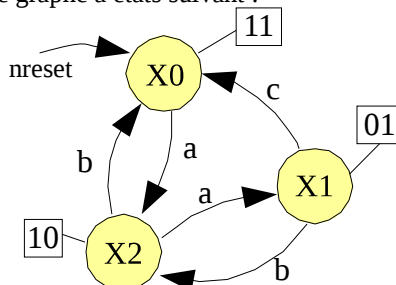


Final MI41

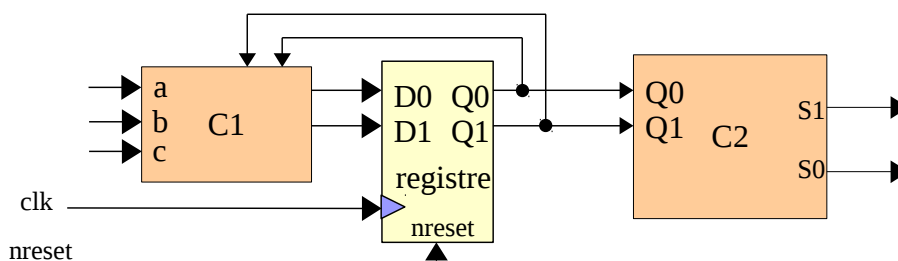
Durée 1h30, documents autorisés : fiche récapitulative ASM ARM

1. Machine à états (5pts)

On considère la machine à états décrite par le graphe à états suivant :



La structure de la machine à état est la suivante :



1. Donnez la description vhdl de la machine à états complète
2. Synthèse : le codage des états choisi est le suivant : X0 : 11, X1 : 01, X2 : 10, donnez les équations logiques des deux systèmes combinatoires C1 et C2.

2. ASM (5, 5 et 5 pts)

Les fonctions suivantes sont à écrire en assembleur ARM. Les fonctions doivent suivre les conventions d'appel standard ARM APCS.

1. Ecrire une fonction de d'attente qui scrute la valeur d'un bit d'une case mémoire : si le bit N^on est à 0 l'exécution reste bloquée dans la fonction et attend que ce bit passe à 1 avant de revenir, si le bit N^on est déjà 1 l'exécution doit revenir de la fonction immédiatement.

prototype C : void attend_bit_n(int* a, int n) ;

2. Fonction décrite par le code C suivant :

```

extern int dummy1(int x) ;

int dummy(int a, int b)
{
    int res ;
    if (a > b)
        res = dummy1(a) - dummy1(b)
    else
        return -1 ;
}
  
```

3. Fonction de filtrage décrite par le code C suivant :

```

int filtre (int * a, int * b, int N)
{
    int i, res ;
    res = 0 ;
    for (i=0;i<N;i++){
        res = res + a[i] * b[i] ;
    }
    return res ;
}
  
```

Ce code réalise $\sum_0^{N-1} a_i b_i$ où les valeurs a_i et b_i sont placées dans les tableaux d'entiers (32bits) a et b de taille N.