Hoạt động của Dispatcher:

- Dispatcher giữ vai trò như để phân bổ các block cho đến các core, khi được khởi tạo, nó gửi tín hiệu reset tất cả các core và các thành phân của core.

- Quét qua tất cả các core, nếu core đó đã được reset thì cấp cho nó 1 block, block++, nếu core có tín hiệu hoàn thành block rồi (thể hiện qua core\_start && core\_done) thì reset core đó.

Data Memory controller và Instruction memory controller:  
- Dùng để quán lý truy cập từ core (LSU và fetcher) tới global memory

- Có 5 trạng thái: IDLE, READ\_WAITING, WRITE\_WAITING, READ\_SERVED và WRITE\_SERVED, về cơ bản thì read và write giống nhau ở cách hoạt động,

- Đầu tiên ở trạng thái IDLE, nó sẽ quét qua tất cả các (channels), channel ở đây là các cổng nối giữa controller và global memory, nếu channel đó rảnh, nó sé tìm xem có consumer nào có read hay write request hay không, nếu có thì gửi tín hiệu mem\_read/ mem\_write tương ứng tới global memory, sau khi global memory trả về tín hiệu mem\_read/ mem\_write ready, nếu consumer đó không còn gửi tín hiệu read/write request nữa thì channel vào trạng thái IDLE.

- CORE:

+ Scheduler dùng để quản lý state của core đó, gồm các state là IDLE, FETCH, DECODE, REQUEST, WAIT, EXECUTE, WRITEBACK. Ở state IDLE nếu nhận tín hiệu start từ dispatcher của gpu thì sẽ tới giai đoạn FETCHING.

+ Fetcher sẽ hoạt động ở giai đoạn FETCHING, lúc này thì sẽ gửi tín hiệu mem\_read\_request tới Instruction memory controller để lấy instruction dựa theo “current\_pc” và đợi kết quả từ controller trả về, xong giai đoạn fetch

+ Tiếp theo là giai đoạn decode, điều này sẽ do 2 khối llà decoder và control unit đảm nhiệm, decoder sẽ giải mã ra địa chỉ của các thanh ghi rd, rs, rt và immediate nếu instruction là dành cho câu lệnh immediate. Control unit sẽ giải mã opcode(instruction 15:12) để biết được đây là lệnh gì, add, sub, mul, div ,… và sau đó trả về các tín hiệu tương ứng, ví dụ nếu là opcode ADD thì sẽ ra RegWrite, ALUControl để điều khiển khối ALU

+Tiếp theo tới giai đoạn REQUEST, là khối LSU sẽ gửi tín hiệu tới data\_mem\_controllẻr nếu gặp tín hiệu MemRead hoặc MemWrite và chuyển sang 1 state gọi là WAITING, đợi kết quả từ controller trả về. Song song với đó thì register cũng chuẩn bị giá trị cho rs và rt.

khi kết quả đã trả về hết rồi thì core sẽ chuyển sang giai đoạn EXECUTE, giai đoạn mà khối alu sẽ đi vào hoạt động, ở đây chỉ có các lệnh cơ bản chứ chưa làm lệnh xịn, chỉ có các lệnh là add, sub, mul ,div.

+Tới giai đoạn WRITEBACK, , tùy thuộc vào instruction có phải là lệnh ProgramReturn hay không mà sẽ thoát chương trình hoặc cập nhật current\_Pc sang tiếp theo, đồng thời lúc này nếu lệnh hiện tại có yêu cầu RegWrite thì cập nhật vào register[rd\_address]