

TD n°3

Le micro-code

Exercice 1 : Squelette du micro-code

Réalisez, dans le langage symbolique du LMA proposé en cours, le squelette du micro-code interprétant un LM utilisant le code expansif (= qui n'est pas sur un nombre de bits fixé à l'avance) suivant :

000 à 101	6	instructions à code sur 3 bits
1100 à 1110	3	instructions à code sur 4 bits
11110, 11111	2	instructions à code sur 5 bits

Cela comprend :

- le séquençement par défaut,
- la récupération de l'instruction,
- l'identification (micro-codée) de l'instruction,

mais pas le traitement spécifique de l'instruction.

Exercice 2 : Modes d'adressage

2.1 Donnez les μ -instructions de récupération et stockage dans OP1 d'un opérande pour les modes d'adressage ci-dessous. On supposera que l'adresse a été préalablement lue et stockée dans TMP. On sautera en X (adresse à définir ultérieurement) après récupération de l'opérande.

2.2 Donnez les μ -instructions d'écriture d'un résultat pour les modes d'adressage ci-dessous. On supposera que l'adresse a été préalablement lue et stockée dans TMP et que la valeur à écrire est stockée dans RES. On sautera en Y (adresse à définir ultérieurement) après écriture du résultat.

		Opération à effectuer	
Mode d'adressage	Contenu de TMP	Récupération d'opérande	Ecriture de résultat
Immédiat	opérande	OP1 = TMP	
Registre	0 ou 1	Si TMP = 0 alors OP1 = E sinon OP1 = F	Si TMP = 0 alors E = RES sinon F = RES
Direct	adresse mémoire	OP1 = mem[TMP]	mem[TMP] = RES
Indirect par registre	0 ou 1	Si TMP = 0 alors OP1 = mem[E] sinon OP1 = mem[F]	Si TMP = 0 alors mem[E] = RES sinon mem[F] = RES
Indirect	adresse mémoire	OP1 = mem[mem[TMP]]	mem[mem[TMP]] = RES
Local	décalage d'adresse	OP1 = mem[PLP + TMP]	mem[PLP + TMP] = RES
Indexé / registre de base	adresse + registre*	OP1 = mem[adr(TMP) + reg(TMP)]	mem[adr(TMP) + reg(TMP)] = RES
Local Indirect	décalage d'adresse	OP1 = mem[mem[PLP + TMP]]	mem[mem[PLP + TMP]] = RES

*Le bit de poids faible de TMP donne le registre (E ou F) contenant le décalage. Les autres bits donnent l'adresse de base :

