Architecture 2
Exercices

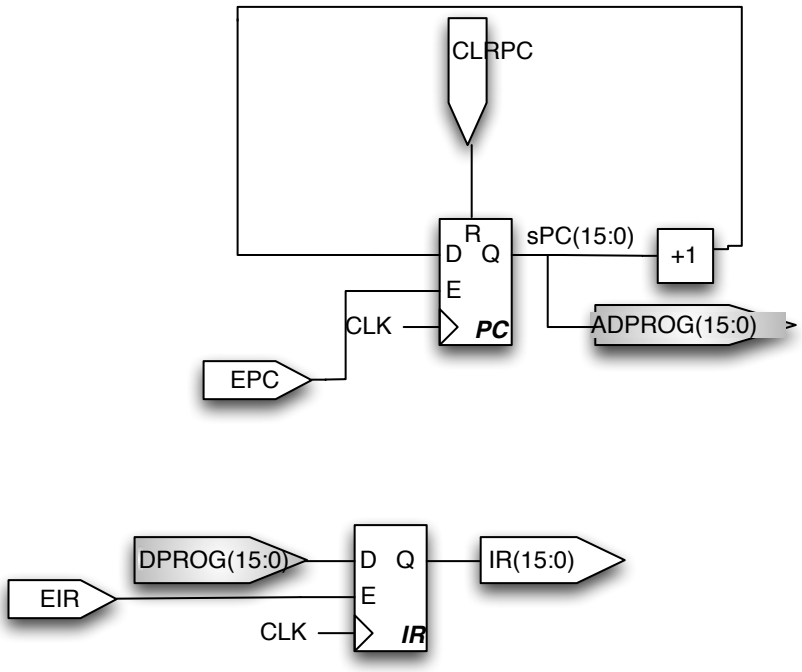
ENSIMAG 1A

Année scolaire 2009–2010

Ex. 1 : Processeur embryonnaire

Compléter les codes VHDL suivants en vu de munir ce processeur de l'instruction reset du cahier des charges.

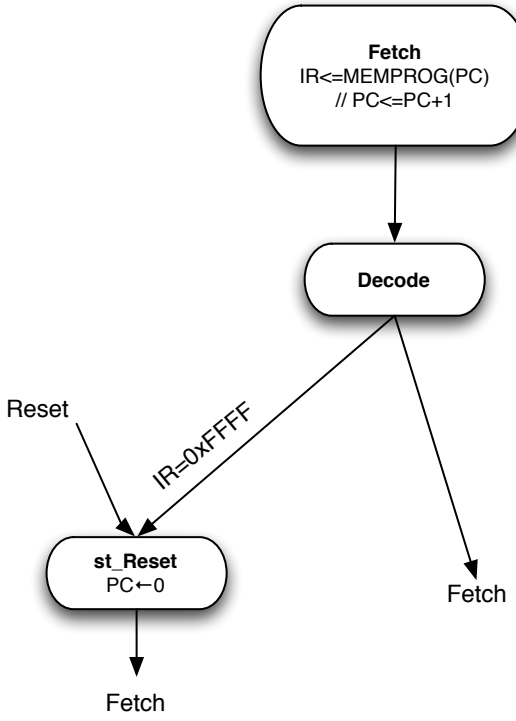
Question 1 Compléter le code de la PO correspondant au schéma suivant (partie nécessaire au reset) :




```
entity PO is
    Port ( DPROG : in std_logic_vector(15 downto 0);
           ADPROG,IR : out std_logic_vector(15 downto 0);
           CLK,EIR,EPC,CLRPC : in std_logic);
end PO;

architecture Behavioral of PO is
component Reg
    Port ( D : in std_logic_vector(15 downto 0);
           Q : out std_logic_vector(15 downto 0);
           CLK,E,RST : in std_logic);
```

```
end component;
signal sPC,sPCD : std_logic_vector(15 downto 0);
begin
        ADPROG<=                    ;
        sPCD<=                    ;
        cPC: Reg
        port map (D=>          ,Q=>          ,CLK=>          ,E=>
,RST=>          );
        cIR: Reg
        port map (D=>          ,Q=>          ,CLK=>          ,E=>
,RST=>          );
end Behavioral;
```

