Center for Embedded Computer Systems
University of California, Irvine

System-Level Modeling and Simulation of an Elevator Control System

Daniel Castellanos, Rainer Dömer

Technical Report CECS-07-04
June 25, 2007

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-3425, USA
(949) 824-8059

{dcastell, doemer}@uci.edu
http://www.cecs.uci.edu/

System-Level Modeling and Simulation of an Elevator Control System

Daniel Castellanos, Rainer Dömer

Technical Report CECS-07-04
June 25, 2007

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-3425, USA
(949) 824-8059

{dcastell, doemer}@uci.edu
http://www.cecs.uci.edu

Abstract

The design complexity of embedded systems is rapidly increasing as more functionality and higher performance is demanded. As a result, the design process must be broken down into multiple steps to achieve system requirements. In this paper, we employ such a design process by presenting a case study for the modeling and simulation of an Elevator Control System (ECS). Our system model combines instantiations of Elevator Control Units (ECU) which communicate through a central broadcast channel implemented by a Controller Area Network (CAN) bus. Our generic model was specified using a System Level Design Language (SLDL), SpecC. In particular, our ECS was customized to simulate the behavior of four elevator shafts operating in a ten story building. The model was successfully specified and simulated at three abstraction levels: the specification, transaction, and bus functional level.

Contents

List of Figures

List of Acronyms

BFM Bus Functional Model. A wire accurate and cycle accurate model of a bus.

CAN Controller Area Network. A serial communications protocol with a focus for automotive application.

ECS Elevator Control System. An embedded system that models an elevator system by instantiating multiple elevator control units.

ECU Elevator Control Unit. An individual system that performs a specific task for an elevator.

SLDL System Level Design Language.

SOC System-on-Chip.

TLM Transaction Level Model. A model of a system in which communication is described as transactions, abstract of pins and wires.

System-Level Modeling and Simulation of an Elevator Control System

Daniel Castellanos, Rainer Dömer

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-3425, USA

{dcastell, doemer}@uci.edu
http://www.cecs.uci.edu

June 25, 2007

Abstract

The design complexity of embedded systems is rapidly increasing as more functionality and higher performance is demanded. As a result, the design process must be broken down into multiple steps to achieve system requirements. In this paper, we employ such a design process by presenting a case study for the modeling and simulation of an Elevator Control System (ECS). Our system model combines instantiations of Elevator Control Units (ECU) which communicate through a central broadcast channel implemented by a Controller Area Network (CAN) bus. Our generic model was specified using a System Level Design Language (SLDL), SpecC. In particular, our ECS was customized to simulate the behavior of four elevator shafts operating in a ten story building. The model was successfully specified and simulated at three abstraction levels: the specification, transaction, and bus functional level.

1 Introduction

The massive success of portable electronic devices has initiated a demand for complex and computationally intensive embedded systems. The time to market and functionality of these devices has become highly competitive. Companies urge system designers to develop better systems in a shorter time. Consequently, the design process becomes a greater challenge.

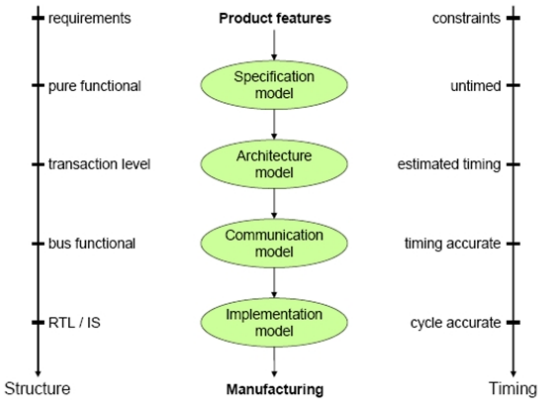

Figure 1: Abstraction Levels in Embedded System Design [2].

Under these circumstances, designers model a System-on-Chip (SOC) using multiple levels of abstraction. Figure 1 describes the typical design flow of an embedded system. First, a system must be proposed with well-defined requirements which define the system behavior. This forms a specification model. Models are further refined until a synthesizable representation is developed. Embedded systems are also designed by exploiting hierarchy. Hierarchical design is critical with large systems because it reduces complexity and eases debugging.

System Level Design Languages (SLDL), such as SpecC [1] and SystemC [3], provide a viable solution to modeling and designing highly complex systems. SLDLs assist the design flow described in Figure 1 by supporting modular design, executability, synthesizability, and simplicity.

1.1 Case Study: An Elevator Control System

This paper will describe the modeling process of an Elevator Control System (ECS) using SpecC. The proposed system will act as a set of elevators typically found in a high-rise building. The ECS will be a distributed embedded system, consisting of a set of communicating Elevator Control Units (ECU) implemented as a System-on-Chip (SOC). Our design will be described at roughly the first three levels illustrated in Figure 1. In addition, the ECS will be modeled as a set of Elevator Control Unit (ECU)s, each ECU representing a subcomponent of the system. A hierarchy of subcomponents will be modeled and validated individually.

1.2 Paper Organization

The following section describes the design of the ECS. Our case study will design the ECS using three levels of abstraction: the specification model, the Transaction Level Model (TLM), and the Bus Functional Model (BFM). The behavior of each ECU is described separately. Section 3 will then discuss the simulation environment and parameters used to validate the ECS model.

2 Elevator Control System

In this section, we will describe the design of an Elevator Control System (ECS). The purpose of this section is to give some insight on how a large system can be broken into subsystems and incorporated to work together. The design process can be described in four parts. First, each subsystem is specified and modeled. Second, a hierarchical structure is constructed using subcomponents. Third, a communication channel is integrated to allow subsystems to communicate with each other. Lastly, each subsystem is integrated and modeled as one system.

2.1 System Overview

The ECS is modeled as a set of multiple ECUs, all running in parallel. All ECUs are individual systems, but behave as subsystems embedded within the ECS. Each ECU is responsible for a distinct task within the system. The design complexity is reduced by modeling each ECU separately. All ECUs are typical of an elevator system. Table 1 lists the eight ECUs modeled.

Table 1: Elevator Control Unit Descriptions

	Elevator Control Unit	Functionality
1	Floor Panel	Panel at each floor and each shaft with up-down controls
2	Floor Display	Display of current floor and direction at each floor
3	Floor Door	Control unit to open-close doors at each floor
4	Car Panel	Panel in each car with request controls
5	Car Display	Display of current floor and direction in each car
6	Car Door	Control unit to open-close door at each car
7	Motor Controller	Control unit for the motor atop each shaft
8	Main Control Unit	Central control to control the entire ECS

2.2 Modeling ECUs

A specification model requires well defined requirements and behaviors. The ECUs are carefully modeled using these requirements. The ECS consists of eight distinct ECUs. The structure and behavior of each ECU is described below.

2.2.1 Door Units

The floor and car door units maintain control of the elevator doors. Their structure and behavior are similar, but placed differently within the ECS. For example, an elevator shaft can consist of only one car door and several floor doors, one for each floor. The task of the door units is simply to open and close the door when signaled. The structure of these systems consists of an output port to signal close or open. These events are triggered by listening to a bidirectional communication port. Figure 2 illustrates the structure of the door units.

