Name: Aria Pahlavan
EID: ap44342

1. Changes you made to the state diagram. Include a picture similar to the state machine that shows the new states you added.

First, I added all the states required to properly translate virtual pages. To do so, first I have added state 17 in which I load the MAR with PTBR and the page number of the virtual address, and then in state 48 I load the PTE (i.e. M[MAR]) into MDR. Second, in states 50 and 57, I do the checks for protection and page-fault, respectively. Then in states 53 and 45, I update the reference bit of the PTE, as well as the modify bit if it's a write access. Finally, in state 47, I create the physical address of the desired location by combining the PFN from PTE and offset bits from VA.

To keep the unaligned and protection exceptions functional, I moved the the unaligned exception check before the protection exception in the flow because the latter has lower priority. It is important to note that the unaligned exception check is done on the virtual address and the protection exception is done using the PTE of the virtual address. Also, to identify whether or not the access is a write, I set the 15th bit of the TEMP register to 1 if the access to be performed is a write (in states 3 and 7).

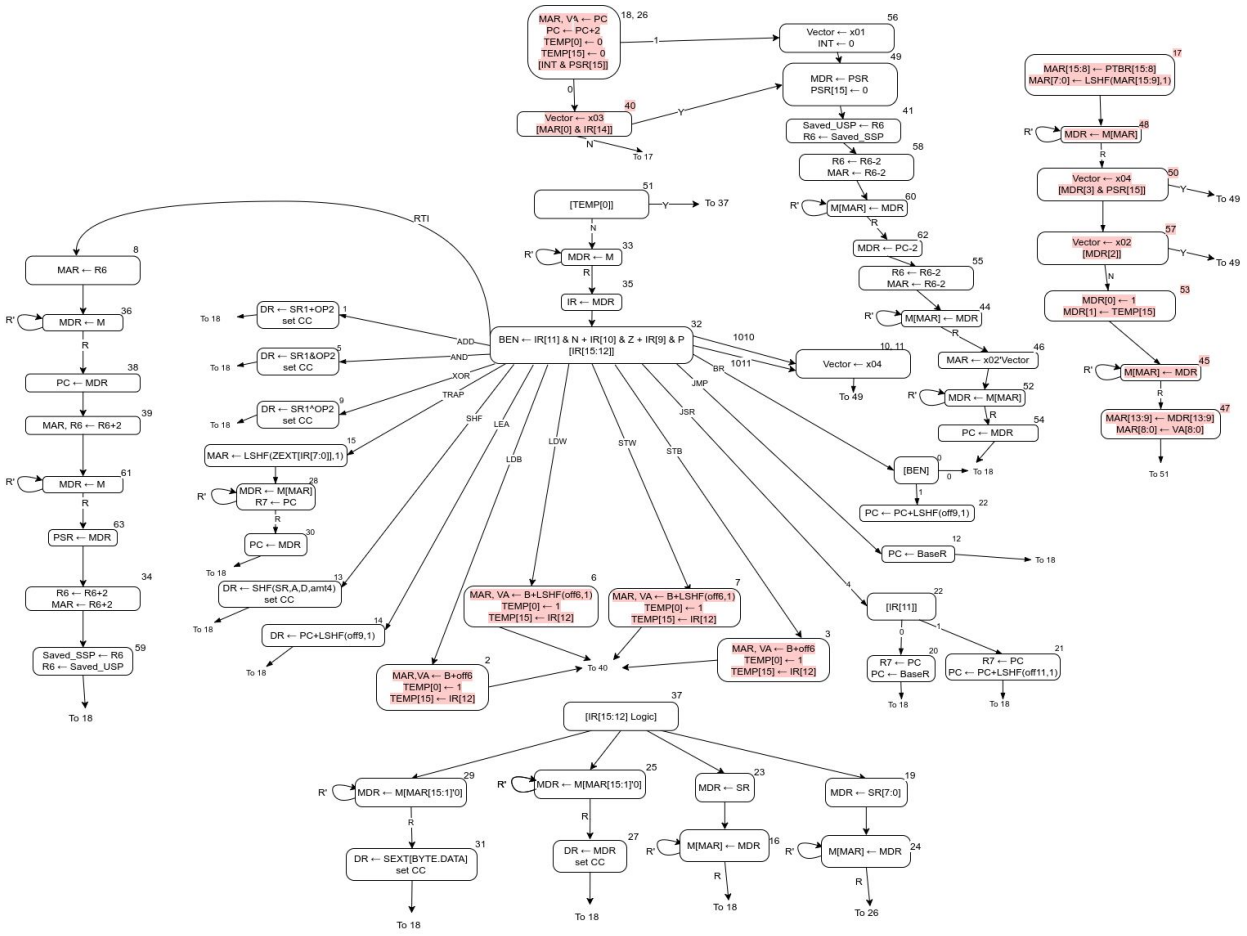

Figure 1. Interrupt and Exception flow

2. Changes you made to the datapath. Clearly show the new structures added, along with the control signals controlling those structures.

