

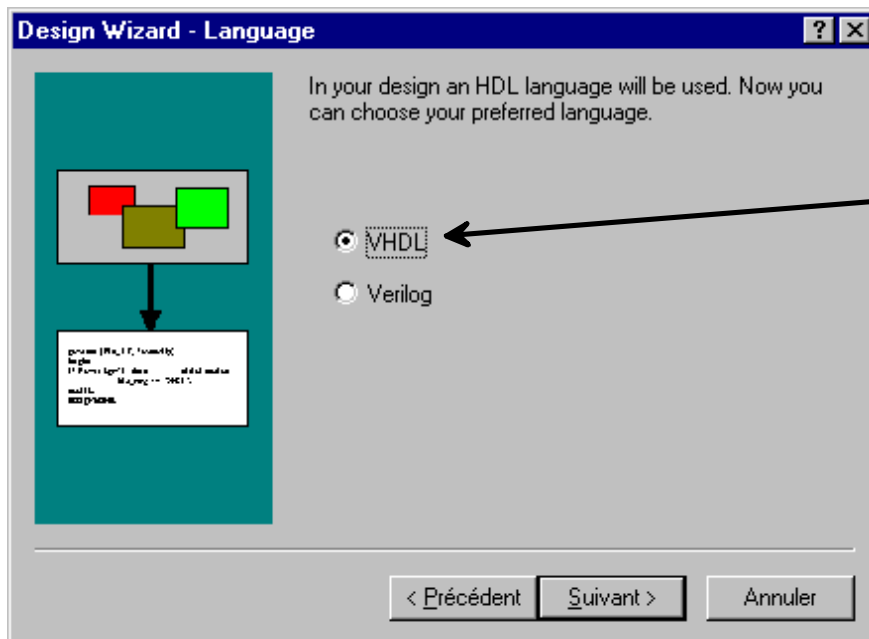
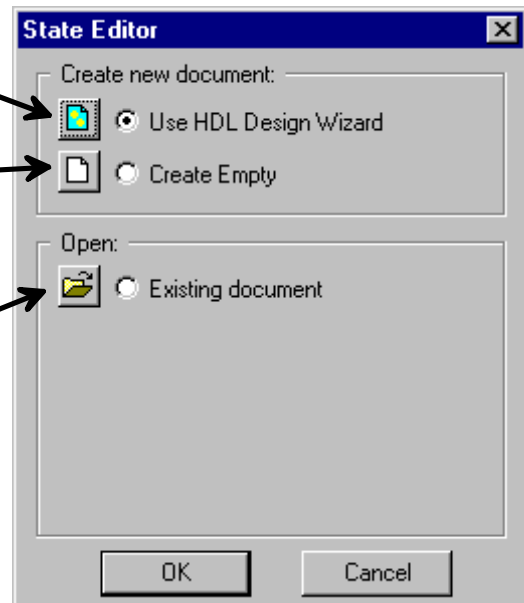
SYNTHESE DE MACHINES D'ETATS

OUVERTURE DE L'EDITEUR DE MACHINE D'ETATS :

