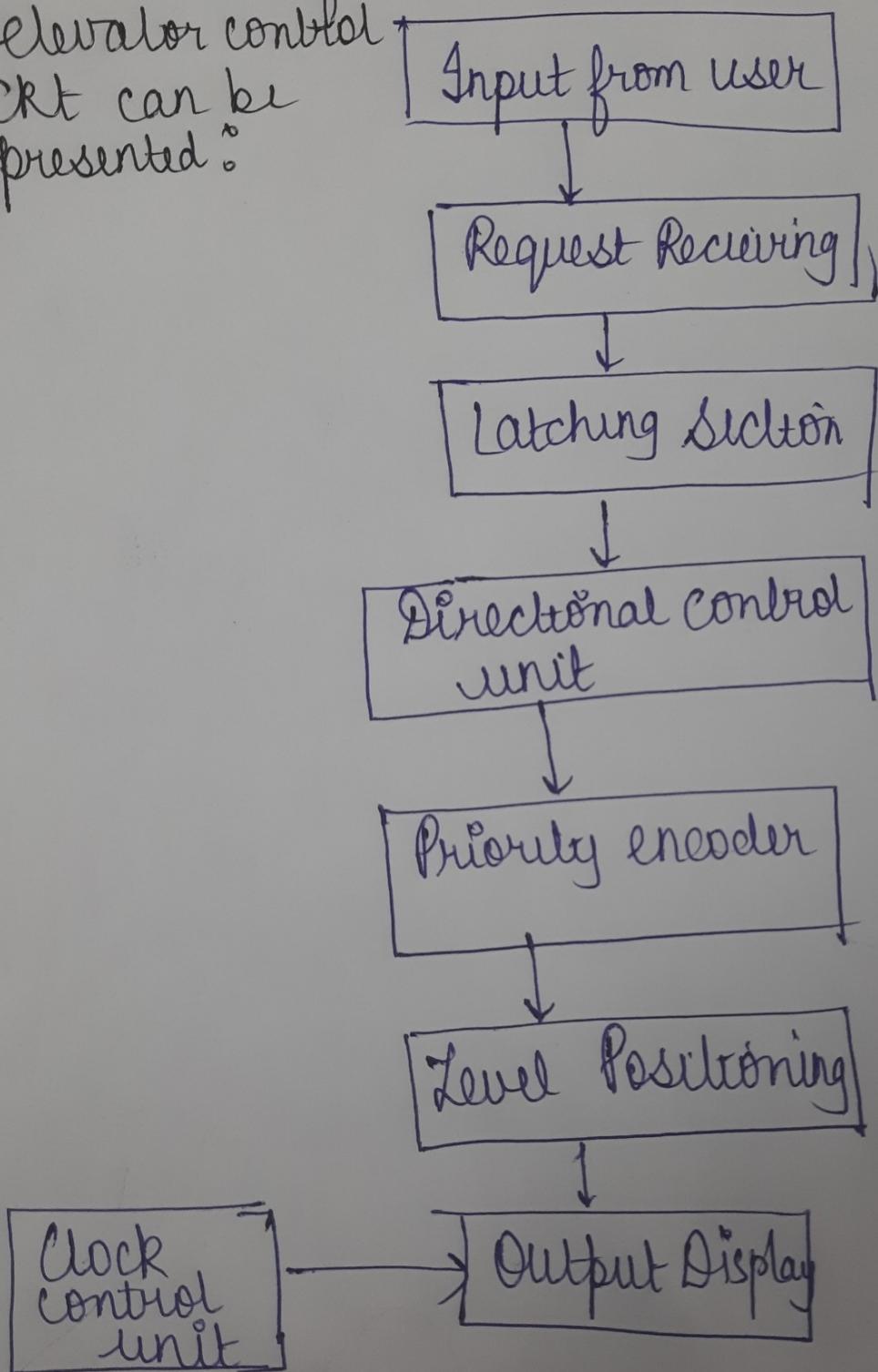


FUNCTIONAL DIAGRAM

→ The entire functional block diagram of the elevator control CRT can be presented:



Position of the elevator	Symbol	Event
② At 1 st floor	STATUS-1ST	<p>GF is pressed from inside</p> <p>2F is pressed from inside</p> <p>3F is pressed from inside</p> <p>4F is pressed from inside</p> <p>G_UP is pressed from outside (Ground floor)</p> <p>1_UP is pressed from outside (1st floor)</p> <p>2_UP is pressed from outside (2nd floor)</p> <p>3_UP is pressed from outside (3rd floor)</p> <p>4_UP is pressed [4th floor]</p> <p>1_DOWN is pressed from outside (1st floor)</p> <p>2_DOWN is pressed from outside (2nd floor)</p> <p>3_DOWN is pressed from outside (3rd floor)</p> <p>4_DOWN is pressed from outside</p>

Design of Elevator

Table 1: Sample inputs and Requested Instructions

Position of user	Symbol	Requested Instruction
Inside of elevator	GE	To go to ground floor
Inside of elevator	1F	To go to 1 st floor
Inside of elevator	2F	To go to 2 nd floor
Inside of elevator	3F	To go to 3 rd floor
Inside of elevator	4F	To go to 4 th floor
Outside the elevator	G UP	To go upwards from ground floor
Outside the elevator	1-DOWN	To go downward from 1 st floor
Outside the elevator	1-UP	To go upward from first floor
Outside the elevator	2-UP	To go upward from 2 nd floor
Outside the elevator	3-UP	To go upward from 3 rd floor
Outside the elevator	3-DOWN	To go downward from 3 rd floor
Outside the elevator	4-DOWN	To go downward from 4 th floor

