Projet ANR VALMEM

Délivrable : D3.1

Titre : *Définition d'un modèle d'automates temporisés pour l'analyse de circuits mémoire*

Auteurs : E. André, E. Encrenaz, L. Fribourg

Version : 1

Date : *25 janvier 2008*

VALMEM : Validation fonctionnelle et temporelle des mémoires embarquées décrites au niveau transistor par des méthodes formelles

Définition d'un modèle d'automates temporisés pour l'analyse de circuits mémoire

E. André, E. Encrenaz, L. Fribourg

25 janvier 2008

1 Description du document

Deliverable D3_1, fourni par le LSV.

Ce document décrit différentes représentations des composants élémentaires d'un circuit mémoire sous la forme d'automates temporisés et propose un modèle sur lequel nous baserons notre méthode de vérification.

2 Eléments constitutifs

Après les étapes d'extraction fonctionnelle et analyse de timing, les éléments constitutifs des mémoires embarquées sont :

- des portes logiques,
- des bascules transparentes (latch),
- des blocs fonctionnels abstraits (sense amplifier).

3 Différents modèles temporels

Les modèles temporels sont décrits sous la forme de STG (State-Transition Graph cf. livrable D1.2) :

- **le modèle de l'inverseur généralisé** associe un triplet pour chaque front de sortie $< d\uparrow, D\uparrow, S\uparrow >$ et $< d\downarrow, D\downarrow, S\downarrow >$, déterminant le délai minimum (resp. maximum et la pente, i.e. la raideur du front) entre un front survenant sur un signal d'entrée et le front de sortie induit.

- **le modèle entrée-sortie** associe un triplet $< d, D, S >$ pour chaque couple $<$ *signal d'entrée, signal de sortie* $>$.

- **le modèle complet** associe pour chaque couple $<$ *signal d'entrée, signal de sortie* $>$ un ensemble de triplets $< c, d, S >$ représentant respectivement la configuration d'entrée, le délai entre le front survenant sur le signal d'entrée et le front de sortie, et la pente du front du signal de sortie. C'est ce dernier modèle qui sera extrait par l'équipe du LIP6 et fourni au LSV.

4 Différentes représentations des composants sous forme d'automates temporisés

4.1 Le modèle des automates temporisés

. Les automates temporisés sont un modèle proposé par [1] permettant de représenter des actions et du temps quantitatif. Nous reprenons la définition de [8]).

Definition
Un automate temporisé est défini par un tuple $\mathcal{A} = (\Sigma, Q, \mathcal{X}, I, \mathcal{S})$ *tel que :*

- Σ *est un ensemble fini d'actions*
- Q *est un ensemble fini de locations (ou états de contrôle)*
- $\mathcal{X}$ *est un ensemble fini d'horloges*
- I *est un invariant, affectant à chaque location* $q \in Q$ *une expression de la forme :* $\wedge x \leq u$ *pour certaines horloges* $x \in \mathcal{X}$ *et* $u \in \mathbb{N}$.
- $\mathcal{S}$ *est une relation de transition (ou step), composée d'élements de la forme* (q, g, a, r, q')*, où :*
  - $q, q' \in Q$ *les locations source et destination de la transition,*
  - $a \in \Sigma$*, l'action*
  - $r \in \mathcal{X}$ *est l'ensemble des horloges à remettre à zéro lors du franchissement de la transition, et*
  - $g : \mathcal{X} \to \mathbb{B}$ *est la garde de la transition : elle est exprimable sous forme d'une conjonction de termes de la forme* $x \geq l$*, pour certaines horloges* $x \in \mathcal{X}$ *et* $l \in \mathbb{N}$.

Une *valuation des horloges* v est une fonction de $\mathcal{X}$ dans $\mathbb{R}+$, et une configuration de l'automate est un couple (q, v), composé d'une location de l'automate et d'une valuation v. Nous notons $v + d$ la valuation d'horloges obtenue à partir de la valuation v dans laquelle toutes les valeurs des horloges ont été augmentée de d. Nous notons également ρ la fonction reset qui affecte à 0 les horloges de r et laisse les autres inchangées.

Un *pas* de l'automate peut avoir deux formes :

- action discrète : $(q, v) \xrightarrow{a} (q', v')$ pour une transition (q, g, a, ρ, g, q') de $\mathcal{S}$ telle que $v' = \rho(v)$ et v satisfait g.
- avancée du temps : $(q, v) \xrightarrow{d} (q, v + d)$ pour un délai $d \in \mathbb{R}+$ tel que $v + d$ satisfait I_q.

