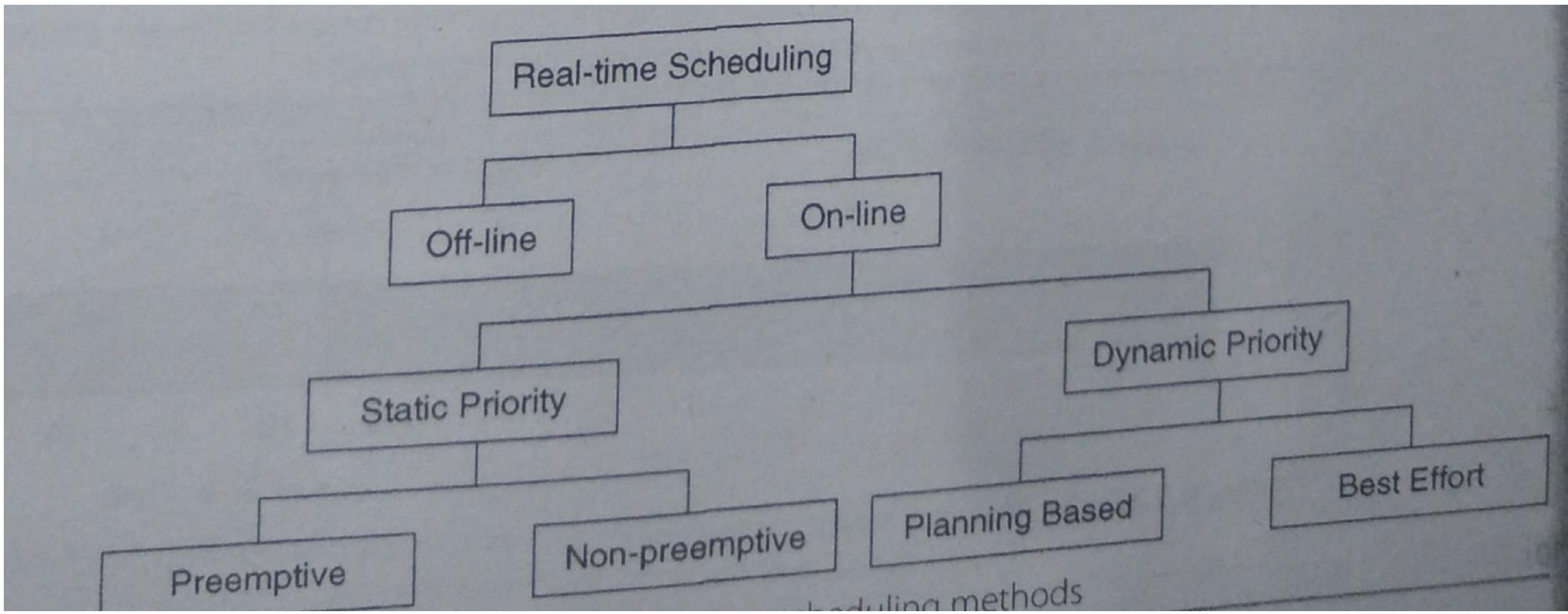
Real-Time Operating System
Chapter 8

Embedded System Design

UCS614

USE F5 to listen the audio embedded in PPT

Real-time Scheduling Algorithms



Real-time Scheduling Algorithms

- Off Time Scheduling
  - Generate scheduling information prior to system execution
  - Scheduling is based on knowledge of release time, deadlines and execution time for all the tasks
  - This is deterministic system model
  - Characteristics of the tasks are known ‘a priori’
  - Disadvantage is the inflexibility

Real-time Scheduling Algorithms

- On Line Scheduling
  - Parameters of the task and the number and types of tasks are not known a priori
  - Scheduler must accommodate dynamic changes in the user demand and availability of resources.
  - Possibly not able to make best use of all resources

Real-time Scheduling Algorithms

Example:

Tasks	Priority	Period	CPU Burst
T_1	1	7	2
T_2	2	17	4
T_3	3	24	8

Schedule the tasks

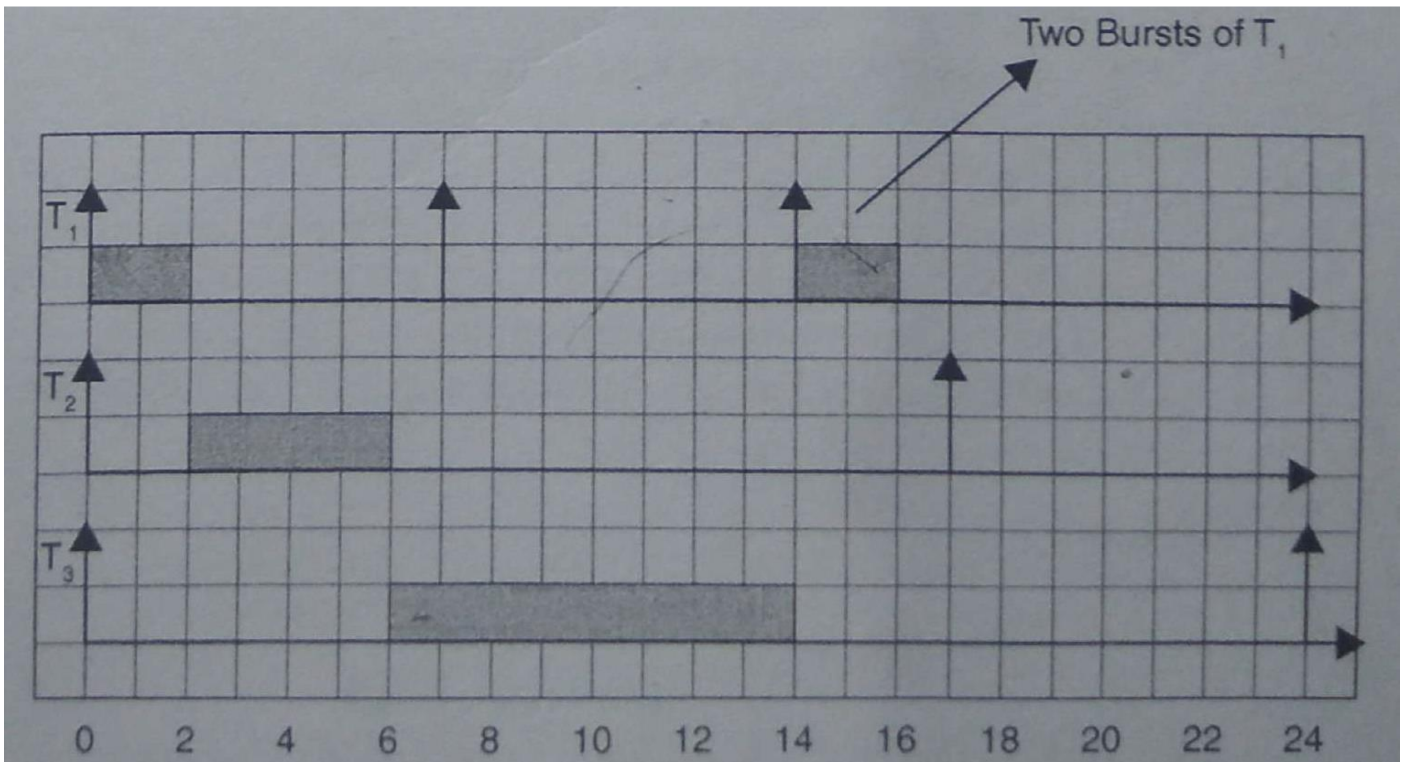
i) Without pre-emption

ii) With pre-emption

Real-time Scheduling Algorithms

Solution

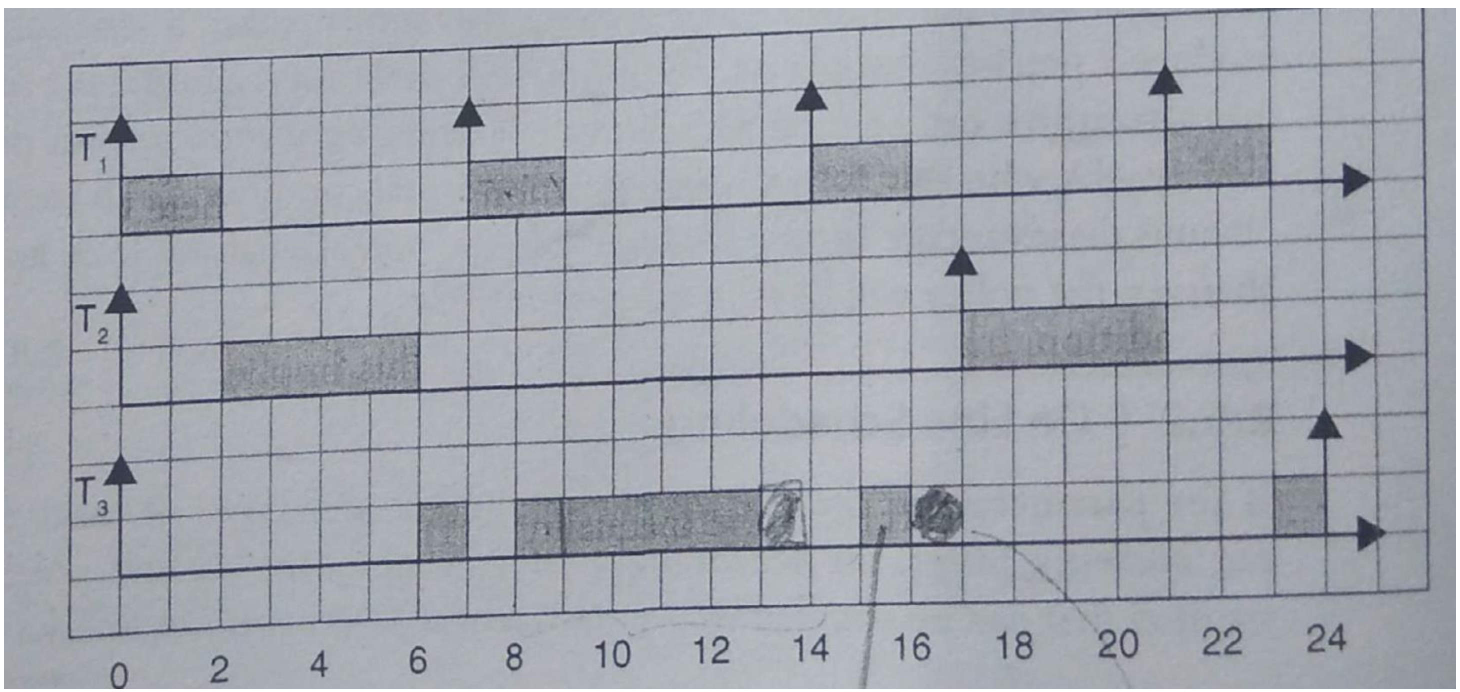
Without pre-emption



Real-time Scheduling Algorithms

Solution

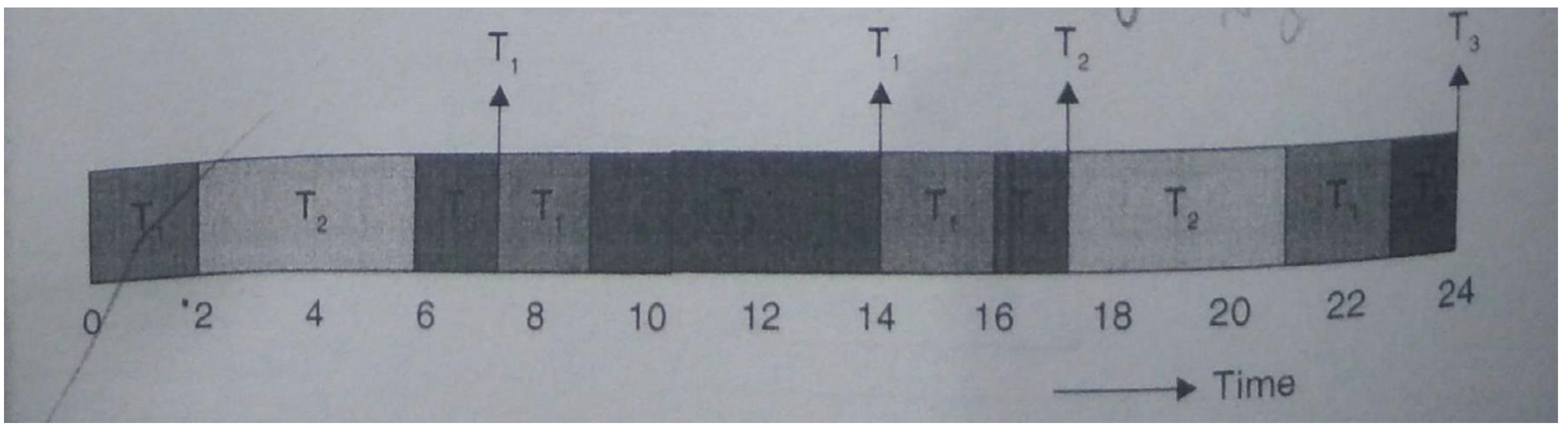
With pre-emption



Real-time Scheduling Algorithms

Solution

With pre-emption



Rate Monotonic Algorithm

- Assigning priorities as a monotonic function of the rate of a (periodic) process
  - Period increases, the priority decreases
  - Process of lowest period will get the highest priority
- Sufficient condition for ‘scheduling’ using the RM algorithm

$$\sum_{i=1}^{n} C_i / P_i \leq n(2^{1/n} - 1)$$

Rate Monotonic Algorithm

$$\sum_{i=1}^{n} C_i / P_i \leq n(2^{1/n} - 1)$$

Table 8.2 | Rate Monotonic Schedulable Bound (RHS of Inequality 8.1)

Task Set Size (n)	Schedulable Bound
1	1
2	0.828
3	0.780
4	0.757
5	0.743
6	0.735
------	-----
infinity	ln2

Rate Monotonic Algorithm

- RM algorithm uses static priority with pre-emption.

