

VALMEM

Traduction automatique d'une description
comportementale VHDL + annotations
temporelles en automates temporisés

A. Bara, E. Encrenaz

LIP6

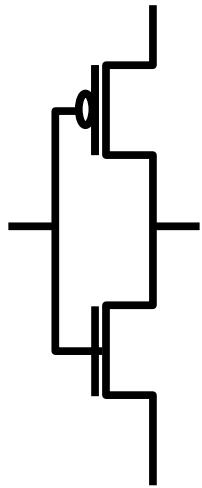
Situation dans la méthode

```
a <= b or c xor d;  
process (a,b)  
begin  
  if a then...
```

VHDL comportemental

Abstraction fonctionnelle

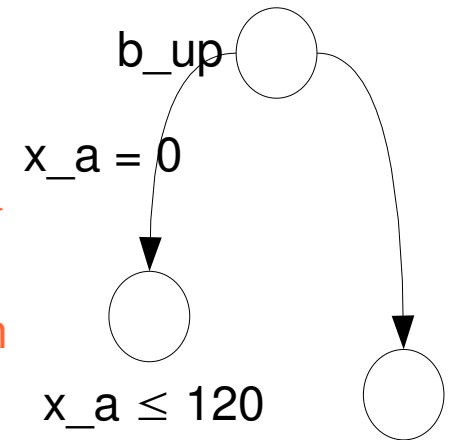
Caractérisation temporelle



Transistor netlist



Traduction



Automates temporisés (Hytech)

```
b_up c_0 d_0 a_up 120  
b_dn c_1 d_1 a_dn 100
```

Annotations temporelles

Syntaxe VHDL acceptée

- Architecture : signaux in /out de type bit
- Entity : VHDL comportemental
 - Affectations concurrentes
 - Processus (forme restreinte)
 - (Pas d'instanciation de composants internes)

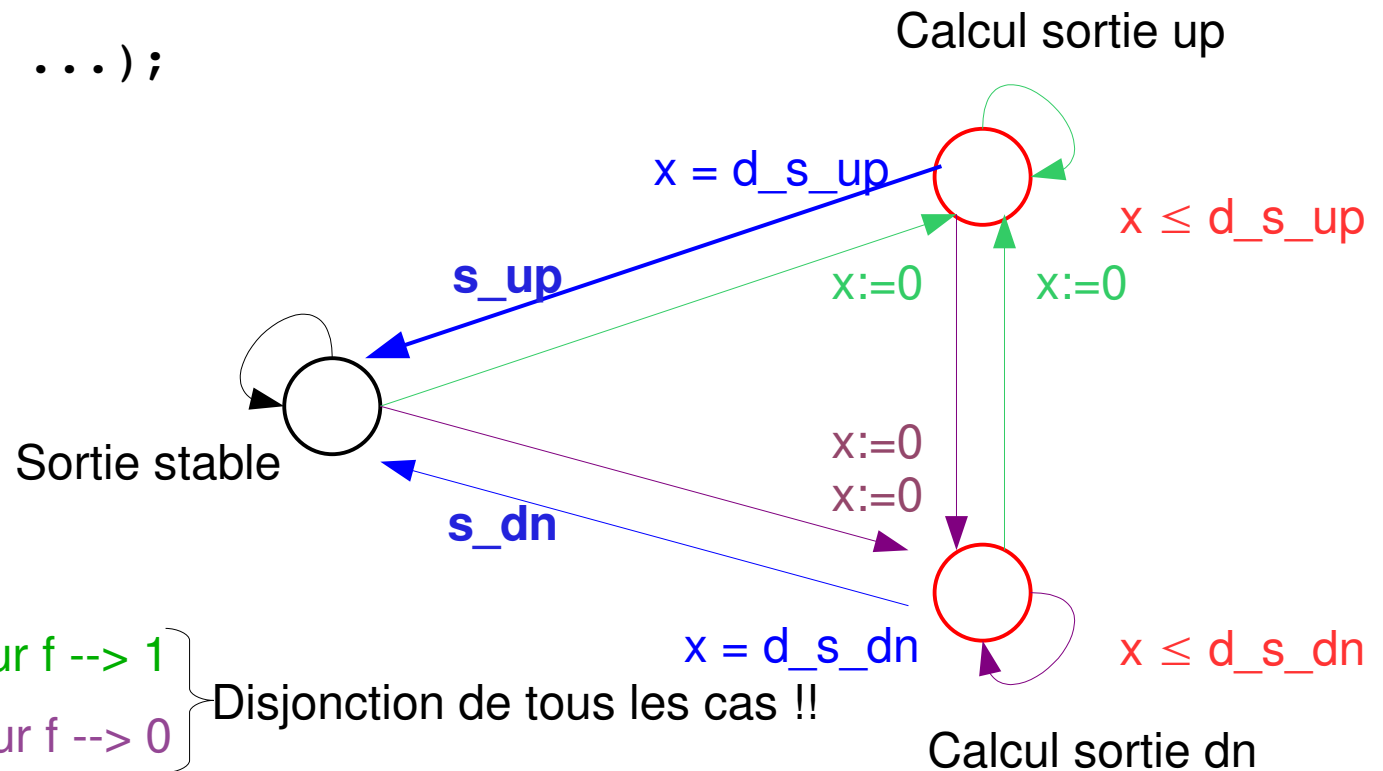
Modèle d'automates temporisés

- Réseau d'automates concurrents
- Horloges
- Variables discrètes
 - Stocke la valeur courante d'un signal
 - Permet la détection des fronts ('event / 'transaction)
- Paramètres
 - Délais de propagation des fronts

