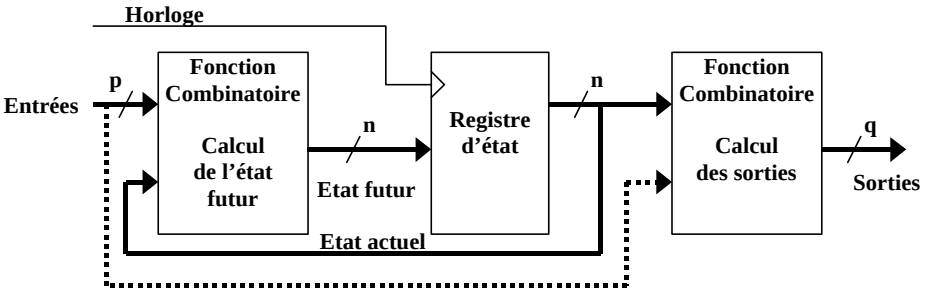
1 LES FONCTIONS SÉQUENTIELLES

La différence essentielle entre une fonction combinatoire et une fonction séquentielle réside dans la capacité de cette dernière de «se souvenir » des événements antérieurs : une même combinaison des entrées, à un certain instant, pourra avoir des effets différents suivant les valeurs des combinaisons précédentes de ces mêmes entrées.

Pour traduire cet effet de mémoire on introduit la notion d'**état interne** de la fonction, l'action des entrées est alors de provoquer d'éventuels changements d'**état**, la situation qui suit le changement de l'une des **entrées** dépend de l'état **précédent**. Si le nouvel état est différent du précédent on dit qu'il y a eu une **transition**.

1.1 LES MACHINES SYNCHRONES À NOMBRE FINIS D'ÉTATS



Une machine à états (M.A.E.) en anglais Finite State Machine (F.S.M.) est un système dynamique, qui peut se trouver, à chaque instant, dans une position parmi un nombre fini de positions possibles. Elle parcourt des cycles, en changeant éventuellement d'état lors des transitions actives de l'horloge. L'architecture générale d'une machine à état est présentée ci-dessous.

1.1.1 HORLOGE, REGISTRE D'ÉTAT ET TRANSITIONS

Le registre d'état, piloté par son horloge, constitue le cœur d'une machine à états. Les autres blocs fonctionnels sont à son service.

Il est constitué de n bascules synchrones. Son contenu représente **l'état actuel** de la machine. Il s'agit d'un nombre codé en binaire sur n bits. L'entrée du registre d'état constitue **l'état futur**, celui qui sera chargé lors de la prochaine transition active de l'horloge. Le registre d'état est la **mémoire** de la machine.

La taille du registre d'état fixe le nombre d'états accessibles. Si **n** est le nombre de bascules, le nombre d'états $\mathbf{N = 2^n}$.

1.1.1.2 *L'HORLOGE*

Le rôle de l'horloge est de fixer les instants où les transitions entre états sont prises en compte. Entre deux fronts consécutifs de l'horloge, la machine est figée en position mémoire.

1.1.2 LES DIFFÉRENTES ARCHITECTURES

La figure 2 présente l'architecture générale d'une machine à états. Suivant la façon dont les sorties dépendent des états et des commandes, on distingue deux types de machines à états : les machines de **Moore** et les machines de **Mealy**. Dans les premières les **sorties** ne dépendent que de **l'état actuel** (la liaison en trait interrompue est absente), pour les secondes les **sorties** dépendent de **l'état actuel et des entrées** (la liaison en trait interrompu est présente).

