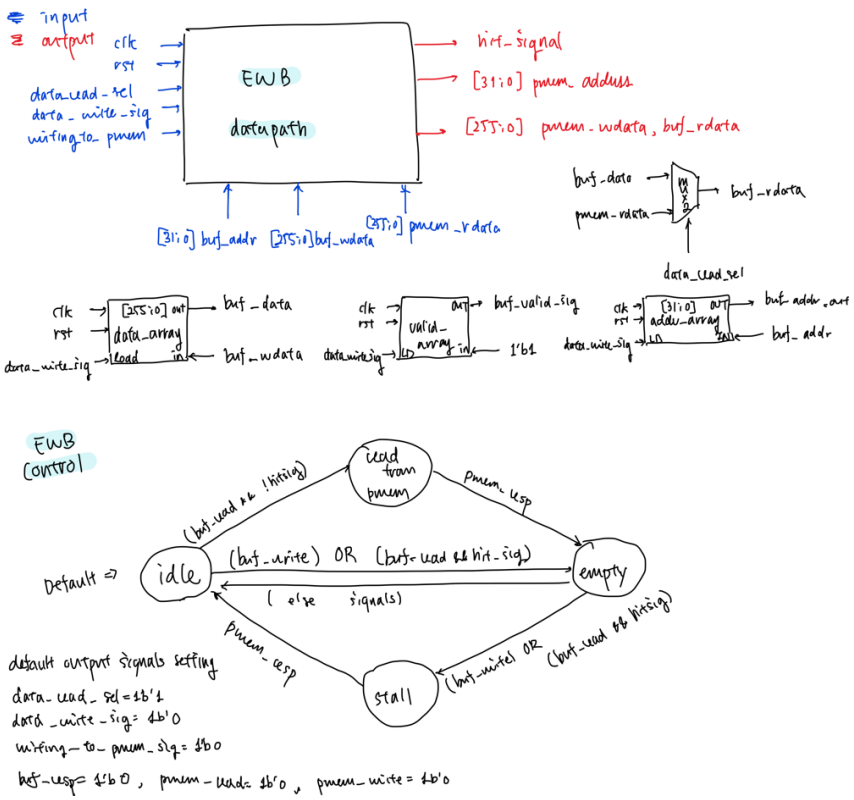
Advanced Feature Proposal and Design

Proposal: we have talked about these proposal during the weekly meeting with TA. Details are all approved, but we still need them to be designed and handed in.

1. Eviction Write Buffer



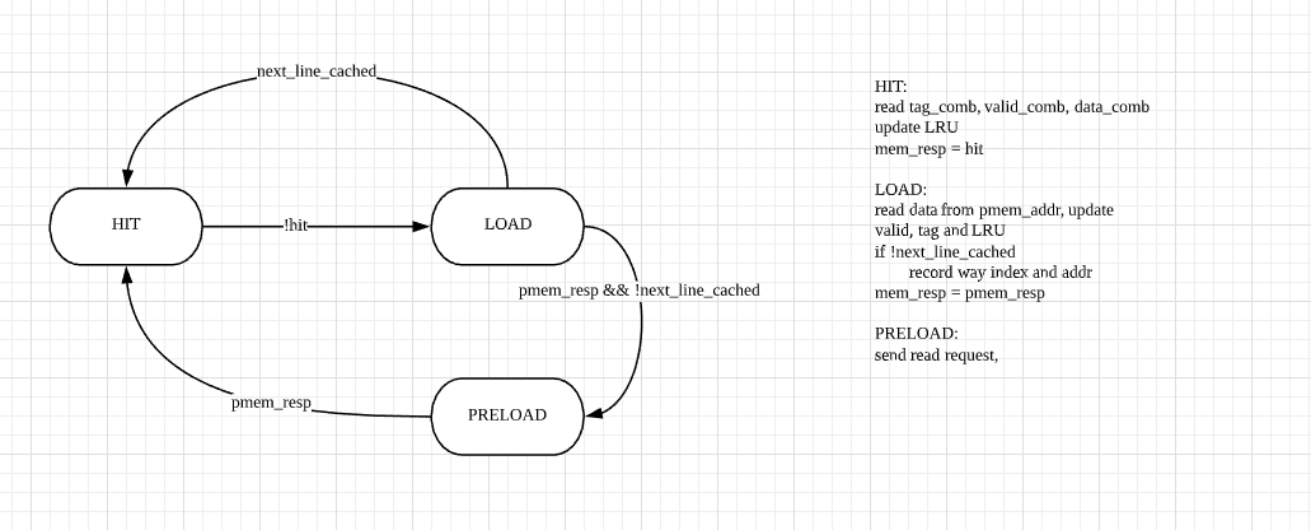

```
idle: begin

    /* write */
    if (buf_write) begin
        data_write_sig = 1;
        buf_resp = 1;
    end

    /* read */
    else if (buf_read && hit_sig) begin
        /* read data from buffer */
        data_read_sel = 0;
        buf_resp = 1;
    end
end

/* Reads data from Physical Memory */
read_from_pmem: begin
    pmem_read = 1;
    if (pmem_resp)
        buf_resp = 1;
end

/* ewb in  Stall state, indicating */
stall: begin
    pmem_write = 1;
    writing_to_pmem_sig = 1;
end

/* Reads Buffer  */
ewb_empty: begin
    if (buf_read && hit_sig)
        data_read_sel = 0;
end
```

EWB datapath: we use One Mux2 and Three registers; registers will be different in their sizes. We name this mux as a data_read_mux that would choose between to read from the buffer or from the physical memory depending on the select signal, which is a data_read_sel. We need two more combinational logic here. Considering that at WB stage, upon a WB request, we want to change the corresponding address to store such data at this buffer, not in actual physical memory. Assign pmem_address = ewb_address. Thus, one combination logic sets the physical memory address depending on writing_to_pmem signal and another logic would be made for hit signal decision just like how we implemented in cache design such that here we bring write buffer's data into the pmem_data. Basically it is another level of register to hold a data before accessing physical memory.

EWB Control: Composed of total four states as drawn above. Next stage logic is also drawn above. Control logic takes Six input signals such as clk, rst, buf_read, buf_write, hit_sig, pmem_resp and would output Six control signals, which are data_read_sel, data_write_sig, writing_to_pmem_sig, buf_resp, pmem_read and pmem_write. State actions are as above picture(Right Side).

Prefetching:

We implemented basic one-block-look-ahead prefetchingin our instruction cache. The state transition and s ignals are shown below. The way_index is used as an indicator of which cache way that the data should be prefetched into.



Note that in our design, the Icache will be able to respond to read/write hit during the preload stage e xcept for the last cycle, where data is fetched from pmem and conflict can potentially happen.