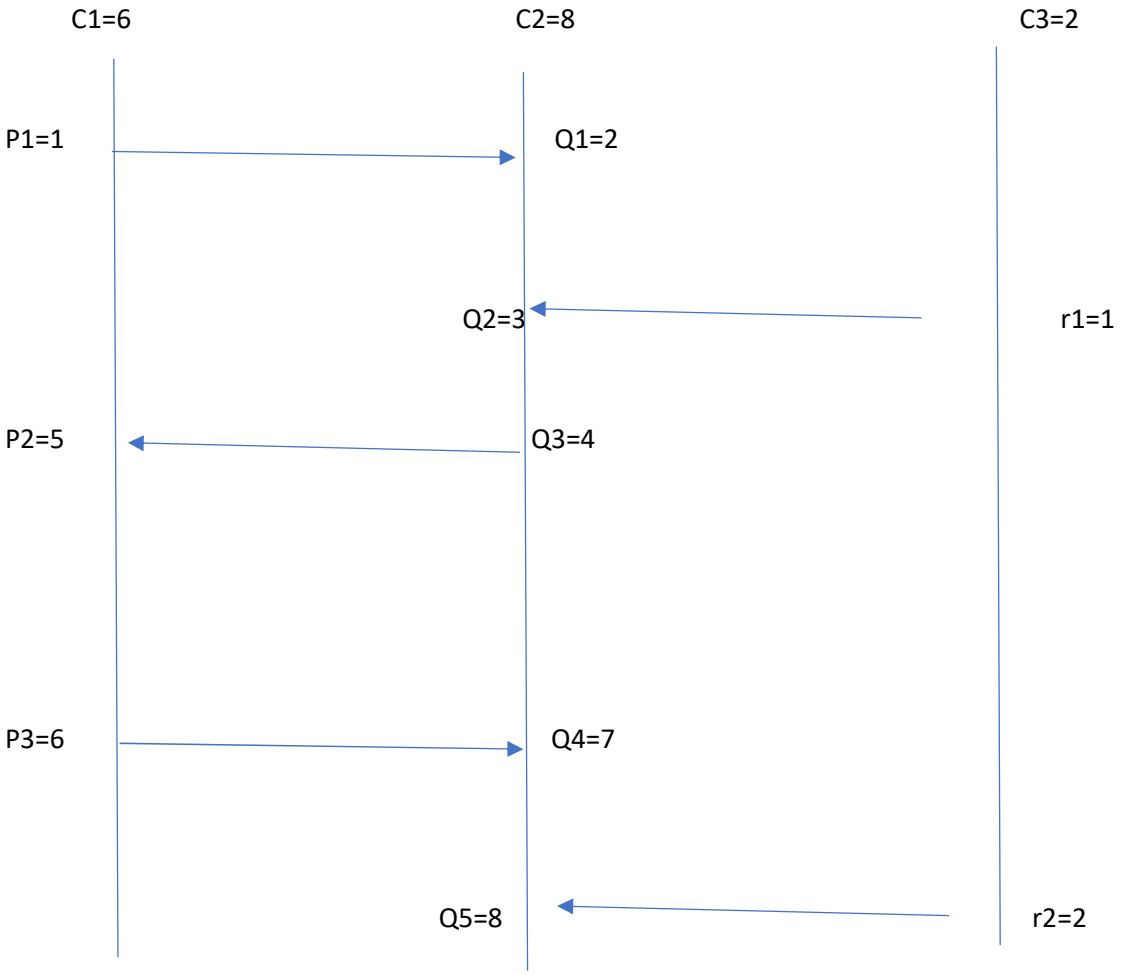
HOMEWORK 4

RISHI KUMAR SONI 1001774020

6.3.Figure 6.28 shows a computation of processes P, Q, and R using asynchronous communication. The messages in this computation are p1→q1, r1→q2, q3→p2, p3→q4, and r2→q5.

(a) Determine the integer timestamp of each event in the computation above.



Order is :