Rate Monotonic Algorithm

Example 2:

Table 8.3

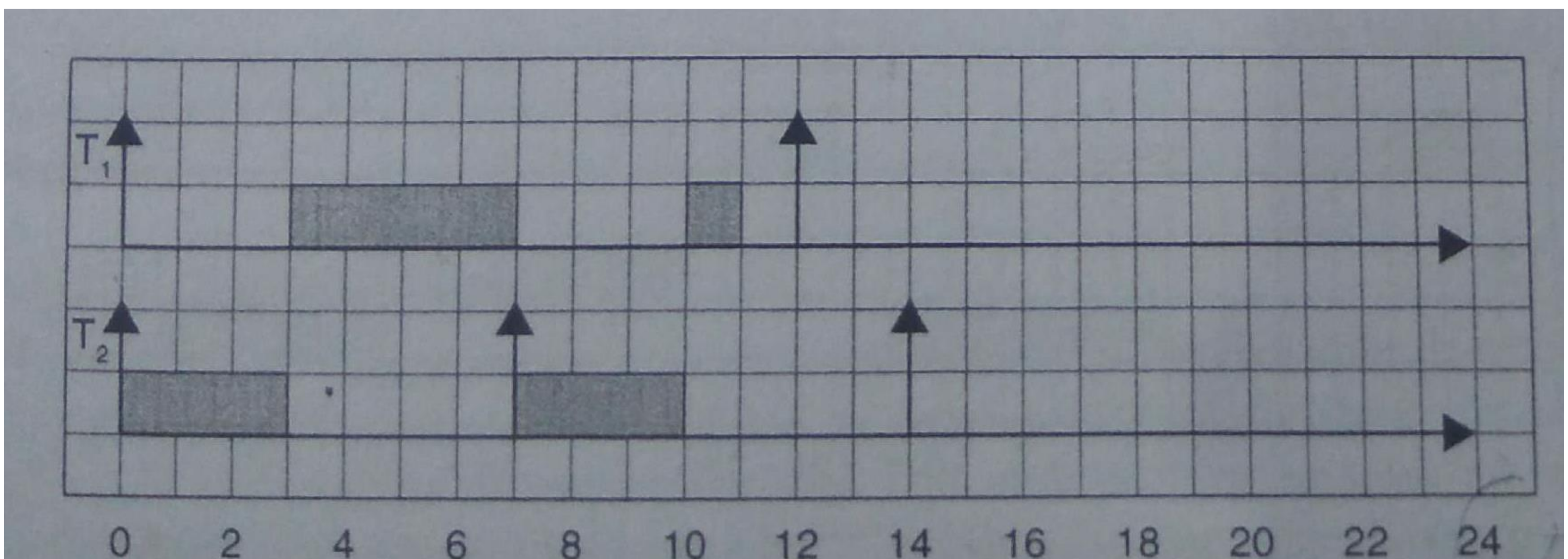
Tasks	Period	CPU Burst
T_1	12	5
T_2	7	3

Rate Monotonic Algorithm

Solution:

CPU Utilization 5/12 + 3/7 =0.844

RHS of inequality = 0.828



Rate Monotonic Algorithm

Example 3:

Table 8.4

Tasks	Period	CPU Burst
T_1	15	4
T_2	12	2
T_3	20	5

Rate Monotonic Algorithm

Solution:

CPU Utilization 4/15 + 2/12 + 5/20 =0.684

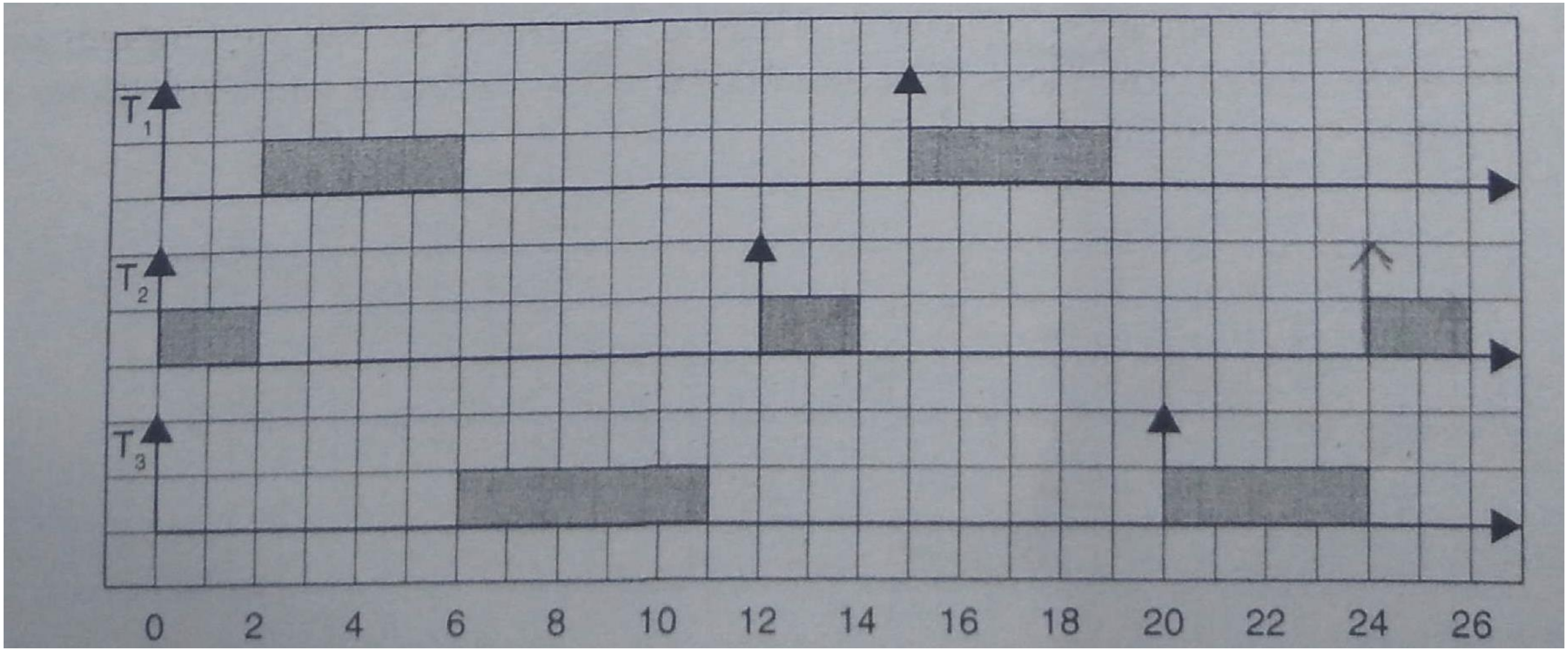
RHS of inequality = 0.782

Since LHS < RHS, sufficient condition is satisfied.

The task set is definitely schedulable.

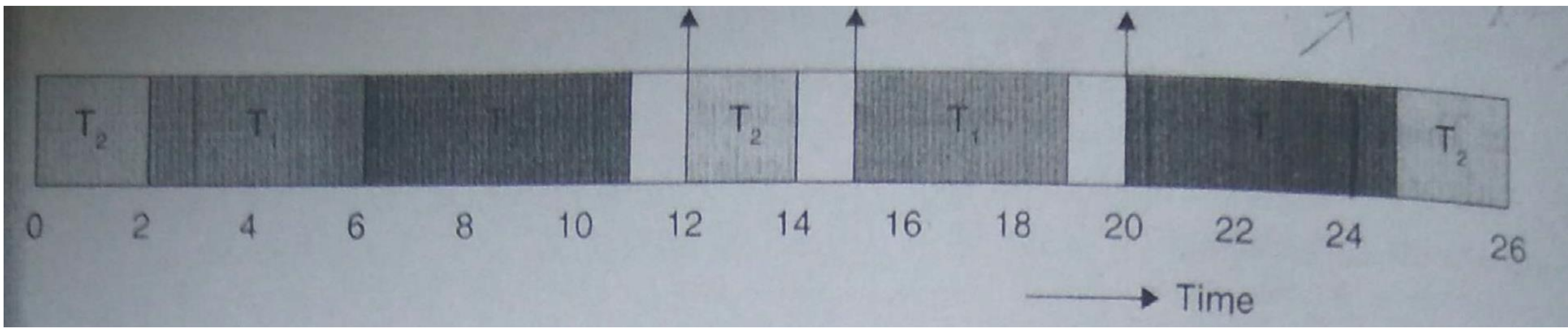
Rate Monotonic Algorithm

Solution:



Rate Monotonic Algorithm

Solution:



Rate Monotonic Algorithm

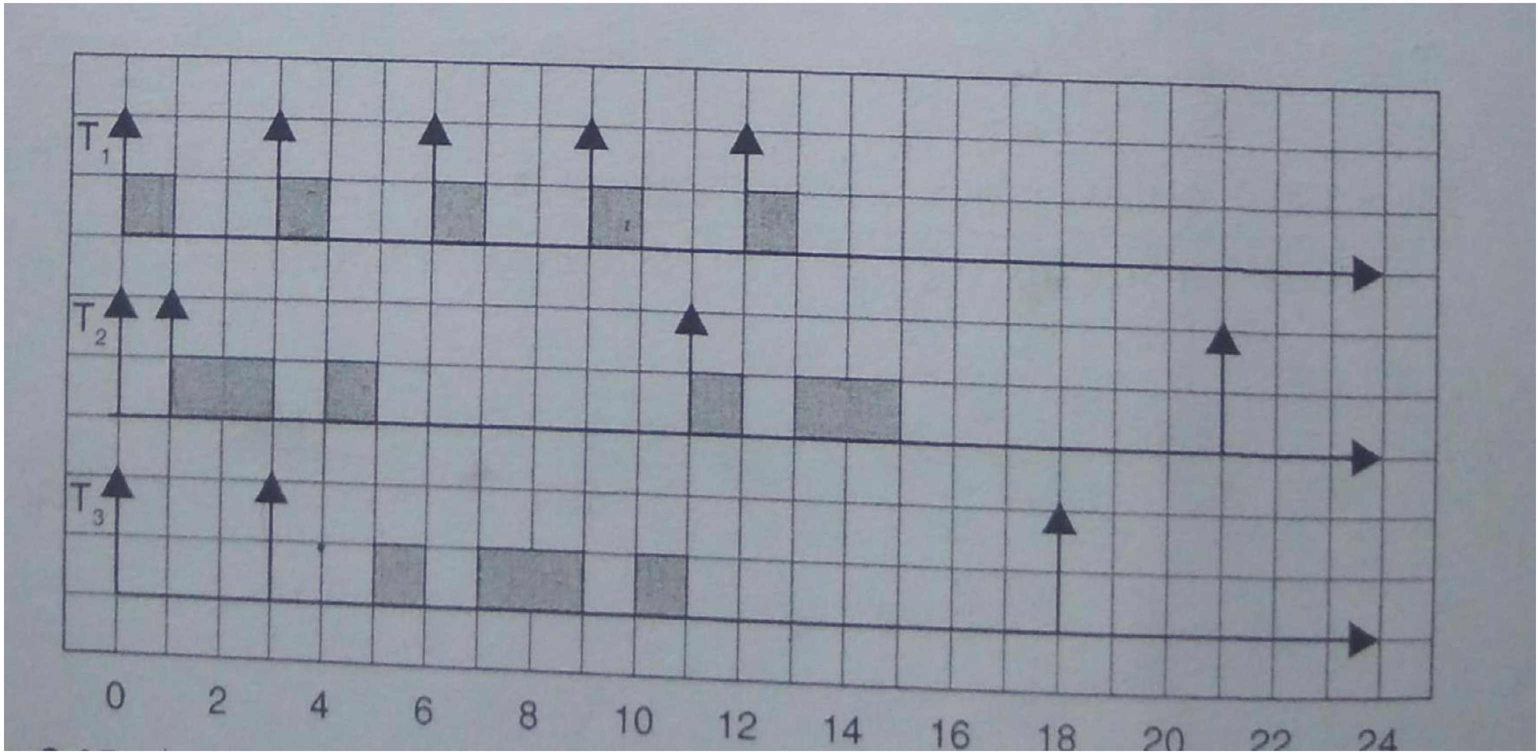
Example 4:

Table 8.5

Tasks	Period	CPU Burst	Release Time
T_1	3	1	0
T_2	10	3	1
T_3	15	4	3

Rate Monotonic Algorithm

Solution:



Earliest Deadline First

- Dynamic priority allocation
- Priority changes at run time
- Highest priority task is one that has closest deadline
- Task that can not be scheduled using RM, can be scheduled by EDF

