

TP3 ALM : Réalisation d'une machine algorithmique

5.3 Modification de l'algorithme

Algorithme d'un modulo à partir de soustractions :

```
modulo(a,b) {  
  |   tant que a>=b  
  |   |   a = a-b;  
  |   return a ;  
}
```

Algorithme modifié:

```
code <- 1  
|   tant que exp>0  
|   |   si exp && 1==1 alors  
|   |   |   code <- code*m ;  
|   |   |   tant que code >= n  
|   |   |   code <- code - n;  
|   |   m <- m*m;  
|   |   tant que m>=n  
|   |   |   m <- m-n;  
|   |   exp <- exp decal(1);
```

Pour initialiser la variable CODE à 1, il suffit d'ajouter un registre Set a Code .

Si Set = true alors la première bascule du registre Code sera initialisée à 1 et les autres seront affectés à 0. (on fait cela avec les variables set et reset des autres bascules).

Pour évaluer la condition "si exp > 0" dans le Lustre, on observe le registre Z de exp. Cette condition est vérifiée si et seulement si Z(exp) est différent de 0.

Pour évaluer la condition "si exp && 1==1", il faut regarder le bit de poids le plus faible de la variable exp et puis comparer ce dernier à la valeur de 1 en effectuant un AND entre exp[0] et la valeur 1.

Pour évaluer les conditions "tant que code >= n" et "tant que m >= n" dans le Lustre, il suffit de vérifier que la valeur du registre N des différences code - n et m - n soit égale à 0.

5.4 Partie opérative

Schéma du circuit de la partie opérative :

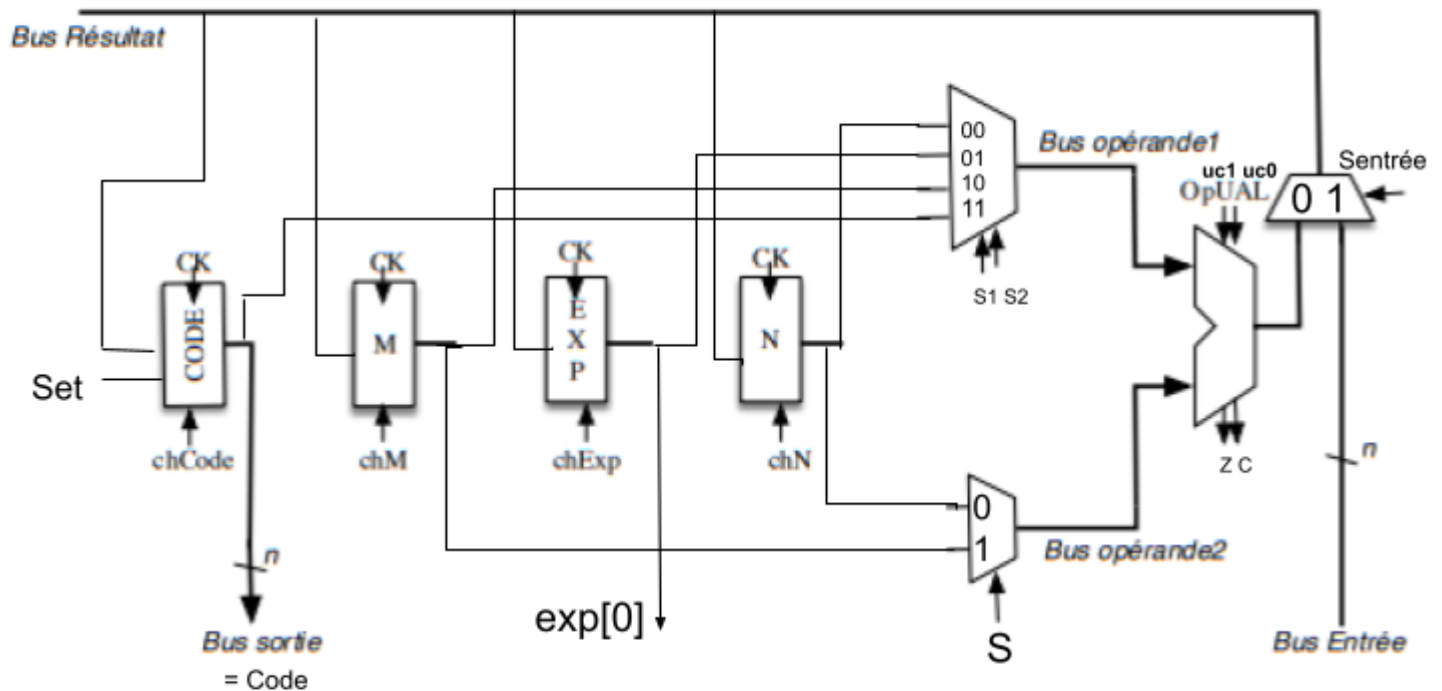


Figure 1: Schema de la partie operative

Les entrées de la partie opérative sont :

- chCode, chM, chExp et chN
- Le registre Set de CODE qui permet d'initialiser CODE a 1.
- la variable S du BusOperande2
- les variables S1 et S2 du BusOperande1
- les variables uc1 et uc0 et les registres Z et C de l'OpUAL
- la variable Sentrée du bus d'entrée.

(les entrées de la partie opérative seront ensuite les sorties de la partie contrôle).

