Delft University of Technology

Faculty of Electrical Engineering,
Mathematics and Computer Science

System Validation Project

Authors:

Kaustubh Agarwal

Ana Aldescu

Aniket Ashwin Samant

Şerban Vădineanu

September 2018

Contents

1 Introduction **1**

2 Requirements **2**
2.1 System Components 2
2.2 Sequences 3
2.2.1 Wafer to lamp 3
2.2.2 Wafer to output stack 3
2.3 System Requirements 3
2.4 Functional Requirements 4

3 Interactions **5**
3.1 User Inputs 5
3.2 Actuators 5
3.3 Robots 6
3.4 Sensors 6

4 Architecture **7**

5 Translated Requirements **9**

6 Modeling the system **10**

7 Verification **11**

8 Conclusions **12**

Chapter 1

Introduction

The System Validation course of TU Delft proposes a project assignment concerning the design of a controller for a small distributed embedded system. The goal of the assignment is to design, model, and verify a control system for a simplified machine that prepares wafers for the production of microchips. Following the different stages of this project, we should be able to properly formulate system requirements and represent the model using labeled transition systems. This document presents the constituent phases of the project. Chapter 2 presents the components, the operating sequences and the requirements of the entire system. In chapter 3 we present the interactions taking place between the components defined in chapter 2. The architecture of the system is shown in chapter 4. Chapter 6 presents the modeled behaviour of all the controllers presented in the architecture. In chapter 5 the model is verified with the translated requirements. Conclusions and other remarks are presented in chapter 8.

Chapter 2

Requirements

The first step of the design process is to identify the requirements for the wafer projecting system we are considering. In order to define clear requirements, we have first identified the constituent components of the system and two main operating sequences.

This chapter describes the main components of the system, the manner in which these components form processes inside the system and the global requirements of the system.

2.1 System Components

- An UV lamp L
- A vacuum chamber
- Two airlocks A1, A2
- Two sets of doors
  - Input doors: DI1, DI2
  - Output doors: DO1, DO2
- Three robots
  - Outside the vacuum chamber: R1, R2
  - Inside the vacuum chamber: R3
- Two sets of wafer stacks
  - Input stacks: I1, I2
  - Output stacks: O1, O2
- One lamp sensor - Lamp Empty, Lamp Full
- Two airlock sensors - Airlock Empty, Airlock Full
- Four stack sensors - Stack Empty, Stack Full

2.2 Sequences  

We have identified two main sequences of the process. The first sequence describes the steps needed to move a blank wafer from the input stack to the lamp, while the second sequence describes how a wafer reaches the output stack after being projected by the lamp.

2.2.1 Wafer to lamp

1. Outside robot picks up wafer from input stack
2. Airlock output door opens
3. Robot drops wafer inside airlock
4. Airlock output door closes
5. Airlock input door opens
6. Inside robot picks up wafer from airlock
7. Inside robot drops wafer to lamp
8. Lamp turns on and projects the wafer

2.2.2 Wafer to output stack

1. Lamp turns off
2. Inside robot picks up wafer from lamp
3. Inside robot drops wafer inside airlock
4. Airlock input door closes
5. Airlock output door opens
6. Outside robot picks up wafer from airlock
7. Outside robot drops wafer to output stack

2.3 System Requirements  

Based on the sequences described above, we have identified a set of requirements, which were grouped with respect to the components we previously distinguished.

1. Airlocks:
   (a) DI1/DI2 can only be opened if DO1/DO2 is closed.
   (b) DO1/DO2 can only be opened if DI1/DI2 is closed.

2. Output stacks:
   (a) A wafer can only be placed into the output stack only if it was projected by the lamp.
3. Lamp:
   (a) The lamp can project only a wafer at a time.
   (b) The lamp can project a wafer only once.
4. Robots:
   (a) Robots can pick-up/drop only one wafer at a time.
   (b) R1/R2 can only drop a wafer into O1/O2 if the stack is not full.
   (c) R1/R2 can only pick-up a wafer from I1/I2 if the stack is not empty.
   (d) R1/R2 can only pick-up a wafer from A1/A2 if DO1/DO2 is open.
   (e) R1/R2 can only drop a wafer in A1/A2 if DO1/DO2 is open.
   (f) R3 can only pick-up a wafer from A1/A2 if DI1/DI2 is open.
   (g) R3 can only drop a projected wafer in A1/A2 if DI1/DI2 is open.
   (h) After picking-up a wafer from A1/A2, R3 can only drop it into the lamp.
   (i) After picking-up a wafer from the lamp, R3 can only drop it into A1/A2.
   (j) R1 can pick-up wafers from I1 and drop them only into A1.
   (k) R2 can pick-up wafers from I2 and drop them only into A2.
   (l) R1 can pick-up wafers from A1 and drop them only into O1.
   (m) R2 can pick-up wafers from A2 and drop them only into O2.

