Fetch -> issue . issue should check fetch cycle start

Issue-> execeution . execution should check issue cycle start

Write-> commit. Commit should check issue cycle start only for non branch/st ops. Maybe halt and unkown as well.

**Load/Store** – David : change issue & write to operate with load/store buffers only . no int resv stat in between.

Yossi : execution – take from load/ store as well for address calc.

**Check qj/qk in exec:**  david : change issue of qj/qk of non rob.. and check if there’s another reference.

Yossi: check if qj & qk = -1 and not vj/vk with min val.

Yossi: ld/st also check qj & qk = -1 and calc address by Vj+Vk, put the result in address.

Commit store: value is the float register id to read value from. (Float.floatToIntBits(value)

Load : Float.intToFloatBits(value)

12. why is ReadyStRow needed? On write update ROB’s dest and set ready to true.

Commit in store , cycle committed at first cycle . Clean store buffer after memdelay is finished.

Fetch – add instruction before pc increment

Jump & BTB – in issue flush everything like a false prediction if jump wasn’ t taken and add to BTB

Memory aliasing – when attempting to start load, check rob table from load’s rob backwards to head to find destination of the same address to a store op. If one exists, try to execute the next load op.Otherwise, continue with the load.