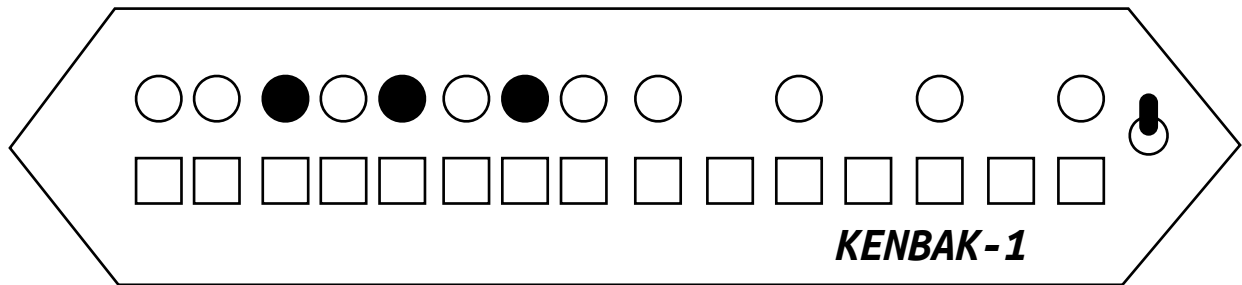


Réplique du *Kenbak-1*



Manuel de programmation simplifié

*Gilles Devillers
Gabriel Bouchard
Sophie Enguehard
Quentin Tessier*

Préface

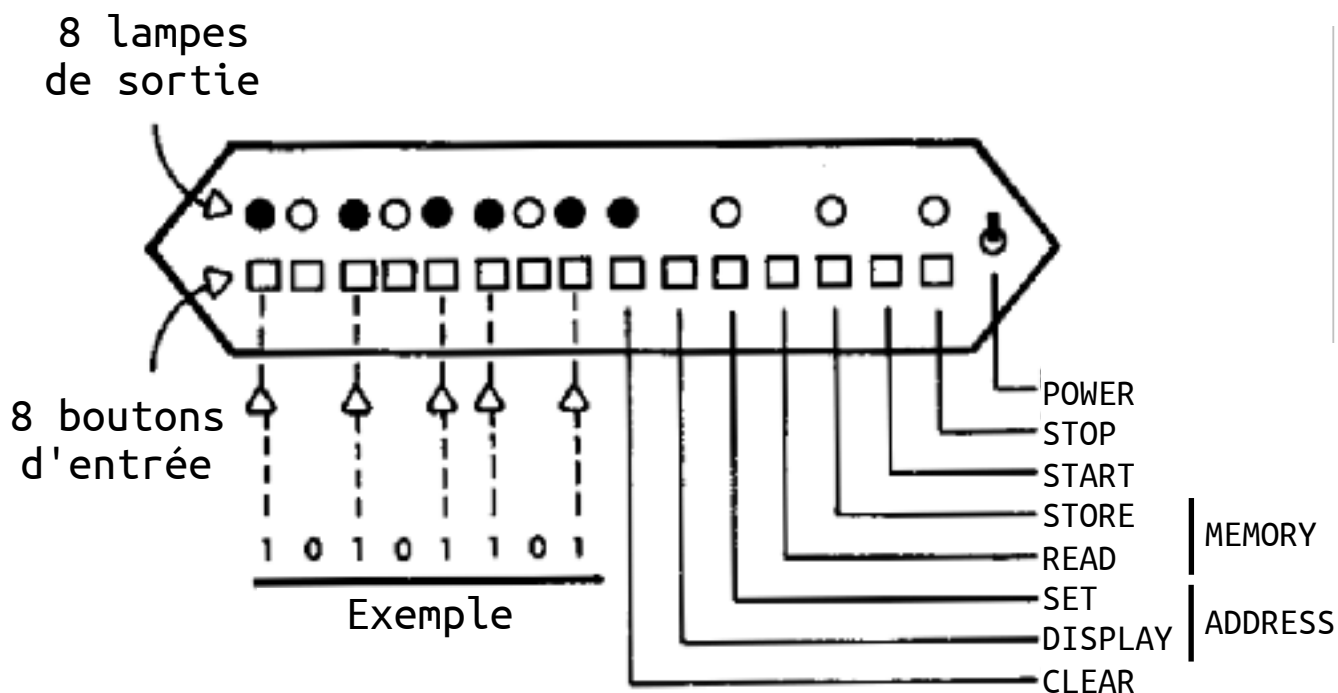
Ce manuel est un résumé de l'utilisation du *Kenbak-1*, traduit en partie du manuel d'origine : *Programming Reference Manual* écrit par *John Blankenbaker*, concepteur du Kenbak-1. Le lecteur doit avoir une connaissance basique des ordinateurs numériques.

Introduction

Le Kenbak-1 est un ordinateur à usage général avec une mémoire série de 256 octets. La machine utilise des nombres avec complément à deux en interne. Les instructions Halt, No-op et Shift ne nécessitent qu'un octet. Toutes les autres instructions nécessitent deux octets et peuvent commencer sur des adresses paires ou impaires.

Sauf indiqué, les nombres utilisés dans la suite de ce manuel seront représentés sous forme octale.

Utilisation de l'interface



- **POWER** : Allumage du *Kenbak-1*.
- **STOP** : Arrête l'exécution du programme.
- **START** : Lance l'exécution du programme.
- **STORE MEMORY** : Stocke le nombre entré à l'adresse pointée par le registre d'adresse.
- **READ MEMORY** : Lis le nombre entré à l'adresse pointée par le registre d'adresse.
- **SET ADDRESS** : Stocke l'entrée dans le registre d'adresse.
- **DISPLAY ADDRESS** : Affiche la valeur du registre d'adresse.
- **CLEAR** : Efface l'entrée.