2 OUTILS DE DESCRIPTION

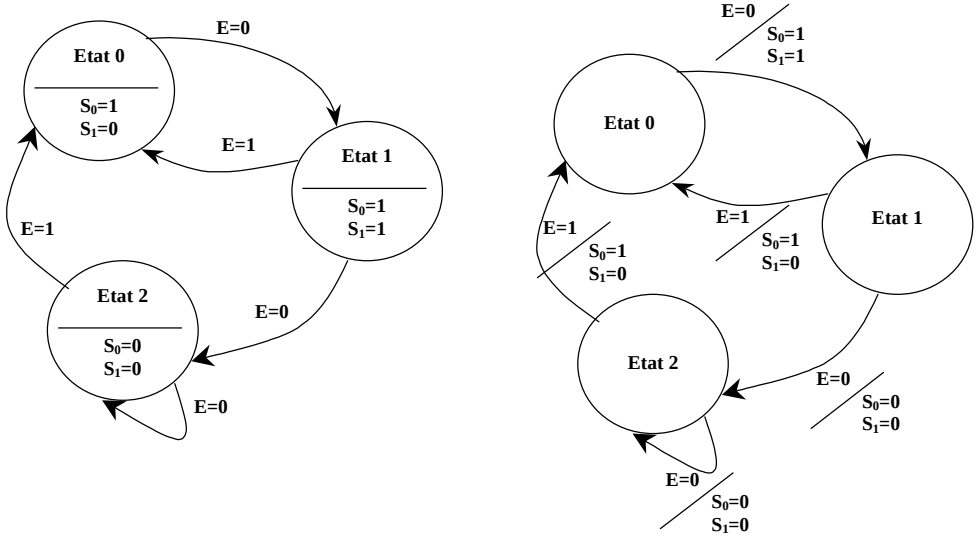
Si l'outil d'analyse et de synthèse des fonctions combinatoires est la table de vérité, le diagramme de transition constitue l'outil privilégié pour l'analyse et la synthèse des fonctions séquentielles.

2.1 LE DIAGRAMME DE TRANSITION

- On associe à chaque valeur possible du registre d'état, une case.
- L'évolution du système est représentée par des flèches représentant les transitions.
- Pour qu'une transition soit activée il faut que les trois conditions suivantes soient vérifiées :

1. Le système se trouve dans l'état «source » considéré
2. La condition de réalisation sur les entrées est vraie
3. Un front actif de l'horloge survient

Pour les machines de **Moore** les sorties évoluent **après l'activation de la transition**. Les valeurs des sorties seront représentées dans les cases du diagramme.



Pour les machines de **Mealy** les sorties évoluent **après l'évolution des entrées**. Les valeurs des sorties seront représentées sur les flèches du diagramme.

2.2 DU DIAGRAMME AUX ÉQUATIONS

Le passage du diagramme de transition aux équations est indispensable si on veut synthétiser la machine à états avec des circuits standards. L'outil permettant le passage explicite du diagramme aux équations est **la table de transitions et d'états**.

C'est une table de vérité constituée :

en entrée de l'état **actuel** du registre d'état
des **entrées** de la machine à états

en sortie de l'état **futur** du registre d'état
des **sorties** de la machine à états

Les équations des sorties du registre d'état sont ensuite adaptées au type de bascules utilisées.

Comme pour les fonctions combinatoires la complexité du problème croit de façon exponentielle avec le nombre d'états et le nombre d'entrées.

2.3 DESCRIPTION VHDL

Le langage VHDL offre de multiples possibilités pour traduire le fonctionnement d'une machine à états. Nous ne nous intéresserons qu'à la description *comportementale*. D'une façon générale seules seront envisagées les fonctions séquentielles synchrones.

Le processus qui décrit le fonctionnement d'une machine à états comporte deux structures imbriquées : le traitement des commandes et le traitement de l'état de départ de chaque transitions.

- Les commandes se prêtent bien à une modélisation par des structures hiérarchiques **if … elsif … else … end if**.
- Les états se prêtent bien à une modélisation en terme d'aiguillage par les structures **case … when … when others … end case**.

Le registre d'état est matérialisé par deux éléments :

1) Un signal interne de type **bit_vector** ou **integer** déclaré de façon à être codé sur n chiffres binaires.
2) Un processus, activé par le *seul signal d'horloge* qui est l'unique endroit où le signal d'état subit une affectation.

