

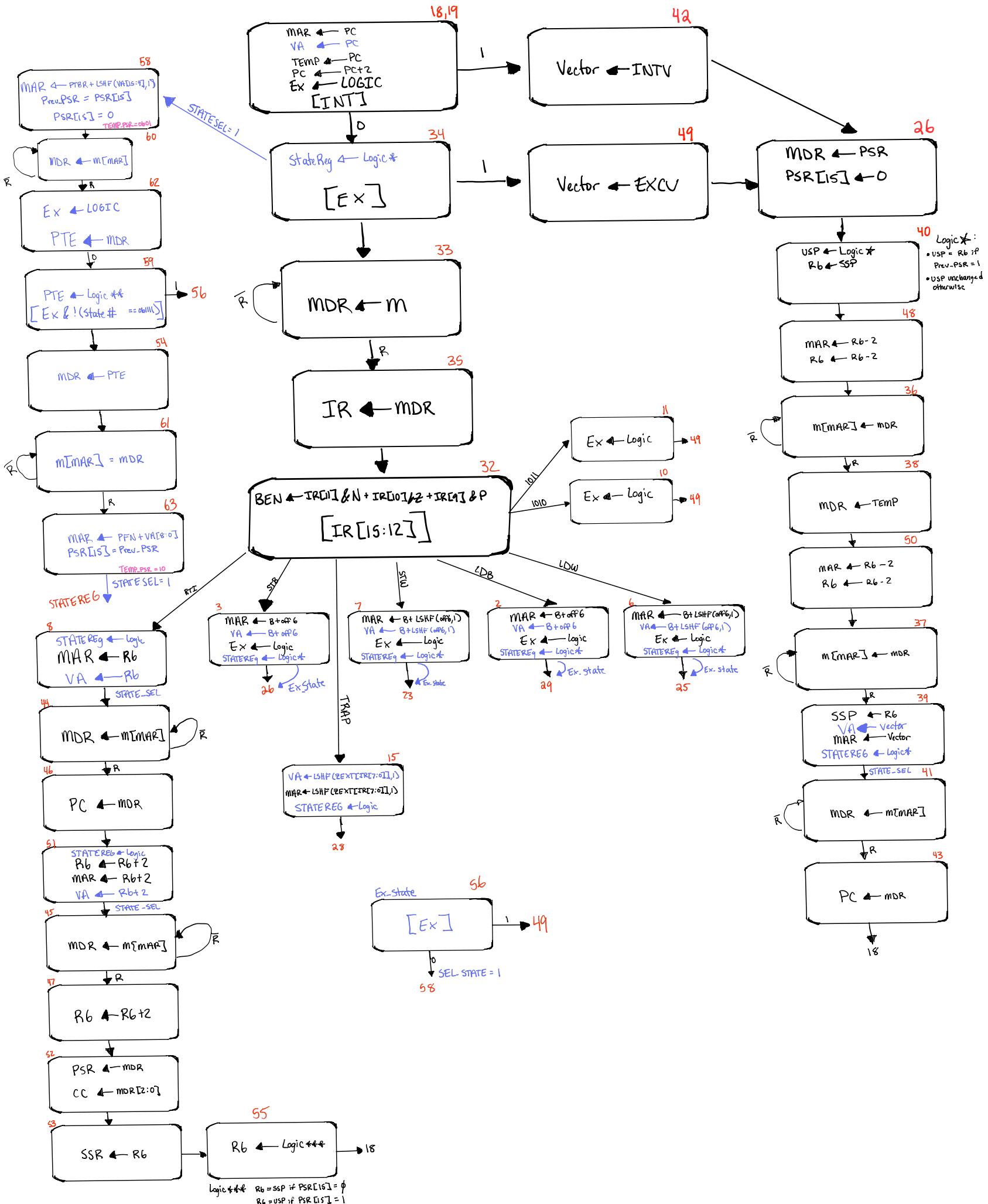
Logic * : Set STATE REG as next state for current instruction

Logic **: set M bit
Set R bit

Ex STATE = state 62

* STATE_SEL causes the sequencer to look @ different states.

Available States:
57



Avery Atchley

Lab 5

State Diagram Changes:

- **Total States Used:** 63/64

For this lab, I changed up how exceptions were handled compared to Lab 4. In Lab 4, I had a state for each instruction that checked the EX flag, causing unnecessary usage of states that carried out the same function. To address this issue, thus freeing up additional states for the translation routine, I created a standalone state, state 56, that checked the exception flag. From this state, we either go into the ESR or to the translation state. The instructions that check for an exception (LDW, LDB, STW, STB) use the EX_STATE control signal, which augments the microsequencer to save the next state for each instruction listed above, then move to state 56.

For the instructions which load MAR, the VA register is also loaded. From there, the translation routine is triggered to switch the value loaded into the MAR to a physical address. Before switching to this routine, the STATREG is loaded to hold the next state of the instruction prior to switching to the translation routine.

The translation routine takes the MAR value and converts it into a physical address. This routine also checks for exceptions with the PTE for the instruction attempting to execute. To accomplish this task, PSR[15] is temporarily set to 0 to prevent protected access exceptions. Once the routine has finished, the previous value held in PSR[15] is restored. We then return back to the next state set before the routine was taken.

Datapath Changes:

The datapath modified the existing exception vector, increasing the number of bits from 3 to 4 to accommodate for page faults. The existing logic was also modified slightly to accommodate better handling for interrupts—acknowledging the interrupt flag after we arrive back at state 18 compared to taking the interrupt as soon as the flag was detected.

A VA and PTBR register were also added to accommodate for the translation routine. For these registers, further logic was added to accommodate for different functions—muxes, LSHF, and ZEXT—needed for logic in the state diagram.

Lastly, a PTE register was added to allow for the modification of the M and R bits in the PTE. This register also allows for the PFN to be placed onto the bus as well.

Microsequencer Changes:

The microsequencer was altered to accommodate switching to 4 new predetermined states—ESR, Ex_State, Translation State, and STATREG—based on the control signals EX, EX_STATE, and STATE_SEL, which are inputs into a 2-bit MUX. The STATREG value is set when LD_STATREG is enabled, thus it is not defined at all times. The ESR state (state 49), Ex_State (state 56), and Translation state (state 58), however, are all defined and non-changing. An additional mux is used to select between the Translation State and STATREG values, using a STATREG control signal, which when enabled selects the STATREG state, and when disabled selects Translation state.

Control Signals:

- LD_VA: Loads VA register
- VAMUX: Selects between 3 options to modify VA for output to the BUS
- GATE_VA: Outputs selected VAMUX value onto the BUS
- LD_PTE: Loads PTE register
- LD_M: Loads M bit of PTE
- WRITE: Selects between 0 and 1 for M bit of PTE -> register set in software
- LD_R: Loads R bit of PTE
- SETR: Selects between 0 and 1 for R bit of PTE
- GATE_PFN: Outputs PFN value, plus offset dictated in the VA, onto the BUS
- GATE_PTE: Outputs PTE onto the BUS
- STATESEL: Causes microsequencer to shift focus from COND and J bits to either translation state register or STATEREG value.
- TEMP_PSR: 2-bit signal meant to dictate if PSR[15] should be saved to PREV_PSR or if PSR[15] should be loaded with PREV_PSR.
- STATEREG: Dictates when the STATEREG value should be loaded into NEXT_LATCHES.STATE_NUMBER.
- LD_STATEREG: Load STATEREG register
- EX_STATE: Dictates when the microsequencer should load the EXSTATE state instead of COND and J bits.