

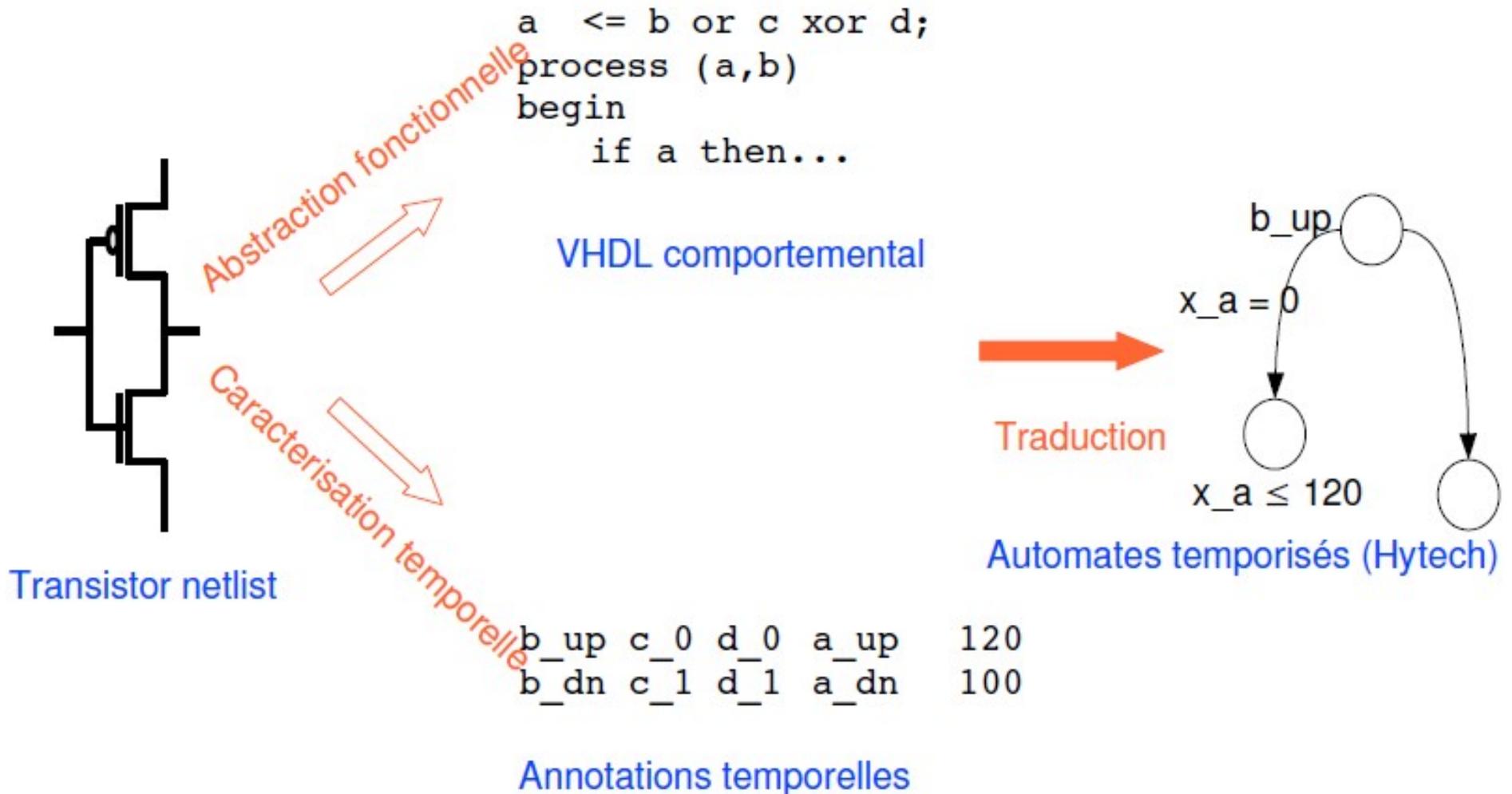
VALMEM

Vérification avec UPPAAL
de circuits décrits en VHDL avec délais