Appel de la saisie assistée
des entrées sorties de la machine

Démarrage de la saisie sur une
feuille vierge

Ouverture d'un ancien document

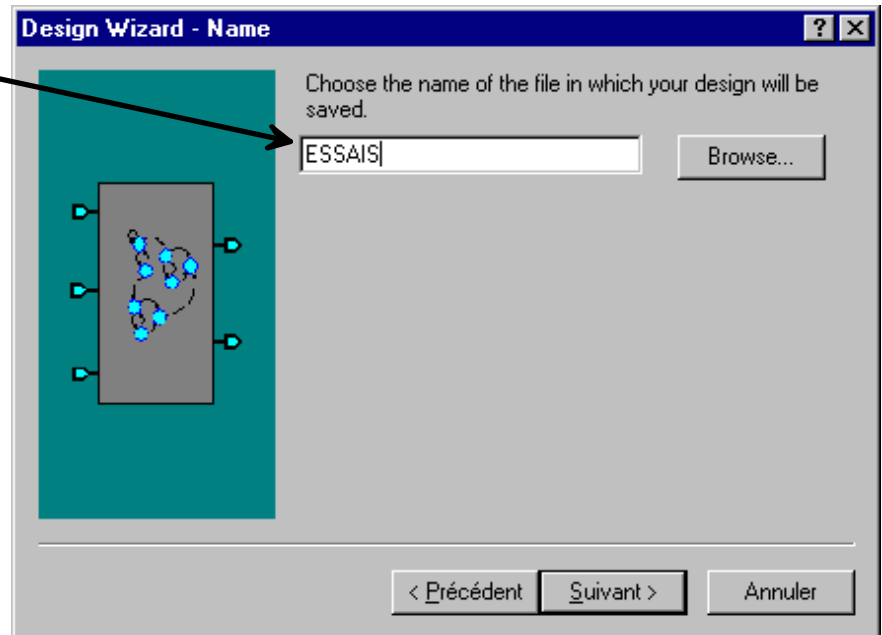


Choix du langage
cible utilisé

SYNTHESE DE MACHINES D'ETATS

SAISIE DU NOM DE LA MACHINE D'ETAT

On entre le nom de la machine d'état

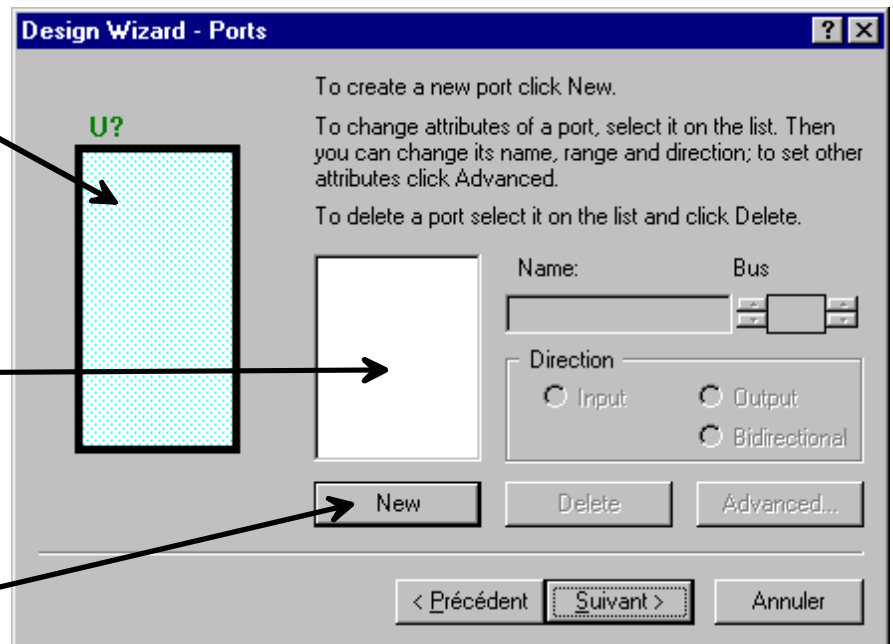


SAISIE DES PORTS

