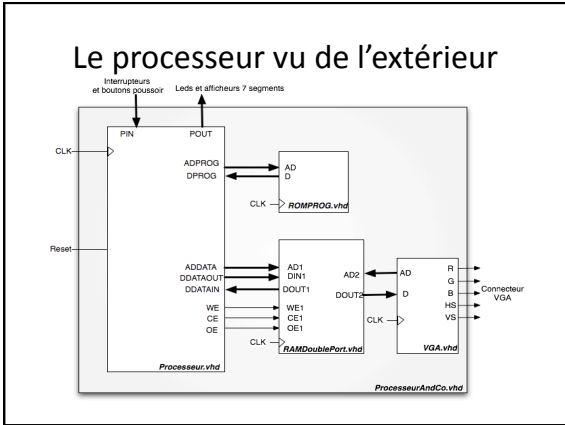


Architecture 2 : Cours 2

Présentation du processeur à concevoir



Exemple d'accès mémoire synchrone

```

Entity ROMPROG is
  port( AD : in std_logic_vector(15 downto 0);
        D : out std_logic_vector(15 downto 0);
        CLK: in std_logic);
end ROMPROG;

Architecture Behavioral of ROMPROG is
  ...
  Begin
    m(0)<="D400"; -- IN R0
    m(1)<="D600"; -- OUT R0
    m(2)<="FFFF"; -- reset

    process(CLK)
    begin
      if (CLK'event AND CLK='1') then
        D <= m(conv_integer(AD));
      end if;
    end process;
  end Behavioral;
  
```

Le jeu d'instruction (RISC) 1/2

- Instructions entrées/Sorties
 - IN Rd **Rd ← PIN**
 - OUT Rb **POUT ← Rb**
- Opérations entre registres
 - Op Rd,Ra,Rb **Rd ← Ra op Rb**
- Accès mémoire
 - LW Rd,Ra **Rd ← MEM(Ra)**
 - SW Rd, Ra **MEM(Ra) ← Rd**
 - LI Rd,Imm **Rd ← ext(Imm)**

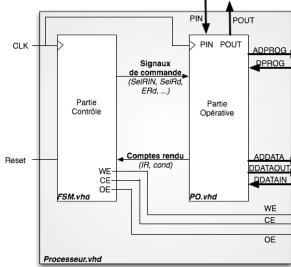
Le jeu d'instruction (RISC) 2/2

- Saut
 - BRcc Ra,Rb **Si Ra respecte cc alors PC ← PC+Rb**
 - BAcc Ra,Rb **Si Ra respecte cc alors PC ← Rb**
 - BRlcc Ra,Imm **Si Ra respecte cc alors PC ← PC+Imm**
- Saut à des sous programme
 - BRL Rd,Rb **Rd ← PC // PC ← PC+Rb**
 - BAL Rd,Rb **Rd ← PC // PC ← Rb**

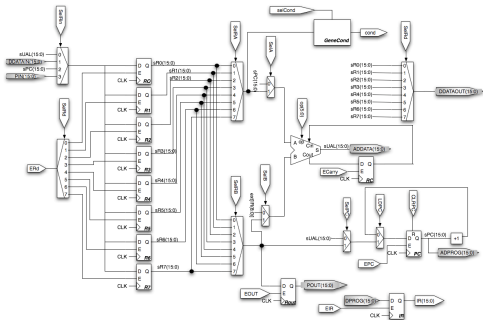
Exemple de programme

```
@ Instruction
0: BCL:      IN %R0
1:          LI %R1,21
2:          LI %R2,SSPROG
3:          BAL %R7,%R2
4:          OUT %R3
5:          BRlinc BCL
10: SSPROG: ADD %R3,%R0,%R1
11:          BAinc %R7
```

Construction du processeur



Chemin de données (PO)



Machine à état fini (FSM ou PC)