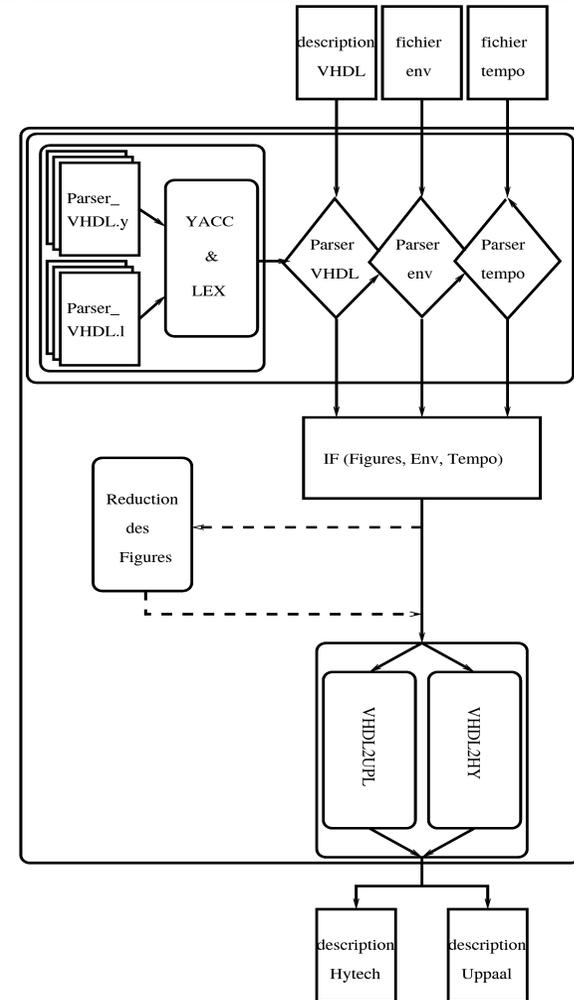
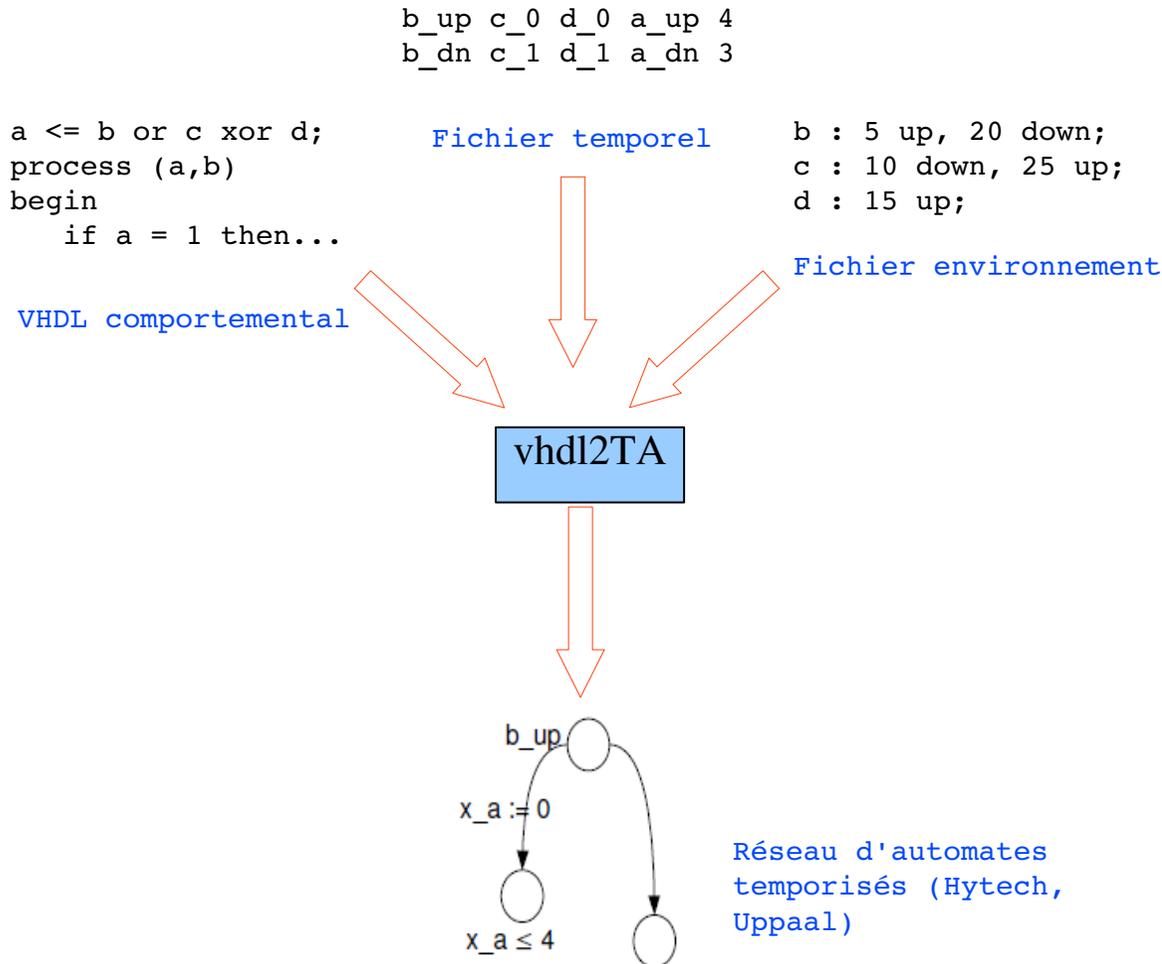
A. BARA, E. ENCRENAZ