Les broches d'entrées et sorties sont dessinées ici

La liste des entrées sorties sera affichée dans cette fenêtre

Cliquer sur new pour entrer une nouvelle entrée ou une sortie

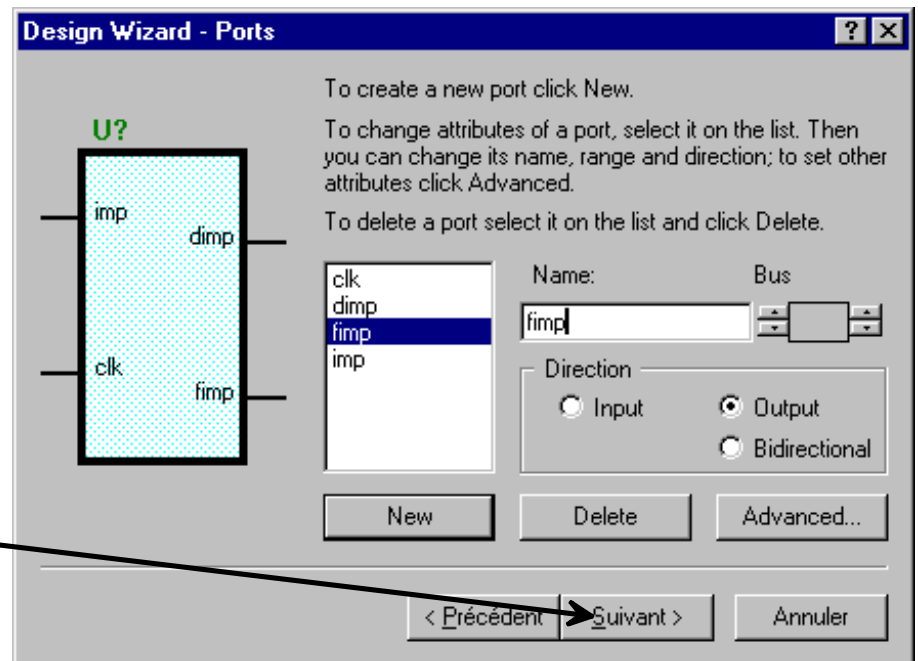


SYNTHESE DE MACHINES D'ETATS

FIN DE LA SAISIE DES PORTS

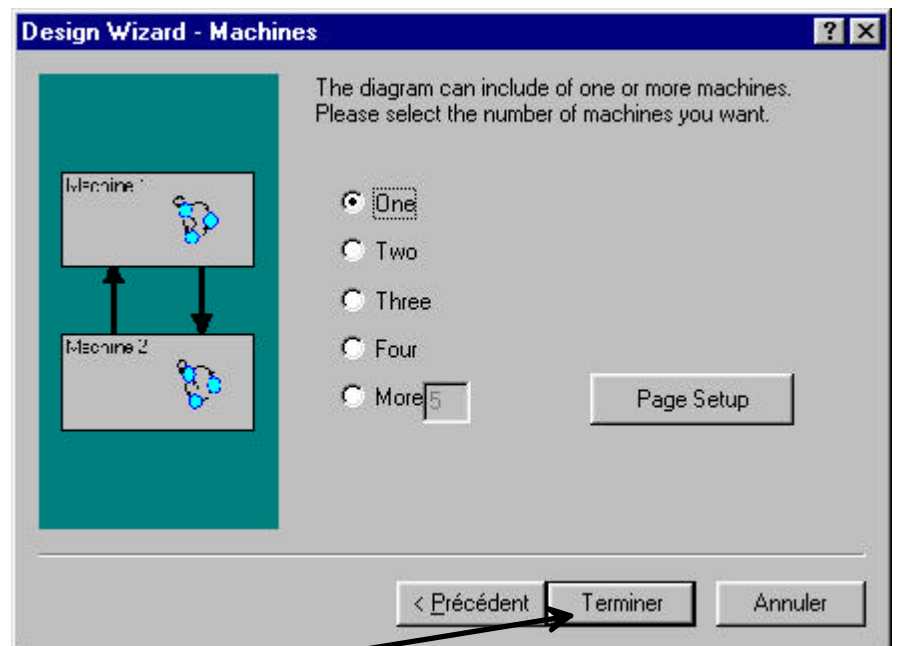
Les signaux d'entrées
et sorties sont saisis

