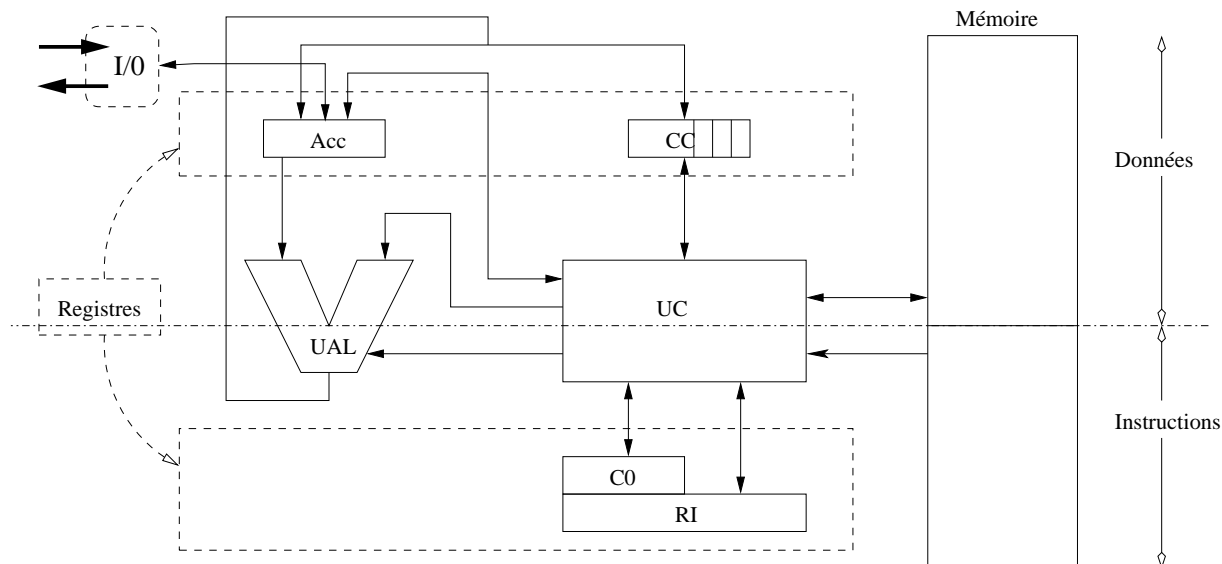


(CM 3) Un processeur simplifié : micro-architecture et langage d'assemblage

1 Schéma de fonctionnement



2 Jeu d'instructions : contrôle / calcul

Num	Code	Commande	arg.	description
0	0000 / 0	LOAD	M, V	charge dans Acc la valeur spécifiée (par l'argument)
1	0001 / 1	STORE	M	recopie la valeur contenue dans Acc dans l'emplacement mémoire spécifié
2	0010 / 2	In / Out	-	Entrée / Sortie, selon le type : 1 pour <i>In</i> ; 0 pour <i>Out</i>
3	0011 / 3	CMP	M, V	compare Acc et la valeur spécifiée : modifie CC, selon $val - Acc$; ne modifie pas Acc
4	0100 / 4	JZ	V	saut conditionnel vers l'instruction dont le numéro est spécifié en argument : saut <i>ssi</i> le <i>flag</i> ZF est à 1
5	0101 / 5	JC	V	saut conditionnel (si CF est à 1)
6	0110 / 6	JMP	V	saut direct (non conditionnel)
7	0111 / 7	END	-	fin du code assembleur : sortie

Les instructions de contrôle ne modifient pas CC, sauf CMP.

Num	Code	Commande	arg.	description
8	1000 / 8	ADD	M, V	ajoute la valeur spécifiée (par l'argument) à Acc
9	1001 / 9	SUB	M, V	retranche la valeur spécifiée à Acc
10	1010 / a	MUL	M, V	multiplie Acc par la valeur spécifiée
11	1011 / b	MOD	M, V	calcule dans Acc le reste de la division euclidienne de Acc par la valeur spécifiée
12	1100 / c	DIV	M, V	calcule dans Acc le quotient de la division euclidienne de Acc par la valeur spécifiée
13	1101 / d	AND	M, V	ET logique, bit à bit
14	1110 / e	OR	M, V	OU logique, bit à bit
15	1111 / f	NOT	-	négation logique, bit à bit