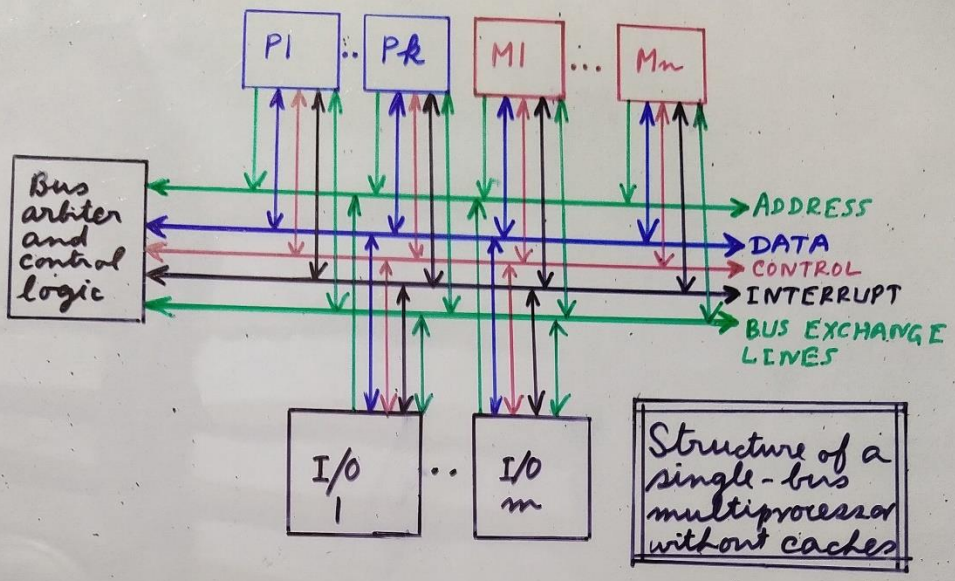
Notes-06

BUS Arbitration

Informal: In a single-bus architecture, the bus is essentially a highway through which data travels between processors and memory modules. Since this is a shared entity, rules must be enforced to avoid chaos; we have to decide who will get control of the bus for data transfer. This is achieved by the BUS ARBITER.

SINGLE SHARED BUS



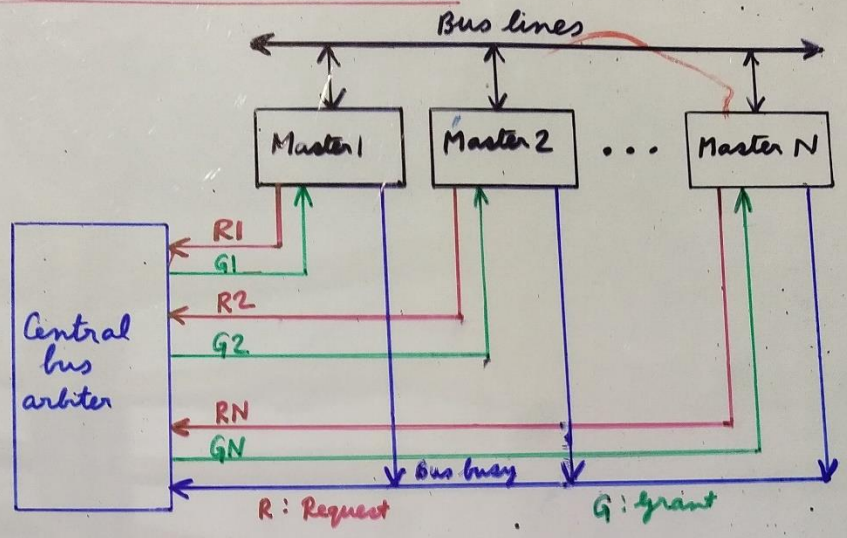
Structure of a single-bus multiprocessor without caches

Bus arbiter logic — allocates the bus in the case of several simultaneous bus requests.

Bus exchange lines — comprise typically one or more bus request lines by means of which the processors or other temporary bus masters can request bus allocation.

According to the state of the bus request lines and the applied bus allocation policy, the arbiter grants one of the requestors via the grant lines.

