

## ***TP2 : INSTRUCTIONS DE TRANSFERT, ARITHMÉTIQUES ET LOGIQUES***

**Durée estimée :** 2 heures

### **2.1 Objectifs détaillés**

- Maîtriser tous les modes d'adressage (immédiat, direct, étendu, indexé, inhérent).
- Comprendre l'impact des opérations arithmétiques sur le registre d'état (Condition Code Register - CCR).
- Savoir manipuler des nombres sur 16 bits avec un processeur 8 bits.
- Utiliser les opérations logiques pour le masquage de bits.

### **2.2 Rappel des instructions principales**

- **Transfert :** LDAA, LDAB, STAA, STAB, TAB, TBA, TAP, TPA
- **Arithmétique :** ABA, SBA, CBA, ADDA, ADDB, ADDD, SUBA, SUBB, SUBD, INCA, INCB, DECA, DECB, NEGA, NEGB
- **Logique :** ANDA, ANDB, ORA, ORB, EORA, EORB, COMA, COMB
- **Décalages :** ASLA, ASLB, ASRA, ASRB, LSLA, LSLB, LSRA, LSRB, ROLA, ROLB, RORA, RORB

### **2.3 Manipulations détaillées**

#### ***Partie A : Modes d'adressage***

1. **Adressage immédiat :**
  - Programme : LDAA #\$25 (charger la valeur 25 en A)
2. **Adressage direct (page zéro) :**
  - Stocker \$25 à l'adresse \$0020, puis LDAA \$20 (charger le contenu de \$0020)
3. **Adressage étendu :**
  - LDAA \$2000 (charger le contenu de l'adresse \$2000)
4. **Adressage indexé :**
  - Initialiser X avec \$2000, puis LDAA 0,X (charger le contenu pointé par X)

#### ***Partie B : Opérations arithmétiques 8 bits***

1. Écrire un programme qui additionne deux nombres 8 bits (sans retenue) :
  - ✓ \$0100 : 86 10 LDAA #\$10
  - ✓ \$0102 : 8B 20 ADDA #\$20
  - ✓ \$0104 : 97 30 STAA \$30
  - ✓ \$0106 : 3F SWI
1.
  - Vérifier le résultat (\$30) à l'adresse \$0030.
2. **Gestion de la retenue :**
  - Additionner \$FF et \$01.
  - Observer le flag Carry (C) dans le CCR après l'opération.

### **Partie C : Opérations 16 bits**

1. Additionner deux nombres 16 bits stockés en mémoire :

- Nombre 1 : \$1234 aux adresses \$40 (poids fort) et \$41 (poids faible)
- Nombre 2 : \$5678 aux adresses \$42 et \$43
- Résultat à ranger aux adresses \$44 et \$45 (avec gestion de la retenue finale)
- Algorithme :
  - Charger l'octet faible du nombre 1 dans A
  - Ajouter l'octet faible du nombre 2
  - Stocker résultat faible
  - Charger l'octet fort du nombre 1 dans A
  - Ajouter l'octet fort du nombre 2 avec la retenue de l'addition précédente (ADCA)
  - Stocker résultat fort

### **Partie D : Opérations logiques**

1. **Masquage** : Écrire un programme qui ne conserve que les 4 bits de poids faible d'un nombre stocké en \$50.
  - Solution : LDAA \$50 puis ANDA #\$0F et STAA \$51
2. **Test de bits** : Vérifier si le bit 3 d'un nombre est à 1.
  - Solution : LDAA \$50 puis ANDA #\$08 et tester le flag Zéro.