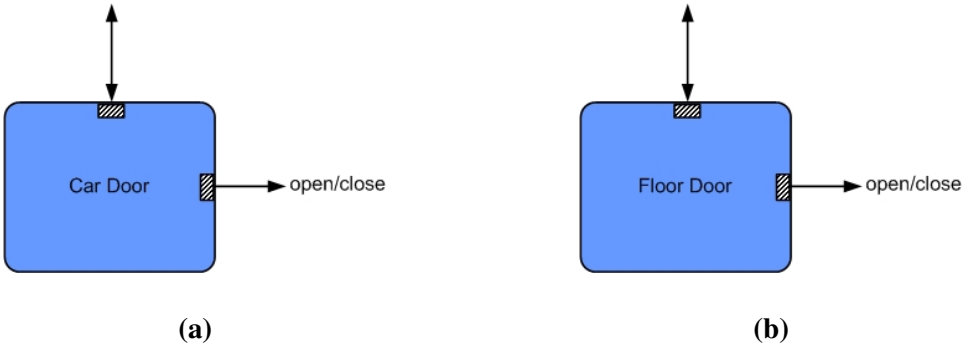

Figure 2: Car and Floor Door Units.

2.2.2 Display Units

Floor and car display units are distributed similarly as the door units. A floor and car display system continuously displays the location and the direction of movement, if any, on a monitor. These two systems consist of three output ports. One to indicate the floor number, and two for direction. Direction can be identified with only one port, but a non-moving state requires the addition of the second port. The internal behavior is also triggered by a bidirectional communication port. Figure 3 shows the structure of the floor and car display units. When an update is received, the display units are updated accordingly.

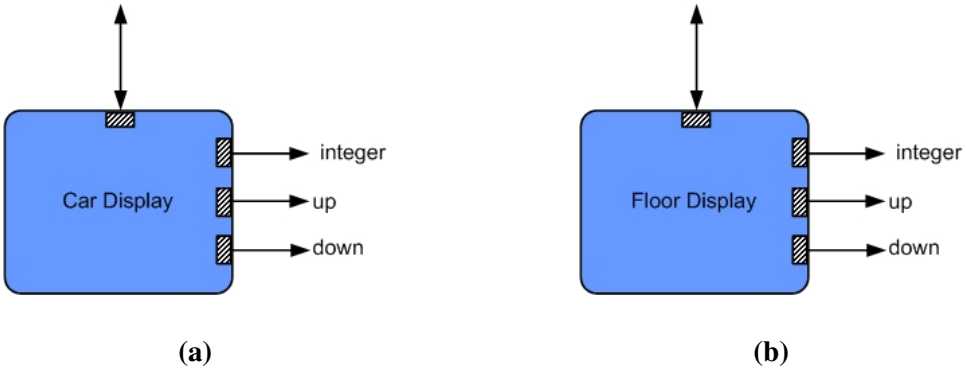

Figure 3: Car and Floor Display Units.

2.2.3 Panel Units

The floor panel's purpose is to dispatch elevator requests from outside the elevator. It consists of two buttons that act as inputs to specify the direction of the request. Floor panels are typically designed with a LED light as a means for acknowledgement. Therefore, a floor panel unit can be modeled with two input ports and two output ports for requests and LEDs respectively. Internally,

the floor panel constructs a message with the request and sends it out. A floor request panel receives messages to update the LED request buttons. Therefore, a bidirectional port is also required for communication purposes.

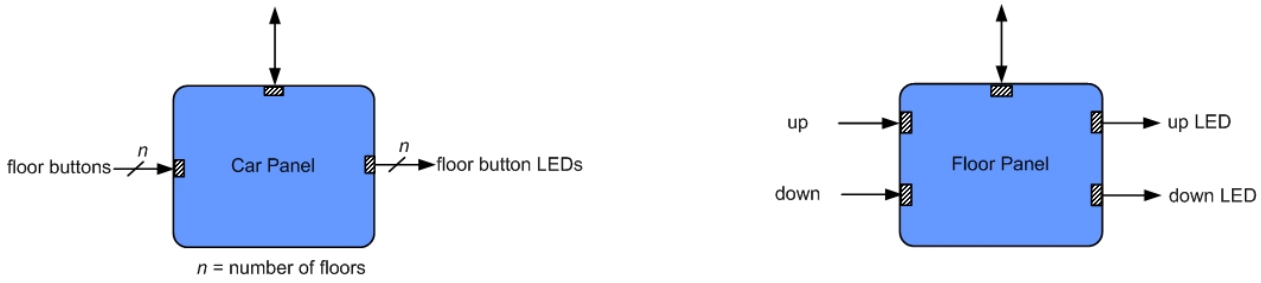

Figure 4: Car and Floor Panel Units.

The structure of the car panel is different compared to the floor panel. Figure 4 shows the structural differences. In the car panel, instead of buttons to identify direction, the buttons represent floor numbers. The floor numbers are represented by a bus-like wired port. In essence, the input port is a bundle of wires, where each wire represents a floor number. A similar output port is needed for each LED light used for acknowledgement (as with the floor panel). The port is continuously monitored. A state change in any of the wires triggers a request. The bidirectional communication port is used to send out requests and to update the state of the LED lights for each button.

2.2.4 Motor Controller

The motor controller unit manages the position of the elevator car. Structurally, it is formed with two output ports used to enable/disable the motor and to indicate the direction of movement. Figure 5 shows the structure of the motor controller. The output ports are designed as inputs to the motor that moves the elevator. It must work closely with the motor to identify its location. The motor controller must also work closely with the main control unit. This interaction is conducted through the bidirectional communication port. The motor controller and the main control unit communicate through this port to ensure requests are completed. By request of the main control unit, the motor controller determines the direction to move in order to satisfy the request. It disables the motor when it has moved to a different floor to allow the main controller to determine if it should continue moving. However, if the elevator is located at the requested floor, it composes a series of messages to indicate its floor location, its direction and announces it has arrived at a requested floor.

2.2.5 Main Controller

The main control unit is the most complex ECU. It is the centralized control element of the ECS. It works closely with all processing elements by delegating and handling all requests that arrive in

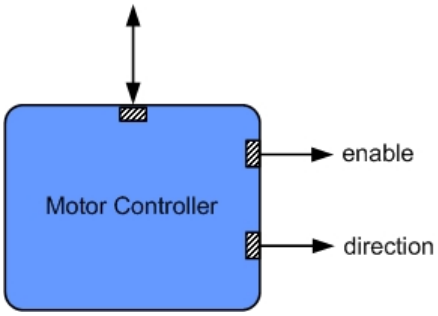

Figure 5: Motor Controller Structure.

the system. The motor controller and main control unit process most of the control of the ECS. The main controller structure, however, is much simpler. Figure 6 shows the structure of the main controller. It consists of only one bidirectional communication port. In contrast, the behavior is more complex. The overall range of supported features that are designed in each ECU can vary from very simple to highly complex. A main control unit must keep track of requests and manage stability. The focus of this study is to properly model and describe the basic behavior of an elevator control system. Therefore, the main control unit only supports essential features.

