

# MICROKIT 09

Séduits par la présentation de la maquette Microkit 09 dans le numéro 10 de Led du mois de septembre, vous l'avez réalisée en suivant pas à pas les indications données dans le numéro 11 de Led d'octobre\*. Mais que peut-on faire avec cet ensemble de puces savantes et comment les faire travailler ? C'est le but des articles qui suivront, de vous aider à réaliser leur dressage programmé.

**P**our commencer, intéressons-nous aux puces, chefs de file de la carte centrale, dont le synoptique est donné en figure 1. La présentation de chacune d'entre elles a déjà été faite dans le premier article. Il importe maintenant de les faire fonctionner. Pour cela nous vous proposons un « menu » de programmes d'initiation.

## COMMENT TRAVAILLE L'UNITE CENTRALE ?

Examinons rapidement le synoptique de la carte centrale (fig. 1) et celui du circuit de l'unité centrale microprocesseur 6809 (fig. 2). Nous allons écrire dans la mémoire RAM (circuit 6116) une suite d'instructions codées. L'unité centrale 6809 va aller chercher les codes-instructions les uns après les autres, les décoder et enfin réaliser l'opération qu'ils indiquent. Pour cela « l'unité centrale 6809 » dispose de quatre parties fonctionnelles :

- des registres qui contiennent soit les adresses où l'unité centrale ira chercher les codes des instructions, soit les adresses où l'unité centrale ira chercher les données à traiter ou ira les ranger après traitement.

Le circuit 6809 en comporte six : PC, X, Y, U, S et DPR ;

- des registres où sont placées les données à traiter ou traitées. Le circuit 6809 en comporte deux : A et B, pouvant contenir chacun des nombres de huit chiffres binaires. Ces deux registres, appelés aussi accu-

mulateurs couplés ensemble constituent un registre de données 16 bits, appelé alors D ;

- une « unité arithmétique et logique » (en anglais ALU) qui traite les données en effectuant des opérations élémentaires (voir la liste du tableau 1). A cette ALU est associé un ensemble de cinq bascules-indicateurs appelées « flags » (« drapeaux » en français), C, V, Z, N et H qui, avec d'autres bascules-indicateurs forment le registre d'état

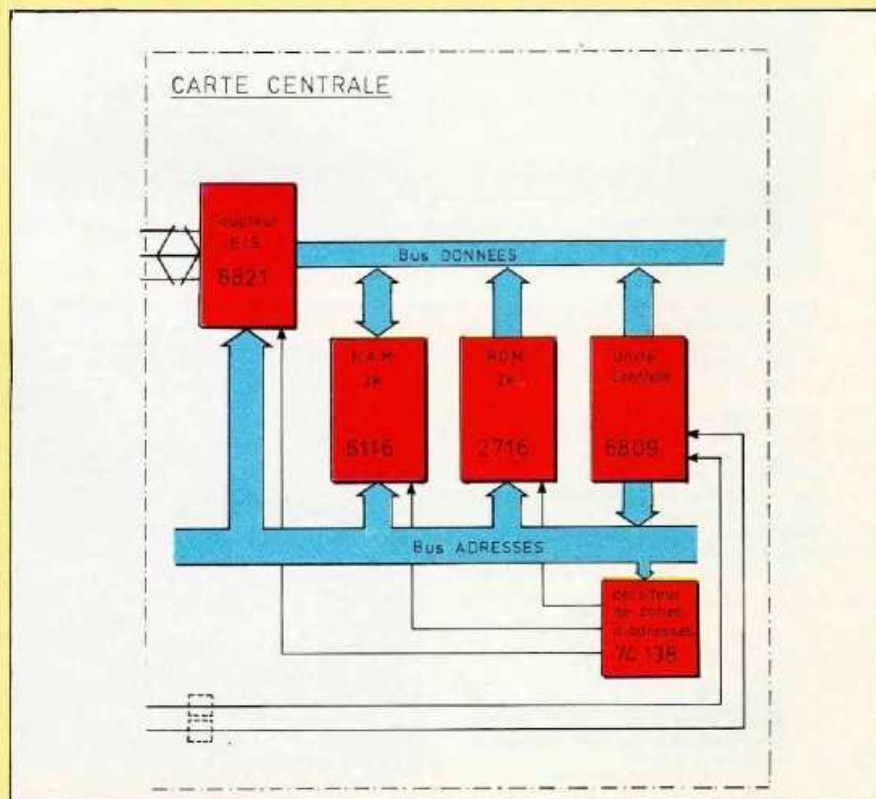
CCR qui nous renseigne sur l'état de l'unité centrale après chaque instruction ;