3 LA DÉMARCHE DE CONCEPTION POUR IMPLANTER UN PRGRAMME DANS UN CPLD

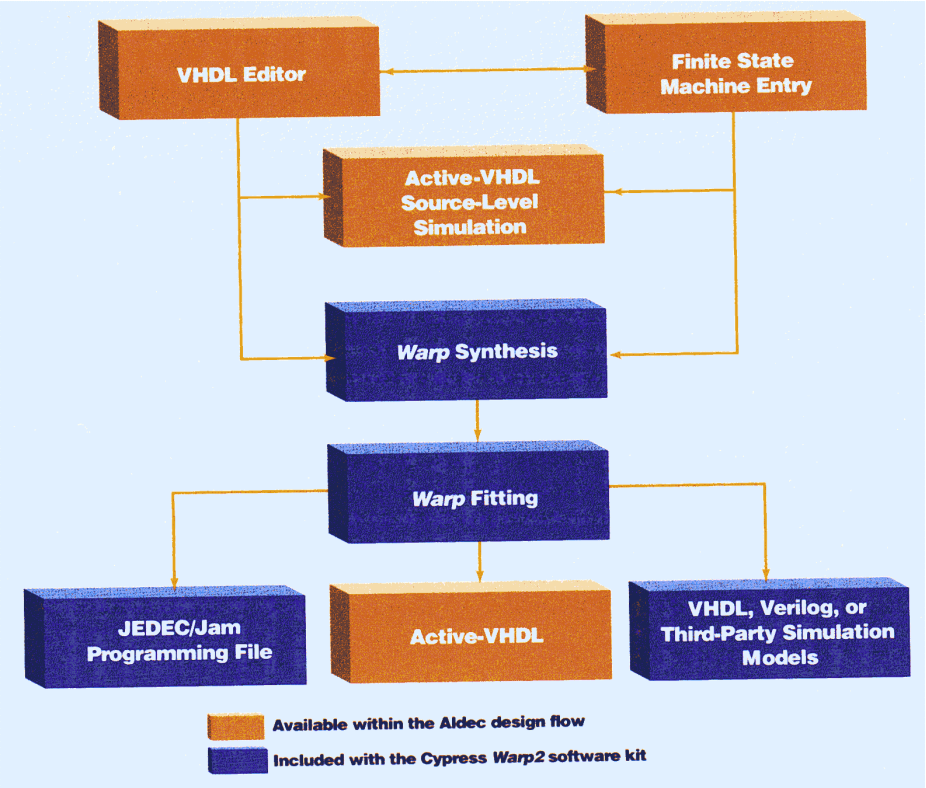


Figure 1 : La démarche à effectuer pour réaliser un CPLD

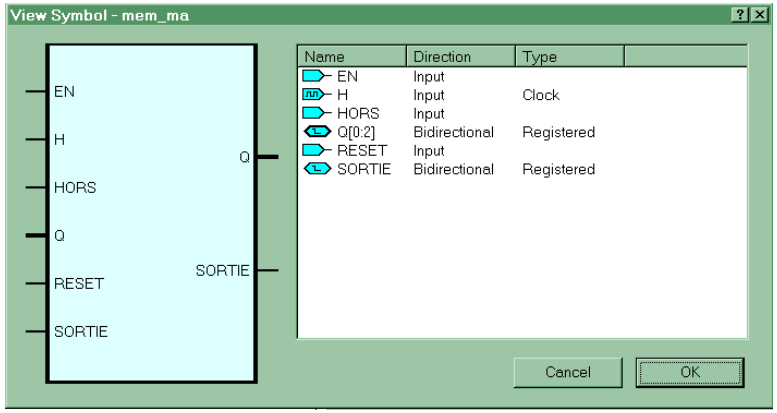


Figure 2 : Définition des Entrées et Sorties

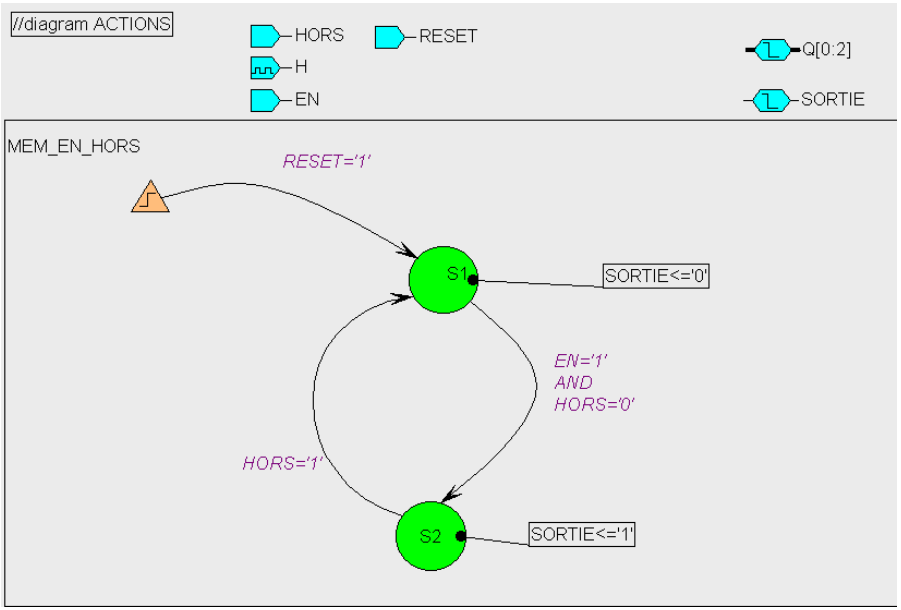


Figure 3 : Machine Mémoire avec priorité à la mise hors mémoire

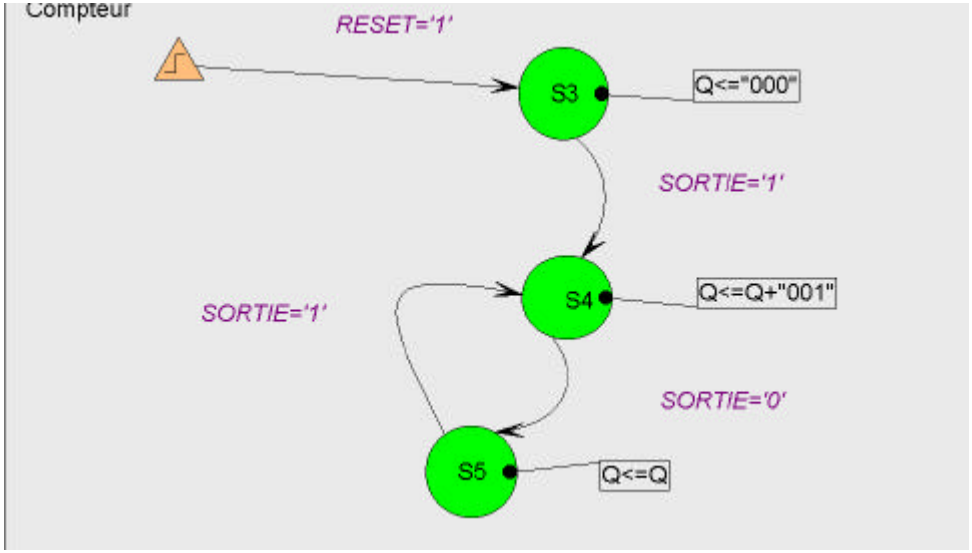


Figure 4 : Le compteur

3.2 CODAGE VHDL

```
--
--  File: D:\Mon_travail\iufm\VHDL\cours_99\mem_ma.vhd
-- created: 11/19/99 12:17:01
--  from: 'D:\Mon_travail\iufm\VHDL\cours_99\mem_ma.asf'
--  by fsm2hdl - version: 2.0.1.45
--
library IEEE;
use IEEE.std_logic_1164.all;
library Cypress;
use Cypress.std_arith.all;
```

Figure 5 : Les librairies

```
entity mem_ma is
  port (EN: in STD_LOGIC;
        H: in STD_LOGIC;
        HORS: in STD_LOGIC;
        RESET: in STD_LOGIC;
        Q: inout STD_LOGIC_VECTOR (0 to 2);
        SORTIE: inout STD_LOGIC);
end;
```