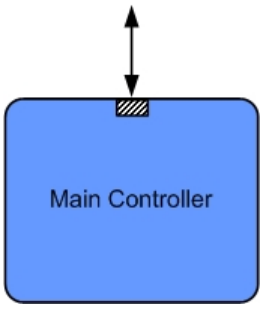

Figure 6: Main Controller Structure.

The main control unit performs three basic tasks. First, it keeps track of all requests made by the floor and car panels. It uses this information to verify if a request has been completed. Second, it is responsible for making decisions and assigns commands. In a request initiated by a floor panel, the main control unit determines which elevator will service the request. Therefore, the main control unit also monitors every elevators floor location. Lastly, it is also responsible for notifying all ECUs once a request has been completed.

Our implementation of the main controller is primitive in design and imposes some limitations. For example, car requests have priority. Therefore, if all shafts are active, no elevator car will ever arrive to a floor request. If multiple car requests are made in a car shaft, only the first request is serviced. Clearly, these limitations have solutions which can be added in future work. Ultimately the design of the main controller can be expanded to support a multitude of features. For simplicity sake, these features are omitted from this implementation.

2.3 System Integration

System integration issues can be simplified if a hierarchical structure is implemented. The modularity of these models also makes adding features easier. In the ECS, multiple units can be instantiated. This requires the ECU structures to include ports for identification purposes. To support this, the port can be declared with a hardwired identification number. For example, a car panel in elevator three can have a three as a shaft ID. Furthermore, a floor panel will also need an identifying floor number. This can be modeled by adding yet another port dedicated to identify the floor ID. In the ECS two main hierarchical components are grouped, the floor and car units. A floor unit consists of a panel, display, and a door. Similarly, a car unit is composed of these units as well. These structures are illustrated in figure 7. For clarity, the ports unique to each structure have been omitted, showing only the identification and communication port.

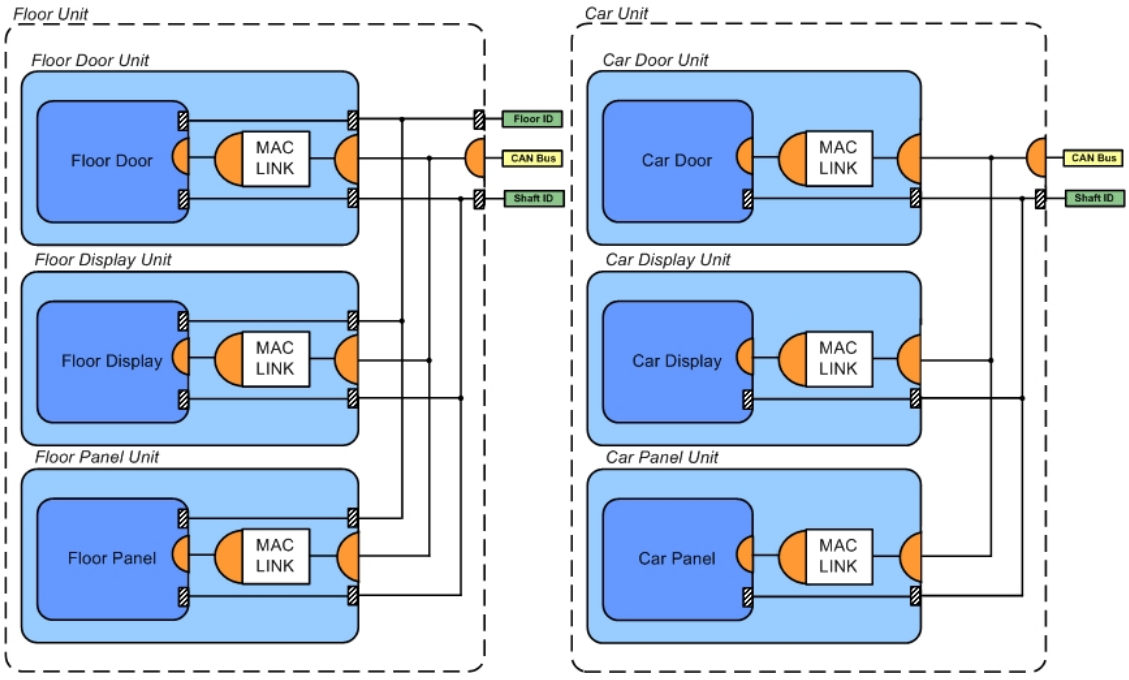

Figure 7: The Floor and Car Unit Hierarchical Structure.

Hierarchical modeling can be further applied. The car unit and floor unit can be subcomponents of a new structure, a shaft unit. In this module, the motor control unit is also integrated within the shaft unit. An ECS system can modeled by instantiating a shaft unit and a main controller. Multiple elevators can be declared by instantiating multiple shaft units. Therefore by implementing a hierarchical model, a generic model is designed.

2.3.1 ECU Communication

It is clear that ECUs heavily rely on a bidirectional communication port. This is the primary communication medium of all ECUs. The ECS model employs a broadcast channel which in our implementation is a Controller Area Network (CAN) bus [4]. The CAN bus is a widely implemented broadcast communication protocol that is widely adopted for automotive purposes. The specification model is then refined to a Transaction Level Model (TLM). The TLM also utilizes the CAN bus model as a communication channel. In fact, each ECU is wrapped around a structure that implements the MAC link channel. In Figure 7, the bidirectional port has been replaced with a channel interface.

Further refinement of the ECS can be applied using the CAN bus model. In the Bus Functional Model (BFM) the timing of bus can be properly modeled. The structure of the BFM becomes slightly different. A communication channel is no longer shared by all subcomponents. Instead, the CAN bus communication port is embedded within each ECU and connected via a bus. This model can accurately simulate the timing of the ECS.

The CAN bus supports the broadcast of messages to all units within the ECS. A message structure is defined and supported by all ECUs. Nine message types were designed for ECU communication. Each message is broadcast to all ECUs (except for the one who created the message). The ECUs continuously listen through the communication channel for any incoming messages. Although, all ECUs receive every message, each unit is responsible for identifying if it is the proper destination. The shaft and floor identification number are compared to those specified in the message. If a message is not addressed to a specific ECU, it is simply ignored. The following are the nine messages an ECU can receive and/or transmit.

Floor Request A floor request message is created by a floor panel as a response to a user. The message consists of the floor ID and the direction. It is used by the main control unit and other floor panels sharing the same floor ID.

Car Request A car request is similar to a floor request, but originates from within the elevator. Therefore, the elevator shaft ID is specified. This message consists of a shaft ID and a destination floor number. This message is destined to the main control unit.

Open Door An open door request is created by the motor controller when it is signaled by the motor that it has arrived at a requested floor. The message is composed of the floor ID and shaft ID. It signals both the car door and the floor door units.

Close Door A close door message, does as named, closes the car and floor doors. Its message contents are the same as an open door message.

Car Go To The car go to request is a message created by the main control unit. This message is created in response to a floor or car request message. This message consists of a shaft ID, and floor ID and is sent to the motor controller.