2.4 Functional Requirements

The following functional requirements ensure the movement of the wafers throughout the system.

1. From the input stack, a wafer should be able to reach the lamp.
2. From the lamp, a wafer should be able to reach the output stack.

Chapter 3

Interactions 

This chapter presents the interactions that take place between the components we have described in chapter 2.

3.1 User Inputs

The following actions can be performed by a user upon the components of the system. We describe this type of interactions as a function which takes a parameter "a" that represents the target of the user interaction, where I1, I2 are input stacks, O1, O2 are output stacks and DO1, DO2, DI1, DI2 are airlock doors.

user_fillStack(a):	(a: I1, I2)	Fill the corresponding stack with wafers.
user_emptyStack(a):	(a: O1, O2)	Empty the corresponding stack.
user_repairDoor(a):	(a: DO1, DO2, DI1, DI2)	Repair the corresponding door.

3.2 Actuators  

The actuators can be instructed by the following commands which can be sent by the controller. The first parameter represents the target of the command, while the second one represents the corresponding action to be taken. An exception is made in the case of the lamp, as there is only one component of this type in the whole system, therefore the first parameter will be omitted. DI1, DI2 are airlock input doors; DO1, DO2 are airlock output doors. The doors can be instructed to open or close, while the lamp can be instructed to turn on or off, therefore the values of parameter "p".

a_setDoor(a, p):	(a: DI1, DI2, DO1, DO2 – p: OPEN, CLOSED)	Sets the doors to open or close.
a_setLamp(p):	(p: ON, OFF)	Turns the lamp on or off.

3.3 Robots

Robots can be ordered to move to a position, pick up a wafer or drop a wafer. In the following definitions, the parameter "r" represents the robot in question (i.e. R1, R2, or R3). Parameter "l" shows the location to which the robot is instructed to move, where I1, I2 are input stacks, O1, O2 are output stacks, A1, A2 are airlocks, L is the lamp.

robot_moveToLocation(r, l):	(r: R1 .. R3 – l: I1, I2, O1, O2, A1, A2, L)	Instructs a robot to move to the specified location.
robot_pickUpWafer(r):	(r: R1 .. R3)	Instructs a robot to pick up a wafer from its current location.
robot_dropWafer(r):	(r: R1 .. R3)	Instructs a robot to drop a wafer at its current location.

3.4 Sensors

The following commands can be used by the safety controller to read out the current data from the sensors. The parameter "a" is the target of the command, with functions having specific targets. If a command has no target, there is only one sensor of that type and will be used automatically. Parameter "r" represents the possible values that the sensors can return.

sense_inputStack(r):	(a: I1, I2 – r : Empty, Not-Empty)	Checks whether the Input stack is empty or not.
sense_outputStack(r):	(a: O1, O2 —r: Full, Not_Full)	Checks whether the Output stack is full or not.
sense_airlock(r):	(a: A1, A2 — r : Wafer, No_wafer)	Checks whether the Airlock contains a wafer or not.
sense_lamp(r):	(r: Wafer, No_wafer)	Checks whether the Lamp contains a wafer or not.

Chapter 4

Architecture

The controller system's architecture has been modelled such that the safety and liveness requirements are always met, by using the various sensors, actuators, and other components that are present in the system.

The main assumptions made here for simplicity are as follows. The robots perform the instructed actions without any error. The lamp performs the projecting process without malfunctioning. The other components may require the user to act upon them based on certain sensor data and the corresponding user inputs described as interactions in chapter 3.

