Old Tech : Cosmac / RCA1802

D.Moreaux

14 juillet 2018

1 Introduction

Le processeur RCA1802 est un processeur 8 bits apparu en 1976. L'architecture avait été créée en 1970 par Joseph Weisbecker sous la forme de deux composants et d'un système nommé FRED (Flexible Recreational Educational Device).

Malheureusement, suite à différents problèmes internes à RCA, il ne fut disponible qu'en 1975 ce qui lui fera rater le train des ordinateurs pour particuliers.

Une version spéciale résistante aux radiations a été réalisée, ce qui a vallu au RCA1802 de se retrouver dans plusieurs projets spaciaux tels que le télescope Hubble, les sondes Galileo et Magellan et des satellites mis en orbite.

La revue Popular Electronics a proposé le schéma du Cosmac Elf, un système minimal basé sur le RCA1802.

On trouve aussi le RCA1802 dans des flippers et bornes d'arcade, des instruments scientifiques et des voitures Chrysler.

2 Le langage du RCA1802

Le Cosmac possède 16 registres 16 bits (R0 à R16) et un accumulateur 8 bits D.

Deux registres P et X servent à sélectionner les registres 16 bits servant de pointeur d'exécution (P) et d'index (X). R0 sera utilisé pour les accès DMA et R1 comme pointeur d'exécution pour les interruptions.

La variable Q sert également de sortie 1 bit (elle est reliée à une broche du processeur) et 4 broches sont liées aux flags EF1 à EF4 qui peuvent être testés directement.

Les instructions permettent généralement de faire des opérations entre D et une valeur en mémoire de faire des transferts entre les parties hautes et basses des registres et D, ...

11 des instructions prennent le quartet de poids faible comme paramètre.

0n	LDN Rn	Lit Memoire(Rn) dans D
1n	INC Rn	Rn=Rn+1
2n	DEC Rn	Rn=Rn-1
4n	LDA Rn	Lit Mémoire(Rn) dans D, Rn=Rn+1
5n	STR Rn	Ecrit D dans Mémoire(Rn)
8n	GLO Rn	D=Rn.0
9n	GHI Rn	D=Rn.1
An	PLO Rn	Rn.0=D
Bn	PHI Rn	Rn.1=D
Dn	SEP n	P=n
En	SEX n	X=n

Les instructions 3x concernent les sauts courts. Les conditions sont inversées pour les valeurs 38 à 3F. L'octet qui suit indique le LSB de la nouvelle adresse (saut inconditionnel, si Q=1, si D=0, si DF=1 ou si EF1..4=1)

Les instructions 6x concernent les sorties (61-67) et entrées (69-6F).

Les instructions Cx concernent les sauts et skips longs.

Les instructions 7x et Fx ainsi que quelques valeurs isolées (00, 60, 68) propose les opérations logiques, arithmétiques et quelques autres instructions spécialisées (Transferts vers Mémoire(RX), chargement d'une valeur immédiate, Q=1 ou Q=0, sauvegarde/récupération de X et P,...)

Aucun mécanisme spécifique n'est prévu pour les sous-programmes. On s'en sort généralement en utilisant R2 comme pointeur de pile, R3 comme pointeur d'instruction normal, R4 et R5 comme pointeurs d'exécutions pour des fonctions CALL et RETURN. En changeant P en 4 ou 5, on appelle les fonctions CALL et RETURN qui effectueront l'appel de sous-programme et le retour.

3 Programme d'exemple

0000	F8	01	LDI 01	Initialiser la variable pointée par R2 à 1
0002	B2		PHI R2	
0003	A2		PLO R2	
0004	E2		SEX 2	
0005	52		STR R2	Début boucle
0006	FF	80	SMI 80	Compare D avec 0x80. Si égaux, Q=1
0008	3A	0B	BNZ 0B	
000A	7B		SEQ	
000B	02		LDN R2	
000C	FF	01	SMI 01	Compare D avec 0x01. Si égaux, Q=0
000E	3A	11	BNZ 11	
0010	7A		REQ	
0011	02		LDN R2	Récupérer la variable
0012	31	16	BQ 16	Selon Q, Shift Left ou Shift Right
0014	FE		SHL	
0015	38		NBR	Skip next
0016	F6		SHR	
0017	52		STR R2	resauver la variable
0018	64		OUT4	envoyer sur les LED
0019	22		DEC R2	
001A	F8	09	LDI 09	Boucle d'attente en utilisant R4
001C	B4		PHI R4	
001D	24		DEC R4	
001E	94		GHI R4	
001F	3A	1D	BNZ 1D	
0021	02		LDN R2	
0022	30	05	BR 05	On boucle

Pour encoder le programme, suivre la procédure ci-dessous :
— WAIT haut, CLEAR bas
— WRITE haut
— WAIT bas (LOAD)
— mettre le premier octet sur les interrupteurs
— enfoncer IN
— répéter les opérations précédentes pour entrer tout le programme
— WAIT haut, CLEAR bas
— CLEAR haut (RUN)