

# COURS VHDL suite

## Verification sur les tests :

**ASSERT** : permet d'afficher des messages a l'écran.

Syntaxe : ASSERT condition REPORT message <SECURITY>

exemples :

ASSERT FALSE REPORT « Tjs a lecran » SEVERITY note ;

REPORT « CHECKING... »

ASSERT s='1' REPORT « Sortie a 1 » SEVERITY warning ; --genere un warning

ASSERT s='1' REPORT « Sortie a 1 » SEVERITY failure ; --genere une erreur et arrete l'execution.

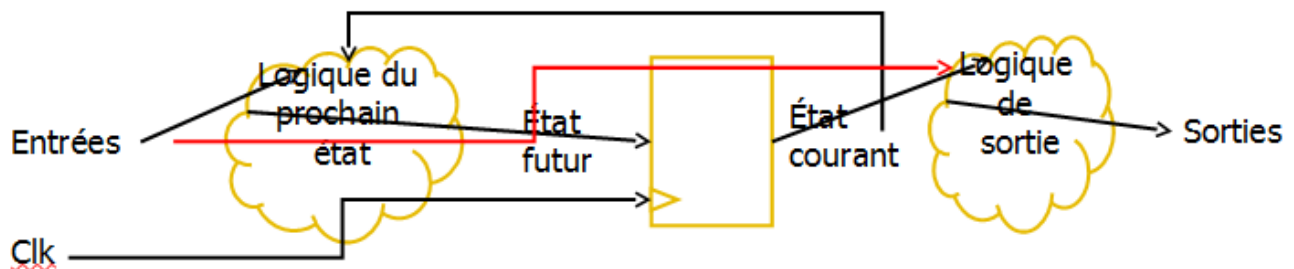
Ex : Additionneur d'1 bit et des 4 bits

# Machines a états

Une machine a états finis est un modèle mathématique de calcul utilisé dans de nombreux domaines.

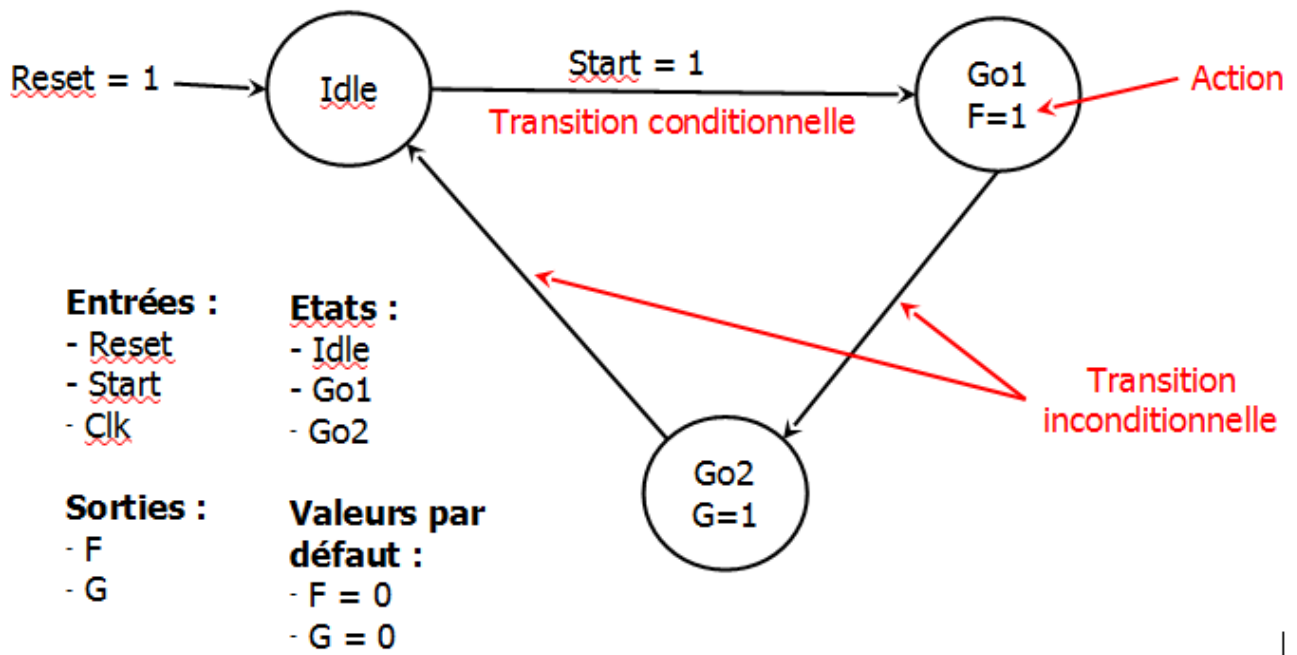
Une machine a états est dans un état a la fois et possède un nombre d'états fini. L'état ou se trouve s'appelle état courant, l'état qui suit état futur. Le passage d'un état courant a un état futur se réalise a une condition et s'appelle transition.