LIP6

Traduction VHDL + temps en automates temporisés



Outil VHDL2TA



Architecture de l'outil

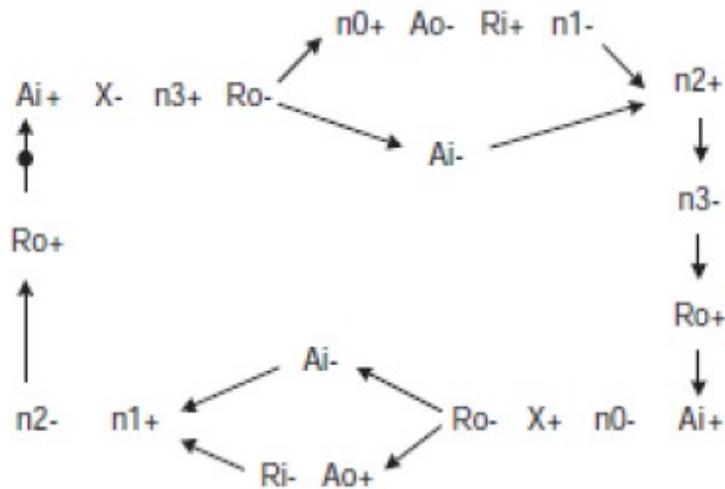
Etudes de circuits (1/4)

- Evaluation de VHDL2TA sur circuits de la littérature
 - Description VHDL + timings des circuits
 - Traduction automatique en automates temporisés (syntaxe HyTech ou UPPAAL)
 - Construction (semi-manuelle) du graphe des zones
 - Vérification automatique de propriétés TCTL

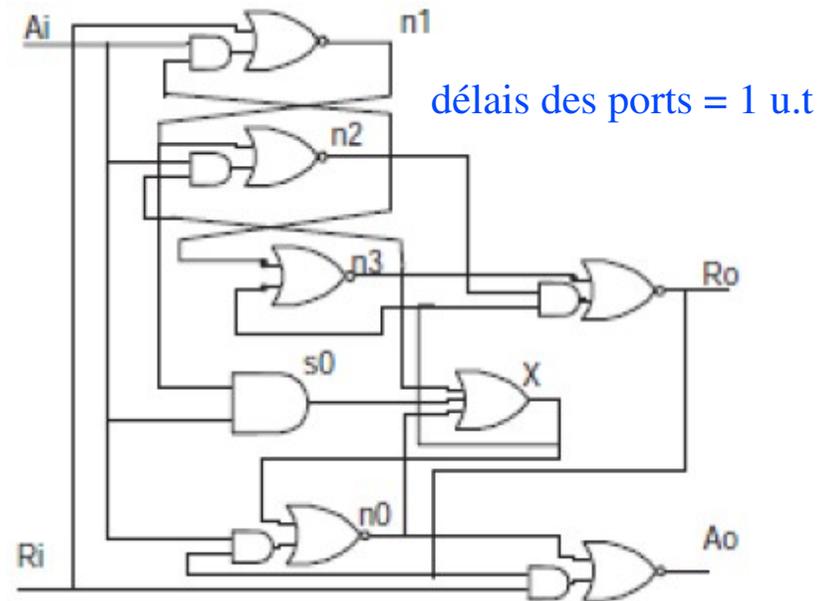
circuit's name	#comb / process	#TA (M+E)	#clocks	#var	#locations	#transitions	uppaal time
and-or [10] [3]	2 / 0	2+1	2	4	9	10	< 0.1s
chu150 [15]	7 / 0	7+3	7	10	41	56	< 0.1s
converta [16]	8 / 0	8+2	8	10	28	30	< 0.1s
desynch [8]	11 / 0	11+2	11	11	34	44	< 0.1s
D flip-flop [9] [2]	5 / 0	5+1	5	7	10	10	< 0.1s
half [5]	7 / 0	7+2	7	9	33	53	< 0.1s
mp-for-pkt [13]	10 / 0	10+1	10	14	38	45	< 0.1s
Muller C-element	1 / 0	1+1	1	3	7	7	< 0.1s
qr42 [11]	4 / 0	4+1	4	6	17	16	< 0.1s
rcv-setup [17]	13 / 0	13+1	13	17	46	49	< 0.1s
rpdfc [12]	8 / 0	8+1	8	12	32	34	< 0.1s
sbuf-read-ctl [14]	5 / 0	5+1	5	8	19	20	< 0.1s
sbuf-send-ctl [4]	14 / 0	14+3	14	17	80	108	< 0.2s
trigger [6]	2 / 0	2+2	2	4	11	12	< 0.1s
spsmall-blueb-lsv1 [7]	20 / 4	27+1	27	29	173	232	< 0.2s
spsmall-blueb-lsv2 [18]	25 / 6	31+1	31	35	248	338	< 0.3s