First I have added the VA and PTBR registers as instructed in the Lab description. However, I did not connect them to the BUS directly. Instead, I connected them to two new register, PTE and PA, to hold the page table entry and physical address at any moment depending on the values of registers VA and PTBR. PTE is created by combining the bits PTBR[15:8] and VA[15:9] to create a 16-bit value. Similarly, PA is created by combining the bits 00, MDR[13:9], and VA[8:0] to create a 16-bit value. PTE and PA are guarded by GatePTE and GatePA. I have also added a UpdateMDR signal to MDR register, and when it is set the reference bit (i.e. bit 0) of MDR is set to 1.

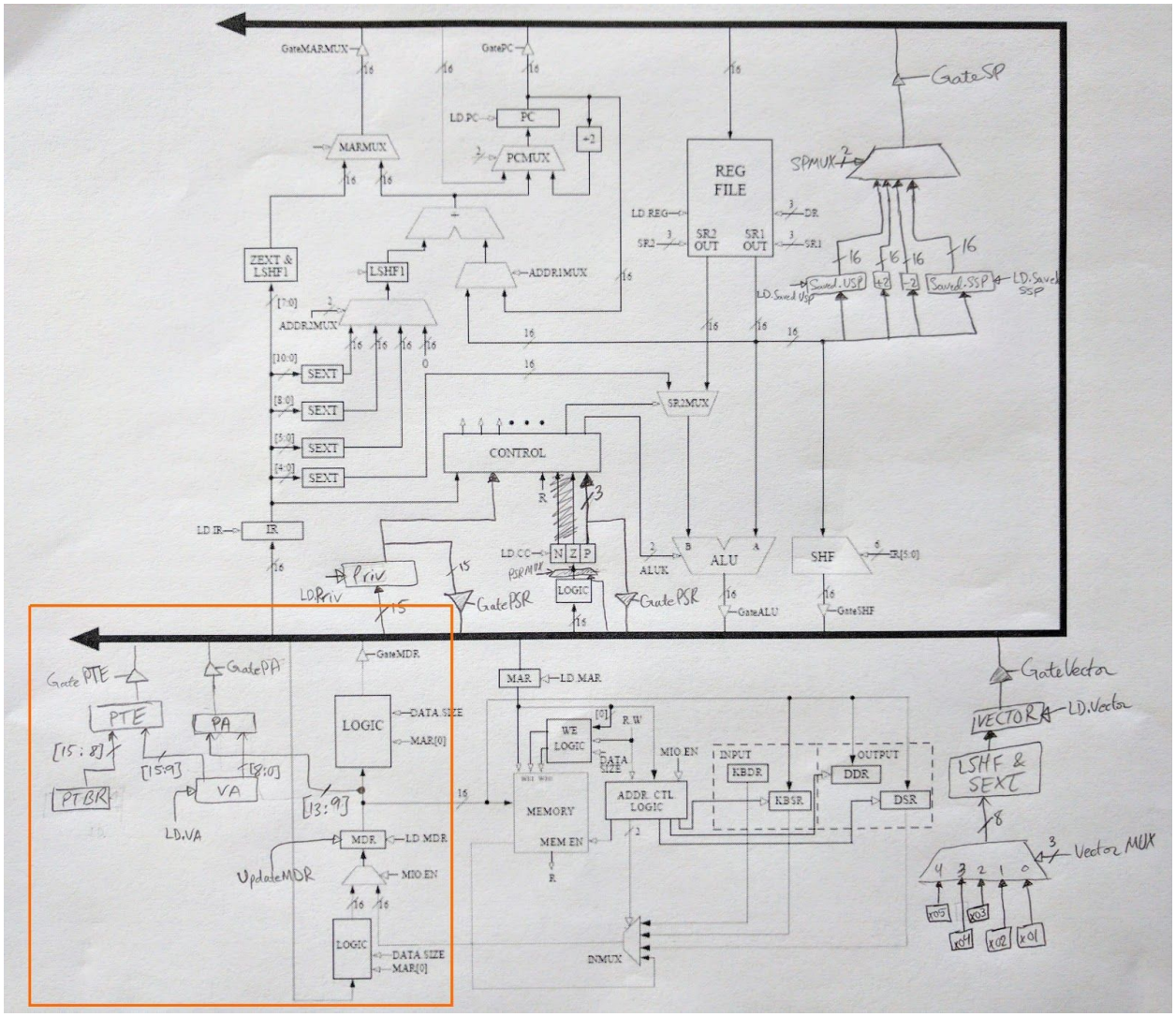

Figure 3. Data Path

3. New control signals you added to each microinstruction. Briefly explain what each control signal is used for.

Explanation provided in previous answer.