Une *étape composée* correspond à une avancée du temps (éventuellement de durée nulle) suivie d'une action discrète : $(q, v) \xrightarrow{d,a} (q', v') \equiv (q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v')$.

Une exécution d'un automate $\mathcal{A}$ à partir d'une configuration (q_0, v_0) est une séquence finie d'étapes composées et se terminant par une avancée du temps. $\xi : (q_0, v_0) \xrightarrow{d_1,a_1} (q_1, v_1) \xrightarrow{d_2,a_2} \ldots (q_k, v_k) \xrightarrow{d_*} (q_k, v_k + d_*)$. On note également $(q_0, v_0) \xrightarrow{\xi} (q', v')$ l'exécution ξ.

Definition
Un réseau d'automates temporisés est défini par $\mathcal{A} = \mathcal{A}^1 \,||\, \mathcal{A}^2 \,||\, \ldots \mathcal{A}^n$ *où chaque automate local* $\mathcal{A}^i$ *est de la forme* $\mathcal{A}^i = (\Sigma^i, Q^i, \mathcal{X}^i, I^i, \mathcal{S}^i)$*. Les ensembles de locations sont disjoints entre eux et les ensembles d'horloges sont disjoints entre eux. L'automate produit* $\mathcal{A}$, obtenu à partir de la composition parallèle des automates locaux $\mathcal{A}^i$ est defini tel que : $\mathcal{A} = (\Sigma, Q, \mathcal{X}, I, \mathcal{S})$ avec $\Sigma = \bigcup_{i=1}^{n} \Sigma^i$, $Q = \Pi_{i=1}^{n} Q^i$, $\mathcal{X} = \bigcup_{i=1}^{n} \mathcal{X}^i$. L'état de contrôle global est noté $\overline{q} = (q^1, ..., q^n)$ et la valuation globale est notée $\overline{v} = (v^1, ..., v^n)$.

La sémantique du réseau d'automate est définie comme suit :

- action discrète : $(q, v) \xrightarrow{a} (q', v')$ telle que, pour tout i : ou bien $a \in \Sigma^i$ et $(q^i, v^i) \xrightarrow{a} (q^{i'}, v^{i'})$, ou alors $a \notin \Sigma^i$ et alors $(q^{i'}, v^{i'}) = (q^i, v^i)$.
- avancée du temps : $(q, v) \xrightarrow{d} (q, v + d)$ pour un délai $d \in \mathbb{R}+$ tel que $v + d$ satisfait $\wedge_{i=1}^{n} I_q^i$.

La vérification par model-checking consiste à décider si un système, modélisé sous la forme d'un automate, satisfait sa spécification, représentée sous la forme d'un ensemble de propositions logiques. La vérification par modèle de propriétés qualitatives (sans représentation quantitative

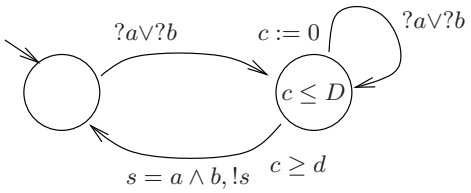

FIG. 1 – Automate temporisé associé à une porte AND à deux entrées a et b et une sortie s. Le délai de propagation est compris dans l'intervalle $[d, D]$. Les valeurs des entrées a et b et de la sortie s sont stockées dans des variables globales.

du temps) est un problème largement exploré (cf. [4] pour un tour d'horizon des méthodes classiquement utilisées). Son extension aux systèmes intégrant une notion de temps quantitatif et modélisés sous la forme d'automates temporisés a été proposée par [1] et constamment améliorée depuis. Une présentation détaillée peut être trouvée dans [2].

4.2 Modélisation des éléments constitutifs des mémoires par automates temporisés

Nous présentons différentes modélisations possibles des éléments constitutifs des circuits mémoire, en nous focalisant sur le respect de la fonctionalité et du délai associé à la production d'une sortie pour chaque composant. En particluier, nous ne considérons pas la pente des signaux[1].