Figure 6 : Entity

```
architecture mem_ma_arch of mem_ma is

...
...

end mem_ma_arch;
```

Figure 7 : Architecture

```
-- SYMBOLIC ENCODED state machine: Compteur
type Compteur_type is (S3, S4, S5);
signal Compteur: Compteur_type;

-- SYMBOLIC ENCODED state machine: MEM_EN_HORS
type MEM_EN_HORS_type is (S1, S2);
signal MEM_EN_HORS: MEM_EN_HORS_type;

begin
--concurrent signal assignments
--diagram ACTIONS;
```

Figure 8 : Les types

```
MEM_EN_HORS_machine: process (H)

begin

if H'event and H = '1' then
    if RESET='1' then
        MEM_EN_HORS <= S1;
        SORTIE<='0';
    else
    case MEM_EN_HORS is
        when S1 =>
            SORTIE<='0';
            if EN='1'
                AND
                HORS='0' then
                MEM_EN_HORS <= S2;
            end if;
        when S2 =>
            SORTIE<='1';
            if HORS='1' then
                MEM_EN_HORS <= S1;
            end if;
        when others =>
            null;
    end case;
    end if;
end if;
end process;
```

Figure 9 : Process de la Mémoire

```
begin

if H'event and H = '1' then
    if RESET='1' then
        Compteur <= S3;
        Q<="000";
    else
    case Compteur is
        when S3 =>
            Q<="000";
            if SORTIE='1' then
                Compteur <= S4;
            end if;
        when S4 =>
            Q<=Q+"001";
            if SORTIE='0' then
                Compteur <= S5;
            end if;
        when S5 =>
            Q<=Q;
            if SORTIE='1' then
                Compteur <= S4;
            end if;
        when others =>
            null;
    end case;
    end if;
end if;
end process;
```

Figure 10 : Process Compteur

3.3.1 DESIGN EQUATIONS (12:38:07)

```
q(0).D =
      /reset * compteurSBV_1.Q * /q(0).Q * q(1).Q * q(2).Q
    + /reset * compteurSBV_1.Q * q(0).Q * /q(2).Q
    + /reset * compteurSBV_1.Q * q(0).Q * /q(1).Q
    + /reset * compteurSBV_0.Q * q(0).Q
q(0).C =
      h
q(1).D =
      /reset * compteurSBV_1.Q * /q(1).Q * q(2).Q
    + /reset * compteurSBV_1.Q * q(1).Q * /q(2).Q
    + /reset * compteurSBV_0.Q * q(1).Q

q(1).C =
      h

q(2).D =
      /reset * compteurSBV_1.Q * /q(2).Q
    + /reset * compteurSBV_0.Q * q(2).Q

q(2).C =
      h

sortie.D =
      /reset * mem_en_horsSBV_0.Q

sortie.C =
      h

compteurSBV_0.D =
      /reset * compteurSBV_1.Q * /sortie.Q
    + /reset * compteurSBV_0.Q * /sortie.Q

compteurSBV_0.C =
      h

compteurSBV_1.D =
      /reset * sortie.Q

compteurSBV_1.C =
      h

mem_en_horsSBV_0.D =
      /reset * en * /hors
    + /reset * mem_en_horsSBV_0.Q * /hors

mem_en_horsSBV_0.C =
      h


Completed Successfully
```

Device: CY37032P44
Package: CY37032P44-125AC

```
                40  :  Not Used
                41  :  Not Used
                42  :  Not Used
                43  :  Not Used
                44  :  Not Used
                 1  :  Not Used
                 2  :  Not Used
                 3  :  Not Used
                 8  :  Not Used
                 9  :  Not Used
                10  :  Not Used
                11  :  Not Used
                12  :  Not Used
                13  :  Not Used
                14  :  Not Used
                15  :  Not Used
                18  :  q(0)
                19  :  Not Used
                20  :  Not Used
                21  :  Not Used
                22  :  Not Used
                23  :  Not Used
                24  :  sortie
                25  :  q(2)
                30  :  (compteurSBV_0)
                31  :  (mem_en_horsSBV_0)
                32  :  (compteurSBV_1)
                33  :  Not Used
                34  :  Not Used
                35  :  Not Used
                36  :  Not Used
                37  :  q(1)
                 4  :  h
                 7  :  hors
                26  :  en
                27  :  reset
                29  :  Not Used
```

Information: Macrocell Utilization.