Etudes de circuits (2/4)

- Exemple : Converta [SGKS00]



Spécification STG du contrôleur *converta*



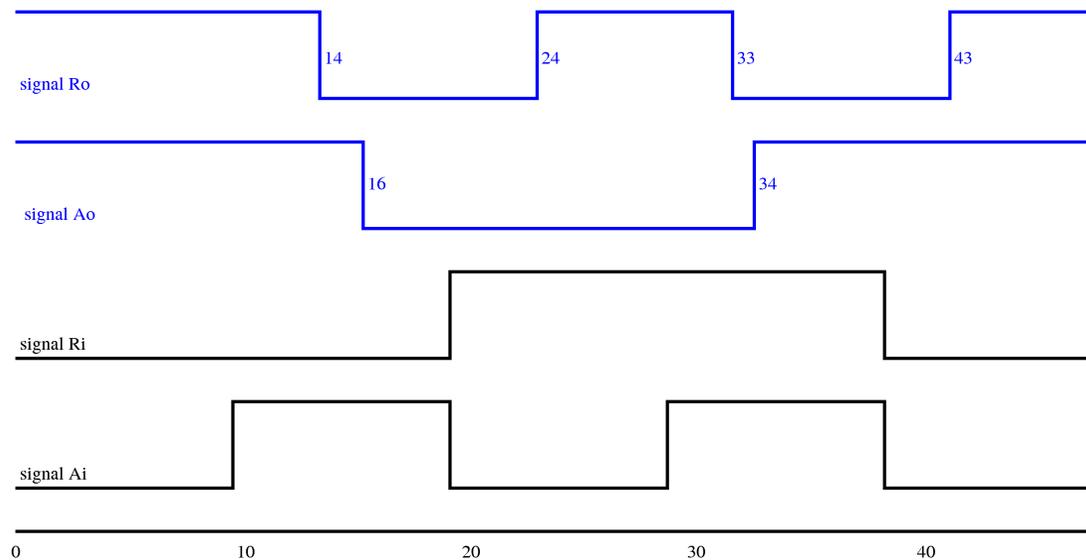
Implémentation du contrôleur *converta*

- Test avec l'environnement suivant

Ai : 10 up, 20 dn, 30 up, 40 dn;
 Ri : 10 up, 40 dn;

Etudes de circuits (3/4)

- Comportement des signaux de sortie



Ro : 14 dn, 24 up, 33 dn, 43 up;
Ao : 16 dn, 34 up;
Ri : 10 up, 40 dn;
Ai : 10 up, 20 dn, 30 up, 40 dn;