Cliquer ici pour
continuer



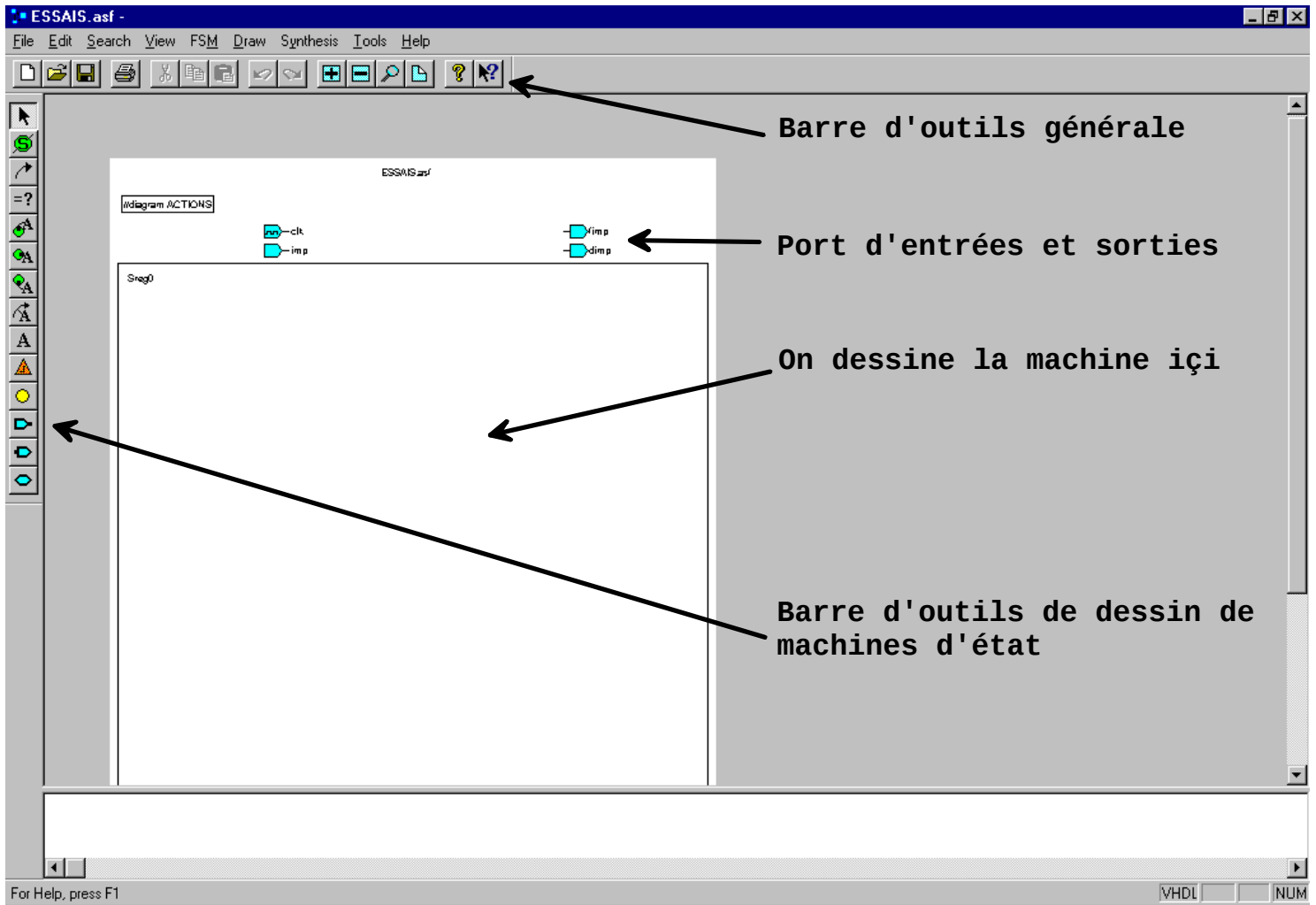
CHOIX DU NOMBRE DE MACHINES D'ETATS

Cliquer ici pour
continuer



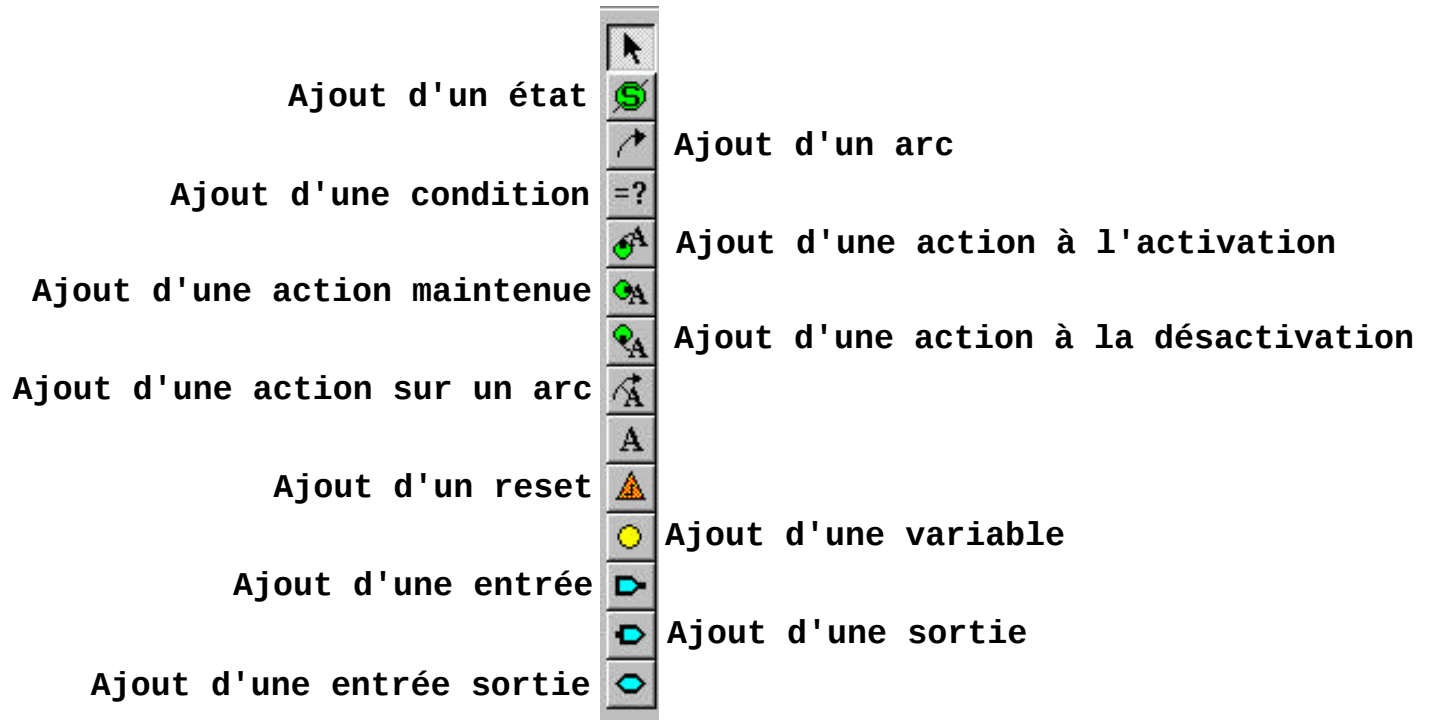