CENTRALIZED ARBITRATION



Protocol of allocating the bus:

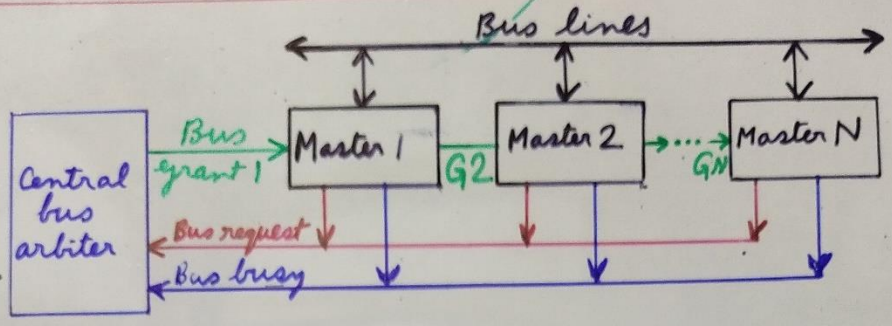
(i) Master$_i$ requests the bus by activating its dedicated bus request line.

(ii) If the bus busy line is passive, that is, no other master is using the bus, the arbiter immediately allocates the bus to the requestor by activating the grant$_i$ line. The requestor deactivates its request line and activates the bus busy line, prohibiting the allocation of the bus for other requests. After completing the bus transaction, the requestor deactivates the bus busy line.

(iii) When the bus busy line is active, the arbiter does not accept any bus requests.

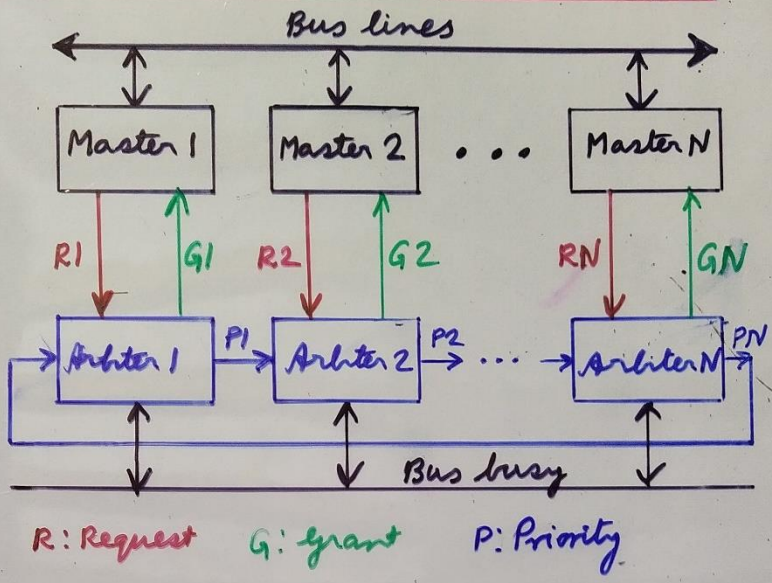
(iv) When several request lines are active by the time the bus busy line becomes passive, the arbiter can use any of the following bus allocation policies: fixed priority, rotating priority, round robin, least recently used, first come first served.

DAISY-CHAINED BUS ARBITRATION



- One of the most popular organizations of arbiter logics is daisy-chaining.
- There is only one shared bus request line. All the masters use this line to indicate their need to access the shared bus.
- The arbiter passes the bus grant line to the first master and then it is passed from master to master, creating a chain of masters.
- The priority of a master is determined by its position in the grant chain. The closer it is to the arbiter, the higher its priority.
- A master can access the shared bus if the bus busy line is passive and its input grant line is active.
- When the master does not require the bus and receives an active grant line, it activates its output grant line enabling the next master to use the bus.

DECENTRALIZED ROTATING ARBITER



- Daisy-chained scheme lacks fairness.
- Rotating arbiter eliminates this drawback.
- Arbiter$_i$ is allowed to grant its coupled master unit if master$_i$ has activated its bus request line, the bus busy line is passive, and the priority $(i-1)$ input line is active.
- If master$_i$ has not activated its bus request line, arbiter$_i$ activates its output priority$_i$ line.
- Selecting the lowest priority unit:
  - daisy-chained arbiter → the master farthest away.
  - rotating arbiter → the master that releases the bus.