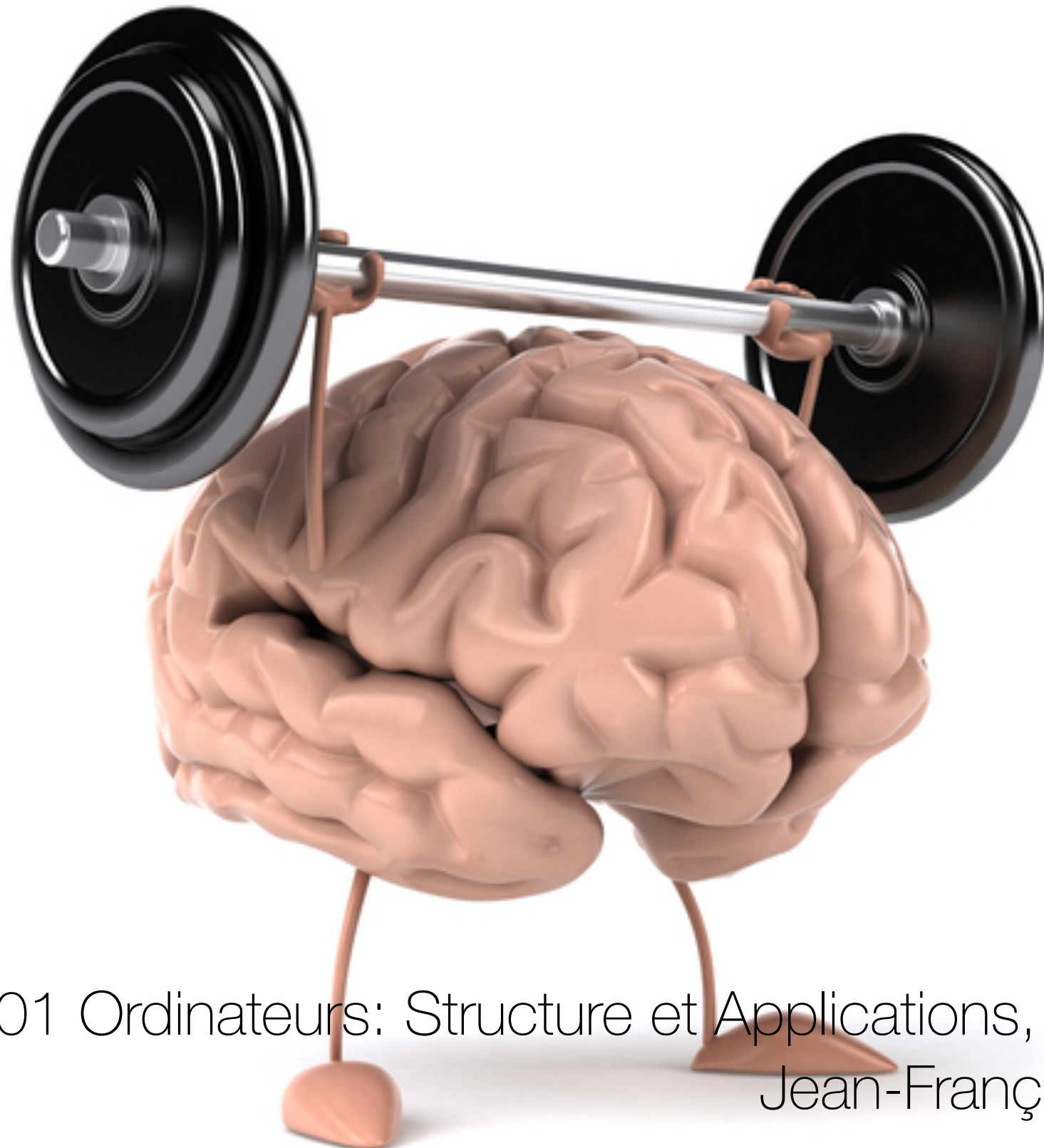


Révision #1

Introduction à la structure interne des ordinateurs



GIF-1001 Ordinateurs: Structure et Applications, Hiver 2016
Jean-François Lalonde

Format des nombres

- 4 points importants:
 - Tout, tout, tout en binaire!
 - Nombre fini et prédéterminé de bits
 - Hexadécimal = binaire
 - Il faut connaître la « recette »!
- Exemples de « recettes »:
 - complément-2 (nombres entiers signés)
 - IEEE 754 (nombres rationnels)
 - ASCII (chaîne de caractères)

Questions

- En complément-2 sur 4 bits:
 - quelles sont les valeurs minimales et maximales pouvant être représentées?
 - -8 à 7
 - $6+3=?$
 - $0110 + 0011 = 1001$, donc -7. Les bits de signe sont différents, donc il y a débordement.
- Exprimez la chaîne de bits suivante en hexadécimal: 01101001
 - 0x69
- Exprimez la chaîne de bits suivante en binaire: 0xA3
 - 10100011

Mémoires & bus

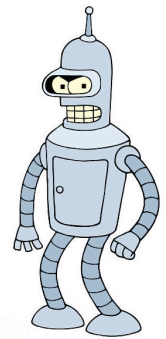
- 2 nombres importants:
 - taille des mots (bus de **données**)
 - nombre de mots (bus **d'adresses**)

mémoire de 2^{16} adresses

Adresse	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
0x0000								
0x0001								
0x0002								
0x0003								
0x0004								
0x0005								
0x0006								
0x0007								
0x0008								
0x0009								
0x0010								
0x0011								
0x0012								
0x0013								
0x0014								
0x0015								
...								
...								
0xFFFF								

taille des mots = 8 bits = 1 octet

Adressage

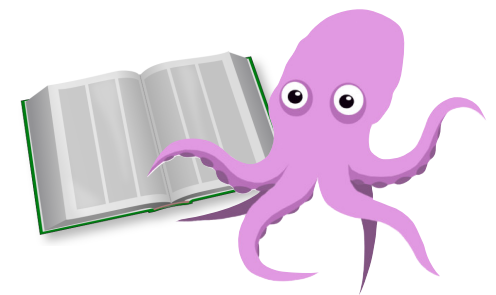
- En « memory-mapped », on adresse tous les périphériques de la même façon
 - on utilise une **adresse** (ex: TP1)
- En « port-mapped », on adresse les périphériques différemment de la mémoire
 - on utilise une **instruction** spécialisée (ex: )

Questions

- Combien de mots en mémoire puis-je adresser si le bus de données et d'adresse ont tous deux 32 bits?
 - 2^{32}
- Un système « memory-mapped » possède un bus d'adresse de 16 bits. Les 4 bits les plus significatifs (MSB) sont utilisés pour le décodeur d'adresse. Combien de périphériques différents puis-je adresser?
 - $2^4 = 16$
- Ce même système possède une mémoire RAM. Si cette mémoire stocke des mots de 8 bits, quelle est la taille maximale de cette mémoire, en kilo-octets?
 - $16 - 4 = 12$ bits d'adresses parviennent à la mémoire
 - $2^{12} \times 8 \text{ bits} = 2^{12} \times 1 \text{ octet} = 2^{12} \text{ octets} = 2^2 \times 2^{10} \text{ octets} = 4 \text{ Ko}$

Intro à l'assembleur

- MOV vs LDR/STR
 - MOV = registres
 - LDR/STR = mémoire
- Instruction = binaire
 - Une autre recette!
 - Opcode, arguments
- Instruction = séquence de micro-instructions
 - Utilise le matériel pour exécuter les instructions
 - μ -instructions déterminées par la mémoire de μ -instructions



Questions

- Dans le programme suivant, quel est l'équivalent binaire des instructions aux adresses 0x0 et 0x4?

Adresse	Instruction
0x0	MOV R3 #0x0
0x1	MOV R0 #0x71
0x2	LDR R1 [R0]
0x3	SUB R1 #0x1
0x4	JZE R1 #0x6
0x5	JZE R3 #0x3
0x6	STR R1 [R0]

- MOV R3 #0x0 = 0x4300
- JZE R1 #0x6 = 0xF106

Jeu d'instructions

Mnémonique	Opcode	Description
MOV Rd Rs	0000	Écriture de la valeur du registre Rs dans le registre Rd
MOV Rd Const	0100	Écriture d'une constante dans le registre Rd
ADD Rd Rs	0001	Addition des valeurs des registres Rd et Rs et insertion du résultat dans le registre Rd
ADD Rd Const	0101	Addition de la valeur du registre Rd avec une constante et insertion du résultat dans Rd
SUB Rd Rs	0010	Soustraction de la valeur Rs à l'intérieur de registre Rd.
SUB Rd Const	0110	Soustraction d'une constante à l'intérieur du registre Rd
LDR Rd [Rs]	1000	Chargement d'une valeur se trouvant à l'adresse Rs de l'ordinateur dans un registre.
STR Rd [Rs]	1001	Écriture de la valeur d'un registre à l'adresse Rs de l'ordinateur.
JZE Rc Const	1111	Saut à l'instruction située à l'adresse identifiée par la constante, mais seulement si Rc = 0 (sinon, cette instruction n'a aucun effet).
JZE Rc [Rs]	1011	Saut à l'instruction située à l'adresse Rs seulement si Rc = 0 (sinon, cette instruction n'a aucun effet).

Format des instructions

Opcode	Argument 1	Argument 2
4 bits	4 bits	8 bits

Questions

- Qu'est-ce que ce programme fait?

Adresse	Instruction
0x0	MOV R3 #0x0
0x1	MOV R0 #0x71
0x2	LDR R1 [R0]
0x3	SUB R1 #0x1
0x4	JZE R1 #0x6
0x5	JZE R3 #0x3
0x6	STR R1 [R0]

Jeu d'instructions

Mnémonique	Opcode	Description
MOV Rd Rs	0000	Écriture de la valeur du registre Rs dans le registre Rd
MOV Rd Const	0100	Écriture d'une constante dans le registre Rd
ADD Rd Rs	0001	Addition des valeurs des registres Rd et Rs et insertion du résultat dans le registre Rd
ADD Rd Const	0101	Addition de la valeur du registre Rd avec une constante et insertion du résultat dans Rd
SUB Rd Rs	0010	Soustraction de la valeur Rs à l'intérieur de registre Rd.
SUB Rd Const	0110	Soustraction d'une constante à l'intérieur du registre Rd
LDR Rd [Rs]	1000	Chargement d'une valeur se trouvant à l'adresse Rs de l'ordinateur dans un registre.
STR Rd [Rs]	1001	Écriture de la valeur d'un registre à l'adresse Rs de l'ordinateur.
JZE Rc Const	1111	Saut à l'instruction située à l'adresse identifiée par la constante, mais seulement si Rc = 0 (sinon, cette instruction n'a aucun effet).
JZE Rc [Rs]	1011	Saut à l'instruction située à l'adresse Rs seulement si Rc = 0 (sinon, cette instruction n'a aucun effet).

- Ce programme:
 - lit la valeur à l'adresse 0x71
 - il la décrémente jusqu'à temps qu'elle atteigne la valeur de 0
 - il stocke le résultat (0) à la même adresse