Description	Used	Max
Dedicated Inputs	1	1
Clock/Inputs	3	4
I/O Macrocells	7	32
PIM Input Connects	10	78
	21 /	115 = 18 %

	Required	Max (Available)
CLOCK/LATCH ENABLE signals	1	6
Input REG/LATCH signals	0	5
Input PIN signals	4	5
Input PINs using I/O cells	0	0
Output PIN signals	7	32
Total PIN signals	11	37
Macrocells Used	7	32
Unique Product Terms	15	160

Messages:

```
-------------------------------------------------------------------------------
-
Signal Name | Delay Type  |   tmax    | Path Description
-------------------------------------------------------------------------------

reg::(compteurSBV_1)[32]
inp::reset
              tS              5.5 ns     1 pass
inp::sortie.Q
              tSCS            8.0 ns     1 pass
out::compteurSBV_1
              tCO             6.5 ns
-------------------------------------------------------------------------------
reg::q(1)[37]
inp::reset
              tS              5.5 ns     1 pass
inp::compteurSBV_1.Q
              tSCS            8.0 ns     1 pass
out::q(1)
              tCO             6.5 ns
-------------------------------------------------------------------------------

Worst Case Path Summary
-----------------------

                    tS = 5.5 ns for q(0).D
                  tSCS = 8.0 ns for q(0).D
                   tCO = 6.5 ns for q(0).C



Summary:
                 Error Count = 0      Warning Count = 0

Completed Successfully
```

```
This file was created on 11/19/1999 at 12:38:08
by C37KFIT.EXE    21/SEP/1998  [v4.02 ] 5.1 IR 14

 CY37032P44*
QP44*                Number of Pins*
QF15034*             Number of Fuses*
F1*                  Note: Default fuse setting 1*
G0*                  Note: Security bit Unprogrammed*
NOTE DEVICE CY37032P44*
NOTE PACKAGE CY37032P44-125AC*
NOTE PROPERTY BUS_HOLD ENABLE*
NOTE PINS  q(0):18 sortie:24 q(2):25 q(1):37 h:4 hors:7 en:26 reset:27 *
NOTE NODES compteurSBV_0:140 mem_en_horsSBV_0:141 compteurSBV_1:142 *
NOTE NODES *

High/Low Power*

L7501
11011111111111111111111111111111101111111111111111110111111111111111111111111
11111111111011011111111111111111101111111111111111101111111111111111111111111
10111111111101111111111111111111101111111111111111101111111111111111111111111
01111111111010111111111111111111101111111111111111101111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111

111111110
111111110
111111110
111111110
111111110
111111110
111111110
111111110
111111110
111111110
* Note: PIM Interconnect LAB 2*
L15001
0* Note: High/Low Power*
L15002
1100* Note: 33 *
L15006
1100* Note: 34 *
1100* Note: 37 *
L15022
0000* Note: 37 *
L15026
1* Note: BUS_HOLD_ENABLE/DISABLE*
L15027
000000*Note: STATIC FUSES *
CBADD*               Note: Fuse Checksum*
U1000000010000010 *  Note: User Device ID *
 CA4D
```