- enfin, un circuit de décodage des instructions associé à un circuit séquenceur, qui reçoit et génère tous les signaux nécessaires à la commande de l'ensemble de la carte.

## PROGRAMME N° 1 :

**09 + 01 = ?**

... Si vous avez répondu « dix », vous avez perdu, mais vous pouvez conti-



# UN MONTAGE PLEIN DE PUCES

nuer à jouer.

Pour comprendre le fonctionnement de l'unité centrale, nous allons utiliser le Microkit 09 comme une calculatrice.

1. Nous allons introduire directement, à l'aide du clavier de la carte périphérique, deux nombres dans l'unité centrale :

- le premier nombre sera écrit dans le registre B ;
- le second nombre sera écrit dans le registre X.

Pour cela il suffit d'appuyer sur la touche Reg puis sur la touche 2B et de taper le nombre qu'on désire inscrire.

**Attention :** Le nombre que vous écrivez dans B sera interprété par l'unité centrale comme un nombre hexadécimal, c'est-à-dire que chaque chiffre peut varier de 0, 1, ... à ...9, A, B, C, D, E, F (F étant équivalent à 15). Les lecteurs qui ne sont pas familiers avec ce code numérique peuvent se reporter au tableau 3 de conversion en fin de l'article.

Ecrire de même le deuxième nombre dans le registre X en remarquant qu'il comporte jusqu'à quatre chiffres hexadécimaux.

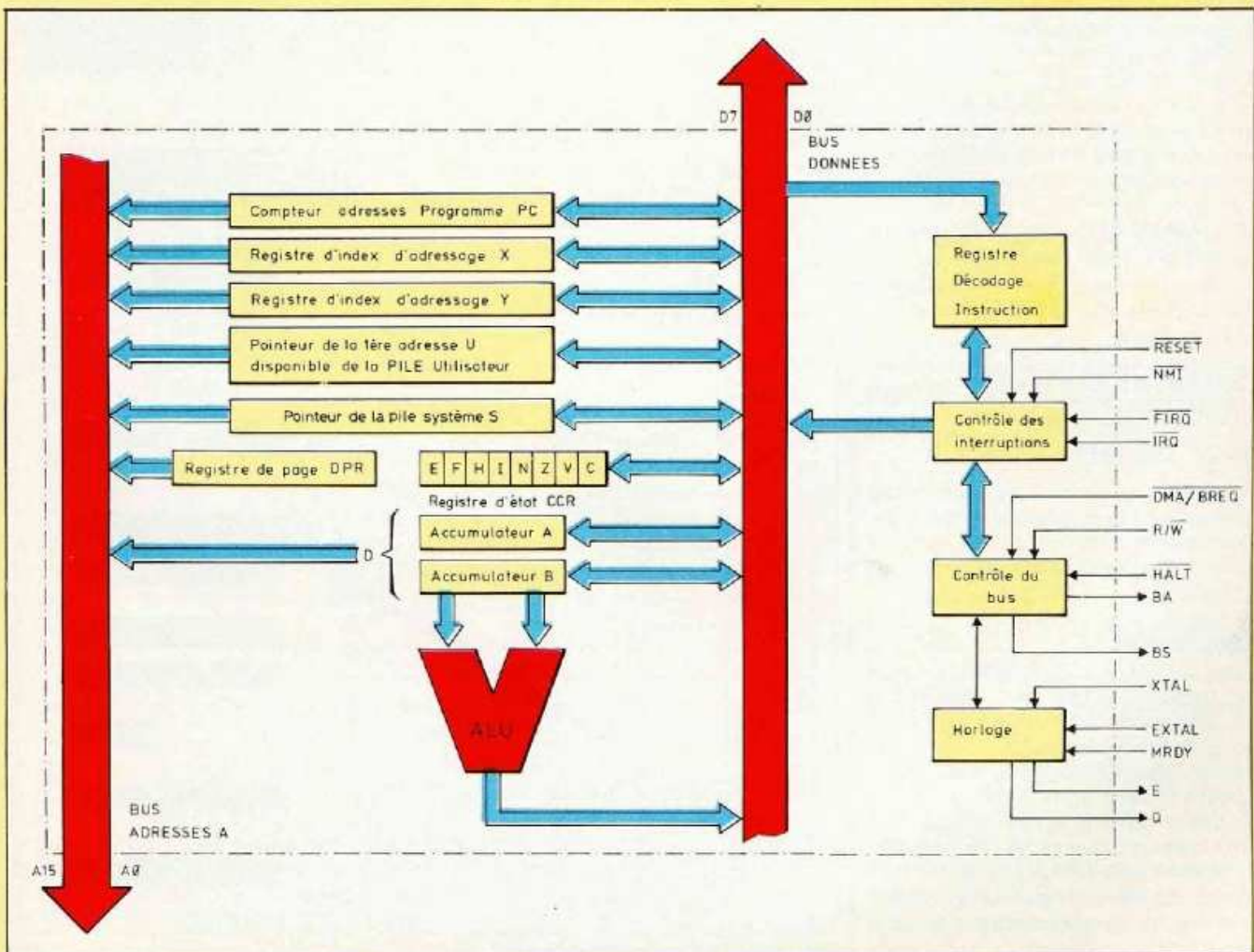
2. Il ne reste plus qu'à rédiger un petit programme en langage machine, demandant à l'unité centrale d'addi-

tionner les deux nombres contenus dans les registres B et X et de placer le résultat dans X.

Le constructeur du circuit 6809 Motorola a, pour cela, établi toute une liste de codes-instructions interprétables par sa machine (voir tableau 1 en fin d'article).

**Le tableau 1 de codes-instruction 6809 est une référence à conserver précieusement pour toute la suite de la série d'articles.**

En effet, nous trouvons en première ligne de ce tableau un code 3A qui, interprété par la partie « décodage instructions » du circuit 6809, provoquera l'addition des nombres conte-



# UN MONTAGE PLEIN DE PUCES

nus dans B et X. Le mode d'adressage est appelé « Inhérent » car la machine sait alors que cette instruction s'adresse aux registres B et X. Le programme 1 se rédige comme suit :

Adresse-Programme	Langage machine code-instruction	Langage Assembleur symbolique opération		Commentaires
0 0 1 0	3 A		ABX	Addition des deux nombres
0 0 1 1	3 F		SWI	Fin du programme - Retour au moniteur

3. Ecrire ce programme en mémoire :  
— utiliser les touches Mem, puis 0010 ;

— y inscrire le premier code 3A ;  
— appuyer sur la touche Inc ; le moniteur pointe alors 0011 ;  
— inscrire en 0011 le code 3F.

4. Lancer le programme :

— appuyer sur RST ;  
— puis sur GO, 0010.

5. Vérifier la bonne exécution du programme en allant examiner le contenu du registre X où doit se trouver le bon résultat, à l'aide des touches Reg et 4X.

Les lecteurs qui ne sont pas encore familiarisés avec l'utilisation du clavier du kit se reporteront à la description détaillée dans le numéro 11 de Led d'octobre.

## COMMENT TRAVAILLE L'UNITÉ CENTRALE AVEC LES MÉMOIRES ?

Pour que l'unité centrale « s'y retrouve », il faut lui indiquer dans les instructions codées :

— soit où aller chercher les données à traiter ;

— soit où aller stocker les résultats du traitement.

Dans ce but le circuit 6809 possède dix modes d'adressage, c'est-à-dire dix façons de coder les adresses, ce qui fait de lui le plus puissant des microprocesseurs 8 bits. Il n'est pas question dans cette série d'articles d'étudier dans le détail chacun des dix modes d'adressage. Nous ne présenterons que certains d'entre eux. Les ouvrages cités en bibliographie à la fin de l'article permettront de compléter cette introduction.

## PROGRAMME N° 2 : ADRESSAGE ÉTENDU

La façon la plus simple de donner une adresse est de l'écrire « en clair », c'est-à-dire à l'aide d'un nombre de quatre chiffres hexadécimaux : c'est le mode d'adressage « étendu ».

Exemple : transférer une donnée de la case-mémoire n° 0210 à la case-mémoire n° 0230. En se rappelant que ce transfert ne peut avoir lieu directement mais que la donnée doit transiter par l'unité centrale, on peut illustrer ce processus de la façon suivante (fig. 3).

L'ordinogramme et le programme correspondants sont simples à écrire :

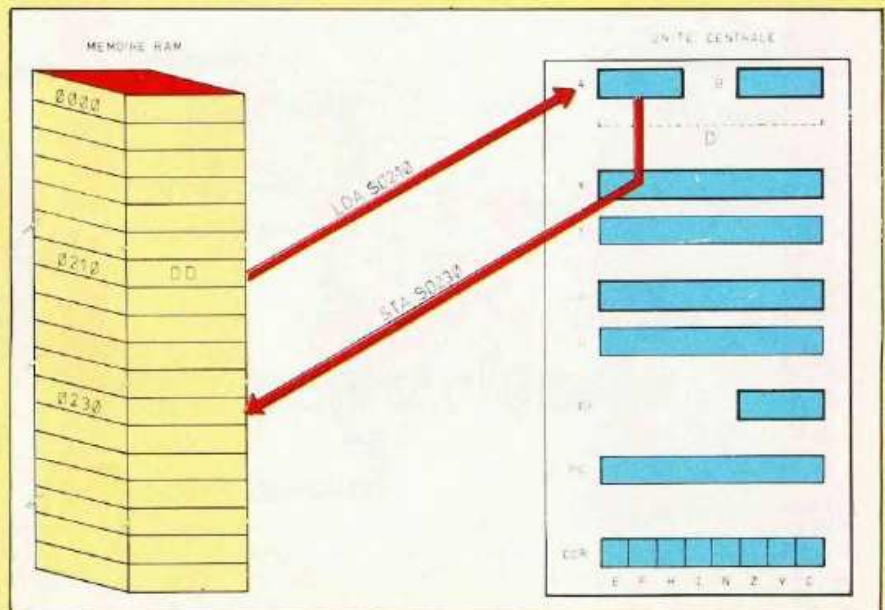
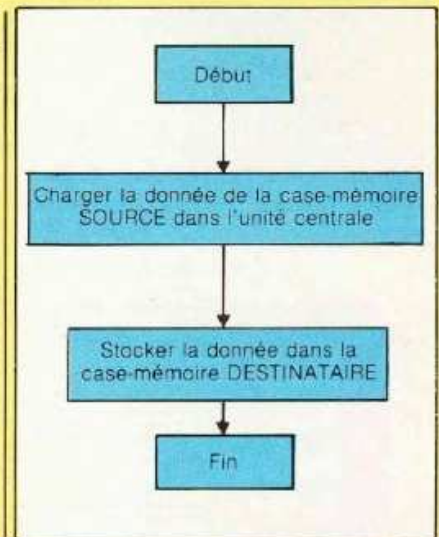


Fig. 3 : Transfert en adressage étendu.

# KIT - 10 P

Adresse-programme	Langage machine code-instruction	Langage Assembleur symbolique opération opérandes		Commentaires
0,0   2,0	B 6   0,2   1,0	LDA	\$0210	Amène la donnée dans l'unité centrale
0,0   2,3	B 7   0,2   3,0	STA	\$0230	Stockage dans nouvel emplacement-mémoire
0,0   2,6	3, F   , , , ,	SWI		Arrêt - Retour au moniteur (visu registres)
				Fin

Il ne vous reste plus qu'à entrer le programme en code-machine, à le lancer et à vérifier le transfert de la donnée en comparant le contenu des cases-mémoires 0210 et 0230 avant et après exécution du programme.

### Remarques

1. Pour trouver le code correspon-

dant aux instructions de chargement et de stockage en mode d'adressage étendu, il suffit de consulter le tableau 1.

2. Avec le Microkit 09, seul le langage machine (code hexadécimal) est utilisable. Le langage Assembleur symbolique sera disponible sur le

micro-ordinateur individuel Micro-comp 09 que nous lancerons mi-1984 (voir réponse au courrier lecteurs en fin d'article).

3. En langage Assembleur, une valeur écrite en hexadécimal est précédée du symbole \$ (dollar).

## PROGRAMMES 3, 4 ET 5 : ADRESSAGE DIRECT

Un autre mode d'adressage consiste à n'écrire dans le programme qu'une partie du numéro de l'adresse (les deux chiffres hexadécimaux de puissance inférieure, à droite), l'autre partie (les deux chiffres hexadécimaux, de puissance supérieure, à gauche) étant contenus dans un

registre dit de « page directe » (DPR « Direct Page Register »). Il suffira alors à la machine de rassembler les deux parties pour reconstituer l'adresse complète. Dans ce mode d'adressage les 64 K, soit 65 536 numéros de cases-mémoires sont ainsi repérés par 256 pages de 256 numéros chacune.

### Programme 3 :

Additionner deux nombres hexadé-

maux entrés au clavier, à l'aide du programme-moniteur dans les cases-mémoires n°s 0240 et 0241. Stocker le résultat en 0242.

Pour cela il nous faut charger les deux chiffres (le nombre 02 de la partie supérieure de l'adresse) 0 puis 2 dans le « registre de page » en utilisant le petit programme suivant. Consulter le tableau 1 pour retrouver les codes utilisés.

Adresse programme	Langage machine code-instruction	Langage Assembleur symbolique opération opérandes		Commentaires
0,0   3,0	C 6   0,2   , ,	LDB	#\$02	Charger "immédiatement" la valeur 02 dans le registre B
0,0   3,2	1, F   9, B   , ,	TFR	B,DP	La transférer pour initialiser le registre de page

Il faut noter qu'il n'existe pas d'instruction de chargement « immédiat » dans le registre DP; dans la suite du programme, il suffira de charger le

nombre correspondant à la partie inférieure de l'adresse dans l'unité centrale et d'utiliser une instruction d'addition en mode d'adressage

direct, la machine se chargeant de reconstituer l'adresse. Les codes utilisés se retrouvent dans le tableau 1 dans la colonne « adressage direct ».

0,0   3,4	9,6   4,0	LDA	<\$40	Chargement du 1 <sup>er</sup> nombre dans unité centrale
, ,   3,6	9, B   4,1	ADDA	<\$41	Addition avec le 2 <sup>e</sup> nombre
, ,   3,8	9,7   4,2	STA	<\$42	Stockage du résultat
, ,   3,A	3, F   , ,	SWI		

Noter que l'adressage direct est symbolisé par < en langage Assembleur. La figure 4 représente le séquençage des opérations effectuées. Chaque fois, l'adresse complète « source » ou « destination » s'obtient en rassemblant les chiffres du registre DP et les chiffres conte-

nus dans le programme. L'intérêt de ce mode d'adressage est de n'utiliser que deux chiffres hexadécimaux d'adresse (soit 1 octet de huit chiffres binaires, chaque « quartet » de quatre chiffres binaires codant un chiffre hexadécimal), ce qui économise de la place pour des

opérations concernant une zone-mémoire dont les numéros d'adresse sont situés dans la même « page ».

### Programme 4

Additionner deux nombres décimaux, entrés par clavier, à l'aide du programme-moniteur. Ce programme implique :