

Corrigé Série TD n°5

Exercice n°1 :

- a) écriture du meilleur code assembleur équivalent pour le code suivant (les variables A, B, C et D sont initialement en mémoire).

A=B+C

B=B+C

D=B-A

Accumulateur (1,1)	Mémoire-Mémoire (3,3)	Pile (0,0)	Chargement – Rangement (3,0) (registre/registre)
Load @B Add @C Store @A Load @B Add @C Store @B Load @B Sub @A Store @B	Add @A,@B,@C Add @B,@B,@C Sub @D,@B,@A	Push @B Push @C Add Pop @A Push @B Push @C Add Pop @B Push @B Push @A Sub Pop @D	Load R1, @B Load R2, @C Add R3, R1, R2 Store R3, @A Load R4, @B Load R5, @C Add R6, R4, R5 Add R6, R1, R2 Store R6, @B Load R1, @B Load R2, @A Sub R7, R6, R3 Store R7, @D
Nbre Ins: 7	Nbre Ins: 3	Nbre Ins: 10	Nbre Ins: 6
Format Inst: C.Op /@ opérande 1Octet 2Octets	Format Inst: C.Op/@Dest/@Sour1/@Sour2 1Oct 2Octs 2Octs 2Octs	Format Inst: 1 ^{er} Format: C.Op/@ opér 1Oct 2Octs 2 ^{ème} Format: C.Op 1Octet	Format Inst: 1 ^{er} Format: C.Op/N°Reg/@opér 1Octet 4bits 2Octets 2 ^{ème} Format: C.Op/N°Reg/N°Reg/N°Reg 1Oct 4bits 4bits 4bits
Longueur Inst:	Longueur Inst:	Longueur Inst:	Longueur Inst:
3 Octets	7 Octets	1 ^{er} Format:3 Octets 2 ^{ème} Format: 1Oct	1 ^{er} Format: 4 Octets 2 ^{ème} Format:3 Octets

	Accumulateur	Mém-Mém	Pile	Reg/Reg
Nombre d'octets d'instructions lus	7 Inst *3 octets	3 Inst * 7octets	(7inst*3 octs) +(3Inst*1oct)	(3inst*4 octs) +(3Inst*3oct)
	21 Octets	21 Octets	24 Octets	21 Octets
Nombre d'octets de données mémoire transféré.	7 Inst *4 octets	3 Inst *12 octs	7 Inst*4 octs	3 Inst*4 octs
	28 Octets	36 Octets	28 Octets	12 Octets
	49 Octets	57 Octets	52 Octets	33 Octets
Quest c)	Accumulateur	Mém-Mém		Reg/Reg
Quest d)				Reg/Reg

Exercice n°2 :

3 bits	3 bits	3 bits	32 bits
Code – Op	Rd	Rs	Immédiat

Longueur : 41 bits (ou 6 octets)

Pour les 6 instructions qui n'utilisent pas de valeur immédiate :

3 bits	3 bits	3 bits
Code – Op	Rd	Rs

Longueur : 9 bits (ou 2 octets)

Pour les 2 instructions qui utilisent des valeurs immédiates :

3 bits	3 bits	32 bits
Code – Op	Rd/Rs	Immédiat

Longueur : 38 bits (ou 5 octets)

Longueur du programme avec 1 format:

Ni * 41 bits OU Ni * 6 octets

Longueur du programme avec 2 formats :

(6 / 8) Ni * 9 bits + (2 / 8) Ni * 38 bits = 16,25 Ni bits

OU (6 / 8) Ni * 2 octets + (2 / 8) Ni * 5 octets = 2,75 Ni octet

Donc le rapport est :

16,25 Ni / 41 Ni = 0,396 (Calculs en bits) OU 2,75 Ni / 6 Ni = 0,458 (Calculs en octets)

Exercice n°3 :

A=Tex ancien/Tex nouveau avec

Tex = NI*T cycle*CPI.

Hypothèse valable : **T cycle ancien = T cycle nouveau.**

Les programmes seront maintenant plus longs, puisqu'on ajoute une instruction de chargement/rangement dans 40% * 25% = 10% des cas. Le nouveau programme aura 10% plus

D'instructions, donc **NI nouveau = 1,1NI ancien**

Le CPI va changer à cause des chargements rangements.

CPI ancien = 3.2 (0.1*3 + 0.5*3 + 0.4*3.5)

CPI nouveau = 3 (en tenant compte de l'ajout d'instructions qui change les pourcentages de chaque catégorie : (0.1*3 + 0.5*3 + 0.5*3)/1.1)

Réponse : P = 3.2 / 3.3 = 0,969.

La performance diminue de 3,1% ce à quoi on devait s'attendre, puisque lorsqu'on analyse rapidement, on remplace une instruction de 5 cycles par deux instructions de 3 cycles chacune.