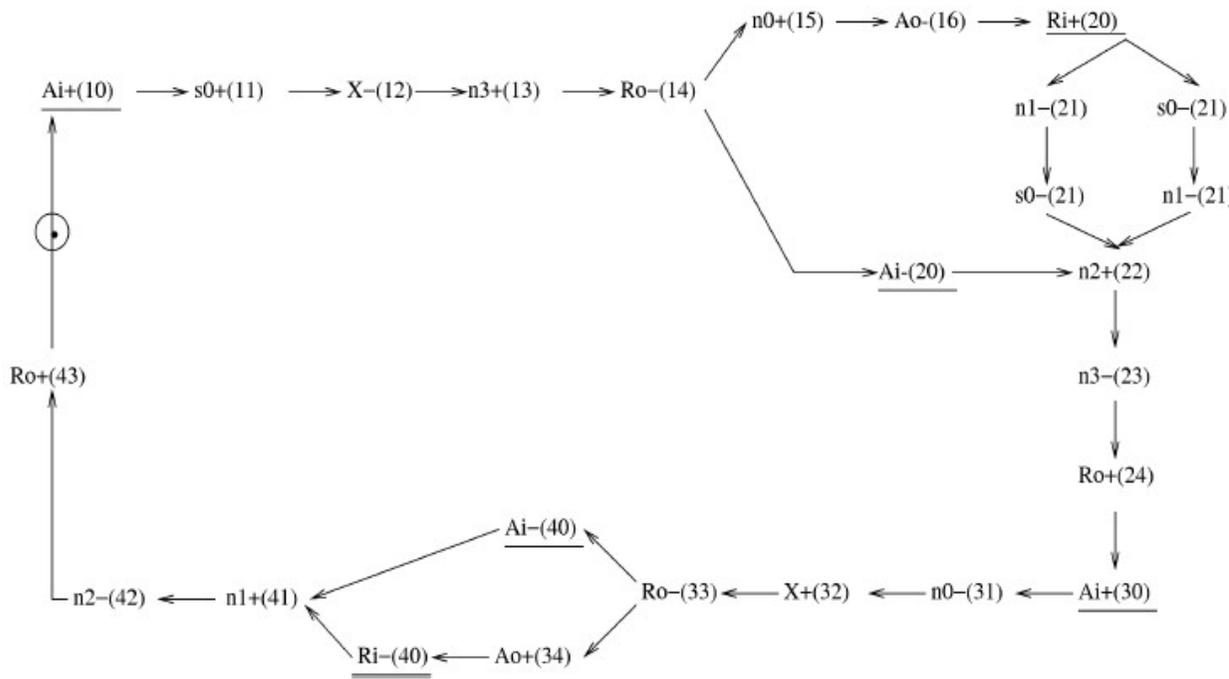
- Propriétés TCTL vérifiées

- $AG (((t > 14 \wedge t < 24) \vee (t > 33 \wedge t < 43)) \Rightarrow Ro = 0) \wedge (((t \geq 0 \wedge t < 14) \vee (t > 24 \wedge t < 33) \vee t > 43) \Rightarrow Ro = 1))$
- $AG ((t > 16 \wedge t < 34) \Rightarrow Ao = 0) \wedge ((t \geq 0 \wedge t < 14) \vee t > 34) \Rightarrow Ao = 1))$

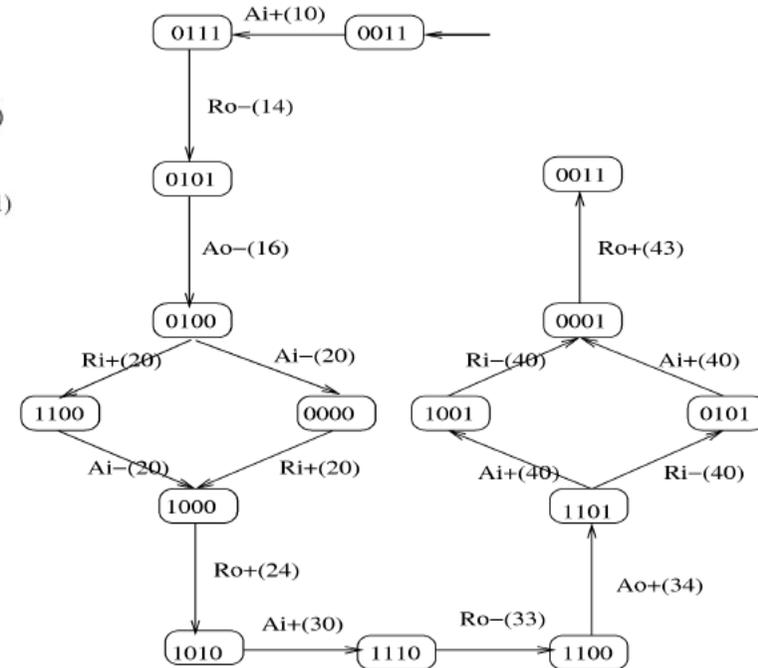
chacune de ces deux propriétés passe en < 0.1 sec

