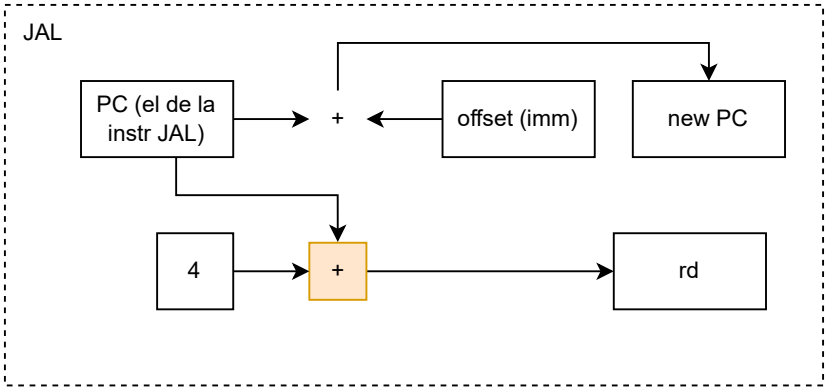
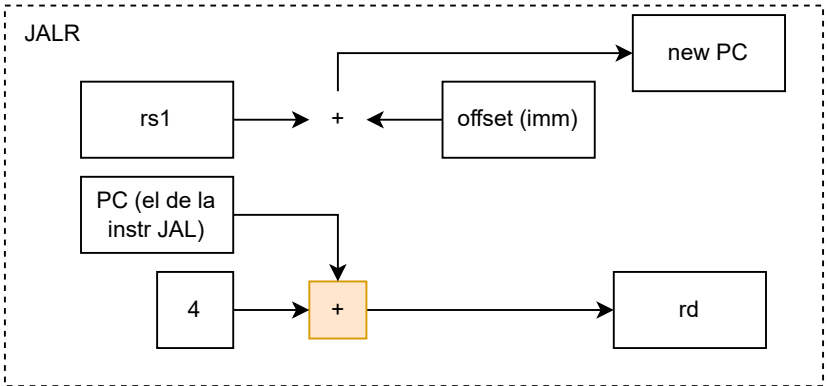


Agregar flag a la ALU de sumar 1 al resultado

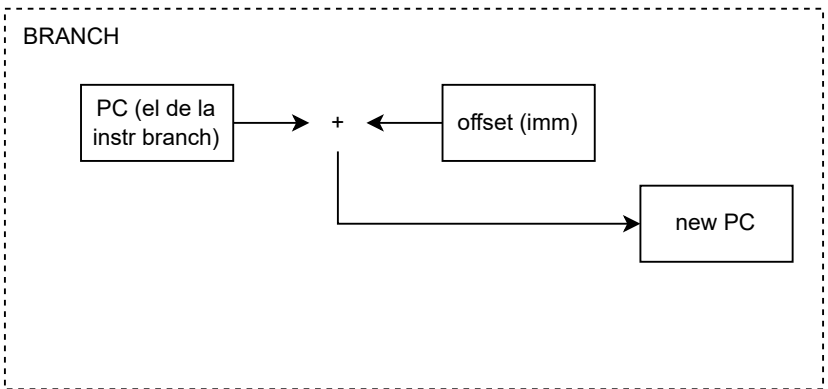
Se podría hacer inmediatamente adentro de fetch, mientras tarde menos de medio período funciona. Se propaga el flag de "sumar 1", la ALU hace imm + offset + 4 y guarda en rd. No hay que frenar el pipeline excepto por lecturas a rd

Flanco 0: la instr que sale del fetch es el JAL. Se calcula in situ el nuevo addr
Flanco 1: se triggerea el fetch a la instr correcta, sale una instr que no se hace. FALSO! no se pierde un flanco
Flanco 2: sale la instrucción válida



Se hace la cuenta, en paralelo, tanto en addr build y jmp ctrl como eventualmente en la ALU.

Flanco 0: la instr que sale del fetch es el JALR
Flanco 1: pasa la decodif, del fetch sale una instrucción que no se hace
Flanco 2: llegan los resultados a addr builder, del fetch sale otra instrucción que no se hace
Flanco 3: se triggerea el fetch a la instrucción correcta, del fetch sale otra instr
Flanco 4: primera instrucción válida que sale del fetch



Propuesta:
Flanco 0: la instr que sale del fetch es el JALR. Se podría predecir in situ según el signo del imm (1 bit) si el salto es hacia adelante o atrás, BTFNT actualizamos el PC.
Flanco 1: del fetch sale una instrucción descartada (si es backward, pq se saltará hacia atrás) o no descartada (si es forward, pq no se saltará hacia adelante). En el caso de que se descarte, se triggerea la lectura al nuevo PC
Flanco 2: del fetch sale una instrucción "útil", o la primera si se tuvo que descartar la anterior, o la segunda si no. Se empieza a hacer una cuenta en la ALU

Flag de forward/backward?