4.2.1 Le bi-bounded inertial delay

C'est le modèle le plus classiquement rencontré dans la littérature portant sur l'analyse de circuits asynchrones (problème TSE [7, 3], model-checking temporisé [6]). Dans ce modèle, le délai de propagation d'un front de l'entrée vers la sortie est compris dans un intervalle $[d, D]$. Cet intervalle peut être obtenu à partir des trois modèles temporels STG cités précédemment : il s'agit des valeurs extrêmes apparaissant sur les valuations des arcs des STG. De fait, il est très proche du modèle STG "inverseur généralisé". Précisément, le modèle inverseur généralisé associe deux intervalles à chaque porte, l'un pour la propagation d'un front montant et l'autre pour la propagation d'un front descendant. Cette distinction selon le sens du front peut être incorporée au modèle sans difficulté.

De plus, ce modèle de propagation est *inertiel* : la survenue de fronts "trop rapprochés" sur les signaux d'entrée n'est pas répercutée sur la sortie de l'élément. Seuls les changements sur les entrées plus lents que le délai de propagation de la porte sont répercutés. L'automate temporisé associé à une porte AND est donné sur la figure 1.

4.2.2 Le modèle utilisé dans le projet BLUEBERRIES

Ce modèle est très proche du bi-bounded inertial delay : le délai de propagation de chaque porte combinatoire est compris dans un intervalle $[d^{\uparrow}, D^{\uparrow}]$ ou $[d^{\downarrow}, D^{\downarrow}]$ selon le sens du front du signal propagé. Par contre, la détermination des bornes de ces intervalles a été obtenue manuellement (par simulation électrique) ; pour chaque porte, seules quelques arêtes du STG

[1] La raideur des fronts d'entrée et de sortie des portes est prise en compte lors de la détermination des délais dans l'étape d'extraction de timing.

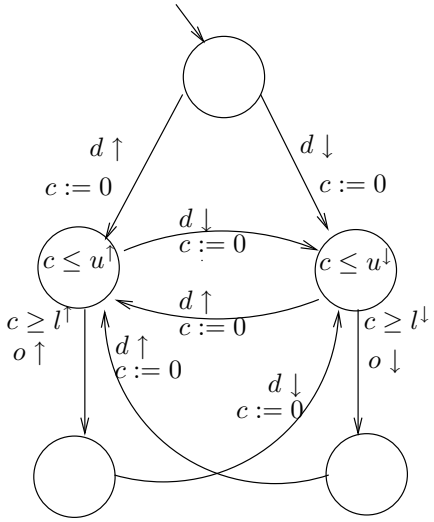

FIG. 2 – Automate temporisé modélisant le délai de propagation d'une porte reporté sur une entrée. L'intervalle de propagation dépend du sens du front propagé, et est compris dans les intervalles $[l^{\uparrow}, u^{\uparrow}]$ et $[l^{\downarrow}, u^{\downarrow}]$.

complet ont pu être caractérisées, et *les bornes des intervalles ont été déterminées à partir de ces données parcellaires.*

Les figures suivantes (2,3) présentent la modélisation d'une porte logique combinatoire suivant ce modèle de délai. Le délai associé à la traversée de la porte est reporté sur les entrées, et modélisé dans un automate distinct de l'automate représentant la fonctionalité de la porte. Une représentation sous la forme d'un unique automate, regroupant le délai de propagation et la détermination de la fonctionalité est possible : il s'agit alors de l'automate *produit* des deux automates distingués ici.

La représentation du latch est donnée sur la figure 4.

4.2.3 Le modèle complet

Ce modèle associe un délai (ponctuel) pour chaque front de signal d'entrée et pour chaque configuration des (autres) signaux d'entrée. Il n'est plus intéressant de distinguer propagation et fonctionalité puisque les deux dépendent de chaque configuration d'entrée. On représente alors la propagation et la fonctionalité sur un unique automate.

Ce modèle est extrêmement précis mais ne permettra pas de traiter des portions de circuit de taille significative : il engendre une combinatoire extrêmement élevée, et un très grand nombre de paramêtres.