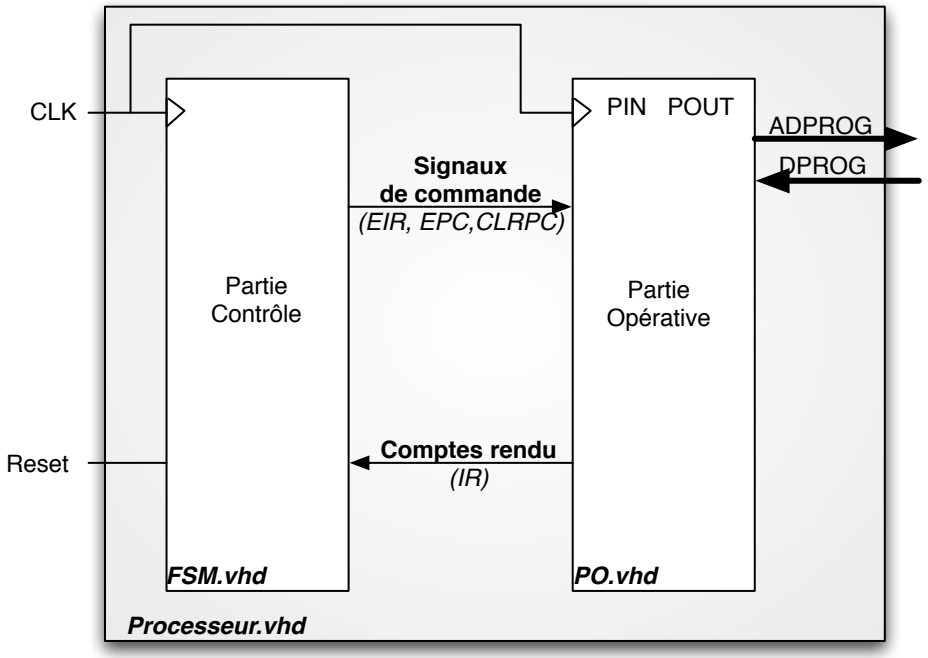
Question 2 Compléter le code de l'automate correspondant au graphe suivant :




```
entity FSM is
    Port ( CLK : in std_logic;
           RESET : in std_logic;
           IR : in std_logic_vector(15 downto 0);
           EIR,EPC,CLRPC : out std_logic);
end FSM;
architecture Behavioral of FSM is
        type StateType is (eReset,fetch,decode);
        signal ETAT_SUIVANT,ETAT_COURANT : StateType;
begin
Registres:
        process (CLK,RESET)
        begin
                if (RESET='1') then
```

```
                ETAT_COURANT<=eReset;
            elsif (clk'event and clk='1') then
                ETAT_COURANT<=ETAT_SUIVANT;
            end if;
        end process Registres ;
-------------------------------------------------------------------

        process (                                                    )
        begin
            EIR<='0';EPC<='0';CLRPC<='0';

            case ETAT_COURANT is
                when eReset =>



                when fetch =>


                when decode =>



                end case;
        end process;

end Behavioral;
```

Question 3 Compléter le code correspondant au processeur



```
entity Processeur is
    Port ( DPROG : in std_logic_vector(15 downto 0);
           ADPROG : out std_logic_vector(15 downto 0);
           CLK,RESET : in std_logic);
end Processeur;

architecture Behavioral of Processeur is
component PO
    Port ( DPROG : in std_logic_vector(15 downto 0);
           ADPROG,IR : out std_logic_vector(15 downto 0);
           CLK,EIR,EPC,CLRPC : in std_logic);
end component;
component FSM
    Port ( CLK : in std_logic;
           RESET : in std_logic;
           IR : in std_logic_vector(15 downto 0);
           EIR,EPC,CLRPC : out std_logic);
end component;

signal IR: std_logic_vector(15 downto 0);
signal EIR,EPC,CLRPC : std_logic;
begin
        cFSM: FSM
        port map (CLK=>          ,RESET=>          ,IR=>          ,EIR=>
,EPC=>       ,CLRPC=>          );
        cDataPath: PO
        port map (DPROG=>          ,ADPROG=>          ,IR=>
,CLK=>          ,EIR=>          ,EPC=>          ,CLRPC=>
);
end Behavioral;
```