On a, d'après le circuit de la partie opérative le codage suivant pour l'UAL:

S1	S2	variable
0	0	N (S = 0)
0	1	EXP
1	0	M (S = 1)
1	1	CODE

5.5 Partie contrôle

- 1) La partie contrôle prend en entrée le bit de poids le plus faible de exp (exp 0) ainsi que registres Z et C de l'opération obtenue avec l'OpUAL et les variables start et reset.

La partie contrôle renvoie donc S,S1,S2,uc1,uc0,Sentree, chEXP, chM, chN, chCode et end.

Schéma de l'automate :

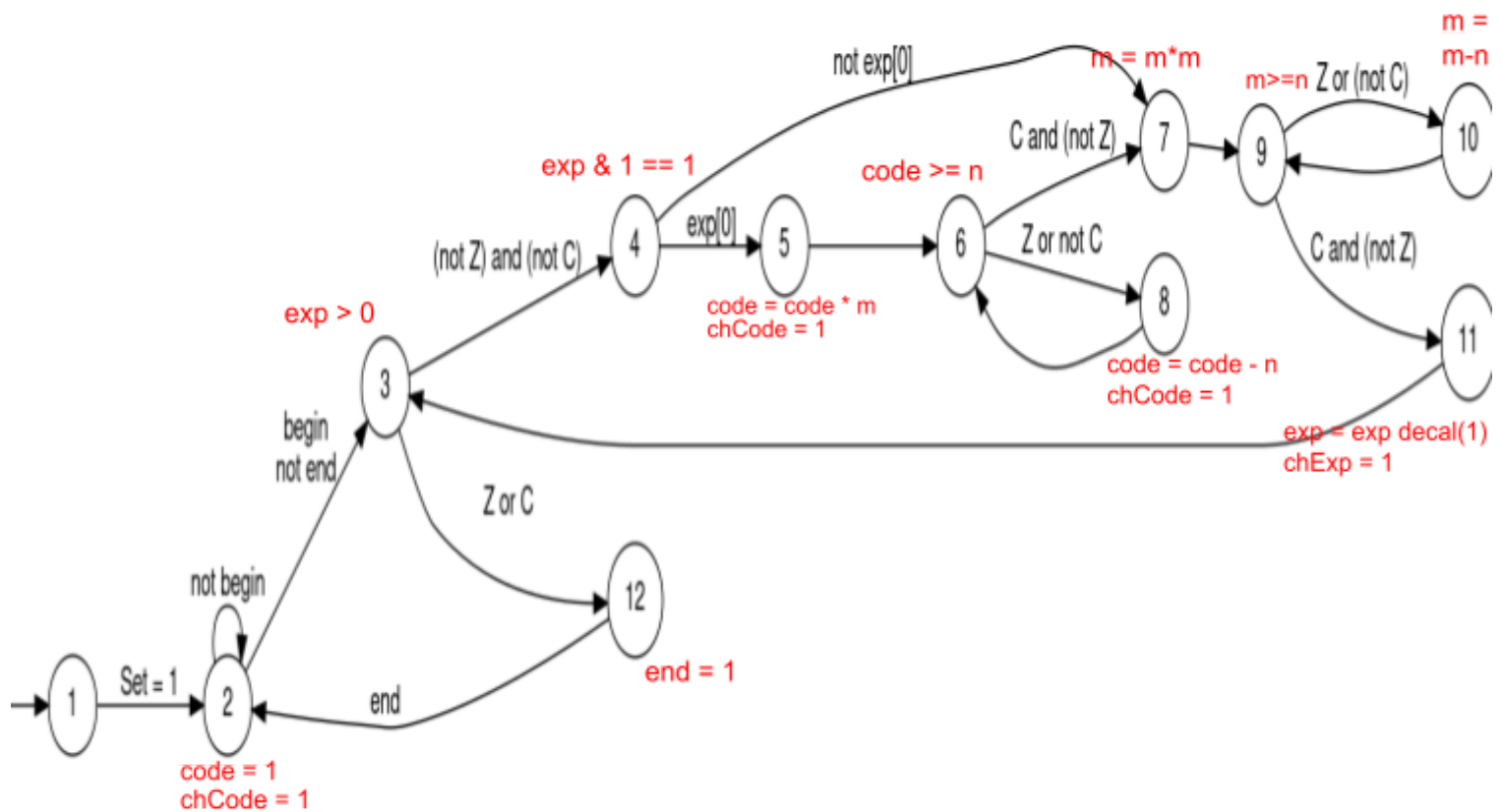


Figure 2 : Automate representant la partie controle

Dans l'automate, le flag C correspond au résultat de l'emprunt des soustractions m-n et code-n.

2)

	S 1	S 2	S	chCo de	chEx p	ch N	ch M	Sentré e	uc1	uc0	Set (misea1)	end
Init	0	0	0	1	0	0	0	1	0	0	1	0
2	1	1		0	0	0	0	0			0	0
3 (exp[0])	0	1		0	0	0	0	0			0	0
4	1	1	1	1	0	0	0	0	0	1	0	0
5	1	1	0	0	0	0	0	0	0	0	0	0
6	1	0	1	0	0	0	1	0	0	1	0	0
7	1	1	0	1	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	1	0	0	0	0	0
10	0	1		0	1	0	0	0	1	0	0	0
Fin				0	0			0			0	1

4) Afin d'initialiser le circuit, l'utilisateur doit charger les variables M, Exp et N avec des entiers (i.e initialiser chN, chExp et chN a 1). Cependant, la variable Code est initialisée au début du programme.