5 Notre choix

Nous avons choisi d'adopter le modèle *bi-bounded inertial delay*, et de procéder si nécessaire à des raffinements pour des portions de circuits particulièrement sensibles. Ce modèle standard est celui qui limite au maximum l'explosion combinatoire. Il est possiblement trop grossier pour vérifier les temps de réponse de circuits particulièrement ajustés ; c'était notamment le cas lors de l'analyse de la mémoire SPSMALL dans le cadre du projet BLUEBERRIES : nous avons été

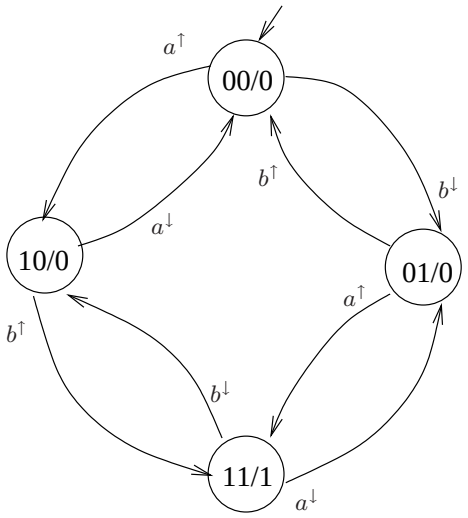

FIG. 3 – Automate (temporisé) modélisant la fonctionnalité d'une porte AND. Les fronts montants et descendants des signaux d'entrée et de sortie sont distingués.

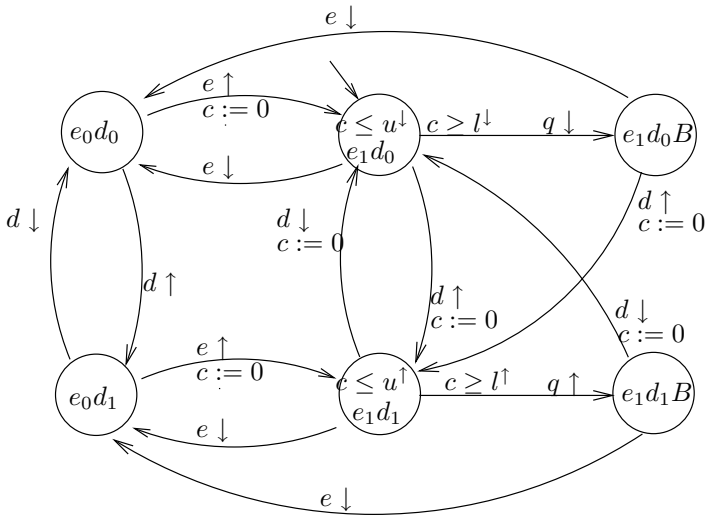

FIG. 4 – Automate temporisé associé à un latch, avec les délais de recopie de l'entrée d sur la sortie q compris dans les intervalles $[l^\uparrow, u^\uparrow]$ et $[l^\downarrow, u^\downarrow]$.

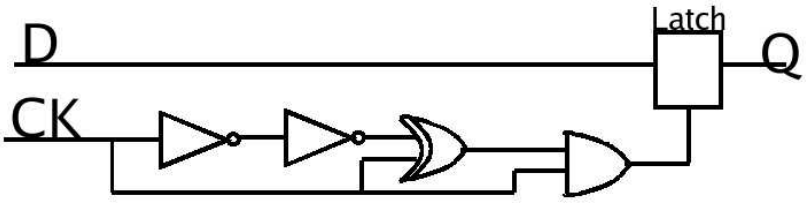

FIG. 5 – Un circuit comprenant un générateur de créneau et une latch.

amené à différencier les intervalles selon le sens du front propagé. Ce raffinement a été suffisant pour prouver les proprités de temps de réponse du corcuit, et pour construire des contraintes garantissant des plages de bon fonctionnement pour plusieurs circuits batis selon la même architecture et implantés dans des technologies différentes.

Nous pouvons poursuivre cette démarche de raffinement : il est en effet possible de connecter l'automate temporisé d'une porte modélisée selon le *bi-bounded inertial delay* en entrée ou en sortie d'une porte modélisée selon le modèle complet (ou un modèle intermédiaire, associant un intervalle de délais pour chaque configuration d'entrée par exemple) car

1. chaque automate dispose de sa propre horloge,
2. les interactions entre automates sont réalisées par des transitions synchronisées, qui correspondent à des transitions d'action et non pas à des transitions d'écoulement du temps.
3. le changement de représentation des délais de propagation n'induit pas de changement d'échelle de temps : tous les automates sont plongés dans le même temps global et toutes les horloges dérivent à la même vitesse.

5.1 Cas d'une sortie connectée vers plusieurs entrées

Si le signal de sortie d'une porte est connecté en entrée de plusieurs portes (disons n, ce signal est dupliqué n fois : l'automate produisant le signal de sortie produit en fait n transitions pour chaque front du signal de sortie, chacune étant synchronisée sur les transition d'une porte aval.