Ici on va étudier 2 genres de machines a états :

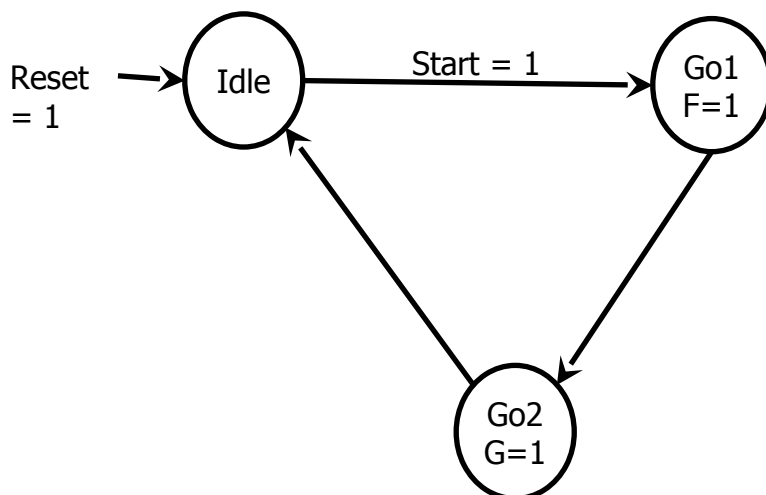


- Machine de Moore → les sorties ne dépendent que de l'état courant.
- Machine de Mealy → les sorties dépendent de l'état courant et des entrées.

(pour une Soit un circuit dont le diagramme d'états est le suivant :



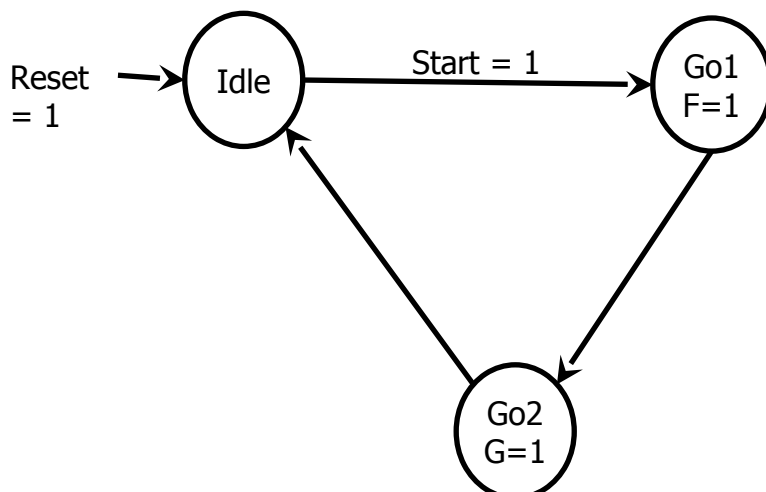
Si on veut utiliser le modèle d'une machine de Moore :



Et le code VHDL :

```
architecture FSM of EXEMPLE is
    type StateType is (Idle, Go1, Go2);
    signal State :: StateType;
begin
    process (Clk, Reset)
    if Reset = '1' then
        State <= Idle;
    elsif Rising_edge(Clk) then
        case State is
            when Idle =>
                if Start = '1' then
                    State <= Go1;
                end if;
            when Go1 =>
                State <= Go2;
            when Go2 =>
                State <= Idle;
            end case;
        end if;
    end process;
    output : process (State)
    begin
        F <= '0';
        G <= '0';
        if State = Go1 then
            F <= '1';
        elsif State = Go2 then
            G <= '1';
        end if;
    end process;
end architecture;
```

Et si veut utiliser une machine de Mealy :



Et le code VHDL :

```
architecture FSM of EXEMPLE is
    --> type StateType is (Idle, Go1, Go2);
    --> signal State : StateType;
begin
    process (Clk, Reset)
    begin
        --> if Reset = '1' then
        -->     --> State <= Idle;
        -->     --> G <= '0';
        -->     --> F <= '0';
        --> elsif Rising_edge(Clk) then
        -->     --> case State is
        -->     --> when Idle =>
        -->     -->     --> if Start = '1' then
        -->     -->     -->     --> State <= Go1;
        -->     -->     -->     --> F <= '1';
        -->     -->     --> end if;
        -->     --> when Go1 =>
        -->     -->     --> State <= Go2;
        -->     -->     --> G <= '1';
        -->     -->     --> F <= '0';
        -->     -->     --> when Go2 =>
        -->     -->     --> State <= Idle;
        -->     -->     --> G <= '0';
        -->     --> end case;
        --> end if;
    end process;
end architecture;
```