Etudes de circuits (4/4)

- Graphes d'états : (Graphe de transition d'états STG + Graphe d'atteignabilité RG)



Les temps d'apparition des signaux du circuit dans la spécification STG



Les temps d'apparition des signaux I/O dans le graphe RG

Analyse de l'architecture (1/10)

SP-SMALL (3x2 bits)

- Architecture complète abstraite automatiquement par LIP6 dans le projet VALMEM
 - 3 mots de 2 bits
 - 62 affectations concurrentes + 30 processus séquentiels
 - 8 signaux non connectés
- Délais de portes (Implémentation):
 - fichiers de timings associés aux portes et aux latches du circuits (donnés par Dominique, Patricia & Pirouz).
 - Extraire les intervalles des délais de propagation des fronts des signaux de sorties des portes et latches du circuit.

Analyse de l'architecture (2/10)

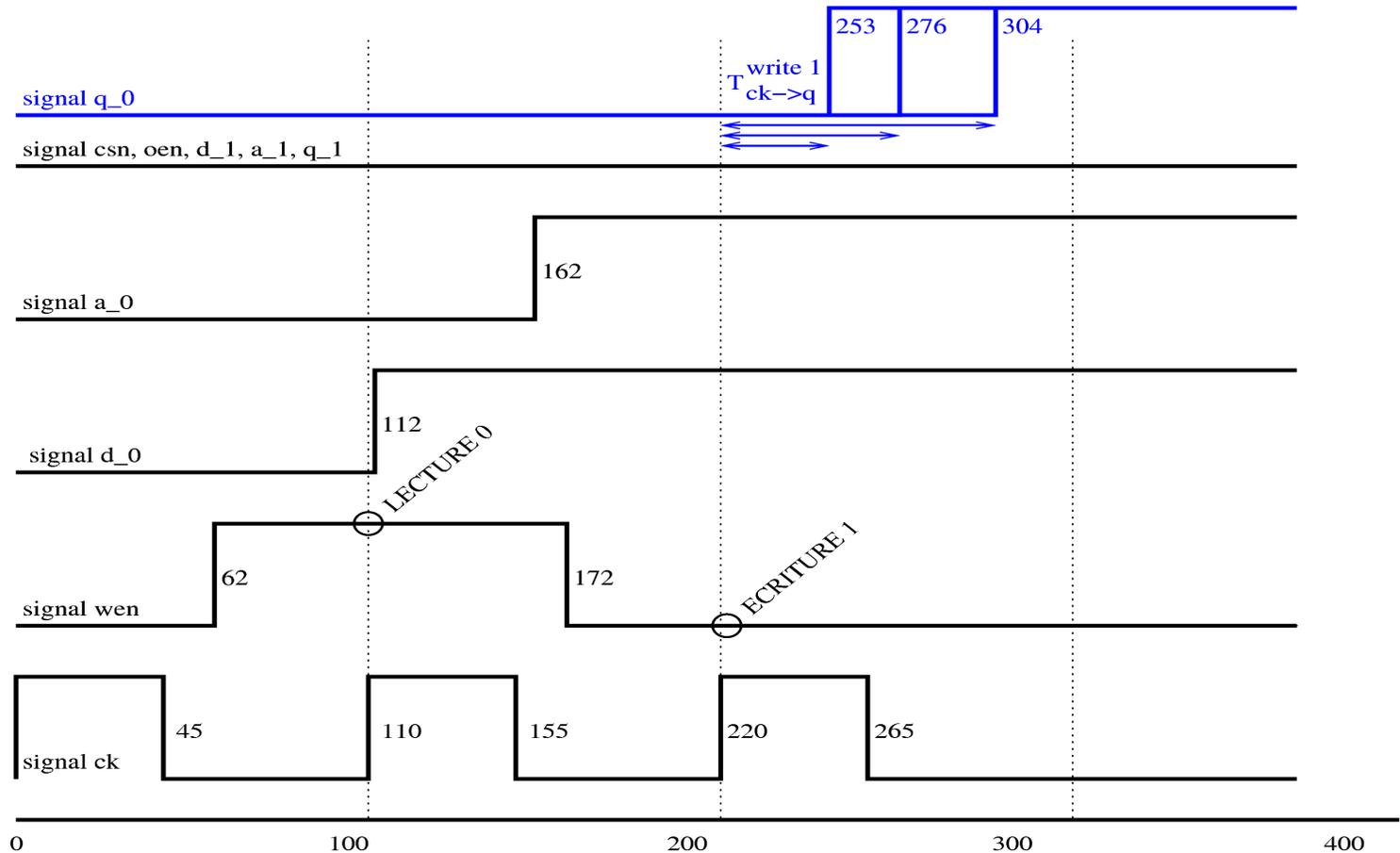
SP-SMALL (3x2 bits)