6 Un exemple

Dans cette section, nous présentons les grandes étapes de l'extraction des contraintes portant sur les délais et garantissant le fonctionnement correct d'un circuit proposé par STMicroelectronics. Le détail des algorithmes sera présenté dans le livrable D3_2.

6.1 Circuit

Le circuit (cf. figure 5) proposé est composé d'un latch mémorisant la valeur du signal d'entrée D lorsque sa commande d'écriture E est activée (écriture sur niveau). La commande d'écriture est réalisée à partir du signal d'horloge, au travers d'un générateur d'impulsions (les deux portes NOT, la porte XOR, la porte AND). Tous ces éléments possèdent des temps de propagation intrinsèques.

Le chronogramme des signaux d'entrées (D et CK) et de sortie (Q) est donné sur la figure 6.

Il met en évidence les plages de stabilité du signal D autour du front montant de l'horloge. Au bout d'un certain temps, nommé $t_{CK \to Q}$, la valeur du signal D est reportée sur la sortie Q.

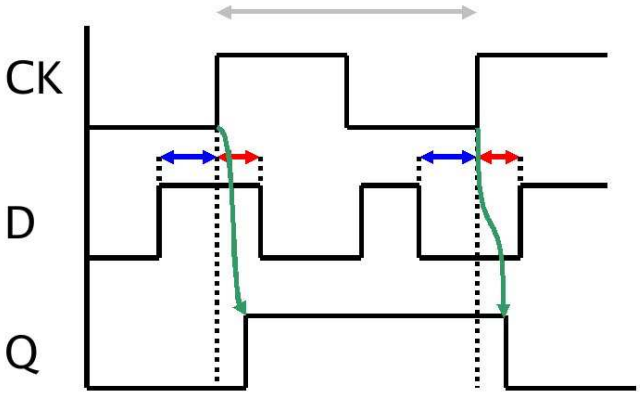

FIG. 6 – Chronogramme des signaux d'entrées et de sortie.

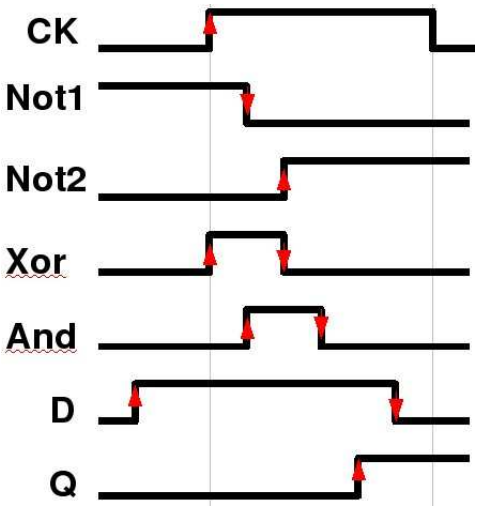

FIG. 7 – Chronogramme des signaux en sortie des portes du circuit.

La suite des fronts des signaux se propageant dans le circuit est décrite sur le chronogramme de la figure 7

Sur cette portion de circuit, on cherche à déterminer les contraintes liant les paramètres de délais associés aux différentes portes afin de garantir que la valeur de la donnée D est correctement recopiée sur la sortie Q.

6.2 Modélisation sous la forme d'automates temporisés

Dans cette première étude, les portes ont été modélisées manuellement sous la forme d'automates temporisés. Seules les transitions marquées en rouge sur le chronogramme précédent ont été représentées. La propagation des délais suit le modèle *bi-bounded inertial delay* : pour chaque porte P, le délai de propagation d'un front survenant sur une entrée de P vers la sortie de P est compris dans un intervalle $[dP\_l, dP\_u]$.

Par ailleurs, on identifie *une classe de mauvais comportements à éviter* : il s'agit des comportements pour lesquels une donnée est présentée sur le signal D, celle-ci est stable autour du

front montant de l'horloge CK, et aboutissant dans un état "mauvais", pour lequel :
- un cycle d'horloge s'est écoulé
- il n'y a pas eu de front montant sur le signal Q.

6.3 Méthode de vérification

La méthode verification suit l'algorithme décrit ci-après.

1. Départ de l'état final mauvais (pas de front montant sur Q),
2. Calcul des états accessibles en arrière,
3. Intersection non vide avec l'état initial : contre-exemple,
4. Obtention d'un ensemble de contraintes,
5. Ajout de la négation de l'une des contraintes au système et retour à l'étape 1.

