

Programmation des systèmes réactifs

Contrôle

ESINSA - Option GSE

26 octobre 2004

Documents : autorisés

Durée : 1h30.

Identification :

1 Détection de fronts

Soit x un flot booléen. On rappelle que le flot qui détecte les fronts montants peut être programmé en LUSTRE de la façon suivante :

```
1 node RisingEdge (x:bool) returns (z:bool);
2 let
3   z = false  $\rightarrow$  x and not pre x;
4 tel
```

1.1 Programmes

Écrire `FallingEdge` et `AnyEdge` des nœuds LUSTRE qui détectent respectivement les fronts descendants et les fronts (aussi bien montants que descendants) du flot booléen passé en entrée.

1.2 Trace d'exécution

Compléter le tableau 1. (Remarque : t remplace true, f remplace false).

Instant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	t	t	t	f	t	f	f	f	t	f	f	t	t	t	f	f	t
RisingEdge(x)																	
FallingEdge(x)																	
AnyEdge(x)																	

TAB. 1 – Traces d'exécution des nœuds LUSTRE `Edge`.

1.3 Équations de récurrence

Donnez les équations de récurrence caractérisant `AnyEdge(x)`.

1.4 Programme ESTEREL

En ESTEREL on peut travailler avec des signaux purs. La convention est qu'un signal présent est considéré comme vrai, un signal absent comme faux. Un front montant est alors le passage d'absent à présent ; un front descendant le passage de présent à absent.

Écrire un module AnyEdge pour détecter tous les fronts.

1.5 Solution en SYNCCHARTS

Dessiner un syncChart AnyEdge pour détecter tous les fronts.

2 Retard inertiel

Un retard inertiel est une forme de filtrage destinée à éliminer les variations trop rapides des entrées. Un retard inertiel d'ordre n ne laisse passer que les variations de niveau de durée supérieure ou égale à n instants. La figure 1 correspond au cas $n = 3$.

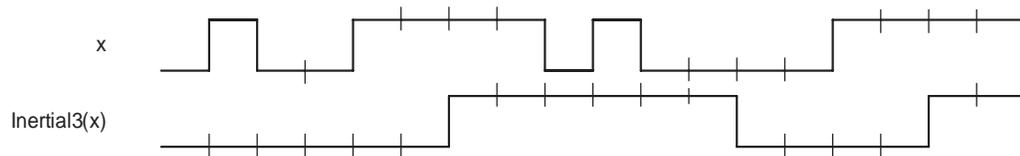


FIG. 1 – Comportement d'un filtre inertiel d'ordre 3.

2.1 Retard inertiel d'ordre 2 en LUSTRE

Le nœud RI2 programme un filtre inertiel d'ordre 2 en LUSTRE.

```

1 node RI2 (x : bool) returns (z : bool);
2 var px : bool;
3 let
4   z = false -> if x and px then true
5     else if not x and not pre x then false
6       else pre z;
7   px = false -> pre x;
8 tel
    
```

Commenter le code et compléter la trace d'une exécution (Table 2).

Instant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	t	t	t	f	t	f	f	f	t	f	f	t	t	t	f	f	t
z=RI2(x)																	
AnyEdge(z)																	

TAB. 2 – Traces d'exécution du nœud LUSTRE RI2.

2.2 Solution en ESTEREL

Question a: Pourquoi le corps de module suivant ne convient-il pas ?

```

1 loop
2   await 2 x;
3   abort
4   sustain z
    
```

```

5     when 2 [not x]
6     end loop

```

Question b: Commenter le code suivant et expliquer son fonctionnement

```

1     trap T in
2     loop
3     await x;
4     pause;
5     present x then
6     exit T
7     end present
8     end loop
9     end trap

```

Question c: En vous inspirant du code précédent, écrire le module ESTEREL RI2 du filtre inertielle d'ordre 2, ayant les signaux purs x en entrée et z en sortie.

3 Vérification

Si le filtre inertielle d'ordre 2 marche correctement il devrait toujours y avoir en sortie au moins deux instants successifs avec la même valeur.

Écrire un programme LUSTRE pour vérifier cette propriété.

Remarque : la propriété peut aussi s'exprimer par le fait que les situations de la figure 2 sont interdites.

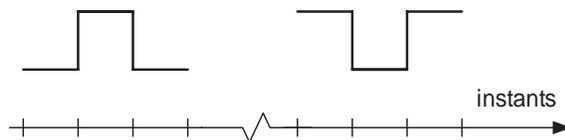


FIG. 2 – Situations interdites.