Principes de traduction (1)

- 1 affectation concurrente : 1 automate temporisé

$s \leq f(a, b, c, \dots);$



Principes de traduction (2)

- 1 process : 1 à 3 automates temporisés

```
process (a,b,c, ...)  
begin  
  if a = 1 then  
    s <= b;  
  elsif e2 then  
    s <= not b;  
end
```

Automate condition 1 (optionnel)

Automate condition 2 (optionnel)

Automate affectation sortie

Insertion des informations temporelles

- Récupération des délais dans le fichier temporel
- Affectation aux paramètres des TA
 - Deux paramètres de délais pour chaque signal
 - Front montant / front descendant
 - Intervalle des valeurs de délai pour toutes les configuration d'entrées et front induisant un front sur le signal de sortie

Modélisation de l'environnement

- Un (réseau d') automate représentant l'évolution des signaux d'entrée est ajouté.
- **IMPORTANT** : la vérification / synthèse de paramètres est valide pour les scénarii couverts par l'environnement
 - Méthode exhaustive en temps pour un ensemble de scénarii prédéfini

Réductions du modèle

- Simplification des chaînes d'inverseurs
- Elimination des automates conditions lorsque les conditions des processus sont réduites à “signal = constante”

Exemples traités

- 3 programmes VHDL
 - Exp1 : quelques portes et registres
 - BLUEB_LSV : abstraction manuelle de SPSMALL lors du projet BLUEBERRIES (1 mot de 1 bit)
 - SPSMALL_1*2 : abstraction fonctionnelle de SPSMALL (2 mots de 1 bit) par LIP6
- Données temporelles non intégrées (tous les délais sont à 1)

Exemple -- Exp1

```
ENTITY exp1 IS
```

```
  PORT (
```

```
    Q_0 : out BIT;
```

```
    CK : in BIT;
```

```
    CSN : in BIT;
```

```
    D_0 : in BIT;
```

```
    vdd : in BIT;
```

```
    gnd : in BIT
```

```
  );
```

```
END exp1;
```

```
ARCHITECTURE RTL OF exp1 IS
```

```
  SIGNAL v_18_E_net81 : BIT;
```

```
  ...
```

```
BEGIN
```

```
  v_18_E_net81 <= not (D_0);
```

```
  v_18_E_net85 <= not (v_18_E_net81);
```

```
  v_18_E_net83 <= not (v_18_E_net85);
```

```
  v_18_E_data_delay_H <= not (v_18_E_net83);
```

VALMEM – réunion du 12/06/2009

```
  v_17_12_10_net13 <= (not(CK) or (v_17_12_10_ext_cs_N and v_18_E_net83));
```

Exemple – Exp1 (cont'd)

```
v_17_12_10_clk_sig_H <= not (v_17_12_10_net13);  
v_17_12_10_net41 <= not (v_17_12_10_clk_sig_H);  
CLK_H <= not (v_17_12_10_net41);  
v_18_E_clk_local_L <= not (CLK_H);  
v_18_E_clk_local_H <= not (v_18_E_clk_local_L);
```

