**Basic RV32I pipelined datapath design document**

**Overview**

**The RISC-V pipeline is broken down into 5 stages, with one step for each stage. The steps are as follows:**

**IF**: Instruction fetch from memory

**ID**: Instruction decode & register read

**EX**: Execute operationor calculate address

**MEM**: Access memory operand

**WB**: Write result back into register

Instead of a state machine, each instruction will generate a ‘control word’ in its decode state to control all of the signals necessary to control all of the pipeline stages which will be passed down the pipeline.

**Control word (ctrl\_word) structure:**

**Load**: load\_regfile

**Mux Selection**: regfilemux\_sel, alumux1\_sel, alumux2\_sel, cmpmux\_sel

**Operations**: cmpop, aluop

**Memory**: dcache\_read, dcache\_write

There are 4 pipeline stage registers between every two pipeline stages, the registers contents are as below:

**IF\_ID**: PC, icache\_rdata

**ID\_EX**: PC, icache\_rdata, ctrl\_word

**EX\_MEM**: ctrl\_word, alu\_out, rs2\_out, rd, br\_en, u\_imm

**MEM\_WB**: ctrl\_word, alu\_out, mdrreg\_out, rd, br\_en, u\_imm