinaisons]

❑ **types d'échanges** →

❑ **registres spéciaux**

- contrôle
 - . RI : reg. d'instruction
 - . CO : compteur ordinal
 -
- calcul
 - . registres de calcul
 - . RE : registre d'état = CC (*Current Context*)

2. Architecture de von Neumann

b. éléments caractéristiques

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

instructions

- ❑ langage machine
 - l'ensemble des instructions reconnues, et son système de codage
 - dépend du processeur

❑ 3 types d'instructions

- transfert de données
- opérations arithmétiques / logiques
- contrôle (tests et *branchements*)

❑ 2 grandes familles :

- processeurs CISC
 - *Complex Instruction Set Computer*
 - une instruction peut effectuer plusieurs opérations élémentaires : chargements depuis la mémoire + calcul + sauvegarde en mémoire
 - 1 à plusieurs cycles selon les instructions
 - 80% des programmes n'utilisent que 20% des instructions
 - exemples : proc. généralistes d'Intel (**x86**, puis **Pentium**, **Core2**, ...), Motorola (**68xx**)
- processeurs RISC
 - *Reduced Instruction Set Computer*
 - chaque instr. n'effectue qu'une opération élémentaire (en temps constant : 1 cycle)
 - faible consommation => regain d'intérêt
 - exemples : **Sparc** (Sun, 1987), **PowerPC** (IBM+Apple+Motorola, 1992), **MIPS** (MIPS Computer Systems, 1985 => calculateur SGI, routeurs CISCO, consoles de jeux Nintendo64 / PlayStation / PSP / ...), **ARM** (Acorn Computers, 1983 => téléphonie, tablettes + processeurs graphiques)

2. Architecture de von Neumann

b. éléments caractéristiques

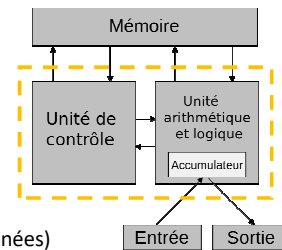
Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

□ Synthèse : caractéristiques d'un processeur

- fréquence d'horloge (en Mhz / GHz)
- micro-architecture (architecture interne)
 - traitement des instructions (vectorisation,)
 - nombre d'UAL, ...
- jeu d'instructions (ISA = *Instruction Set Architecture*)

□ Performance d'un ordinateur

- liée à la performance du processeur
- liée au débit des informations qui circulent (instructions, données)
- limitée par l'élément le plus lent (mémoire, bus, proc.)
- mesure de la performance :
 - fréquence [d'horloge]
 - instructions / s => MIPS
 - instructions en virgule flottante / s (*FLOating point instruction Per Sec.*) => GFLOPS
 - efficacité énergétique => FLOPS / Watt
 - ...



11

Info0201

Ch Jaillet

Introduction à la programmation orientée objet

Chapitre 3 : Notions d'architecture

1. Les différents niveaux de langage
2. Architecture de von Neumann
3. Un exemple simple de processeur
 - a. cahier des charges ; schéma ; description
 - b. synthèse (micro-architecture + ISA)
 - c. analyse / écriture de codes

3. Un exemple simple de processeur a. cahier des charges ; schéma ; description

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

□ Cahier des charges

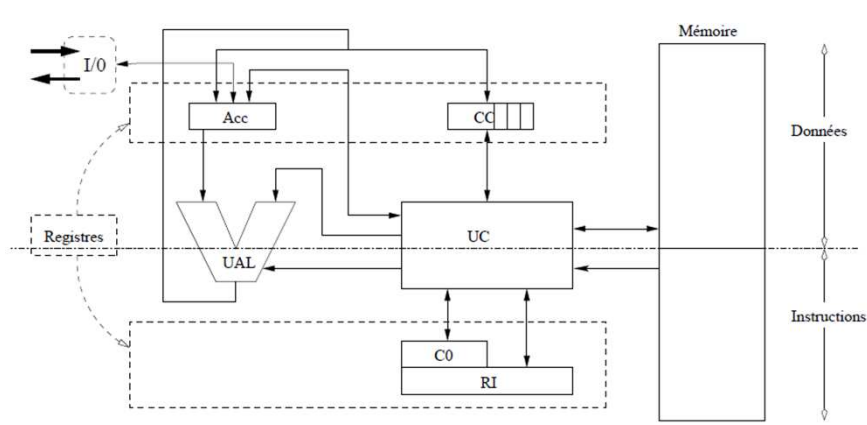
- Architecture simple
 - micro-architecture + jeu d'instructions (RISC !)
 - exemple simplifié, sans rapport avec la réalité du marché
- Ici, registres sur 1 octet (sauf RI : 2 octets)
 - contenu :
 - valeurs : entiers non signés sur 8 bits
 - adresses : non signées, sur 8 bits
 - un seul registre pour faire les opérations : Accumulateur
 - pas d'autre registre pour des besoins de stockage temporaire
 - les données / résultats des calculs doivent être systématiquement sauvegardés

13

3. Un exemple simple de processeur a. cahier des charges ; schéma ; description

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

□ Schéma de fonctionnement



14

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

3. Un exemple simple de processeur

a. cahier des charges ; schéma ; description

UC / UAL

❑ **UC** (Unité de Commande)
=> contrôles

- charge les instructions depuis la mémoire ; les décode
- fait exécuter les calculs
- peut lire et écrire des données :
 - dans / vers l'accumulateur
 - depuis / vers la mémoire

❑ **UAL** (Unité Arithmétique et Logique)
=> calculs

- données en entrée :
 - contenu de l'accumulateur
 - valeur en mémoire ou constante (valeur *immédiate*)
- résultats d'un calcul :
 - valeur, stockée dans l'accumul'
 - information de débordement / ...
 - stockée dans le registre d'état CC
 - permet des comparaisons, pour réaliser des *sauts*

15

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

3. Un exemple simple de processeur

a. cahier des charges ; schéma ; description

Instructions

❑ Ce sont les commandes de base :

- codées en binaire dans la mémoire
- pour les rendre plus "humaines", on les écrit en langage d'assemblage (*mnémoniques du langage*)

❑ Une instruction peut être :

- un accès à une case de la mémoire (lecture/écriture)
- un accès au registre de calcul, accumulateur (lecture/écriture)
- une opération arithmétique (+ , - , ...) ou logique (AND, ...)
- un accès au registre CO (écriture) : saut

❑ Syntaxe de notre langage assembleur :

COMMANDE [ARGUMENT]

(certaines commandes n'ont pas d'argument)

❑ code assembleur = suite d'instructions

- numérotées par ligne à partir de 0

❑ exemples d'instructions :

- LOAD Mem[3]
- ADD 6
- STORE Mem[6]

Voir documentation jointe

16

3. Un exemple simple de processeur

a. cahier des charges ; schéma ; description

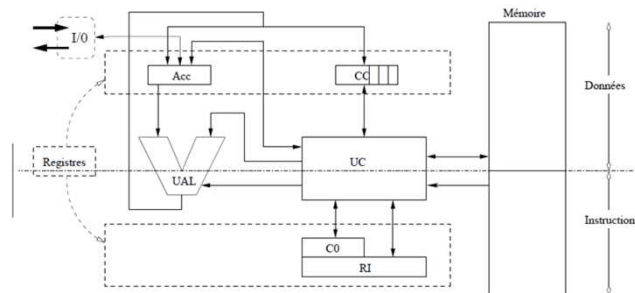
Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Les registres

- Acc (Accumulateur) 1 octet
- CC (Current Context) = RE (Registre d'État) 1 octet
 - mis à jour à chaque instruction de calcul, et par CMP
 - seul 2 bits utilisés, pour les *flags*

. CF : Carry Flag
. ZF : Zero Flag

0 0 0 0 0 0 1 0
← inutile → CF = 0
ZF = 1



- CO (Compteur Ordinal) 1 octet
 - contient l'adresse mémoire de la prochaine instruction à exécuter
- RI (Registre d'Instruction) 2 octets
 - contient l'instruction en cours de traitement

17

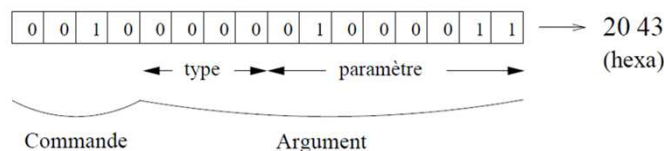
3. Un exemple simple de processeur

a. cahier des charges ; schéma ; description

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Codage des instructions en mémoire

- chaque instruction est codée sur 2 octets (16 bits)
 - commande : 16 valeurs => 4 bits
 - type de l'argument (0 : valeur *immédiate* / 1 : adresse mémoire)
 - 2 valeurs => 4 bits (1 utile)
 - paramètre 256 valeurs => 8 bits



- pas d'argument ? => CONVENTION
 - END, NOT paramètre = 0 ; type = 0
 - In/Out paramètre = 0 ; type = 1 (In) / 0 (Out)

18

3. Un exemple simple de processeur

a. cahier des charges ; schéma ; description

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Codage des instructions en mémoire (suite)

□ exemple

assembleur	hexadecimal	binaire
0 : LOAD Mem[1]	01 01	0000 0001 0000 0001
1 : MUL 31	A0 1F	1010 0000 0001 1111
2 : SUB #FA	90 FA	1001 0000 1111 1010
3 : Out	20 00	0010 0000 0000 0000
4 : END	70 00	0111 0000 0000 0000

□ Exercice :

- Donnez le code d'assemblage correspondant au code machine suivant :

01 0C 30 0A 40 05 00 00 60 06 00 01 11 12 70 00 d0 05

- Que fait ce programme ?

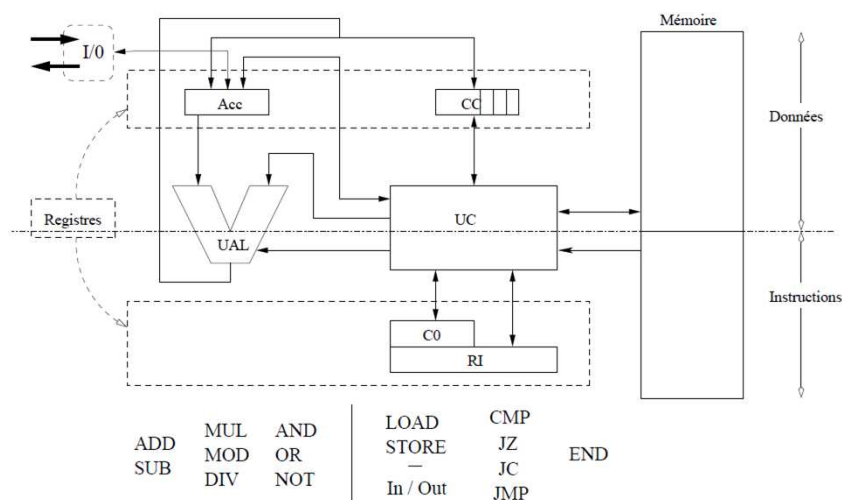
19

3. Un exemple simple de processeur

b. synthèse (micro-architecture + ISA)

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Schéma générale de cette architecture



20

3. Un exemple simple de processeur b. synthèse (micro-architecture + ISA)

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Liste des instructions de contrôle : UAL

Num	Code	Commande	arg.	description
8	1000 / 8	ADD	M, I	ajoute la valeur spécifiée (par l'argument) à Acc
9	1001 / 9	SUB	M, I	retranche la valeur spécifiée à Acc
10	1010 / a	MUL	M, I	multiplie Acc par la valeur spécifiée
11	1011 / b	MOD	M, I	calcule dans Acc le reste de la division euclidienne de Acc par la valeur spécifiée
12	1100 / c	DIV	M, I	calcule dans Acc le quotient de la division euclidienne de Acc par la valeur spécifiée
13	1101 / d	AND	M, I	ET logique, bit à bit
14	1110 / e	OR	M, I	OU logique, bit à bit
15	1111 / f	NOT	-	négation logique, bit à bit

❑ Les instructions de calcul modifient le registre d'état CC

21

3. Un exemple simple de processeur b. synthèse (micro-architecture + ISA)

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

Liste des instructions de contrôle : UC

Num	Code	Commande	arg.	description
0	0000 / 0	LOAD	M, I	charge dans Acc la valeur spécifiée (par l'argument)
1	0001 / 1	STORE	M	recopie la valeur contenue dans Acc dans l'emplacement mémoire spécifié
2	0010 / 2	In / Out	-	Entrée / Sortie, selon le type : In ⇔ type=1 ; Out ⇔ type=0
3	0011 / 3	CMP	M, I	compare Acc et la valeur spécifiée : modifie CC, selon $val - Acc$; pas Acc
4	0100 / 4	JZ	I	saut conditionnel vers l'instruction dont le numéro est spécifié en argument : saut ssi le flag ZF est à 1
5	0101 / 5	JC	I	saut conditionnel (si CF est à 1)
6	0110 / 6	JMP	I	saut direct (non conditionnel)
7	0111 / 7	END	-	fin du code assembleur : sortie

❑ A part CMP, les instructions de contrôle ne modifient pas CC

22

Ch Jaillet

Info0201

Introduction à la programmation orientée objet

Chapitre 3 : *Notions d'architecture*

1. Les différents niveaux de langage
2. Architecture et processeur
- 3. Un exemple simple de processeur**
 - a. cahier des charges ; schéma ; description
 - b. synthèse (micro-architecture + ISA)
 - c. analyse / écriture de codes

Ch Jaillet (URCA)
Info0204 – Ch. 3
Notions d'archi.

3. Un exemple simple de processeur

c. analyse / écriture de codes

Exemples directs :

- trace d'exécution (plusieurs valeurs), organigramme, analyse :
 - ...
- écriture de code assembleur : traitement sur des exemples, analyse, organigramme, code d'assemblage
 - addition de 3 et 7, et écriture du résultat à l'adresse 13
 - addition de 3 et de la valeur à l'emplacement mémoire 18, et écriture du résultat à l'emplacement mémoire 60
 - multiplication par 4 du contenu d'un emplacement en mémoire
 - échange des contenus des emplacements mémoire 18 et 30

Utilisation des instructions de saut :

- ...

24