Instruction extraction

This document is the reference one about the pps_ei module included in the miniMIPS core. The goal of this module is to read the next instruction to be executed.

This stage sends the program counter *PF_pc* from pps_pf stage to the memory.

The stage takes care of different signals which change the result of the outputs. By priority :

- The signal *reset* initializes the outputs (NOP instruction and forbids interruption for this instruction to prevent to stop on an unreal one).
- The signal *stop_all* appears when the pipeline must be stop as the memory accesses are not done. The outputs are saved.
- The signal *clear* makes the *EI_instr* to a null instruction and forbids the interruptions for this instruction by making *EI_it_ok* to '0'. This signal appears when an interruption or a bad prediction occur in the pipeline.
- The signal *stop_e*i is equivalent to *stop_all* and appears when a data hazard is unresolved in the pipeline.
- If none of those signals have appeared then the instruction read in memory and its address are put in the output *EI_instr* and *EI_adr* and the interruptions are allowed (the instruction is a real one).