

Le circuit SBUF-READ-CTL :

Cet exemple est tiré de la thèse de Marco Antonio Peña Basurto. La spécification du contrôleur *sbuf-read-ctl* décrit sous forme d'un graphe STG et son implémentation sont donnés dans les figures 1.a et 1.b respectivement.

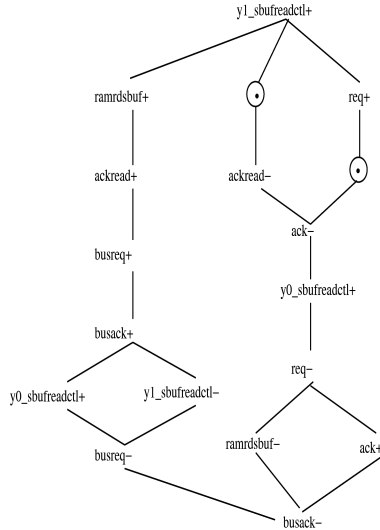


Figure 1: le contrôleur *sbuf-read-ctl*: (a) La spécification du contrôleur *sbuf-read-ctl*. (b) L'implémentation du contrôleur *sbuf-read-ctl*.

Le circuit qui implémente cette spécification est décrit par le programme VHDL suivant :

```

ENTITY sbuf_read_ctl IS
  PORT (
    ack          : out  BIT;
    ramrdsbuf    : out  BIT;
    busreq       : out  BIT;
    req          : in   BIT;
    busack       : in   BIT;
    ackread     : in   BIT
  );
END sbuf_read_ctl;

ARCHITECTURE RTL OF sbuf_read_ctl IS
  SIGNAL y1_sbufreadctl : BIT;
  SIGNAL y0_sbufreadctl : BIT;

BEGIN

  ack <= y0_sbufreadctl and not busack;
  ramrdsbuf <= y1_sbufreadctl or busack;
  busreq <= (not y0_sbufreadctl and busreq) or (ackread and y1_sbufreadctl);
  y1_sbufreadctl <= (y1_sbufreadctl and not busack) or (not ackread and req);
  y0_sbufreadctl <= (req and y0_sbufreadctl) or busack;

END;

```

Nous avons effectué un test sur l'environnement décrit ci-dessous. Les délais de traversée des portes du circuit sont tous définis à une unité de temps.

```

busack : 30 up, 40 down ;
req : 10 up, 50 down ;

```

ackread : 20 up, 60 down ;

La description en Uppaal générée possède les caractéristiques suivantes :

- 5 automates associés aux portes du circuit + les automates d'environnement.
- 5 horloges locales pour les cinq automates associées aux portes du circuit + l'horloge globale.
- 8 variables discrètes associées aux signaux du circuit.
- 10 paramètres associés aux délais des fronts montants et descendants des signaux du circuit.
- 3 x 5 locations des automates associés aux portes du circuit + les locations des automates d'environnement.

Les propriétés mentionnées ci-dessous sont bien vérifiées. Elles expriment le comportement attendu des signaux de sortie qui est conforme à la spécification du contrôleur décrite par le graphe STG.

- AG (((t>0 & t<41) or t>52) => ack = 0) & ((t>41 & t<52) => ack = 1)) .
- AG (((t>0 & t<21) or t>32) => busreq = 0) & ((t>21 & t<32) => busreq = 1)) .
- AG (((t>0 & t<12) or t>41) => ramrdsbuf = 0) & ((t>12 & t<41) => ramrdsbuf = 1)) .

Nous avons constaté que le temps de génération de la description en Uppaal et le temps de la vérification de chacune de ces trois dernières propriétés sont inférieurs à 0,1 sec.

La projection des temps d'apparition des fronts des signaux du circuit dans le graphe STG est donnée dans la figure 2.

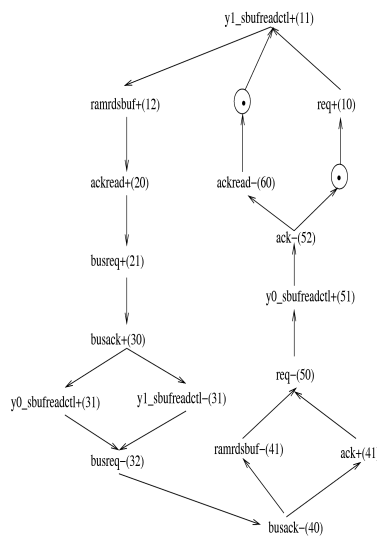


Figure 2 : les temps d'apparition des signaux du circuit *sbuf-read-ctl* dans le graphe STG du circuit.

Comme le montre le graphe construit par simulation, décrit dans la figure 3, la séquence d'évènements est bien conforme à la spécification du module *sbuf-read-ctl* présentée dans la figure 1.

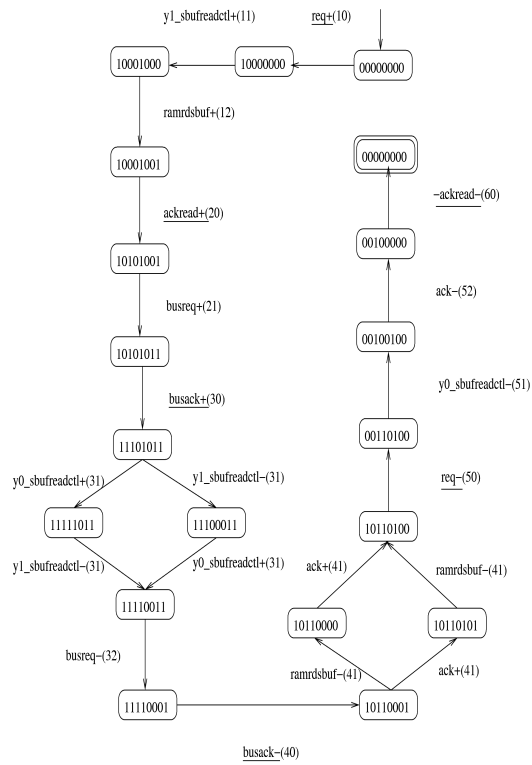


Figure 3 : le graphe d'accessibilité construit par Uppaal.

Bibliographie

[P.03] M. A. Peña. *Relative timing based verification of concurrent systems*. Department of Computer Architecture, Technical University of Catalonia, Barcelona, Spain, April 2003.