- Traduction :
 - 18087 lignes de code description UPPAAL.
 - 92 automates (portes & latches) + 24 automates (Gardes des process) + 1 automate Env.
 - 93 horloges + (92+24) var + 92x4 délais.
 - Pour UPPAAL, les délais sont instanciés aux bornes des intervalles calculés précédemment
 - Temps de traduction VHDL + temps → TA : 60 sec.

Analyse de l'architecture (3/10)

SP-SMALL (3x2 bits)

- Environnement de test :



Analyse de l'architecture (4/10)

SP-SMALL (3x2 bits)

- Analyse avec l'outil Uppaal :

- 92 propriétés TCTL décrivant le comportement des signaux du circuit + 30 propriétés TCTL décrivant l'évaluation des expressions des gardes des processus

$q_0 : [253, 304] \uparrow$  $AG ((t \geq 0 \wedge t < 253) \Rightarrow q_0 = 0) \wedge (t > 304 \Rightarrow q_0 = 1)$

- Vérification formelle avec Uppaal de ces propriétés

- Les 120 propriétés sont vérifiées en 3h en tout, tCK->Q est évaluée en 10 mn
- les intervalles de propagation des signaux dans les portes sont trop grands, ce qui conduit à un temps de réponse global tCK->Q > taaw

$q_0 : [253, 304] \uparrow$  $q_0 : 276 \uparrow$

- divergence des temps
- fausses configurations



supprimer ces configurations

Analyse de l'architecture (5/10)

SP-SMALL (3x2 bits)

- Suppression des fausses configurations dans les fichiers timings
 - compatibilité des signaux + compatibilité des configurations d'entrées/sortie
- Fichiers de timings :
 - *Un fichier timing $T(S)$ est associé à une porte ou latch de sortie S et de fonctionnalité $F(E)$, où $E = \langle E_1, \dots, E_n \rangle$ représente la liste des signaux d'entrées (sensibilité) de la porte ou du latch.*
 - *Un fichier timing $T(S)$ contient une séquence de lignes.*
 - *Une ligne lg du fichier timing $T(S)$, est un triplet $\langle e, s, t \rangle$, tel que :*
 - *$e = \langle e_1, \dots, e_n \rangle$, est une configuration d'entrée avec un seul front ($\forall i e_i \in \{0, 1, 'D', 'U'\}$ et $\exists ! j e_j \in \{'D', 'U'\}$).*
 - *s est la valeur du front de sortie $s \in \{'D', 'U'\}$.*
 - *t représente le temps de propagation de la porte ou du latch.*

Analyse de l'architecture (6/10)

SP-SMALL (3x2 bits)

- Exemple d'un fichier temporel :

- $s \leq \text{not}(e_1 \text{ or } e_2)$

e_1	e_2	s	time
0	u	d	1ps
0	d	u	1ps
u	0	d	2ps
d	0	u	3ps



e_1	e_2	e_3	s	time
0	u	0	d	1ps
0	d	1	u	1ps
u	0	0	d	2ps
d	0	1	u	3ps
0	u	0	u	1ps
0	d	1	d	1ps
1	d	0	u	1ps
1	u	1	d	1ps
0	0	d	1	1ps
0	1	u	0	1ps
1	0	d	0	1ps
1	1	u	1	1ps