- GatePTE
  - YES
  - NO
- GatePA
  - YES

    - NO
- LD_VA
    - YES
    - NO
- UpdateMDR
    - YES
    - NO

4. Changes you made to the microsequencer. Draw a logic diagram of your new microsequencer and describe why you made the changes.

I added a new condition variable (COND3), which helps me determine whether I need to check the BEN bit or MDR[2] for page-fault. Also, I have modified the check for privilege protection to ((PSR[15] AND MDR[3]) OR !PSR[15]).

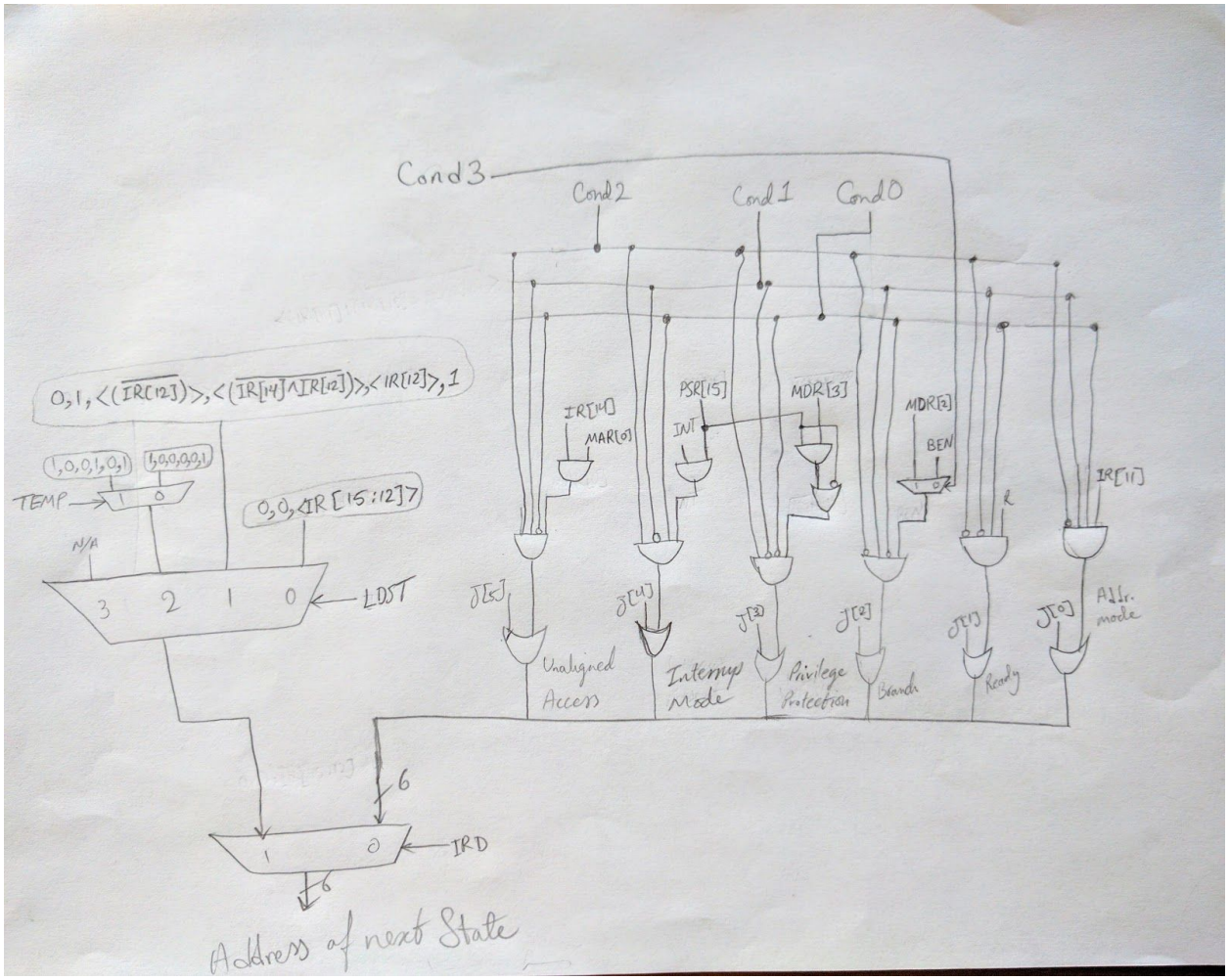

Figure 3. Microsequencer