SYNTHESE DE MACHINES D'ETATS

SAISIE GRAPHIQUE DE LA MACHINE D'ETAT



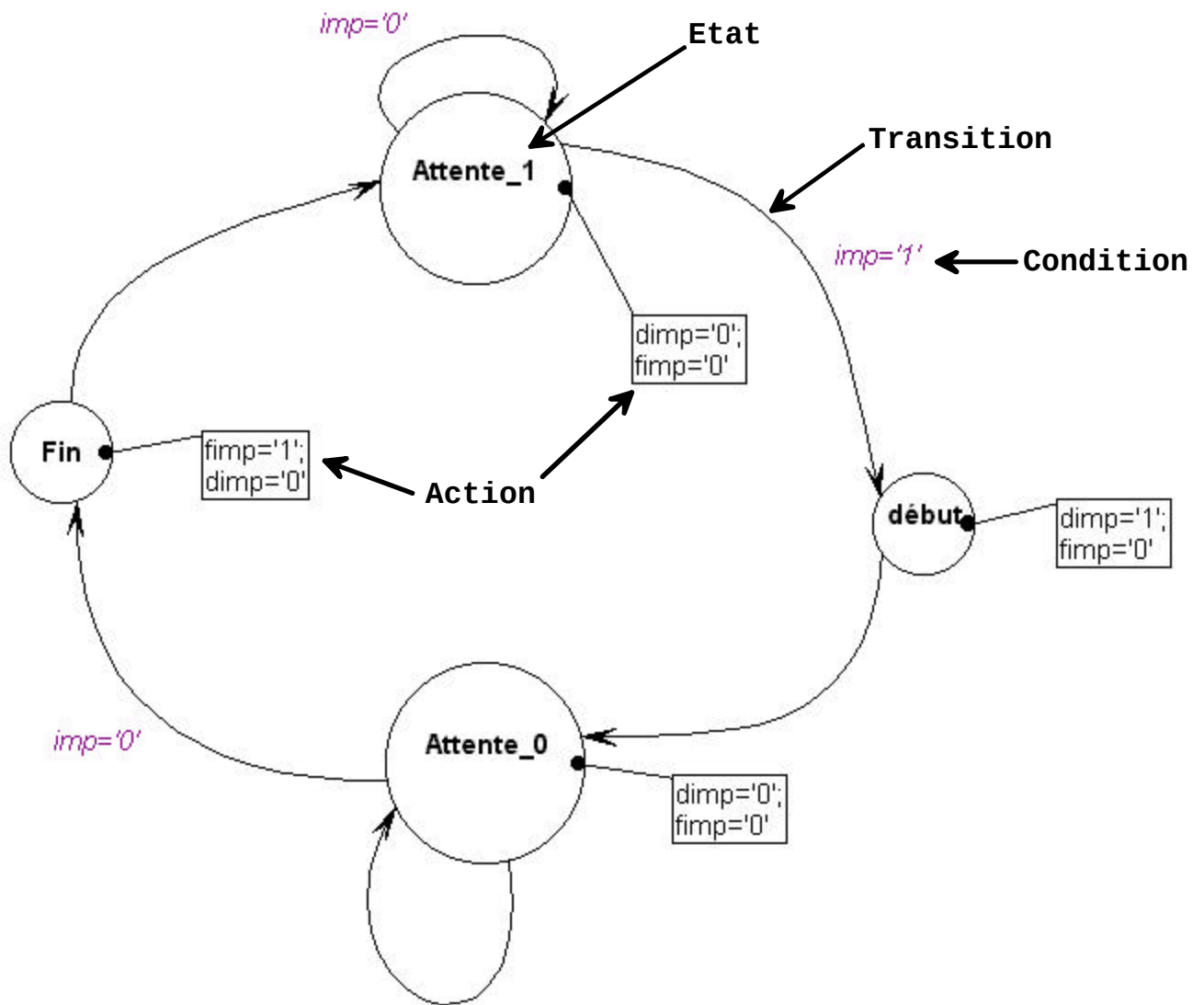
SYNTHESE DE MACHINES D'ETATS