P1

R1

Q1

R2

Q2

Q3

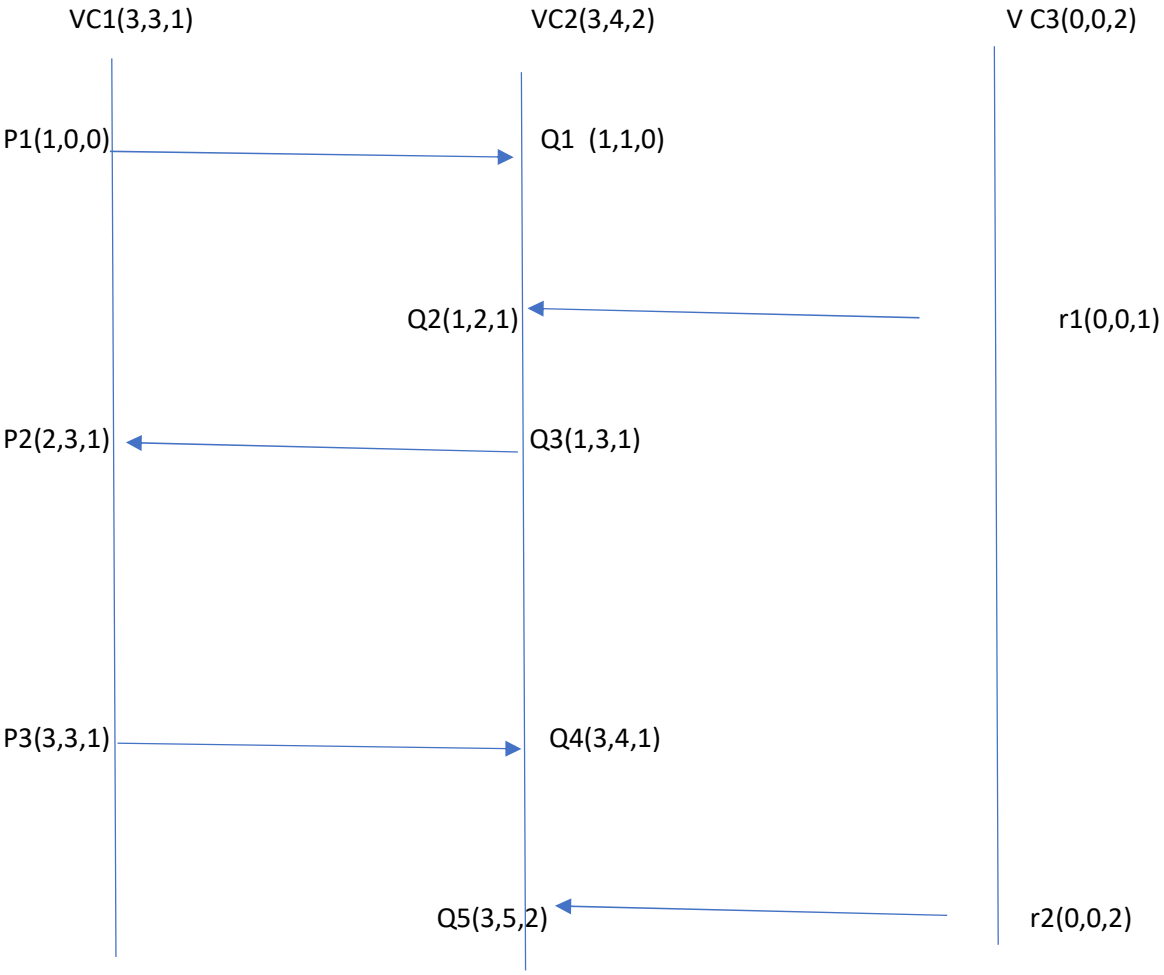
P2

P3

Q4

Q5

b)



(c) Indicate all of the events that are concurrent with P3.

P3 is concurrent with R2 only .

(d) Indicate all of the events that are concurrent with R2.

R2 --→ P1, Q1, Q2, P2, Q3, Q4, P3.

6.5

(b) Determine the vector timestamp of each event in the computation above. Each vector timestamp has three elements, with the first, second, and third elements indicating logical clock values for P, Q, and R, respectively.

do {

remove the first request message (Tj,Mj);

if (Mj > Ej) // Ej is the vector value in the token that

Ti is holding Ti sends the token to Tj and exits the loop;

else Ti discards this useless request;

} until (the request queue is empty);

<u>SOLUTION :</u>

Consider an example where there are 3 threads:

- T1, T2, T3

- Suppose we have Token vector with value: [3,3,3]

- This means that T1 , T2 and T3 already have entered into the critical section 3 times. Now assuming that : T2 is still in the critical section. Now, T1 requests to enter, at this point M1 will be 4 (4th intention to enter the CS). Since, T2 is already in CS, T1 will be in the wait queue. T3 requests to enter in the CS, so M3 will also be 4 at this point.

- T2 is already in the Critical Section.T1 makes a request again. Now, M1 will be 5 (E1 is still 3 since the first request is still pending in the wait queue).T2 exits the critical section and checks for the waiting queue.T1 is first at the queue, so it will be served next.

OLD ALGO

bash

do {

remove the first request message (Tj,Mj);

if (Mj > Ej) // Ej is the vector value in the token that Ti is holding

Ti sends the token to Tj and exits the loop; else Ti discards this useless request;

} until (the request queue is empty);

At this point the value of the M1 is 5 and the E1 is 3

If condition satisfies and T1 is allowed to enter the critical section.

NEW ALOG

bash

do {

remove the first request message (Tj,Mj);

if (Mj == Ej+1) // Ej is the vector value in the token that Ti is holding

Ti sends the token to Tj and exits the loop; else Ti discards this useless request.

} until (the request queue is empty).

When if condition is changed, M1 is 5 and E1 will be 3+1 = 4.5≠4, so condition will fail and it not allow T1 to enter the critical section.

Hence, the implementation ***FAILS.***