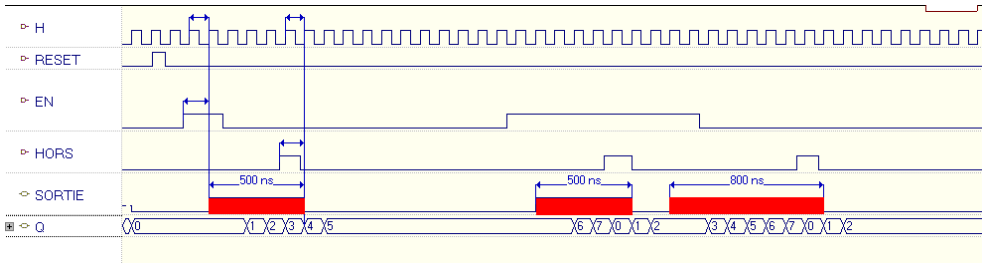


Figure 11 : Le comptage est effectif lorsque la mémoire est active

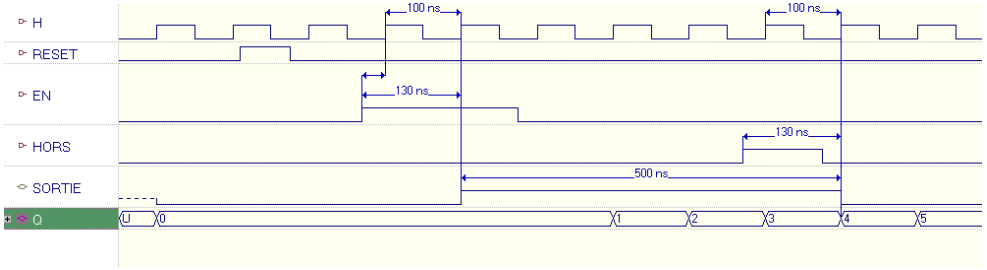


Figure 12 : Observation des retards

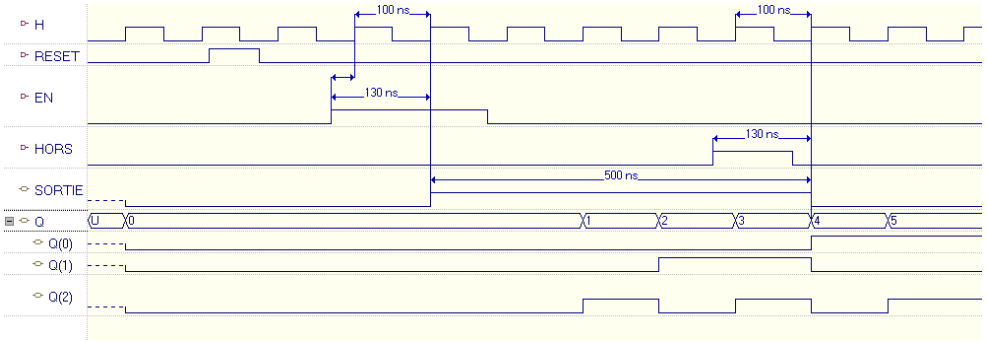


Figure 13 : Observation des retards et de l'état du bus

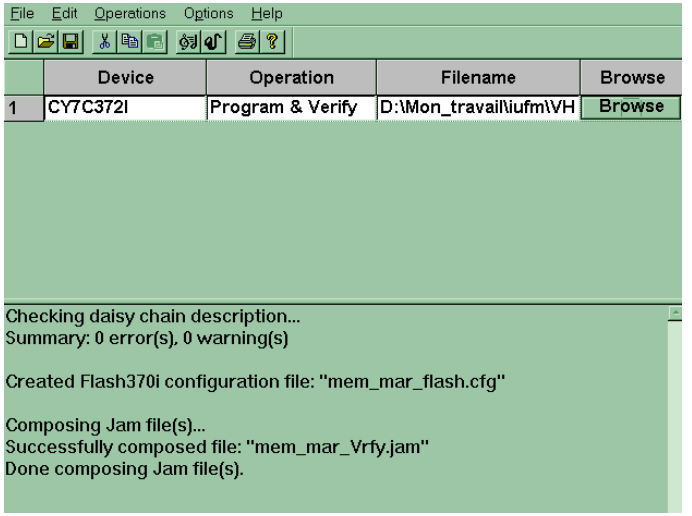


Figure 14 : Transfert du fichier dans la composant