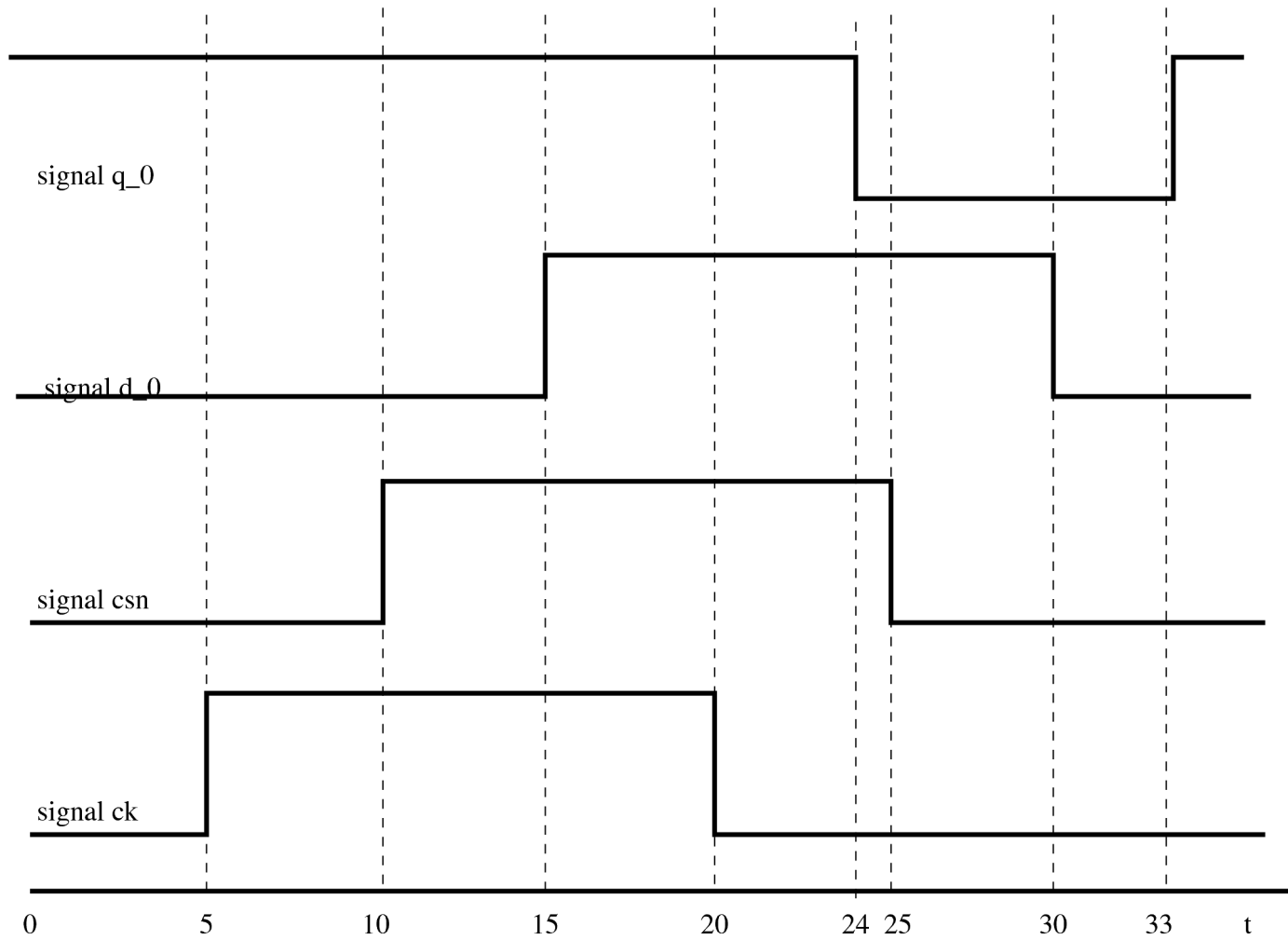
```
v_17_12_10_net96 <= not (CSN);  
v_17_12_10_ext_cs_H <= not (v_17_12_10_net96);
```

```
REG10: PROCESS (CK, v_17_12_10_ext_cs_H)  
BEGIN  
  IF CK = '0' THEN  
    v_17_12_10_ext_cs_N <= not (v_17_12_10_ext_cs_H);  
  END IF;  
END PROCESS;  
REG12: PROCESS (v_18_E_clk_local_L, v_18_E_data_delay_H)  
BEGIN  
  IF v_18_E_clk_local_L = '1' THEN  
    v_18_E_data_delay_H_inv <= not (v_18_E_data_delay_H);  
  END IF;  
END PROCESS;  
END;
```

Traduction de Exp1

- Sans réduction :
 - 16 automates + 1 (env)
 - pb mémoire hytech
- Avec réductions :
 - Réductions chaines de not + gardes des processus
 - 7 automates + 1 (env) / 10 var / 18 paramètres
 - Calcul de traces d'exécution (chronogrammes) pour différents environnements (avec tous les délais à 1)

Exp1 -- env1



Exemple – BLUEB_LSV

- Architecture complète abstraite dans le projet BLUEBERRIES (1 mot de 1 bit) / simplifications manuelles
- Traduction (avec optimisations) :
 - 28 automates / 28 horloges / 32 var / 62 paramètres
 - Actuellement non validée...

Exemple – SPSMALL1*2

- Architecture complète abstraite automatiquement par LIP6 dans VALMEM
- Adaptations préalables à la traduction :
 - bit_vector --> bit
 - retrait de 8 processus (monstrueux)
 - signaux non connectés supprimés
- Traductions :
 - Sans opt : 90 automates / 90 horloges / 112 var / 178 paramètres
 - Avec opt : 56 automates / 56 horloges / 68 var / 134 paramètres
 - Actuellement non validées ...

Limitations actuelles

- Syntaxe VHDL
 - Type bit uniquement (pas de bit_vector / std_logic)
 - Process très limités
- Modèle à 2 délais uniquement
- Combinatoire !!!
 - Taille des automates (combinatoire des transitions)
 - Produit d'automates avant l'analyse : Outil hytech limité à une dizaine d'automates
 - Produit à la volée : UPPAAL / outil ad-hoc