Pour entrer une valeur :

1. Entrer l'adresse de destination avec les boutons d'entrée.
2. Appuyer sur SET ADDRESS pour se positionner à l'adresse entrée (stocker l'adresse entrée dans le registre d'adresse).
3. Appuyer sur CLEAR pour effacer l'entrée.
4. Entrer la donnée (nombre) à stocker en mémoire.
5. Appuyer sur STORE MEMORY pour stocker la donnée à l'adresse mémoire sélectionnée (adresse stockée dans le registre d'adresse).

Structure de la mémoire et adressage

Chacun des 256 octets de la mémoire est adressable. La plage octale des adresses va de 000 à 377 (0 à 255 en décimal). Le Kenbak-1 étant un ordinateur 8 bits, un octet peut contenir une adresse complète.

Neuf emplacements mémoires sont « spéciaux », on les appelle « registres » :

- 000 : Registre A, registre arithmétique principal. Utilisé pour les opérations de chargement, stockage, addition, soustraction, et, ou, négation, décalages et rotations.
- 001 : Registre B, registre arithmétique secondaire. Utilisé pour les opérations de chargement, stockage, addition, soustraction, décalages et rotations.
- 002 : Registre X. Son contenu est utilisé dans le mode d'adressage indexé. Peut être utilisé comme registre arithmétique avec les opérations de chargement, stockage, addition et soustraction.
- 003 : Registre P. Contient l'adresse de la prochaine instruction à exécuter.
- 200 : Registre de sortie. Contrôle les lampes de sortie lorsque l'ordinateur est en mode exécution.
- 201, 202, 203 : Registres de dépassement et de retenue pour les registres A, B et X respectivement. Le bit 0 correspond à un dépassement mémoire et le bit 1 correspond à une retenue.
- 300 : Registre d'entrée. Le nombre entré grâce aux boutons d'entrée est stocké ici en mode exécution comme en mode arrêt.

Représentation des nombres

Plusieurs choix sont possibles pour représenter les nombres : entier, entier signé et décimal signé.

Voici les nombres pouvant être représenté sur un octet :

Représentation	Valeur décimale	Code octal	Code binaire
Entier positif			
Plus grand nombre	+255	377	11 000 000
Plus petit nombre	0	000	00 000 000
Entier signé			
Plus grand nombre	+127	177	01 111 111
Plus petit nombre	-128	200	10 000 000
Décimal signé			
Plus grand nombre	+127/128	177	0,1 111 111
Plus petit nombre	-128/128	200	1,0 000 000

Les conditions de sauts positif, négatif et positif différent de zéro supposent qu'une des représentations signées est utilisée.

Le bit de dépassement est également dépendant des représentations signées.

Les quantités représentées peuvent être étendues en utilisant plusieurs octets.

Modes d'adressage

Les instructions d'addition (ADD), chargement (LOAD), stockage (STORE), négation (LNEG), et (AND), ou (OR) ont cinq modes d'adressage :

1. Constante (Constant) ou immédiat (Immediate) : l'opérande est le contenu du deuxième octet de l'instruction.
2. Mémoire (Memory) : le deuxième octet de l'instruction est l'adresse de l'opérande.
3. Indirect (Indirect) : le deuxième octet de l'instruction est l'adresse de l'adresse de l'opérande.
4. Indexé (Indexed) : le deuxième octet de l'instruction est ajouté au registre X pour former l'adresse de l'opérande.
5. Indirect indexé (Indirect Indexed) : le deuxième octet de l'instruction est utilisé comme pointeur d'adresse vers un autre emplacement mémoire. Le contenu de ce second emplacement est ajouté au registre X pour former l'adresse de l'opérande.

Les instructions de manipulation de bits n'utilisent que le mode d'adressage « Mémoire ».

Les instructions de saut peuvent utiliser les modes d'adressage « Mémoire » (direct) ou « Indirect ».

Instructions

Voici un récapitulatif du jeu d'instructions du Kenbak-1. Pour plus d'informations sur des instructions en particulier, consulter le livret officiel « *Programming Reference Manual* »

Instruction	Chiffres octal du premier octet		
	D--	-D-	--D
Add, Sub, Load, Store	Reg A = 0 Reg B = 1 Reg X = 2	Add = 0 Sub = 1 Load = 2 Store = 3	Constant = 3 Memory = 4 Indirect = 5 Indexed = 6 Ind/Ind = 7
Or, And, Lneg	3	Or = 0 Noop = 1 And = 2 Lneg = 3	Constant = 3 Memory = 4 Indirect = 5 Indexed = 6 Ind/Ind = 7
Sauts	Reg A = 0 Reg B = 1 Reg X = 2 Incondit. = 3	JPD = 4 JPI = 5 JMD = 6 JMI = 7	(≠0) = 3 (=0) = 4 (<0) = 5 (≥0) = 6 (>0) = 7
Test et manipulation de bits	Forcer 0 = 0 Forcer 1 = 1 Sauter 0 = 2 Sauter 1 = 3	Valeur = poids du bit	2
Décalages, rotations (un octet)	Déc. droi. = 0 Rot. droi. = 1 Déc. gauc. = 2 Rot. gauc. = 3	A = 0 B = 4 -----plus----- 1 fois = 1 2 fois = 2 3 fois = 3 4 fois = 0	1
Divers (un octet)	Halt = 0 ou 1 Noop = 2 ou 3	Peu importe	0