Analyse de l'architecture (7/10)

SP-SMALL (3x2 bits)

- Compatibilité des signaux :
 - *Algorithme 1 (fichier temporel T(S))*
 - pour chaque nom signal st apparue dans le fichier temporel T(S) faire*
 - si st ∉ dans VHDL alors*
 - supprimer la colonne*
 - pour chaque ligne lg du fichier T faire*
 - si le front de sortie induit par un front sur une colonne supprimée alors*
 - supprimer la ligne lg.*
 - Absence d'un signal d'entrée sg dans le fichier T(S) ?
 - prise en compte les signaux du fichier T(S) qui ont une relation directe avec ce signal sg (on les supprime pas).

Analyse de l'architecture (8/10)

SP-SMALL (3x2 bits)

- Compatibilité des configurations d'entrées/sortie :

- $BS : \{'D', 'U'\} \longrightarrow \{0, 1\}, BS(x) = \begin{cases} 1 & \text{si } x = 'U' \\ 0 & \text{si } x = 'D' \end{cases}$

- *Algorithme2 (fichier temporel T(S))*

pour chaque ligne lg du fichier T faire

rechercher $j / e_j \in \{'D', 'U'\}$.

remplacer e par $e' = \langle e_i, e_j = 1 - BS(e_j) \rangle$ ('D' par 1 & 'U' par 0*)*

remplacer e par $e'' = \langle e_i, e_j = BS(e_j) \rangle$ ('D' par 0 & 'U' par 1*)*

*si $f(e') \neq 1 - BS(s)$ alors (*$f(e') \neq 0$ si $s = 'U'$ & $f(e') \neq 1$ si $s = 'D'$)**

supprimer la ligne lg.

*sinon si $f(e'') \neq BS(s)$ alors (*$f(e') \neq 0$ si $s = 'D'$ & $f(e') \neq 1$ si $s = 'U'$)**

supprimer la ligne lg.

Analyse de l'architecture (9/10)

SP-SMALL (3x2 bits)

- Exemples de délais réduits :
 - En appliquant Algo1 + Algo2 :*

signal	ancien délai	nouveau délai
<i>v_17_12_10_ext_cs_n</i>	<i>[5, 21]dn, [1, 29]up</i>	<i>[6, 6]dn, [1, 9]up</i>
<i>v_18_e_data_delay_h_inv</i>	<i>[10, 11]dn, [11, 31]up</i>	<i>[10, 11]dn, [17, 17]up</i>
<i>v_18_f_data_delay_h_inv</i>	<i>[10, 11]dn, [10, 30]up</i>	<i>[10, 11]dn, [17, 17]up</i>
<i>v_17_12_10_net180</i>	<i>[5, 22]dn, [1, 31]up</i>	<i>[6, 6]dn, [1, 9]up</i>

- En appliquant que Algo2 (suffisant) :*

signal	ancien délai	nouveau délai
<i>b0</i>	<i>[0, 38]dn, [1, 39]up</i>	<i>[5, 38]dn, [1, 37]up</i>
<i>b1</i>	<i>[0, 8]dn, [0, 39]up</i>	<i>[0, 8]dn, [0, 39]up</i>
<i>b_0</i>	<i>[0, 39]dn, [1, 39]up</i>	<i>[5, 39]dn, [1, 37]up</i>
<i>b_1</i>	<i>[0, 39]dn, [1, 39]up</i>	<i>[5, 39]dn, [1, 37]up</i>

Analyse de l'architecture (10/10)

SP-SMALL (3x2 bits)

- Analyse de la description avec intervalles des délais réduits par Algo 1 et 2.

$$q_0 : [253, 304] \uparrow \quad \longrightarrow \quad q_0 : [258, 298] \uparrow$$

- Cette réduction est encore insuffisante ($taaw = 276$).
- Quelques pistes d'améliorations ?
 - Intégrer les temps (exacts) dans le modèle VHDL :
 - délais ponctuels associés à chaque configuration et non plus intervalle. A l'exécution, seules les configurations fonctionnelles réellement utilisées sont prises en compte.
Modèle VHDL très gros (contient tous les délais pour toutes les configurations)

Merci pour votre attention

Bibliographie

- Voir le site web de l'outil VHDL2TA :
 - www.lsv.ens-cachan.fr/~encrenaz/valmem/vhdl2hytech/index.html