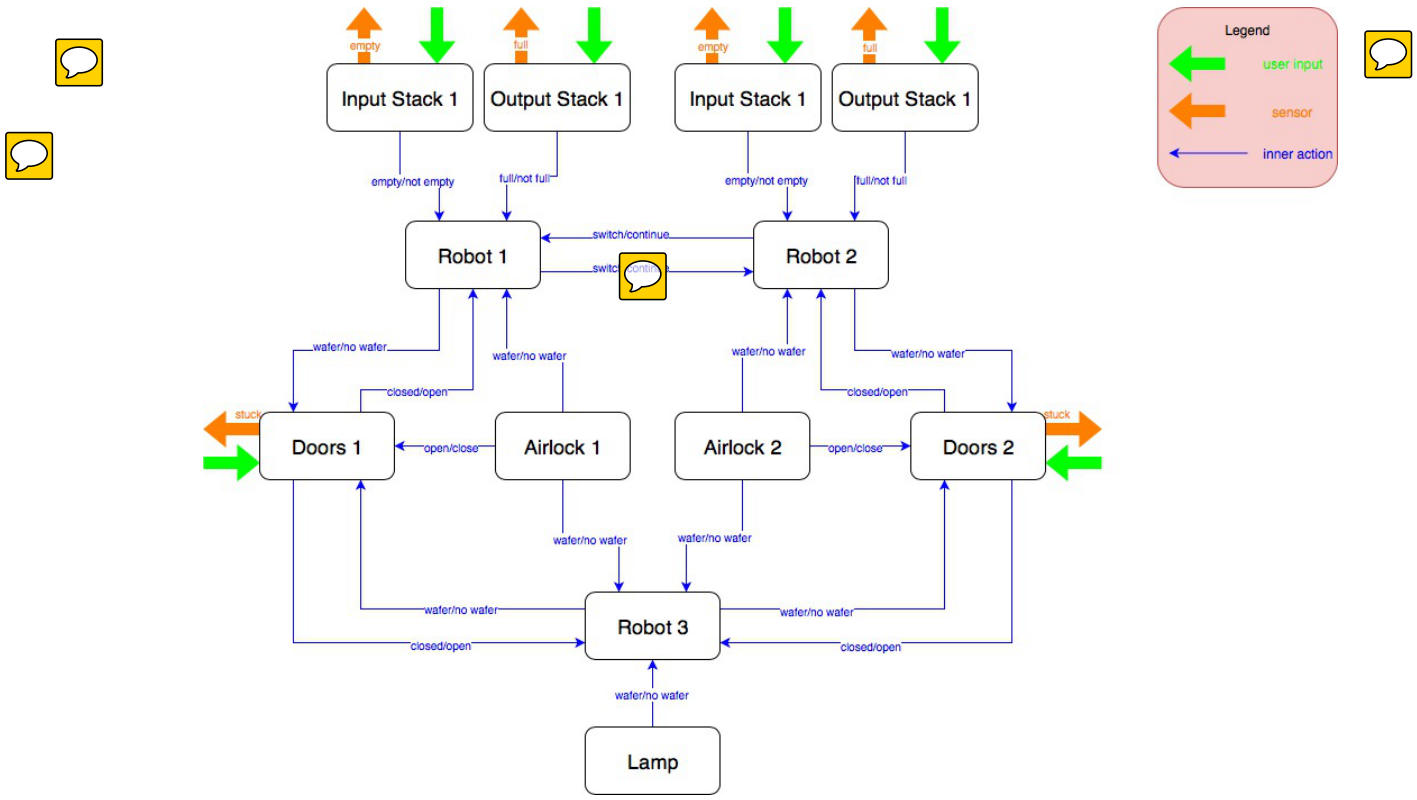

Figure 4.1: System architecture diagram

The architecture of the system is represented graphically using the Figure 4.1. It consists of several components working in parallel to meet the requirements stated in chapter 2.

For the most part, the system is expected to function without the need for any user input, but certain cases have been identified in which user input is needed to meet the system's liveness requirements. These are indicated by green arrows in the diagram, and the corresponding sensor action indicating the need for user input is shown using orange arrows.

The components interact with each other internally, as shown using blue arrows. The information exchanged by them includes sensor data, actuator feedback (to indicate the pass/fail status of operations) and other status data as shown on the arrows. This information is observable but need not be acted upon by the user since the system is automated. It is needed by the components internally to make decisions about their respective actions based on the status of other dependent components.

The system never reaches a state of "completion" since it is expected to run continuously till one of the conditions requiring a user input is met (see chapter 2). Theoretically, if the user continually replenishes the input stack with new wafers and clears the output stack at the same rate, and if the airlock doors do not malfunction, the system can run for an indefinite period of time.

Chapter 5

Translated Requirements

Chapter 6

Modeling the system

Chapter 7

Verification

Chapter 8

Conclusions