1. Work through each instruction

2. Identify additional logic we’ll need

3. What control signals do we need? Which stages will need which ones?

4. Do we want hierarchical or flat design? -- I think hierarchical is a good idea; it may be slightly more complicated, but i think it is worth it for breaking up the work and keeping it looking nice & organized

Fetch:

Fetch instruction from instruction cache

Predict pc (PC+=2)

Decode:

Set the control word.

The following happen in parallel:

-Identify registers, grab operands from register file

-Issue logic of whether instr is ready for execute; if not, send STALL signal to stall decode

& fetch stages (so that both don’t accept new bits before we can move on)

-Calculate BR/JMP target address (this allows single-cycle BR misprediction penalty)

Compute BR condition AFTER registers are read

-IF BR taken or JMP, assign calc’d target to PC predictor

*Control Signals: STALL signal, signals that indicate BR/JMP just glide through next stages?*

Execute: For clarity, split instructions into two latency classes:

1. Register-Register Operation: (single cycle) Adds & Logical

Feed arguments to the ALU

2. Memory Reference: (2 cycles) -- all memory loads obv

ALU adds register & offset to calc address

*Control Signals: ALU signals like choosing operators (immediate val) and ALUop*

Memory Access:

1. Those single cycles just forward the result through this stage to the next

2. Mem Refs do their memory thing

special things with ldb/stb done here?

*Control Signals: Memory control signals set here or in execute? prob exe*

Write Back:

Write results back to register file

QUESTIONS:

How do we make the mem signals work? Should we just use a simple delay to make sure that they don’t go high until the MAR has selected the right address?

How much of our old MAR/MDR model is necessary to retain? Is it ok to just pass a dCache an input signal if it’s already stored in the interstage register?

For LDI/STI, should we use a flipflop to keep track of our “state” since we can’t use an fsm?

Citation:

<http://en.wikipedia.org/wiki/Classic_RISC_pipeline>

Citation for idea for citation:

Genevieve LaBelle