Table II

Position of the elevator	symbol	event
① At group floor	Status-ground	1F is pressed from inside 2F is pressed from inside 3F is pressed from inside 4F is pressed from inside 1-DOWN is pressed from inside 4F is pressed from inside 1-DOWN is pressed from outside [from 1 st floor] 1-UP is pressed from outside [first floor]
		2-UP, 4-UP, 2-DOWN, 3-DOWN-4DOWN
		Similarly 3 rd , 4 th

Table FF

Position of elevator ⑤ At 4 th floor	Symbol	Event
	STATUS - 4 th	
	G_E is pressed	inside
	1F is pressed	inside
	2F is pressed	inside
	3F is pressed	inside
	G_UP is pressed outside	
	1_UP is pressed outside	
	2_UP is pressed outside	
	3_UP is pressed outside	
	4_UP is pressed outside	
	1_DOWN is pressed outside	
	2_DOWN is pressed outside	
	3_DOWN is pressed outside	
	4_DOWN is pressed outside	

Table H

Position of elevator ⑤ At 4 th floor	Symbol STATUS - 4 th	Event
		G_E is pressed inside
		1F is pressed inside
		2F is pressed inside
		3F is pressed inside
		G_UP is pressed outside
		1_UP is pressed outside
		2_UP is pressed outside
		3_UP is pressed outside
		4_UP is pressed outside
		1_DOWN is pressed outside
		2_DOWN is pressed outside
		3_DOWN is pressed outside
		4_DOWN is pressed outside

Position of elevator	symbol	Event
At 3rd floor	STATUS_3RD	<p>GF is pressed inside</p> <p>1F is pressed inside</p> <p>2F is pressed inside</p> <p>4F is pressed inside</p> <p>G_UP is present outside</p> <p>L_UP is pressed outside</p> <p>2_UP is pressed outside</p> <p>3_UP is present outside</p> <p>4_UP is pressed outside</p> <p>1_DOWN is pressed outside</p> <p>2_DOWN is pressed outside</p> <p>3_DOWN is pressed outside</p> <p>4_DOWN is pressed outside</p>

Position of elevator	Symbol	Event
① at 2 nd floor	STATUS 2N	Gf is present from inside
	---	1f is present from inside
	---	2F is present from inside
	---	4f is present from inside
	---	Q_UP is present outside
	---	1_UP is present outside
	---	2_UP is present outside
	---	3_UP is present outside
	---	4_UP is present outside
	---	1_DOWN is present outside
	---	2_DOWN is present outside
	---	3_DOWN is present outside
	---	4_DOWN is present outside

Reuse Logic Patterns by Using atomic subcharts.

The elevator cars used identical request queues. The elevator system chart models their behaviour by using linked atomic subcharts from a library model.

In library model, the chart Elevator implements the logic for a generic elevator car.

To program the subcharts Elevator-A and Elevator-B so that they control the appropriate car, you map data and events in each subchart to the corresponding data and events in the main chart. For instance, for Elevator-B:

- The subchart input floor-request maps to the chart input CAR-B-floor-request.
- The subchart output position maps to the chart output doorBOpen
- The subchart output doorOpen maps to the chart output doorBOpen
- The subchart event CAR-CALL maps to the chart event CAR CALL-B.

Problem statement 8

Manage Request from user Interface

- The elevator system chart consist of three parallel subcharts. Each of these subcharts manage a queue of requests from the UI.
- The Elevator-Manager subchart implements the main control logic for the elevator system. This subchart processes these requests and delegates them to one of the elevator cars, depending on availability and proximity to the request.
- Elevator_A and Elevator_B represent the logic for the two elevator cars. Each car has its own queue that holds all of its floor requests. Floor requests are generated when you click a button inside the elevator car or when you click a button inside the elevator car or when the Elevator-Manager delegates a request from hall queue to the car.
- To convert the duplicate subcharts to atomic subcharts, first make a library atomic subcharts.
- Then use linked instance of this library to replace the duplicate elevator car subcharts.

Manage Request from user Interface

- The elevator system chart consist of three parallel subcharts. Each of these subcharts manage a queue of requests from the UI.
- The Elevator-Manager subchart implements the main control logic for the elevator system. This subchart processes these requests and delegates them to one of the elevator cars, depending on availability and proximity to the request.
- Elevator_A and Elevator_B represent the logic for the two elevator cars. Each car has its own queue that holds all of its floor requests. Floor requests are generated when you click a button inside the elevator car or when you click a button inside the elevator car or when the Elevator-Manager delegates a request from hall queue to the car.
- To convert the duplicate subcharts to atomic subcharts, first make a library atomic subcharts.
- Then use linked instance of this library to replace the duplicate elevator car subcharts.

1. Migrate these functions from the Elevator-Manager subchart into the parent chart.
 - exists_in_queue
 - deregister
 - dequeue

Rename these functions to distinguish them from the functions inside the elevator car subcharts.
2. Using the model Explorer, migrate these variables from the Elevator-Manager subchart into the parent chart.
 - hall_call_queue
 - hall_call_status
3. In the elevator system chart, set the Export chart level Functions chart property to true.
4. Modify the Elevator-Manager and Elevator subcharts to use the migrated functions and variables.
5. Create a library atomic subchart from the Elevator-A subchart, as described in State Multiple Times in a chart. Reuse a state of the atomic subchart to pass the elevator car to the chart, change the scope of the data position from local to own.
6. To enable the position containing subchart