Car Position This message allows the car/floor ECUs to display the current location of an elevator. The message also indicates the direction movement (or if it is stationary). The message is

transmitted by the motor controller and is awaited for by the car/floor panels and the main control unit.

Car Arrived A car arrived message informs the main controller an elevator has arrived at a requested destination. This message sent out by the motor controller. The main control unit uses this message to update the status of pending requests.

Floor Request Granted Upon completion of a request, the main control unit informs the floor panels to update their LED ports.

Car Request Granted Similar to a floor request granted message, but destined to the car panel LED ports.

2.4 Summary

This section has described how each ECU was modeled. The hierarchical design allows reusability by grouping similar components into one. The CAN bus provides an efficient communication medium used to broadcast messages to all ECUs. A complete ECS system was modeled.

In the following section a validation environment is constructed and simulated.

3 Model Validation

In this section, the validation method is described. The generic model of the Elevator Control System (ECS) is instantiated and simulated as four elevator shafts in a ten story building. A test bench was written to ensure the ECS behaves as expected. The following section will describe the simulated behavior, the ECS testing structure, and simulation environment.

3.1 Simulation Environment

The ECS was designed to provide a customizable simulation environment. Specifically, it allows the user to modify the number of floors and the number of elevator shafts. The core of the ECS model is illustrated in Figure 8. Note that the diagram uses the car unit and floor unit modules presented in Section 2. In addition, this figure describes the ECS system with the CAN bus Transaction Level Model (TLM). The CAN bus is distributed to each ECU. For simplicity, the input and output ports of each individual ECU have been omitted.

For simulation purposes, a variable number of floor units and shaft units can be declared. The number of floor units corresponds to the number of floors desired. Likewise, the number of shaft units specifies the number of elevator shafts. Our test model consists of ten floor units and four elevator shafts. All ECUs managed by one main control unit which communicate through the same communication channel.

In Figure 9 a general overview of the testing structure is shown. The structural differences between a TLM and BFM model are clearly shown. To control the simulation environment, the panel ECUs were disabled to prevent them from creating their own requests. A test bench unit was integrated and commands were injected through the CAN bus.

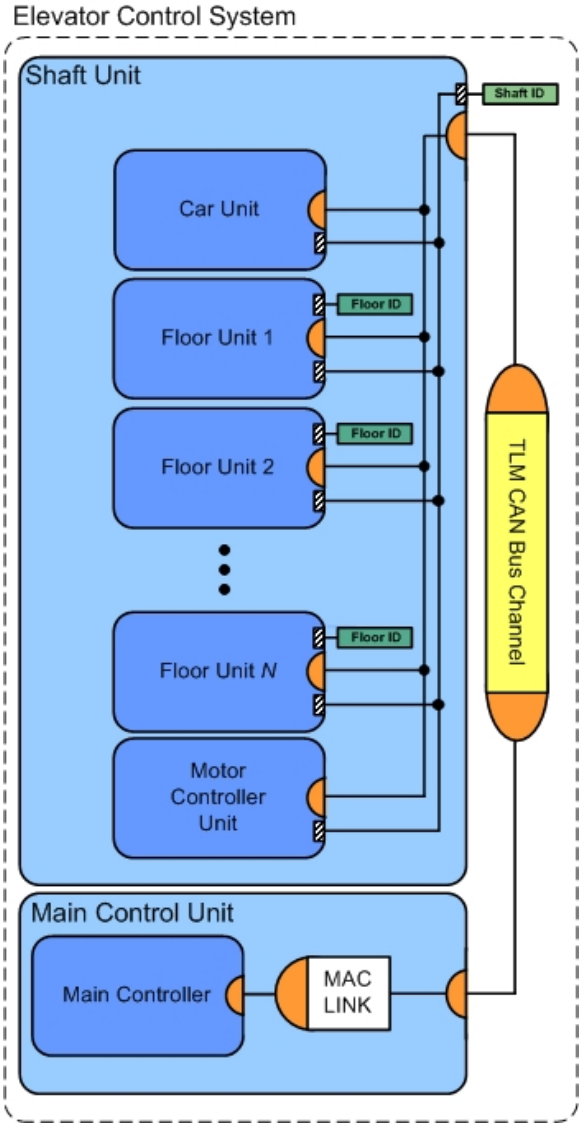

Figure 8: ECS Generic Model.

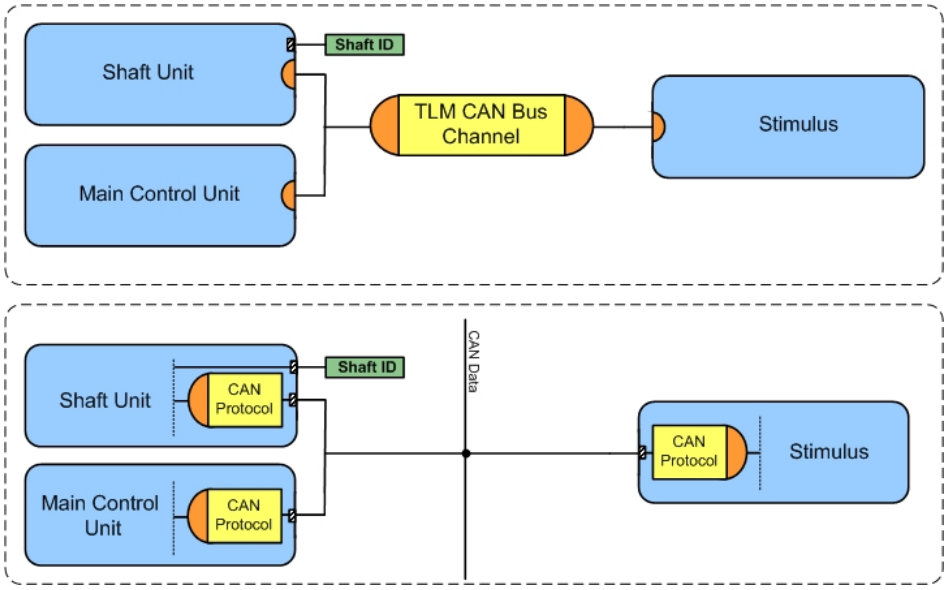

Figure 9: Elevator Control System Test Configuration.

3.1.1 Use Case Description

The ECS initial conditions establish each elevator on the first floor. There are only two ways to request elevator service, that is, through the car or floor panels. Let us assume a user requests to go down while currently at the 5th floor. The user presses the down button. This event triggers the floor panel to light up the LED button. A *floor request* message is composed and broadcast. All ECUs, except the floor panels and main control unit, discard this message. Only the floor panels with the same floor ID use this request and enable their LED buttons. The main control unit, however, plays a more vital role. It assigns the elevator to service the request. It follows by transmitting a *car go to* message. The motor controller proceeds as signaled by the request. The motor controller unit determines the current position of the elevator shaft, and, in this example, it ascends. As it is in motion, it generates a series of car position messages to update the state of the floor and car display units. In addition, the main control unit is informed as the elevator transitions from one floor to the next. As it arrives to the 2nd floor, the main control unit verifies if any other floor requests have been issued at the new floor. A new car go to message is re-issued either with the same request or a new one. This process continues until the elevator arrives at the initially requested floor. The motor controller sends a message to open the doors and informs the main controller to clear its request. The doors close shortly after.

For testing purposes, a monitor was connected to each ECU output port. This allows for quick debugging and verification that every message is managed correctly and the system behaves as intended. A printout of the monitor is shown in the Appendix. The printout shows the response of each ECU as it progresses until the request completed.

In the event that a request is made from inside the elevator, a car request is produced. The process is similar as requesting service by using the floor. Inside however, the car panel directly

defines the elevator and the floor number destination. The main controller then keeps track of this request and prevents any floor requests from being issued to this elevator. It is evident how the main controller plays a critical role in maintaining stability. The ECU exchange messages as described earlier.

The example described is a simple way to show functionality and how the model works overall. The model, however, was tested using multiple commands with elevators running in parallel. The test bench and results from the simpler simulation can be referenced in the Appendix.

Overall, the simulation run successfully validates the ECS model. It is important to note that the simulation of the BFM abstraction took considerably longer than the TLM. Therefore, the TLM model was used as the primary model to validate our system. It is reasonable to assume that the same behavior can be reproducible based on the results of the BFM.

4 Conclusion

This paper has described a case study for the modeling of an Elevator Control System (ECS) using a System Level Design Language (SLDL). The ECS was designed and modeled using a set of Elevator Control Units that interact via a central broadcast communication channel. The communication channel is an abstract model based on the Controller Area Network (CAN) bus. The ECS was modeled at three abstraction levels: the specification model, the Transaction Level Model (TLM), and the Bus Functional Model (BFM).

The ECS was designed to provide a configurable simulation model. The ECS model was validated by simulating four elevator shafts operating in a ten story building. The simulation of the ECS exhibited the expected behavior.

The simulation results also provide some insight into possible future work. The main control unit provides ECS with the basic control to function, but imposes some limitation. In addition, safety considerations were not considered in the design and could be added to the model in the future. Overall, the system model successfully describes an ECS.

5 Acknowledgement

Some portions of our implementation rely on the results of the class EECS 298: System-on-Chip Description and Modeling Winter 2006. In particular, portions of the implementations by Shinko Kawagoe and Seung-mok Yoo were reused.

References

[1] Daniel D. Gajski, Jianwen Zhu, Rainer Dömer, Andreas Gerstlauer, and Shuqing Zhao. *SpecC: Specification Language and Design Methodology*. Kluwer, 2000.

[2] Andreas Gerstlauer, Rainer Dömer, Junyu Peng, and Daniel D. Gajski. *System Design: A Practical Guide with SpecC*. Kluwer, 2001.

[3] Thorsten Grōtker, Stan Liao, Grant Martin, and Stuart Swan. *System Design with SystemC*. Springer, 2002.

[4] Gunar Schirner and Rainer Dömer. Abstract communication modeling: A case study using the can automotive bus. In *Proceedings of the International Embedded Systems Symposium*, Manaus, Brazil, August 2005.

A Appendix

A.1 SpecC Source

Selected SpecC Source has been provided below.

A.1.1 ecs.sh

```
#include <sim.sh>

#define NUM_FLOORS      10
#define NUM_SHAFTS      4

// Clock period for the CAN 0.2 microsec -> 500MHz
#define CAN_CLK_PERIOD 200000ull
#define BUFFER  100


enum ECSMSG_TYPE        /* the types of messages in the ECS system */
{
ECSMSG_CAR_REQUEST,
ECSMSG_OPEN_DOOR,
ECSMSG_CAR_ARRIVED,
ECSMSG_CAR_POSITION,
ECSMSG_FLOOR_REQUEST_GRANTED,
ECSMSG_CAR_REQUEST_GRANTED,
ECSMSG_CLOSE_DOOR,
ECSMSG_CAR_GOTO,
ECSMSG_FLOOR_REQUEST

};

enum direction {UP, DOWN, NONE};

struct ecsmsg   /* define our message data type */
{
    enum ECSMSG_TYPE Type;
    enum direction moving_direction;
    int ShaftID;
    int FloorID;
    bool flag;
};
```

```
// EOF ecs.sh
```

A.1.2 ecs.sc

```
#include <stdio.h>
#include <stdlib.h>
#include "ecs.sh"
import "canBus/network";
import "MainController";
import "ShaftUnit";
import "TestBench";


behavior Main()
{
        /*(BFM) wire for the CAN bus
        signal resolved bit4[1] CAN_DATA;*/

        /* Transaction Level Model (TLM) of the CAN Protocol */
          CanProtocolTLM canProt;

        /*(BFM) pull up resistor for the CAN wire
        PullUpResistor          canPullUp(CAN_DATA);*/

        /*Instantiate (BFM)*/
/*      ShaftUnit   elevator1(1,CAN_DATA);
        ShaftUnit   elevator2(2,CAN_DATA);
        ShaftUnit   elevator3(3,CAN_DATA);
        ShaftUnit   elevator4(4,CAN_DATA);
        MainUnit    controller(CAN_DATA);
        TestBench   message_provider(CAN_DATA);*/

        /*Instantiate (TLM)*/
        ShaftUnit   elevator1(1,canProt);
        ShaftUnit   elevator2(2,canProt);
        ShaftUnit   elevator3(3,canProt);
        ShaftUnit   elevator4(4,canProt);
        MainUnit    controller(canProt);
        TestBench   message_provider(canProt);

        int main(void){
```

```
                par {
        //(BFM)         canPullUp;
                        elevator1;
                        elevator2;
                        elevator3;
                        elevator4;
                        controller;
                        message_provider;
                }

                return 0;
        }
};

// EOF ecs.sc
```

A.1.3 ShaftUnit.sc

```
#include <stdio.h>
#include <stdlib.h>
#include "ecs.sh"
import "canBus/network";
import "FloorDisplay";
import "FloorPanel";
import "FloorDoor";
import "CarUnit";
import "MotorUnit";
import "FloorUnit";

/* (BFM)
behavior ShaftUnit(
        in const int elevator_no,
        inout signal resolved bit4[1] CAN_DATA){
*/

behavior ShaftUnit(
        in const int elevator_no,
        ICanProt canProt){


        /*Instantiate (BFM) --There will be 10 floors per shaft*/
/*      CarUnit                 shaft(elevator_no, CAN_DATA);
```

```
        FloorUnit               funit1(1, elevator_no, CAN_DATA),
                                funit2(2, elevator_no, CAN_DATA),
                                funit3(3, elevator_no, CAN_DATA),
                                funit4(4, elevator_no, CAN_DATA),
                                funit5(5, elevator_no, CAN_DATA),
                                funit6(6, elevator_no, CAN_DATA),
                                funit7(7, elevator_no, CAN_DATA),
                                funit8(8, elevator_no, CAN_DATA),
                                funit9(9, elevator_no, CAN_DATA),
                                funit10(10, elevator_no, CAN_DATA);
        MotorControllerUnit     motorunit(elevator_no, CAN_DATA);
*/
        /*Instantiate (TLM)*/
        CarUnit                 shaft(elevator_no, canProt);
        FloorUnit               funit1(1, elevator_no, canProt),
                                funit2(2, elevator_no, canProt),
                                funit3(3, elevator_no, canProt),
                                funit4(4, elevator_no, canProt),
                                funit5(5, elevator_no, canProt),
                                funit6(6, elevator_no, canProt),
                                funit7(7, elevator_no, canProt),
                                funit8(8, elevator_no, canProt),
                                funit9(9, elevator_no, canProt),
                                funit10(10, elevator_no, canProt);
        MotorControllerUnit     motorunit(elevator_no, canProt);

        void main(void){

                par {
                        shaft;
                        funit1;
                        funit2;
                        funit3;
                        funit4;
                        funit5;
                        funit6;
                        funit7;
                        funit8;
                        funit9;
                        funit10;
                        motorunit;
```

```
                }

        }

};

//EOF ShaftUnit.sc
```

A.1.4 FloorDoor.sc

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "ecs.sh"
#include "sim.sh"
import "canBus/network";


behavior FDoorMonitor(in signal bit[1] door){

  void main(void){

    char string[BUFFER];
     while (1){
          wait door;
          printf("%d sec\t%llu ps\t\t%s\tFloor Door is: %d\n",
          (int) (now()/(1 SEC)),now(),active_path(string,BUFFER-1),
          (bool)door);
    }
  }
};

behavior FloorDoor(
        in const int FloorNo,
        in const int ShaftNo,
        ICanMacLink link,
        out signal bit[1] DoorOpen){

  void main(void){
    struct ecsmsg msg;

    /*
```

```
    Initial Conditions:
    Door should be closed.
    */

    DoorOpen = 0; /*closed*/

    while(1){

      link.msgReceive(NONE, &msg, sizeof(msg));

      switch(msg.Type){

        case ECSMSG_OPEN_DOOR:

                if((msg.ShaftID == ShaftNo) && (msg.FloorID==FloorNo)){
                        DoorOpen=1;
                }

                break;

        case ECSMSG_CLOSE_DOOR:

                if((msg.ShaftID== ShaftNo) && (msg.FloorID==FloorNo)){
                        DoorOpen=0;
                }

                break;
        default:
                break;
        }

     }
  }
};

/*BFM
behavior FloorDoorUnit(
        in const int floorid,
        in const int shaftid,
        inout signal resolved bit4[1] CAN_DATA){
        */
```

```
/*TLM*/
behavior FloorDoorUnit(
        in const int floorid,
        in const int shaftid,
        ICanProt canProt){

  signal bit[1] out1;

  /*(BFM)
  CanProtocolTr canProt(CAN_DATA, CAN_CLK_PERIOD);*/
  CanMacLink canMacLink(canProt);

  FloorDoor floordoor(floorid,shaftid,canMacLink,out1);
  FDoorMonitor fdoor_m(out1);

  void main(void){

    par{

        /*(BFM)
        canProt;*/
        floordoor;
        fdoor_m;
    }
  }
};


// EOF FloorDoor.sc
```

A.1.5 CarDisplay.sc

```
#include <stdlib.h>
#include <stdio.h>
#include "ecs.sh"
#include "sim.sh"
import "canBus/network";

behavior CDMonitor(
        in signal int FloorOut,
        in signal bit[1] Up,
        in signal bit[1] Down){
```

```
  void main(void){

    char string[BUFFER];

    while (1){
      wait FloorOut,Up,Down;
      printf("%d sec\t%llu ps\t\t%s\tFloor: %d, Up:%d Down:%d\n",
      (int)(now()/(1 SEC)),now(),active_path(string,BUFFER-1),
      FloorOut,(bool)Up,(bool)Down);
    }
  }
};

behavior CarDisplay(
        in const int ShaftNo,
        ICanMacLink link,
        out signal int Floor,
        out signal bit[1] MovingUp,
        out signal bit[1] MovingDown){

  void main(void) {

    struct ecsmsg msg;

    /*Initializations: 1st Floor, stationary*/
    MovingUp = 0;
    MovingDown = 0;
    Floor = 1;

    while (1){

    /* Check incoming messages */
      link.msgReceive(NONE, &msg, sizeof(msg));

      switch(msg.Type){

        case ECSMSG_CAR_POSITION:

            if (msg.ShaftID == ShaftNo){
                Floor = msg.FloorID;

                switch(msg.moving_direction){
```

```
                case UP:
                        MovingUp = 1;
                        MovingDown = 0;
                        break;
                case DOWN:
                        MovingUp = 0;
                        MovingDown = 1;
                        break;
                case NONE:
                        MovingUp = 0;
                        MovingDown = 0;
                        break;
                }
            }
            break;

        default:
            break;
        }
    }
  }
};

/*BFM
behavior CarDisplayUnit(
        in const int shaftid,
        inout signal resolved bit4[1] CAN_DATA) {
*/

/*TLM*/
behavior CarDisplayUnit(
        in const int shaftid,
        ICanProt canProt){

  /*Wires connected to monitor*/
  signal int wire1;
  signal bit[1] wire2, wire3;

  /*CanBus Behaviors*/
  // (BFM) CanProtocolTr canProt(CAN_DATA, CAN_CLK_PERIOD);
  CanMacLink canMacLink(canProt);
```

```
  CarDisplay      cardisplay(shaftid, canMacLink, wire1,wire2,wire3);
  CDMonitor       car_display_m(wire1,wire2,wire3);

  void main(void) {

        par{
            //BFM         canProt;
                          cardisplay;
                          car_display_m;
        }
  }
};

//End of CarDisplay.sc
```

A.2 Test Bench and Results

The following is a simple test bench used to demonstrate how the ECS behaves with a floor panel request and a car panel request. A much detailed simulation was conducted, but the monitor output is to large to include in this appendix.

A.2.1 TestBench.sc

```
#include <stdlib.h>
#include <stdio.h>
#include "ecs.sh"
#include  "sim.sh"
import "canBus/network";

behavior MessageSender(ICanMacLink link) {

  struct ecsmsg msg;
  const char testbench[]="Test Bench";

  void main(void) {

        /* Floor Panel Request from 5th floor, user wants go Down*/
        msg.Type= ECSMSG_FLOOR_REQUEST;
        msg.ShaftID=4;
        msg.FloorID=5;
        msg.moving_direction = DOWN;
        waitfor 5 SEC; /*wait for system to initialize*/
```

```
        link.msgSend(129, &msg, sizeof(msg));


        /*Car Panel Request - User in elevator 2 wants to go to 5th floor*/
        msg.Type= ECSMSG_CAR_REQUEST;
        msg.ShaftID=2;
        msg.FloorID=5;
        msg.flag = 1;
        waitfor 15 SEC;
        link.msgSend(129, &msg, sizeof(msg));

      /* Quit the simulation */
        waitfor 100 SEC;
        printf("%d sec\t%llu ps\t%23s\tDONE!\n",
        (int) (now()/(1 SEC)),now(),testbench);
        exit(1);

        }
};

/* BFM MODEL
behavior TestBench(inout signal resolved bit4[1] CAN_DATA) {*/

/*TLM MODEL*/
behavior TestBench(ICanProt canProt){

 // CanProtocolTr canProt(CAN_DATA, CAN_CLK_PERIOD); BFM Only

    CanMacLink canMacLink(canProt);
    MessageSender msender(canMacLink);

  /* main method of the PE */
  void main(void) {

     par {
    //  canProt; BFM ONLY
      msender;
    }
  }
};
```

A.2.2 Simulation Log

```
0 sec    0 ps
         main controller  initialized
0 sec    0 ps
         Test Bench  initialized
0 sec    0 ps
         Test Bench composing FLOOR_REQUEST
0 sec    0 ps
         Test Bench msg composed: TYPE:8, FloorID:5, ShaftID:4, direction:1
5 sec    5000297995232 ps
         main controller received ECSMSG_FLOOR_REQUEST from Floor# 5
5 sec    5000297995232 ps
         Test Bench sent FLOOR_REQUEST
5 sec    5000297995232 ps
         Test Bench      composing CAR_REQUEST
5 sec    5000297995232 ps
         Test Bench      msg composed: TYPE:0, FloorID:5, ShaftID:2, direction:1
5 sec    5000297995232 ps
         Main.elevator1.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 1
5 sec    5000297995232 ps
         Main.elevator2.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 1
5 sec    5000297995232 ps
         Main.elevator3.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 1
5 sec    5000297995232 ps
         Main.elevator4.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 1
5 sec    5000598990416 ps
         Main.elevator1.motorunit.motor GOING UP
5 sec    5000598990416 ps
         Main.elevator1.motorunit.motor Currently at Floor:1 going to:5
5 sec    5000598990416 ps
         Main.elevator1.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
5 sec    5000899985600 ps
         main controller received CAR_POSITION message: Motor 1 is on floor 1
5 sec    5000899985600 ps
         main controller updated the car location for car 1. It is now on floor
1
5 sec    5000899985600 ps
         Main.elevator1.shaft.car_display.car_display_m Floor: 1, Up:1 Down:0
5 sec    5000899985600 ps
         Main.elevator1.funit1.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec    5000899985600 ps
         Main.elevator1.funit2.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec    5000899985600 ps
```

```
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 1, Up:1 Down:0
5 sec   5000899985600 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 1, Up:1 Down:0
15 sec  15001197980832 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 2
15 sec  15001197980832 ps
        main controller updated the car location for car 1. It is now on floor
2
15 sec  15001197980832 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15001197980832 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 2, Up:1 Down:0
```

```
15 sec  15001498976016 ps
        main controller received ECS_CAR_REQUEST from Car 1 to floor
5
15 sec  15001498976016 ps
        main controller sending CAR_GO_TO message to Shaft# 1
15 sec  15001799971200 ps
        Main.elevator1.motorunit.motor GOING UP
15 sec  15001799971200 ps
        Main.elevator1.motorunit.motor Currently at Floor:2 going to:5
15 sec  15001799971200 ps
        Main.elevator1.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
15 sec  15002100966384 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 2
15 sec  15002100966384 ps
        main controller updated the car location for car 1. It is now on floor
2
15 sec  15002100966384 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 2, Up:1 Down:0
15 sec  15002100966384 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 2, Up:1 Down:0
20 sec  20000595990464 ps
        main controller received ECS_CAR_REQUEST from Car 2 to floor
5
20 sec  20000595990464 ps
```

```
        main controller sending CAR_GO_TO message to Shaft# 2
20 sec  20000595990464 ps
        Test Bench sent CAR_REQUEST
20 sec  20000595990464 ps
        Test Bench      Quiting in 100 seconds
20 sec  20000896985648 ps
        Main.elevator2.motorunit.motor GOING UP
20 sec  20000896985648 ps
        Main.elevator2.motorunit.motor Currently at Floor:1 going to:5
20 sec  20000896985648 ps
        Main.elevator2.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
20 sec  20001197980832 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 1
20 sec  20001197980832 ps
        main controller updated the car location for car 2. It is now on floor
1
20 sec  20001197980832 ps
        main.elevator2.shaft.car_display.car_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 1, Up:1 Down:0
20 sec  20001197980832 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 1, Up:1 Down:0
25 sec  25002398961616 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 3
25 sec  25002398961616 ps
```

```
        main controller updated the car location for car 1. It is now on floor
3
25 sec  25002398961616 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002398961616 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25002699956800 ps
        main controller received ECS_CAR_REQUEST from Car 1 to floor
5
25 sec  25002699956800 ps
        main controller sending CAR_GO_TO message to Shaft# 1
25 sec  25003000951984 ps
        Main.elevator1.motorunit.motor GOING UP
25 sec  25003000951984 ps
        Main.elevator1.motorunit.motor Currently at Floor:3 going to:5
25 sec  25003000951984 ps
        Main.elevator1.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
25 sec  25003301947168 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 3
25 sec  25003301947168 ps
        main controller updated the car location for car 1. It is now on floor
3
25 sec  25003301947168 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 3, Up:1 Down:0
```

```
25 sec  25003301947168 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 3, Up:1 Down:0
25 sec  25003301947168 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 3, Up:1 Down:0
30 sec  30001495976064 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 2
30 sec  30001495976064 ps
        main controller updated the car location for car 2. It is now on floor
2
30 sec  30001495976064 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
```

```
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001495976064 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30001796971248 ps
        main controller received ECS_CAR_REQUEST from Car 2 to floor
5
30 sec  30001796971248 ps
        main controller sending CAR_GO_TO message to Shaft# 2
30 sec  30002097966432 ps
        Main.elevator2.motorunit.motor GOING UP
30 sec  30002097966432 ps
        Main.elevator2.motorunit.motor Currently at Floor:2 going to:5
30 sec  30002097966432 ps
        Main.elevator2.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
30 sec  30002398961616 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 2
30 sec  30002398961616 ps
        main controller updated the car location for car 2. It is now on floor
2
30 sec  30002398961616 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 2, Up:1 Down:0
30 sec  30002398961616 ps
```

```
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 2, Up:1 Down:0
35 sec  35003599942400 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 4
35 sec  35003599942400 ps
        main controller updated the car location for car 1. It is now on floor
4
35 sec  35003599942400 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003599942400 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35003900937584 ps
        main controller received ECS_CAR_REQUEST from Car 1 to floor
5
35 sec  35003900937584 ps
        main controller sending CAR_GO_TO message to Shaft# 1
35 sec  35004201932768 ps
        Main.elevator1.motorunit.motor GOING UP
35 sec  35004201932768 ps
        Main.elevator1.motorunit.motor Currently at Floor:4 going to:5
35 sec  35004201932768 ps
        Main.elevator1.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
35 sec  35004502927952 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 4
35 sec  35004502927952 ps
```

```
        main controller updated the car location for car 1. It is now on floor
4
35 sec  35004502927952 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 4, Up:1 Down:0
35 sec  35004502927952 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 4, Up:1 Down:0
40 sec  40002696956848 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 3
40 sec  40002696956848 ps
        main controller updated the car location for car 2. It is now on floor
3
40 sec  40002696956848 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
```

```
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002696956848 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40002997952032 ps
        main controller received ECS_CAR_REQUEST from Car 2 to floor
5
40 sec  40002997952032 ps
        main controller sending CAR_GO_TO message to Shaft# 2
40 sec  40003298947216 ps
        Main.elevator2.motorunit.motor GOING UP
40 sec  40003298947216 ps
        Main.elevator2.motorunit.motor Currently at Floor:3 going to:5
40 sec  40003298947216 ps
        Main.elevator2.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
40 sec  40003599942400 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 3
40 sec  40003599942400 ps
        main controller updated the car location for car 2. It is now on floor
3
40 sec  40003599942400 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
```

```
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 3, Up:1 Down:0
40 sec  40003599942400 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 3, Up:1 Down:0
45 sec  45004800923184 ps
        main controller received CAR_POSITION message: Motor 1 is on floor 5
45 sec  45004800923184 ps
        main controller updated the car location for car 1. It is now on floor
5
45 sec  45004800923184 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45004800923184 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 5, Up:1 Down:0
45 sec  45005101918368 ps
        main controller received ECS_CAR_REQUEST from Car 1 to floor
5
45 sec  45005101918368 ps
        main controller sending CAR_GO_TO message to Shaft# 1
45 sec  45005402913552 ps
        Main.elevator1.motorunit.motor sending CAR_POSITION
45 sec  45005402913552 ps
        Main.elevator1.motorunit.motor_m OFF
45 sec  45005703908736 ps
```

```
        main controller received CAR_POSITION message: Motor 1 is on floor 5
45 sec  45005703908736 ps
        main controller updated the car location for car 1. It is now on floor
5
45 sec  45005703908736 ps
        Main.elevator1.motorunit.motor sending OPEN_DOORS
45 sec  45005703908736 ps
        Main.elevator1.shaft.car_display.car_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit1.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit2.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit3.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit4.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit5.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit6.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit7.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit8.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit9.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45005703908736 ps
        Main.elevator1.funit10.flr_display.floor_display_m Floor: 5, Up:0 Down:0
45 sec  45006004903920 ps
        Main.elevator1.motorunit.motor waiting for 10 sec to close
45 sec  45006004903920 ps
        Main.elevator1.shaft.car_door.cdoor_m   Car Door is: 1
45 sec  45006004903920 ps
        Main.elevator1.funit5.flr_door.fdoor_m  Floor Door is: 1
50 sec  50003897937632 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 4
50 sec  50003897937632 ps
        main controller updated the car location for car 2. It is now on floor
4
50 sec  50003897937632 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
```

```
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50003897937632 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004198932816 ps
        main controller received ECS_CAR_REQUEST from Car 2 to floor
5
50 sec  50004198932816 ps
        main controller sending CAR_GO_TO message to Shaft# 2
50 sec  50004499928000 ps
        Main.elevator2.motorunit.motor GOING UP
50 sec  50004499928000 ps
        Main.elevator2.motorunit.motor Currently at Floor:4 going to:5
50 sec  50004499928000 ps
        Main.elevator2.motorunit.motor_m ON and moving UP-0, DOWN-1 Status: 0
50 sec  50004800923184 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 4
50 sec  50004800923184 ps
        main controller updated the car location for car 2. It is now on floor
4
50 sec  50004800923184 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
```

```
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 4, Up:1 Down:0
50 sec  50004800923184 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 4, Up:1 Down:0
55 sec  55006004903920 ps
        Main.elevator1.motorunit.motor sending CLOSE_DOORS
55 sec  55006302899152 ps
        Main.elevator1.shaft.car_door.cdoor_m   Car Door is: 0
55 sec  55006302899152 ps
        Main.elevator1.funit5.flr_door.fdoor_m  Floor Door is: 0
55 sec  55006603894336 ps
        main controller received ECS_CAR_ARRIVED message from Shaft# 1
55 sec  55006904889520 ps
        main controller sent FLOOR_REQUEST_GRANTED message to floor# 5
55 sec  55006904889520 ps
        Main.elevator1.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 0
55 sec  55006904889520 ps
        Main.elevator2.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 0
55 sec  55006904889520 ps
        Main.elevator3.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 0
55 sec  55006904889520 ps
        Main.elevator4.funit5.flr_panel.panel_m Floor Panel Light up: 0 down: 0
60 sec  60005098918416 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 5
60 sec  60005098918416 ps
        main controller updated the car location for car 2. It is now on floor
5
60 sec  60005098918416 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 5, Up:1 Down:0
```

```
60 sec  60005098918416 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005098918416 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 5, Up:1 Down:0
60 sec  60005399913600 ps
        main controller received ECS_CAR_REQUEST from Car 2 to floor
5
60 sec  60005399913600 ps
        main controller sending CAR_GO_TO message to Shaft# 2
60 sec  60005700908784 ps
        Main.elevator2.motorunit.motor sending CAR_POSITION
60 sec  60005700908784 ps
        Main.elevator2.motorunit.motor_m OFF
60 sec  60006001903968 ps
        main controller received CAR_POSITION message: Motor 2 is on floor 5
60 sec  60006001903968 ps
        main controller updated the car location for car 2. It is now on floor
5
60 sec  60006001903968 ps
        Main.elevator2.motorunit.motor sending OPEN_DOORS
60 sec  60006001903968 ps
        Main.elevator2.shaft.car_display.car_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit1.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit2.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit3.flr_display.floor_display_m Floor: 5, Up:0 Down:0
```

```
60 sec  60006001903968 ps
        Main.elevator2.funit4.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit5.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit6.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit7.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit8.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit9.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006001903968 ps
        Main.elevator2.funit10.flr_display.floor_display_m Floor: 5, Up:0 Down:0
60 sec  60006302899152 ps
        Main.elevator2.motorunit.motor waiting for 10 sec to close
60 sec  60006302899152 ps
        Main.elevator2.shaft.car_door.cdoor_m   Car Door is: 1
60 sec  60006302899152 ps
        Main.elevator2.funit5.flr_door.fdoor_m  Floor Door is: 1
70 sec  70006302899152 ps
        Main.elevator2.motorunit.motor sending CLOSE_DOORS
70 sec  70006600894384 ps
        Main.elevator2.shaft.car_door.cdoor_m   Car Door is: 0
70 sec  70006600894384 ps
        Main.elevator2.funit5.flr_door.fdoor_m  Floor Door is: 0
70 sec  70006901889568 ps
        main controller received ECS_CAR_ARRIVED message from Shaft# 2
70 sec  70006901889568 ps
        main controller sent CAR_REQUEST_GRANTED message to Elevator# 2
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #1 is 0
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #2 is 0
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #3 is 0
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #4 is 0
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #5 is 0
70 sec  70007202884752 ps
        Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #6 is 0
```

```
70 sec   70007202884752 ps
         Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #7 is 0
70 sec   70007202884752 ps
         Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #8 is 0
70 sec   70007202884752 ps
         Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #9 is 0
70 sec   70007202884752 ps
         Main.elevator2.shaft.car_panel.panel_m  CarPanel Button #10 is 0
120 sec 120000595990464 ps
         Test Bench      DONE!
```