

Old Tech : Z-80

D.Moreaux

6 novembre 2018

1 Introduction

Le processeur Z-80 a été créé par Zilog avec un jeu d'instruction compatible avec le jeu d'instructions du processeur 8080 d'iNTEL (mais de nombreuses instructions ajoutées).

Il a équipé un bon nombre d'ordinateurs au début des années 80, dont les Amstrad (464, 664 et 6128), Sinclair (ZX80, ZX81, ZX Spectrum, Spectrum 128K, Sinclair +2, Sinclair +3), MSX et de nombreux autres.

Il proposait un jeu d'instructions très riche, de nombreux registres et permettait un interfaçage très aisé au niveau électronique (horloge simple phase, pas de multiplexage, rafraichissement RAM Dynamique intégré,...)

Sa popularité lui a également valu d'être le processeur autour duquel tournait CP/M, un OS cross-platform qui permettait d'utiliser les mêmes programmes sur de nombreuses machines. Cette disponibilité de CP/M a poussé Apple à proposer une carte "CP/M" incluant le processeur Z-80 pour ses Apple-II et Commodore a équipé les Commodore 128 d'un Z-80 en second processeur (dans les deux cas, à côté d'un 6500)

On le trouve encore de nos jours, entre autres dans des versions micro-contrôleur avec des extensions 32 bits.

2 Le langage du Z-80

Le Z-80 propose un jeu d'instruction très étendu comportant plusieurs centaines d'instructions. Ces instructions sont réparties en 3 *pages*, deux de ces pages étant accédées à l'aide d'instructions préfixes (CB et ED).

Les instructions peuvent prendre de 1 à 4 octets selon le cas, avec des temps d'exécution qui vont de 4 cycles d'horloge à plus de 20 cycles d'horloge.

Il propose également de très nombreux registres (A, l'accumulateur, B, C, D, E, H et L) certains pouvant être regroupés par paires pour former des

registres 16 bits (BC, DE et HL, ce dernier tenant le rôle d'accumulateur 16 bits) qui permettaient d'adresser la totalité de la mémoire.

Des registres d'index IX et IY sont accessibles à l'aide de préfixes et remplaceront HL. Si ils sont utilisés pour accéder à la mémoire, un *offset* de -128 à +127 sera fourni qui sera ajouté au registre dans le calcul de l'adresse.

Un registre SP servira de pointeur de pile, utilisé pour les appels de sous-routines (CALL, RET) ou pour un stockage rapide des registres (PUSH, POP).

Le registre F est composé des drapeaux de l'ALU (Zero, Carry, Sign, Half-Carry, Parity/Overflow) et est souvent couplé au registre A.

Les registres de travail (A, B, C, D, E, H, L) ont également des versions *alternatives* (A', B', C', ...) et les instructions EXX et EX AF,AF' permettent d'échanger ces variables (A et A' sont gérés séparément).

Les instructions disponibles sont également très variées : ADD, SUB (et leurs versions avec Carry ajouté/soustrait : ADC, SBC), CP (compare, un SUB dont le résultat n'est pas sauvé), AND, OR, XOR, des décalages avec ou sans le carry (RL, RR, RLC, RRC, ...) des opérations sur un bit (BIT pour tester, SET et RES pour mettre à 1 ou à 0), INC et DEC (+1 ou -1), des sauts sur différentes conditions, ...

Le Z-80 dispose également d'instructions de copie par bloc (LDIR qui copie de HL vers DE un total de BC octets par exemple) qui peuvent prendre des temps d'exécution assez longs (en fait, tant que BC est non nul, PC n'est pas mis à jour pour pointer sur l'instruction suivante et l'instruction est répétée en boucle).

3 exemple de programme

```
2000 3E 01 LD A, 0x01
2002 D3 40 OUT (0x40),A
2004 06 20 LD B, 0x20
2006 0D      DEC C
2007 20 FD JR NZ, 0x2006
2009 05      DEC B
200A 20 FA JR NZ, 0x2006
200C 07      RLCA
200D 18 F3 JR 0x2002
```

2000 on charge 1 dans le registre A (qui contient les bits à allumer)
2002 on envoie le registre A sur le port 40 de sortie correspondant aux
LED

2004 on charge 0x20 (32) dans le registre B
2006 on diminue C, s'il vaut 0, Z sera actif
2007 si Z est inactifs, on saute en 2006
2009 on diminue B
200A si B n'est pas arrivé à 0, on saute en 2006
200C on fait un décalage du contenu de A vers la gauche (0F RRCA fait
le décalage vers la droite)
200D on retourne en 2002

Pour charger ce code, on pousse sur [ADDR] pour passer en entrée d'adresse,
on tape 2000, on pousse sur [DATA] pour passer en saisie de donnée, on en-
code un octet à la fois en poussant sur [+] pour passer au suivant.

Quand on a terminé, on revient à l'adresse 2000 et on pousse sur [GO]
pour lancer le programme.