LA BARRE DES OUTILS DE SAISIE GRAPHIQUE



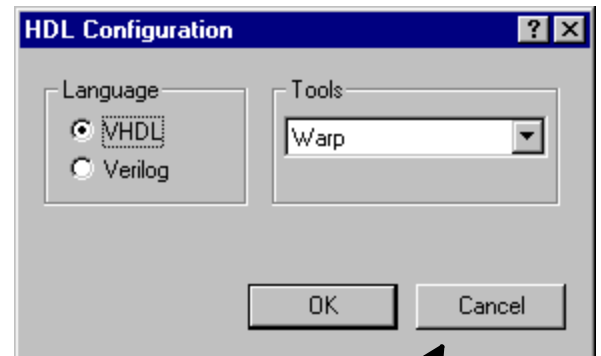
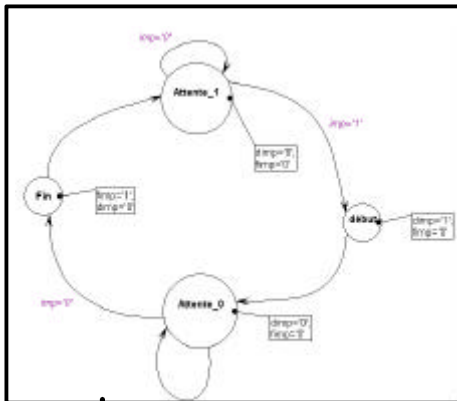
SYNTHESE DE MACHINES D'ETATS

