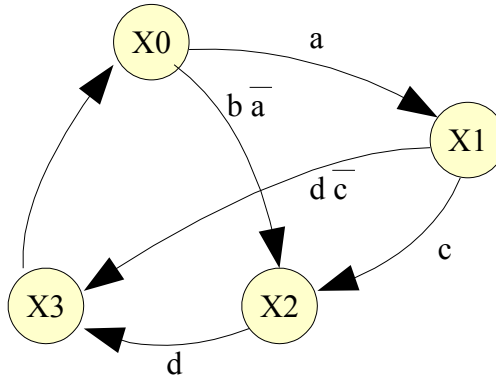


Final MI41

Documents autorisés : récapitulatifs des instructions ARM et transparents de cours
 Durée 2h
 Le code non expliqué et non commenté ne sera pas considéré comme valable

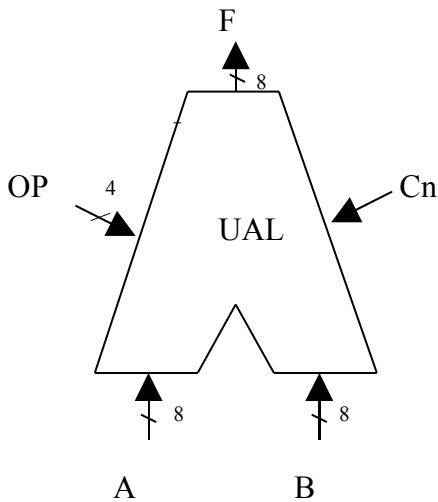
1. Machine à état câblée

- 1, Expliquer et détailler comment réaliser la machine à état permettant de réaliser le graphe à état donné ci après. On détaillera jusqu'aux équations logiques.
- 2, Donner la description VHDL permettant de synthétiser la machine à état câblée dont le comportement est donné par le graphe à état ci-après



2. Description VHDL

Donner la description VHDL (entity et architecture) de l'UAL vue en TD. On ne s'occupe pas de C, V et Z.



OP	Opération F
0 0 0 0	non A
0 0 0 1	A plus B
0 0 1 0	A plus B plus Cn
0 0 1 1	A moins B
0 1 0 0	A moins B moins Cn
0 1 0 1	A OU B
0 1 1 0	A ET B
0 1 1 1	A ⊕ B
1 0 0 0	A
1 0 0 1	LSL
1 0 1 0	LSR
1 0 1 1	ASR
1 1 0 0	ROR
1 1 0 1	RRX
1 1 1 0	-
1 1 1 1	-

Note : Pour les décalages c'est la donnée présente sur A qui est décalée

3. Assembleur

1. Ecrire le code ASM ARM d'une fonction que l'on appellera *umull* réalisant la multiplication non signée de deux entiers 32 bits. Les arguments de la fonction seront transmis par les registres internes r0 et r1. Le résultat sera un entier 32 bits qui sera renvoyé par le registre r0.

Cette fonction devra effectuer la multiplication par additions successives et ne pas utiliser les instructions arm de multiplication. Par ailleurs cette fonction intégrera une fonctionnalité de saturation. A savoir que si le résultat de la multiplication est un nombre supérieur ou égal à 2^{32} , elle devra renvoyer le nombre $2^{32}-1$ (plus grand nombre représentable sur 32 bits).

2. Ecrire le code ASM ARM d'une fonction que l'on appellera *smull* réalisant la multiplication de deux nombres signés. Cette fonction utilisera bien évidemment la fonction *umull* écrite précédemment. Les arguments seront également transmis via les registres interne r0 et r1 et le résultat retourné via le registre r0.