Progress Report:

Roles:

- Phillip: Register Rename / Dispatch
- David: RAT, Functional Unit, RRF, FreeList
- Brian: Reservation Station, ROB, Memory

Functionalities:

1. **Register Rename / Dispatch (Phillip)**:
    - Implemented the overall control logics for register rename / dispatch. It includes stalling logic, retrieving free registers, checking RAT, and sending relevant signals into reservation stations.
4. **RAT, Physical Reg, Func. Unit, RRF, FreeList (David)**:
    - Developed the modules required for different parts of the OoO processor. It includes making latches, queues, and logic regarding these components.
5. **Reservation Station, ROB, Arbiter, Memory (Brian)**:
    - Designed the reservation station, ROB, and Memory functional unit. It includes control logic within the reservation station and the ROB queue along with the functionality for the memory unit. By changing the design from 3 CDB to an arbiter, connected the relevant components.

Testing Strategy:

- Unit tests: Each component (RAT, Phys.Reg, Funct units, RRF, FreeList, RS, ROB, arbiter, memory etc.) was thoroughly tested in isolation using unit tests to verify individual functionalities. Then we did an overall testing to see if we passed the coremark

Timing and Energy Analysis:

- Fmax = 666.666MHz
- Clk period = 1.5ns

Roadmap: Roles:

- Phillip: Basic memory implementation
- David: Basic cache
- Brian: Basic branch predictor

Features/Functionalities:

1. **Basic memory implementation (Phillip)**:
    - Will focus on implementing the basic memory load/store model where the store will be executed on commit and the load will be executed after all loder stores execute.
2. **Basic cache (David)**:
    - Utilize the mp_cache to make minimal changes to just fit into our OoO CPU.
3. **Basic branch predictor (Brian)**:
    - Will implement the branch not taken branch predictor.