Comportement Control Logic

* Lecture inst[31 :0]
* **CLOCK**
* SI (bulle sur R1) alors A1sel = 1 sinon 0
* SI (bulle sur R2) alors B1sel = 1 sinon 0
* SI (inst == BRANCH ou JUMP) alors A2sel = 1 sinon 0
* SI (inst == BRANCH) alors
  + SI (inst == BRANCH UNSIGNED) alors BrUn = 1 sinon 0
  + Faire les disjonctions de cas pour les autres opérations de branchement
* SI (inst == IMMEDIATE) alors B2sel = 1 sinon 0
* ALUSel = instruction code
* **CLOCK**
* SI (inst == STORE) alors MemRW = 1 sinon 0
* **CLOCK**
* SI (inst == LOAD) alors WBSel1 = 0 sinon 1
* SI (inst == JUMP) alors WBSel2 = 1 sinon 0
* SI (inst == BRANCH ou JUMP) alors PCSel = 1 sinon 0
* SI (inst != BRANCH et STORE) alors RegWEn = 1 sinon 0
* **CLOCK**
* rs1[n] == rd[n-1]
* rs2[n] == rd[n-1]
* **NOTE : Faire une disjonction de cas en fonction du type d’instruction (voir Jeu d’instruction.xlsx ou les spec risc v page 148)**
* inst[6 :0] == 0b1100011 || inst[6 :0] == 0b1101111 || inst[6 :0] == 1100111
* inst[6 :0] == 0b1100011
* inst[14 :13] == 0b11
* inst[6 :0] != 0b0110011 (instruction != REGISTER / REGISTER)
* inst[6 :0] == 0b0100011
* inst[6 :0] == 0b0000011
* inst[6 :0] != 0b1101111 || inst[6 :0] == 1100111
* inst[6 :0] != 0b1100011 && inst[6 :0] != 0b0100011