Lorsque l'intersection entre l'état initial et les états accessibles en arrière depuis le mauvais état est *vide*, l'ensemble de contraintes obtenu assure le bon fonctionnement du système.

Les calculs des états accessibles ont été réalisés avec l'outil HyTech [5]. Les contre-exemples et contraintes ont été analysés manuellement. A chaque itération, le contre-exemple retenu est minimal :
- $\mathrm{Pre}^n(\mathrm{Final}) \cap \mathrm{Init} \neq \emptyset$
- $\mathrm{Pre}^{n-1}(\mathrm{Final}) \cap \mathrm{Init} = \emptyset$

Une fois obtenu un ensemble de contrainte éliminant *toutes* les traces menant vers le mauvais état, cet ensemble de contraintes est réduit. Les contraintes redondantes sont supprimées.
L'ensemble final de contraintes non redondantes est le suivant :

```
dHold > dAndUp2_u + dLatchUp_u
dAndUp2_u+dLatchUp_u < dNot1Down_l+dNot2Up_l+dXorDown1Up_l+dAndDown1_l
```

D'autres jeux de contraintes auraient pu être obtenus. Le jeu de contraintes final dépend, à chaque itération, du choix du contre-exemple à évincer et du choix, dans la contrainte associée à ce contre-exemple, de la sous-contrainte niée.

6.4 Interprétation des contraintes

Ces contraintes sont interprétables sur le circuit :
- `dHold > dAndUp2_u + dLatchUp_u` : Le temps de maintien de D doit être supérieur à la somme des temps maxima du front montant du « and » et de franchissement du latch.
- `dAndUp2_u+dLatchUp_u < dNot1Down_l+dNot2Up_l+dXorDown1Up_l+dAndDown1_l` : La somme des temps maxima du front montant du « and » et de franchissement du latch doit être inférieure au temps minimal nécessaire entre le front descendant du « not 1 » et le front descendant du « and ».

7 Conclusion

Nous avons envisagé différentes alternatives pour la représentation des éléments constitutifs des mémoires sous la forme d'automates temporisés. Le point crucial concerne le niveau d'abstraction auquel l'ensemble des délais de propagation est représenté : le LIP6 nous fournira une description suivant le modèle temporel STG complet, mais il nous est apparu que ce niveau est trop précis pour que l'on puisse espérer appliquer les méthodes de vérification par modèles sur des portions de circuit significatives. Nous proposons d'utiliser le modèle standard *inertial bi-bounded*

delay, qui permet d'obtenir des contraintes temporelles pour des circuits asynchrones dans le cas général. Lorsque les circuits sont optimisés pour obtenir de meilleures performances (en temps de réponse par exemple), les portions sensibles du circuit (sur le chemin critique) pourront être modélisées suivant le modèle complet, et combinées aux autres portions du circuits, moins sensibles, modélisées selon le bi-bounded inertial delay.

Références

[1] R. Alur and D.L. Dill. A Theory of Timed Automata. *Theoretical Computer Science 126*, pages 183–235, 1994.

[2] P. Bouyer and F. Laroussinie. Vérification par automates temporisés. In *Systèmes temps-réel 1 : techniques de description et de vérification*, pages 121–150. Hermès, 2006.

[3] S. Chakraborty and D. Dill. Approximate algorithms for time separation of events. In *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, pages 190–194. IEEE Computer Society, 1997.

[4] D. Peled E. Clarke, O. Grumberg. *Model Checking*. The MIT Press, 200.

[5] T. Henzinger, P. Ho, and H. Wong-Toi. A User Guide to HYTECH. In *Int. conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*.

[6] O. Maler and A. Pnueli. Timing analysis of asynchronous circuits using timed automata. In *Int. conf. on Correct Hardware Design and Verification Methods (CHARME)*, volume 987, pages 189–205, 1995.

[7] K. McMillan and D. Dill. Algorithms for interface timing specification. In *Proc. of the IEEE Int. Conf. on Computer Design (ICCD)*, pages 48–51. ACM/IEEE Computer Society, 1992.

[8] R. Ben Salah, M. Bozga, and O. Maler. On interleaving in timed automata. In *Int. conf. on Concurrency (CONCUR)*, volume 2791, pages 465–476, 2006.