SAISIE DE LA MACHINE D'ETAT CI-DESSOUS

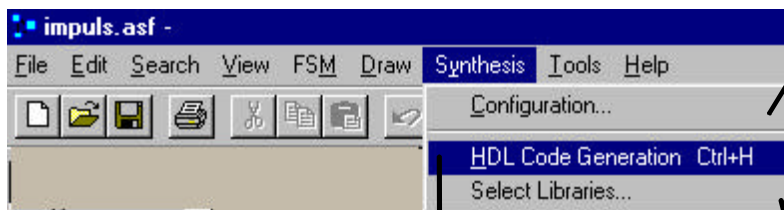


SYNTHESE DE MACHINES D'ETATS

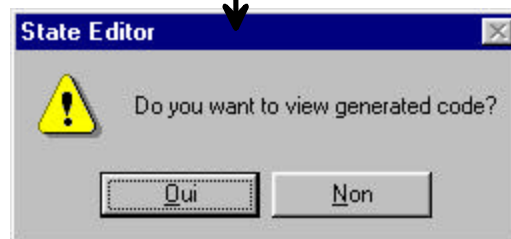
TRADUCTION DE LA MACHINE D'ETAT EN CODE VHDL



Choix du
langage cible



Synthèse



Ajout de nouvelles
bibliothèques
ci-besoin

```
library IEEE;
use IEEE.std_logic_1164.all;

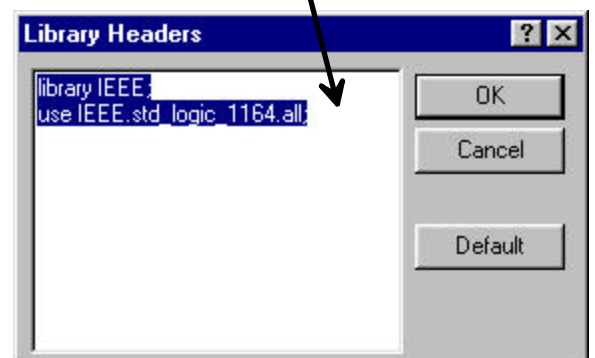
entity impuls is
  port (clk: in STD_LOGIC;
        imp: in STD_LOGIC;
        dimp: out STD_LOGIC;
        fimp: out STD_LOGIC);
end;

architecture impuls_arch of impuls is

  -- SYMBOLIC ENCODED state machine: Sreg0
  type Sreg0_type is (Attente_0, Attente_1, debut, Fin);
  signal Sreg0: Sreg0_type;

  begin
    --concurrent signal assignments
    --diagram ACTIONS;

    Sreg0_machine: process (clk)
    begin
      if clk'event and clk = '1' then
        case Sreg0 is
          when Attente_0 =>
            if imp='1' then
              Sreg0 <= Attente_1;
            elsif imp='0' then
              Sreg0 <= Fin;
            end if;
          when Attente_1 =>
            if imp='1' then
              Sreg0 <= Attente_0;
            elsif imp='0' then
              Sreg0 <= Fin;
            end if;
          when Fin =>
            Sreg0 <= Fin;
          when debut =>
            Sreg0 <= debut;
        end case;
      end if;
    end process;
  end;
```



SYNTHESE DE MACHINES D'ETATS

COMPILATION ET SYNTHESE AVEC GALAXY

