

BFComputer Specifications

BFComputer – abrégé BFCom, ou BFC – est un ordinateur 12-bits fonctionnant avec le langage hétéroclite Brainf*ck en guise de langage d’assembleur. Ainsi, il ne comprend que huit instructions basiques qui seront détaillé dans la prochaine partie. BUT PROJET ETC... AUTRE TRUCS A DIRE ?

Le Brainf*ck et les instructions

Le Brainf*ck est donc composé de huit instructions élémentaires permettant de manipuler une mémoire, le déroulement du programme ainsi qu’interagir avec des entrées/sorties. EXPLICATION JEU INSTRUCTIONS BF

+	Incrémente la valeur de la cellule courante
-	Décrémente la valeur de la cellule courante
>	Incrémente le curseur de la mémoire
<	Décrémente le curseur de la mémoire
[Début d’une boucle
]	Fin d’une boucle
.	Envoie la valeur de la cellule courante à la sortie
,	Écris la valeur de l’entrée dans la cellule courante

L’architecture de l’ordinateur

L’ordinateur doit donc avoir une architecture spécifique pour interpréter ce jeu d’instruction réduit. L’ALU – Arithmetic and Logic Unit – est assez simple car il se résume à incrémenter ou décrémenter la valeur de registres. Cependant, ce qui complexifie la tâche est le système de boucle. Dans un langage d’assembleur normal, il n’existe pas de boucle à proprement parlé mais des ‘sauts’ avec ainsi l’adresse d’arrivée indiquée dans l’instruction. Ici le challenge a été de trouver un mécanisme permettant à l’ordinateur d’aller chercher le début ou la fin de la boucle correspondante. De plus, pour optimiser un peu le tout – bien que cela ne soit pas forcément utile étant donné que dans tous les cas l’ordinateur sera lent, son horloge fonctionnant à X MHz max – chaque instruction de boucle va être stocké dans une mémoire cache parallèle à celle utilisable pour l’utilisateur.

L’architecture peut être décomposée en deux parties : la logique de contrôle, qui permet de décoder l’instruction courante et de l’exécuter ; ainsi que logique de calcul, qui contient les mémoires, la RAM, l’ALU et les entrées/sorties.

La logique de calcul

Les registres

L’ordinateur est constitué de quatre registres :

Nom	Sigle	Utilité
Program Counter	PC	Registre contenant l'adresse de l'instruction courante
Memory Address Register	MAR	Registre contenant le curseur/pointeur de la mémoire
Loop Address Register	LAR	Registre contenant
Loop Counter	LPC	Permet de compter le nombre de boucle rencontré lorsque l'on cherche l'instruction opposée

Le LAR et le MAR sont reliés via un multiplexeur au bus d'adresse de la mémoire RAM lui même relié au bus de données.

L'ALU

L'ALU est en réalité rien d'autre que un registre relié au bus de données et permet d'incrémenter ou décrémenter sa valeur interne.

Les entrées/sorties

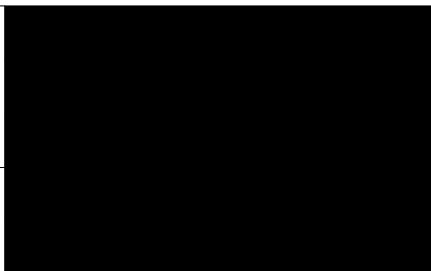
TODO

Le bus de données

Le bus de données est relié à la RAM, l'ALU, le PC ainsi que les entrées/sorties. Il est par défaut mis à zéro et est ainsi utilisé pour initialiser les registres et la RAM à cette valeur.

La logique de contrôle

La logique de contrôle est en soit assez simple : elle réunit tous les éléments nécessaire à la détermination des opérations que la logique de calcul doit effectuer. Nous avons donc pour cela une ROM principale avec 8192 cellules de 25 bits. 8192 cellules correspondent à 13 bits et plus précisément quatre éléments résumé dans le tableau ci dessous :

Element		Bit Index	
Program ROM		#0	
		#1	
		#2	
Phase		#3	
		#4	
Flag Register		LFF : Loop Found Flag	#5
		SLF : Start Loop Flag	#6
		ELF : End Loop Flag	#7
		RF : Reset Flag	#8

		#9
Zero Checkers	LFF : Loop Found Flag / ID : 0	#10
	BZF : Bus Zero Flag / ID : 1	#11

Voici une brève explication de chaque élément :

Program ROM

La ROM qui contient le programme à exécuter. Son contenu peut être changé via le programmeur de ROM dont une partie est aussi dédiée. Les instructions défilent via le PC.

Phase

Registre de 1-bit qui bascule entre 0 et 1 pour les instructions nécessitant plusieurs cycles d'horloge.

Flag Register

Registre de flag indiquant certains états interne de l'ordinateur.

Voici une description de chaque flag : TODO

Zero Checkers

Voir logique de calcul

Les séquences d'instruction

Fetch Cycle

Le « Fetch Cycle » est l'opération qui consiste à récupérer l'instruction suivante dans le but de la décoder puis l'exécuter. Elle est donc réalisée avant chaque instruction.

TODO

La réinitialisation de l'ordinateur

La réinitialisation de l'ordinateur (qui est aussi son initialisation) consiste à mettre tous les registres ainsi que le contenu de la RAM à zéro. Cela consiste à trois étapes distinctes :

1/ Mettre la valeur du bus de donnée (alors à zéro) dans le PC & mettre à zéro la valeur du MAR via ses entrées branchés sur le 0.

2/ Mettre la valeur du PC dans le LAR & celle du bus dans la RAM

3/ Incrémenter le MAR jusqu'à la valeur max (4095) où le RF est réinitialisé & celle du bus dans la RAM

La gestion des boucles

sdfdsqgfdsf