Earliest Deadline First

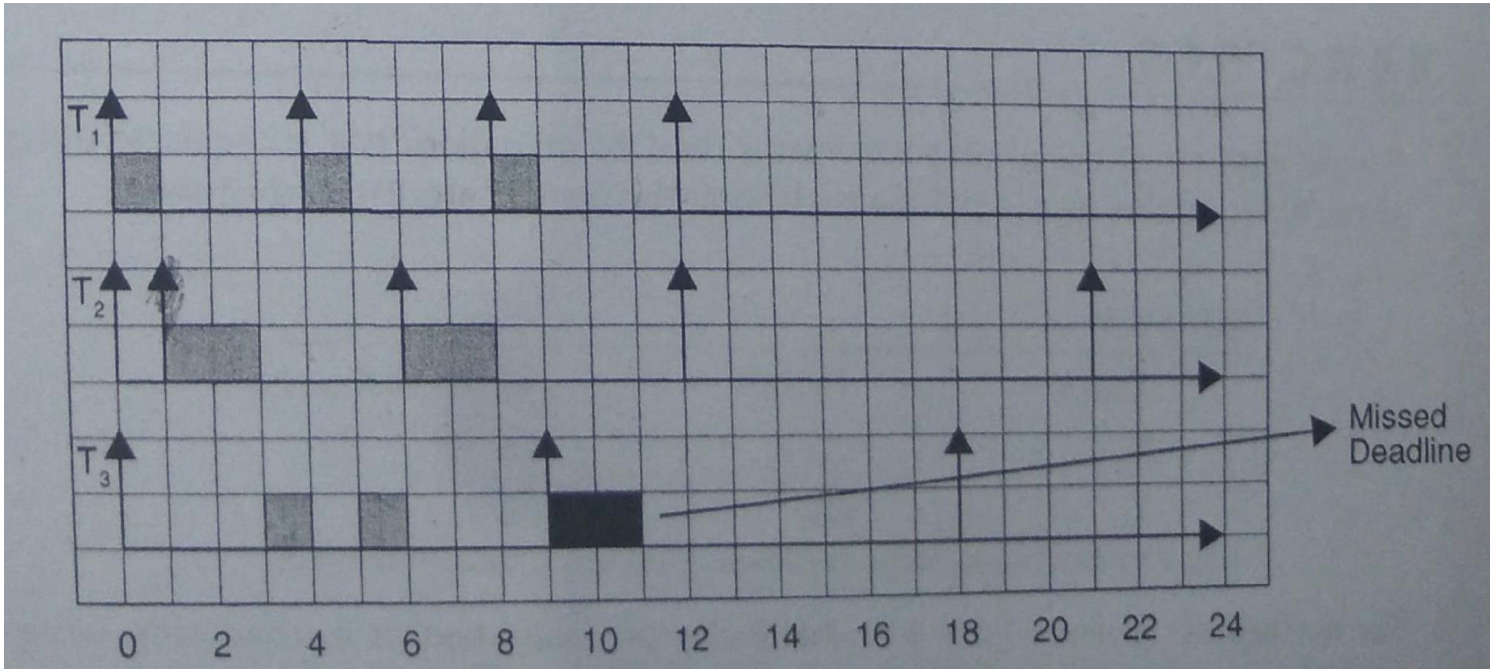
Example 5:

Table 8.6

Tasks	Period	CPU Burst
T_1	4	1
T_2	6	2
T_3	9	4

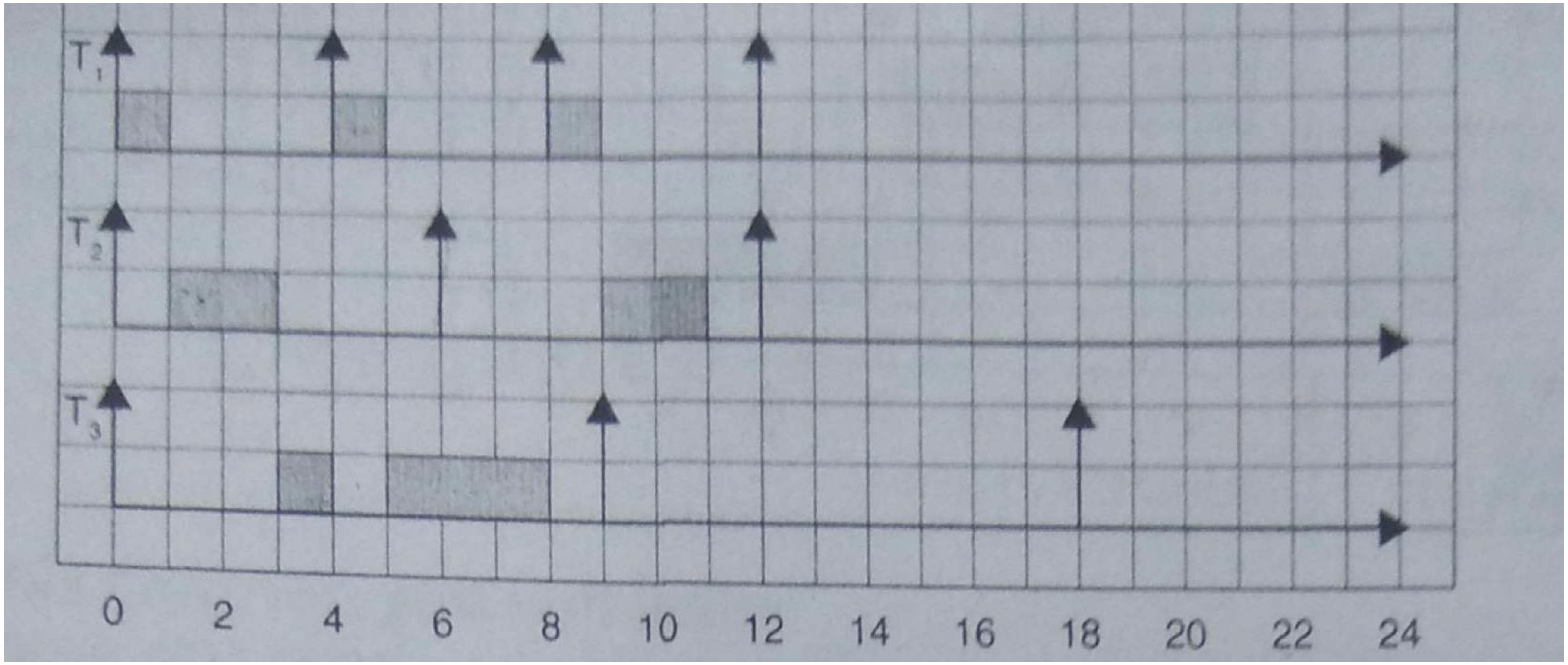
Earliest Deadline First

Solution: Not schedulable using RM



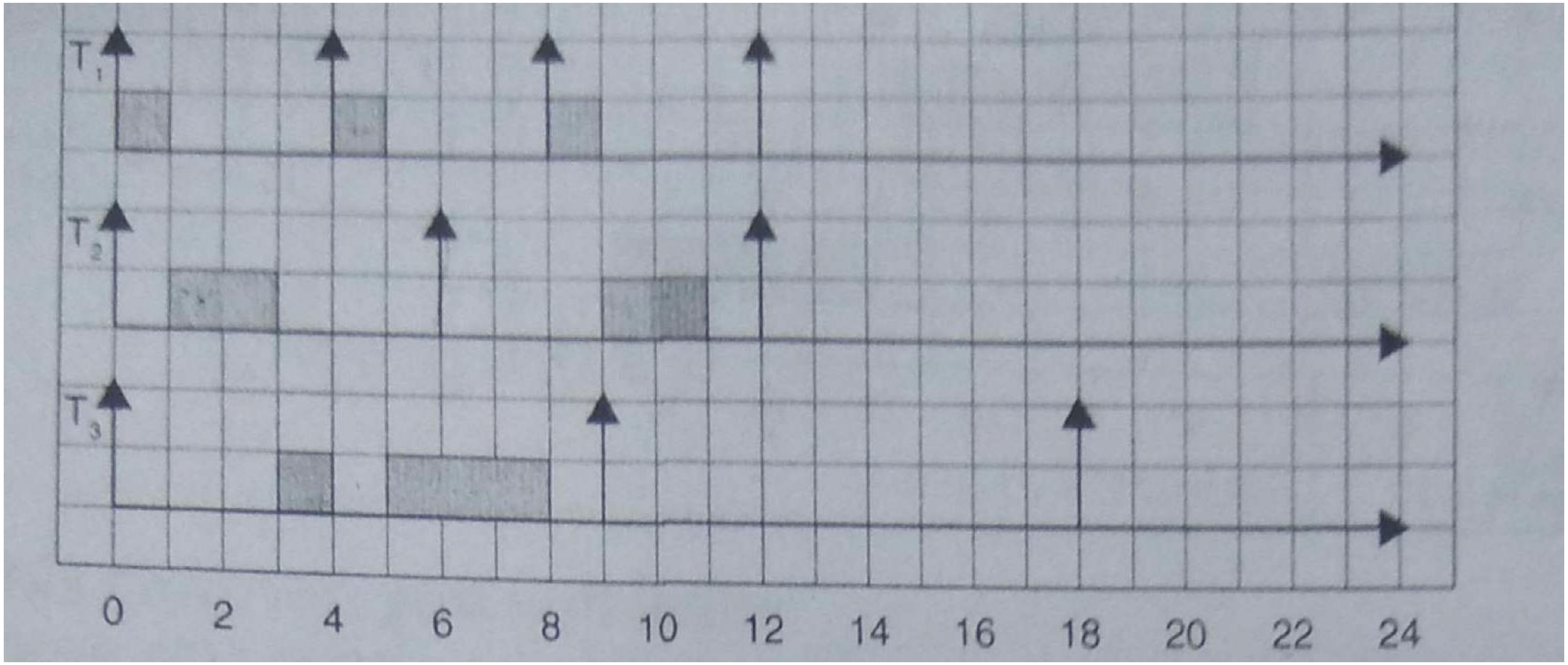
Earliest Deadline First

- Solution: Not schedulable using EDF also. Wrong solution in Book.



Earliest Deadline First

Solution: Not schedulable using EDF also. Wrong solution in Book.



Earliest Deadline First

Solution: