

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO HỆ THỐNG NHÚNG VÀ THIẾT KẾ
GIAO TIẾP NHÚNG

Đề tài: Điều khiển thang máy cao ốc

Nhóm sinh viên thực hiện:

Tên sinh viên	MSSV	Mã lớp
Nguyễn Viết Phong	20186316	129283
Dương Văn Biên	20172426	129283
Nguyễn Văn Chiến	20172431	129283
Đỗ Vũ Thanh Hiền	20182494	129283
Lại Văn Đức	20172475	129283

Giảng viên hướng dẫn: TS. Phạm Văn Tiến

Hà Nội – 12/2021

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH ẢNH	i
DANH MỤC BẢNG	ii
LỜI NÓI ĐẦU	iii
CHƯƠNG 1. GIỚI THIỆU CHUNG (INTRODUCTION)	1
<i>1.1 Giới thiệu chung</i>	<i>1</i>
<i>1.2 Hệ thống thời gian thực (Real-time operating system)</i>	<i>1</i>
<i>1.3 Hệ thống điều khiển thang máy cao ốc.....</i>	<i>1</i>
CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG.....	4
<i>2.1 Mô hình hóa hệ thống sử dụng UML.....</i>	<i>4</i>
2.1.1 Use case Diagram	4
2.1.2 Class Diagram.....	5
2.1.3 State Diagram.....	5
2.1.4 Sequence Diagram	6
<i>2.2 Mô hình hóa hệ thống sử dụng SystemC.....</i>	<i>9</i>
<i>2.3 Kết luận</i>	<i>12</i>
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN).....	13
<i>3.1 Thiết kế phần cứng.....</i>	<i>13</i>
3.1.1 Lưu đồ thuật toán cho bộ điều khiển 1 thang máy.....	13
3.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath).....	15
<i>3.2 Thiết kế phần mềm.....</i>	<i>18</i>
3.2.1 Mục tiêu	18
3.2.2 Lý do lựa chọn phần mềm.....	18
3.2.3 Thiết kế	20
3.2.4 Hướng triển khai	22
3.2.5 Giải quyết các vấn đề gặp phải trong quá trình triển khai	22

3.3 Thiết kế giao tiếp giữa phần cứng và phần mềm	23
3.3.1 Mô tả chi tiết các đầu vào và ra	23
CHƯƠNG 4. KIỂM THỬ (VERIFICATION).....	25
4.1 Kế hoạch kiểm thử.....	25
4.1.1 Kịch bản test	25
4.1.2 Kiểm thử thiết kế phần cứng bộ điều khiển cho 1 thang máy sử dụng ModelSim	25
4.1.3 Triển khai tích hợp hệ thống sử dụng kit FPGA ZC702 và phần mềm Vivado.....	26
4.2 Kết quả và đánh giá.....	29
CHƯƠNG 5. BÁO CÁO CÁ NHÂN.....	30
5.1 Thiết kế phần cứng.....	30
5.1.1 Lưu đồ thuật toán cho bộ điều khiển 1 thang máy.....	30
5.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath)	32
5.1.3 Triển khai tích hợp hệ thống sử dụng kit FPGA ZC702 và phần mềm Vivado.....	35
5.2 Kết quả và đánh giá.....	38
CHƯƠNG 6. KẾT LUẬN	39
TÀI LIỆU THAM KHẢO.....	40

DANH MỤC HÌNH ẢNH

Hình 2.1 Use case diagram của hệ thống thang máy	4
Hình 2.2 Class Diagram của hệ thống thang máy.....	5
Hình 2.3 State Diagram cho phần mềm hệ thống	6
Hình 2.4 Sequence Diagram cho nút bấm ngoài thang.....	7
Hình 2.5 Sequence Diagram cho nút bấm bên trong thang	7
Hình 2.6 Sequence Diagram cho hoạt động mở cửa.....	8
Hình 2.7 Sequence Diagram cho cảnh báo overload	8
Hình 2.8 Sequence Diagram cho cảnh báo cháy.....	9
Hình 2.9 Sơ đồ hệ thống thang máy sử dụng systemC	10
Hình 3.1 Lưu đồ thuật toán mô tả hoạt động thang máy	15
Hình 3.2 Sơ đồ ASMD	16
Hình 3.3 Sơ đồ kết nối khối Control unit và datapath	17
Hình 3.4 Sơ đồ phần mềm hệ thống.....	20
Hình 3.5 Sơ đồ hoạt động tìm kiếm	21
Hình 3.6 Sơ đồ toàn hệ thống giao tiếp giữa vi xử lý và phần cứng	23
Hình 4.1 Kết quả mô phỏng.....	26
Hình 4.2 Tạo project trên phần mềm vivado	26
Hình 4.3 Block design.....	27
Hình 4.4 Kit FPGA ZC702	27
Hình 4.5 Năng lượng tiêu thụ ước tính	28
Hình 4.6 Báo cáo timing	28
Hình 4.7 Tài nguyên tiêu tốn	28
Hình 4.8 Gen bitstream và tạo sdk.....	29

DANH MỤC BẢNG

Bảng 1.1: Bảng thông số kỹ thuật	2
Bảng 2.1 Tín hiệu đầu vào/ đầu ra của elevator controller	11
Bảng 2.2 Tín hiệu đầu vào/ đầu ra của phần mềm hệ thống.....	12
Bảng 3.1 Bảng mô tả thông số kỹ thuật	17

LỜI NÓI ĐẦU

Thời gian gần đây, các Hệ thống nhúng – Thời gian thực được quan tâm nhiều hơn ở Việt Nam, và trên thế giới thì các hệ thống này đã và đang được phát triển mạnh mẽ và là xu hướng thịnh hành ở các nước Công nghiệp vì những lợi ích to lớn, thiết thực mà nó mang lại. Chương trình học môn Hệ thống nhúng và thiết kế giao tiếp nhúng là một phần quan trọng giúp hiểu rõ quy trình thiết kế, đánh giá hệ thống nhúng. Trong báo cáo này, nhóm chúng em triển khai thiết kế và mô phỏng hệ thống nhúng cho Điều khiển thang máy cao ốc. Thiết kế được triển khai bằng ngôn ngữ mô tả phần cứng System Verilog và ngôn ngữ C, mô phỏng kiểm thử trên phần mềm Vivado. Cho ra kết quả hoạt động đúng với yêu cầu bài toán. Chúng em sẽ đi trình bày cụ thể những gì chúng em đã làm được thông qua 4 chương sau:

Chương 1: Giới thiệu chung (Introduction)

Chương 2: Đặc tả hệ thống (System Specification)

Chương 3: Thiết kế hệ thống (System Design)

Chương 3: Kiểm thử (Testing)

Chương 4: Kết luận

Nhóm chúng em xin chân thành cảm ơn thầy Phạm Văn Tiến đã tận tâm hướng dẫn em trong quá trình thực hiện bài tập lớn cũng như hoàn thiện báo cáo này!

CHƯƠNG 1. GIỚI THIỆU CHUNG (INTRODUCTION)

Chương này giới thiệu khái quát và các kiến thức cơ bản về hệ thống nhúng bài toán Hệ thống Điều khiển thang máy cao ốc.

1.1 Giới thiệu chung

Hệ thống nhúng (Embedded Systems) là sự kết hợp của phần cứng và phần mềm máy tính, và có thể bổ sung phần cơ khí hoặc một số bộ phận khác, được thiết kế để thực hiện một chức năng cụ thể. Điều này trái ngược hoàn toàn với máy tính cá nhân như Laptop, Computer, ... Nó cũng bao gồm phần cứng, phần mềm các thành phần cơ khí. Tuy nhiên, máy tính cá nhân không được thiết kế để thực hiện một chức năng cụ thể. Đúng hơn là nó thực hiện nhiều công việc khác nhau, hay có thể sử dụng thuật ngữ máy tính đa năng để phân biệt với máy tính hệ nhúng.

Thông thường, một hệ thống nhúng là một thành phần trong một số hệ thống lớn hơn. Như bài toán ta đang triển khai Điều khiển thang máy cao ốc, một hệ thống nhúng điều khiển giảm tốc, chiều hướng lên xuống thang máy, đóng mở cửa thang, ... Trong vài trường hợp, các hệ thống nhúng này được kết nối bởi một số mạng truyền thông [1].

1.2 Hệ thống thời gian thực (Real-time operating system)

Trong cả bài toán điều khiển chúng ta hay bắt gặp các thuật ngữ “Thời gian thực”. Thời gian thực không phải là thời gian phản ánh một cách trung thực, chính xác thời gian hay yêu cầu hệ thống phải trùng với thời gian thực tế.

Hệ thống thời gian thực được hiểu là các hoạt động của hệ thống phải thỏa mãn về tính tiên định. Tính tiên định là hành vi của hệ thống phải được thực hiện trong một khung thời gian cho trước hoàn toàn xác định, khung thời gian này được quyết định bởi đặc điểm và yêu cầu của hệ thống.

Thực tế cho thấy rằng hầu hết các hệ thống nhúng là các hệ thống thời gian thực và ngược lại các hệ thống thời gian thực là hệ thống nhúng.

1.3 Hệ thống điều khiển thang máy cao ốc

Thang máy là thiết bị vận tải để chở người và hàng hóa theo phương thẳng đứng. Nó là một loại hình máy nâng chuyển được sử dụng rộng rãi trong các ngành sản xuất của

Hệ thống nhúng và thiết kế giao tiếp nhúng

nền kinh tế như khai thác hầm mỏ, trong ngày xây dựng, luyện kim, ... ở những nơi đó thang máy được sử dụng để vận chuyển hàng hóa, sản phẩm, đưa công nhân tới nơi có độ cao làm việc khác nhau, v.v. Nó đã thay thế sức lực của con người và mang lại năng suất cao.

Trong sinh hoạt dân dụng, thang máy được sử dụng rộng rãi trong các tòa nhà cao tầng, cơ quan, khách sạn, trung cư, v.v.

Hệ thống của một thang máy bao gồm:

- Cabin
- Door
- Processor controller
- Monitor
- Button
- Motor
- Sensor

Hệ thống sẽ nhận thông tin người dùng từ Button có thể từ ngoài hành lang hoặc trong cabin, thông tin từ các sensor đưa qua Processor controller để đưa ra các tín hiệu điều khiển Monitor hiển thị chiều lên xuống, tầng mà cabin đang ở đó, Motor nhận tín hiệu điều khiển cabin lên xuống nhanh hay chậm, Door nhận tín hiệu đóng hay mở cửa thang máy.

Các thông số kỹ thuật được mô tả qua bảng sau:

Bảng 1-1: Bảng thông số kỹ thuật

Item	Parameter
Number of Persons	15
Rate capacity (kg)	1150
Rate speed (m/sec)	1.75
Dimensions of the well	2200mm x 2000mm
Cabin Dimensions	1800mm x 1400mm
Wide door	1000mm x 2100mm
Power Supply	1 phase: 220V
Processor	Xilinx Zynq-7000 SoC ZC702
Power of Processor	1.8 – 1.9 W

Hệ thống nhúng và thiết kế giao tiếp nhúng

Frequency of Processor	20 MHz
Fire alarm sensor	2412/24E
Number of control ladders	2
Number of floors	N

Yêu cầu về công nghệ:

- Dễ điều khiển
- An toàn tuyệt đối cho người và thiết bị
- Xử lý các tình huống phát sinh chuẩn xác
- Tiết kiệm điện năng
- Dễ dàng nâng cấp chỉnh sửa, bảo trì
- Đảm bảo tính thời gian thực

CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG

Chương này mô tả tổng quan về hệ thống thang máy mà nhóm thực hiện.

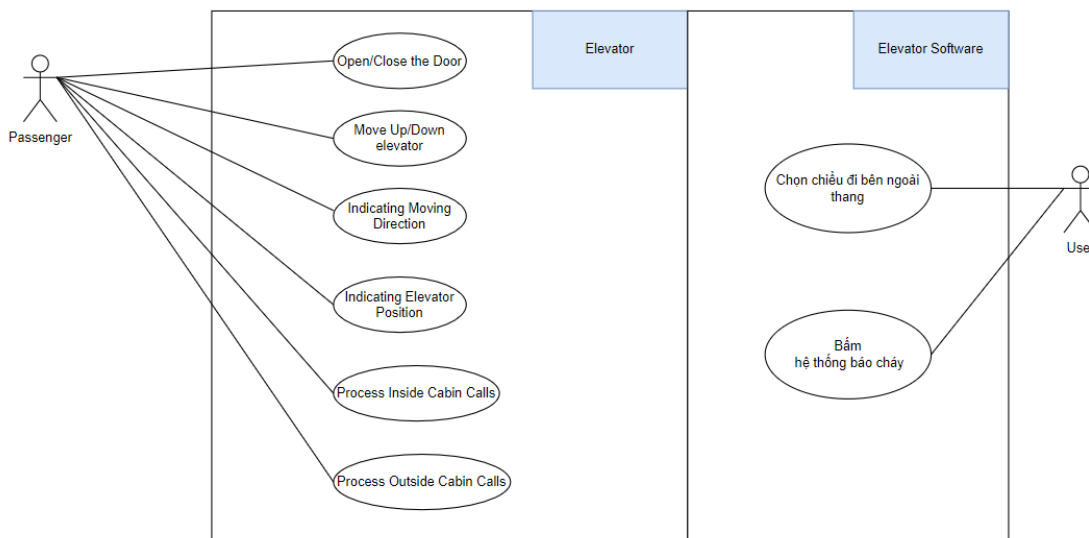
2.1 Mô hình hóa hệ thống sử dụng UML

UML (Unified Modeling Language) là một ngôn ngữ được sử dụng để mô tả các đặc điểm của hệ thống. Dựa vào chỉ tiêu kỹ thuật đã được đề cập trong chương 1, nhóm thực hiện thiết kế hệ thống thang máy sử dụng UML.

2.1.1 Use case Diagram

Tất cả các phần của hệ thống tương tác với con người hoặc các tác nhân tự động sử dụng hệ thống cho một số mục đích và cả con người và tác nhân đều mong đợi hệ thống hoạt động theo những cách có thể dự đoán được. Trong UML, use case diagram được sử dụng để mô hình hóa các hành vi của một hệ thống hoặc một phần của hệ thống.

Dựa theo các yêu cầu kỹ thuật đã được nêu trước đó, use case diagram của hệ thống thang máy được thể hiện ở hình dưới.



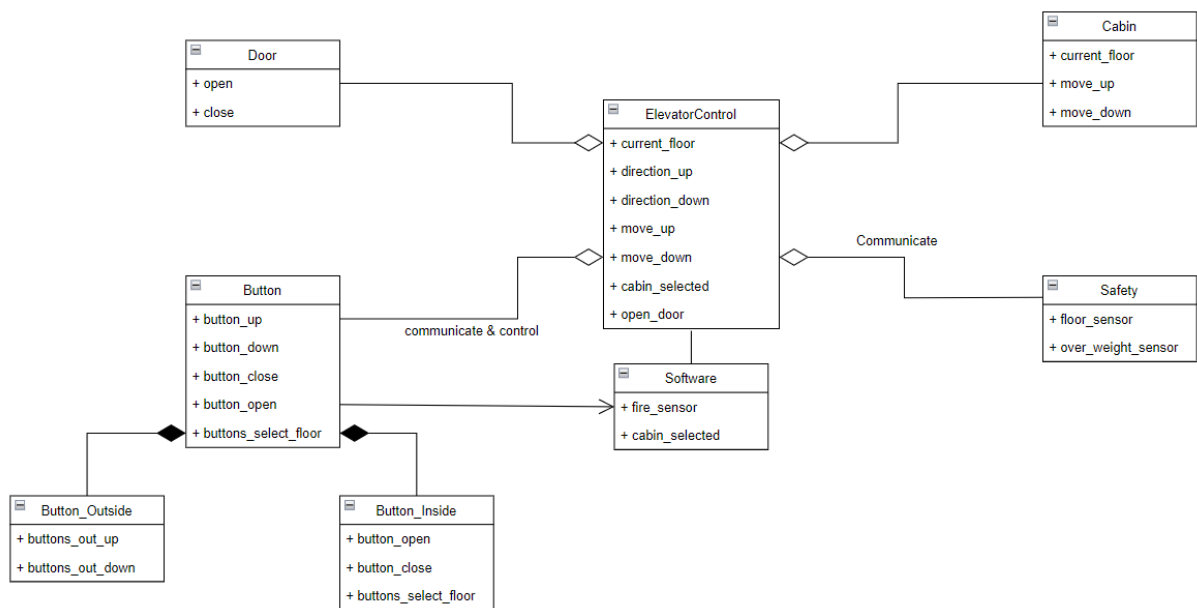
Hình 2.1 Use case diagram của hệ thống thang máy

Tác nhân duy nhất trong hệ thống thang máy là hành khách hay người sử dụng thang máy, đây là vai trò của con người khi tương tác với hệ thống. Người sử dụng tương tác với hệ thống thang máy bằng cách chọn tầng đi chuyển, báo cháy, chọn chiều đi chuyển. Người sử dụng cũng có thể đưa ra quyết định có vào hay rời thang máy hay không dựa vào chiều đi chuyển của thang máy và vị trí của cabin. Do đó, biểu đồ use case cho thấy tác nhân có

mối quan hệ với các trường hợp của hệ thống: Open/Close the Door, Move Up/Down elevator, Indicating Moving Direction, Indicating Elevator Position, Process Inside Cabin Calls, Process Outside Cabin Calls. Ngoài ra, khi người dùng chọn chiều đi bên ngoài thang máy hoặc bấm hệ thống báo cháy, phần mềm hệ thống sẽ xử lý và đưa ra thông tin.

2.1.2 Class Diagram

Biểu đồ lớp (Class Diagram) sẽ làm rõ các lớp, giao diện và mối quan hệ giữa chúng. Biểu đồ lớp liên quan đến mô tả hệ thống một cách tổng quan bao gồm các thuộc tính, hoạt động trong một lớp. Các thuộc tính và hoạt động có thể có của một hệ thống thang máy được thể hiện trên hình dưới đây.

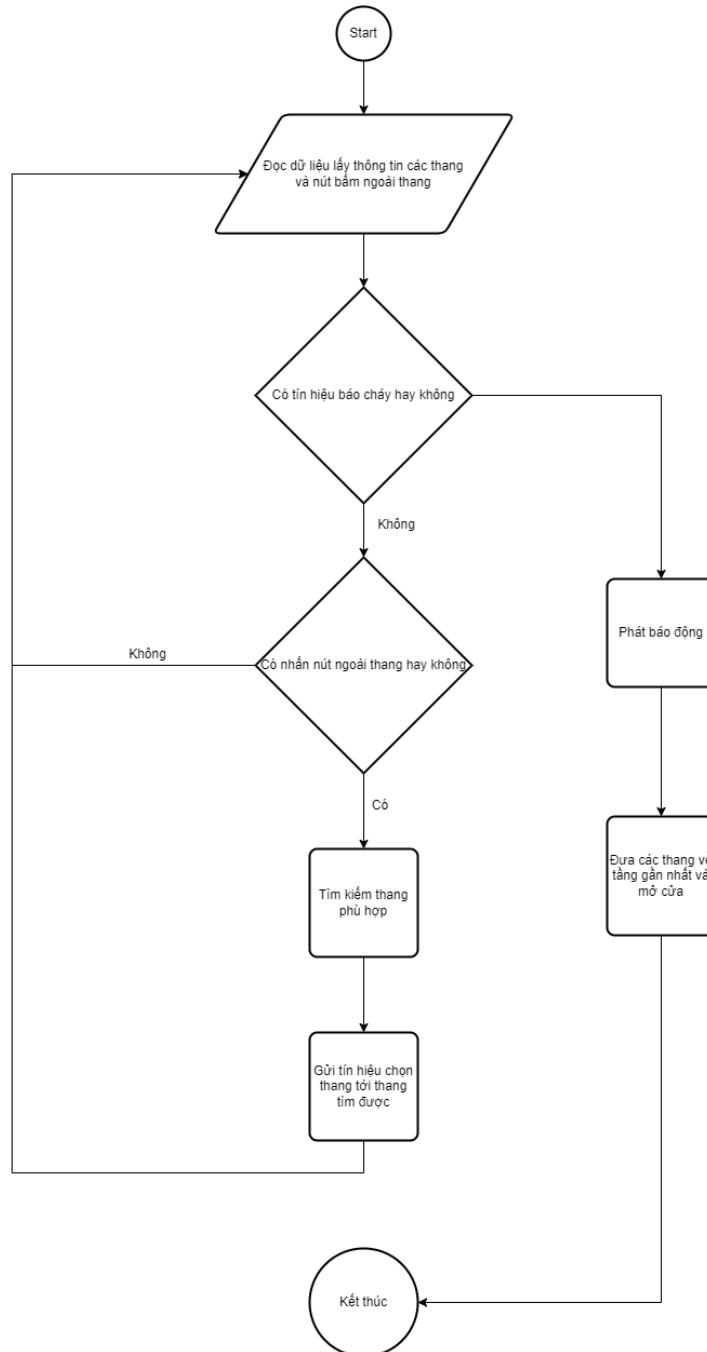


Hình 2.2 Class Diagram của hệ thống thang máy

Từ hình trên, hệ thống thang máy bao gồm hai phần chính là Elevator controller và Software. Elevator là thành phần chính của hệ thống, giữ nhiệm vụ nhận tín hiệu từ các đầu vào của thang máy như nút bấm, cảm biến, ..., đồng thời tương tác với phần mềm để giải quyết một số tình huống như người dùng bấm cảnh báo báo cháy, bấm chọn thang bên ngoài thang máy, hoặc cảm biến phát hiện có cháy.

2.1.3 State Diagram

Biểu đồ trạng thái (state diagram) để hiện luồng hoạt động của một thành phần hệ thống. Dưới đây là biểu đồ trạng thái của phần mềm hệ thống.



Hình 2.3 State Diagram cho phần mềm hệ thống

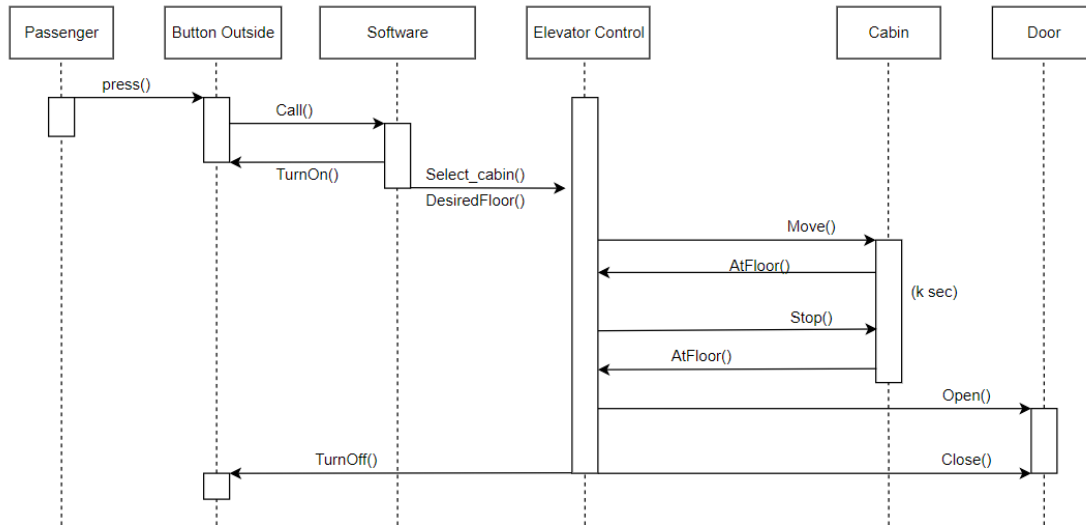
Phần mềm hệ thống có nhiệm vụ chọn thang khi có người dùng bấm chọn thang bên ngoài thang máy hoặc đưa ra cảnh báo khi có tín hiệu báo cháy.

2.1.4 Sequence Diagram

Sơ đồ trình tự (Sequence diagram) thể hiện tương tác giữa một tập hợp các đối tượng của hệ thống. Sơ đồ trình tự bao gồm tất cả các thông báo cho một phần hoặc một trường

hợp sử dụng. Ở đây, nhóm sử dụng sơ đồ trình tự để thể hiện sự tương tác giữa phần cứng và phần mềm, và các tương tác cho trường hợp cảnh báo của hệ thống.

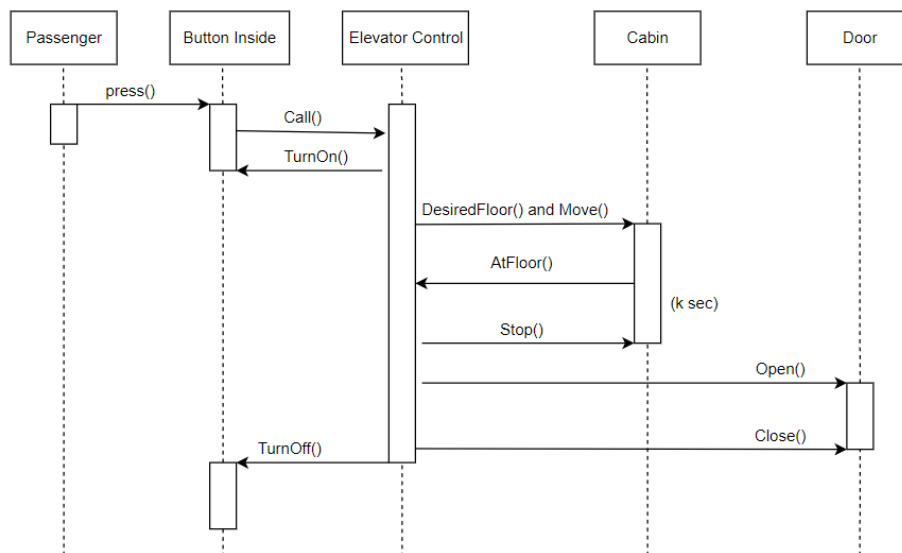
Sequence diagram cho nút bấm bên ngoài thang:



Hình 2.4 Sequence Diagram cho nút bấm ngoài thang

Dựa vào sơ đồ trên, khi người dùng bấm nút bên ngoài thang máy, phần mềm hệ thống nhận tín hiệu và xác định thang phù hợp. Sau đó, elevator controller thực hiện điều khiển thang nếu thỏa mãn điều kiện. Đồng thời, khi đến vị trí tầng có yêu cầu, cửa thang máy phải mở một khoảng thời gian để người dùng đi vào thang.

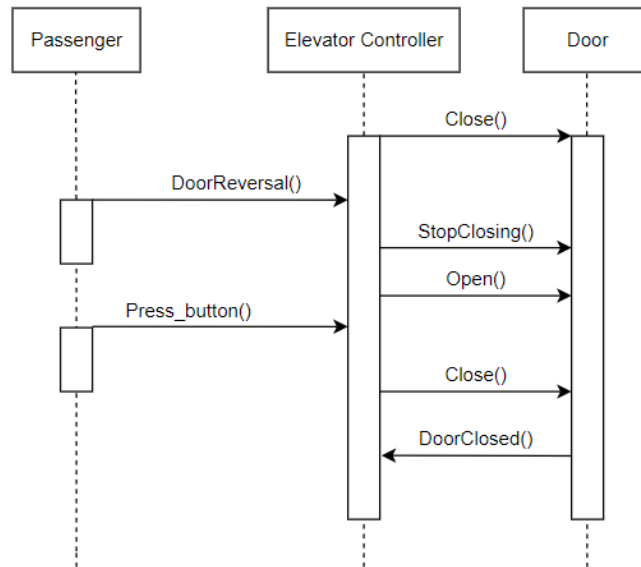
Sequence diagram cho nút bấm bên trong thang:



Hình 2.5 Sequence Diagram cho nút bấm bên trong thang

Đối với các tín hiệu khi người dùng bấm nút bên trong, elevator controller sẽ nhận tín hiệu và thực hiện hoạt động như hiển thị tầng được chọn, chiều di chuyển, đóng cửa và di chuyển thang phù hợp với yêu cầu của người dùng.

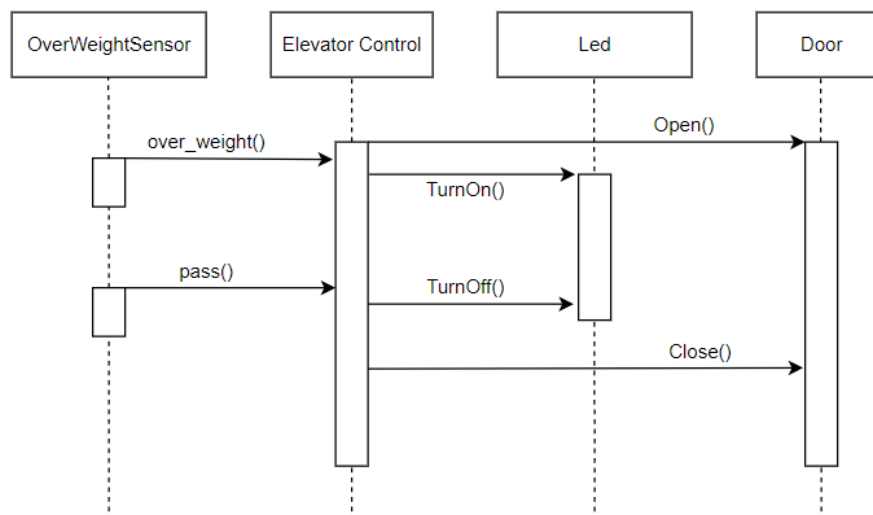
Sequence cho hoạt động mở cửa (Door Reversal):



Hình 2.6 Sequence Diagram cho hoạt động mở cửa

Bên cạnh việc thực hiện các hoạt động mở, đóng cửa bình thường, có một số trường hợp khi thang đang đóng cửa thì người dùng nhấn nút mở cửa cho một mục đích nào đó. Khi đó, elevator controller sẽ nhận tín hiệu và thực hiện theo yêu cầu của người dùng nếu phù hợp điều kiện đề ra.

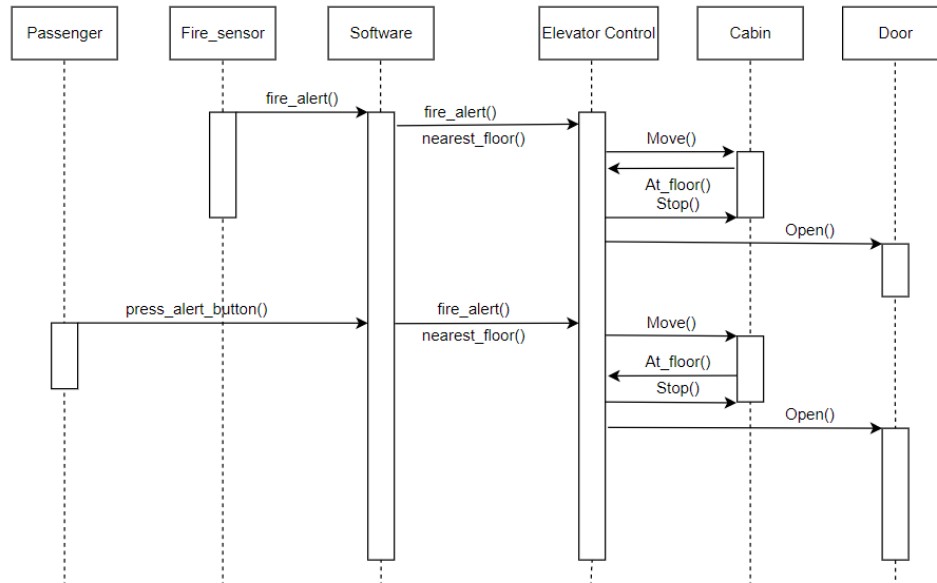
Sequence diagram cho cảnh báo overload:



Hình 2.7 Sequence Diagram cho cảnh báo overload

Hình bên trên thể hiện sequence diagram cho cảnh báo quá tải (overload). Elevator controller sẽ nhận tín hiệu từ cảm biến trọng lượng (overweight sensor), sau đó kiểm tra điều kiện của thang máy. Nếu quá trọng lượng quy định (bị overload), controller sẽ đưa ra cảnh báo và cửa thang máy sẽ không được đóng lại cho đến khi nhận được tín hiệu thỏa mãn yêu cầu từ cảm biến.

Sequence diagram cho cảnh báo cháy:

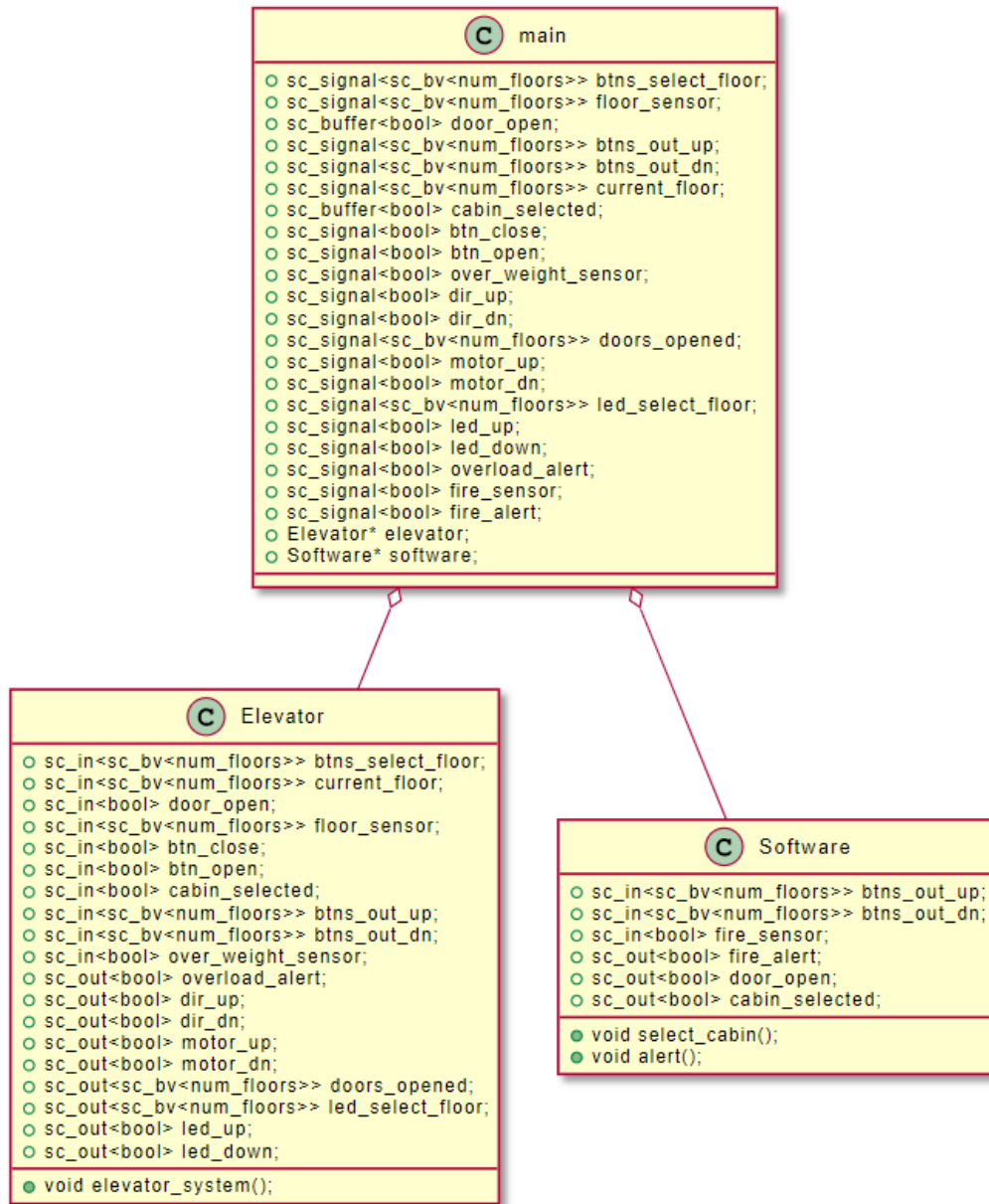


Hình 2.8 Sequence Diagram cho cảnh báo cháy

Hình trên thể hiện sequence diagram cho cảnh báo cháy. Đối với trường hợp cảnh báo cháy, phần mềm hệ thống sẽ nhận tín hiệu từ cảm biến hoặc khi người dùng bấm nút báo cháy. Sau đó, nếu có xảy ra cháy, đưa ra cảnh báo và thực hiện lựa chọn tầng gần nhất với vị trí của thang máy. Elevator controller nhận tín hiệu từ phần mềm hệ thống, thực hiện di chuyển thang về tầng gần nhất và mở cửa thang cho người dùng di chuyển ra ngoài.

2.2 Mô hình hóa hệ thống sử dụng SystemC

Ngoài việc sử dụng UML, nhóm còn chọn systemC là ngôn ngữ thứ hai để thiết kế và mô hình hóa hệ thống thang máy. Sơ đồ hệ thống của hệ thống thang máy được mô tả trên hình dưới đây.



Hình 2.9 Sơ đồ hệ thống thang máy sử dụng systemC

Dựa vào các yêu cầu kỹ thuật ở phần trước, nhóm thiết kế các đầu vào và đầu ra của hệ thống thang máy như được thể hiện trên hình. Class main chứa các tín hiệu đầu vào và đầu ra của hệ thống phục vụ cho việc mô phỏng. Class Elevator chứa phần điều khiển hệ thống (elevator controller) và class Software chứa tín hiệu thang được chọn và cảnh báo cháy. Các mô tả về đầu vào và đầu ra của hai class được thể hiện trong bảng dưới đây.

Bảng 2-1 Tín hiệu đầu vào/ đầu ra của elevator controller

Tên	Đầu vào/ Đầu ra	Mô tả
btns_select_floor	input	Tín hiệu tầng được chọn khi người dùng nhấn nút bên trong thang
current_floor	input	Vị trí của thang hiện tại
door_open	input	Tín hiệu mở cửa khi có cháy
floor_sensor	input	Tín hiệu cảm biến tầng
btn_close/btn_open	input	Tín hiệu đóng/mở thang máy
btns_out_up/btns_out_dn	input	Tín hiệu mở thang máy
cabin_selected	input	Thang phù hợp khi có người dùng bấm nút di chuyển bên ngoài thang
over_weight_sensor	input	Tín hiệu từ cảm biến trọng lượng
overload_alert	output	Cảnh báo quá tải
dir_up/dir_dn	output	Hướng di chuyển của thang
motor_up/motor_dn	output	Thang di chuyển lên hoặc xuống
doors_opened	output	Cửa thang được mở
led_select_floor	output	Hiển thị tầng được chọn
led_up/led_down	output	Hiển thị hướng thang di chuyển

Bảng 2-2 Tín hiệu đầu vào/ đầu ra của phần mềm hệ thống

Tên	Đầu vào/ Đầu ra	Mô tả
btns_out_up/btns_out_dn	input	Tín hiệu chọn đi lên/xuống bên ngoài thang máy
fire_sensor	input	Tín hiệu từ cảm biến báo cháy hoặc người dùng bấm nút báo cháy
fire_alert	output	Tín hiệu cảnh báo cháy
door_open	output	Tín hiệu mở cửa khi có cháy
cabin_selected	output	Thang được chọn phù hợp để thực hiện yêu cầu bên ngoài thang

Ngoài ra, các hàm `elevator_system()`, `select_cabin()` và `alert()` sẽ được triển khai để mô phỏng hệ thống thang máy.

2.3 Kết luận

Trong chương này, nhóm đã trình bày việc mô hình hóa hệ thống thang máy sử dụng hai công cụ là UML và SystemC. Từ đó, nhóm có thể hình dung tổng quan về hệ thống và các phần cần thiết để đáp ứng chỉ tiêu kỹ thuật đã nêu ra ở Chương 1. Công việc triển khai hệ thống được trình bày ở chương tiếp theo.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN)

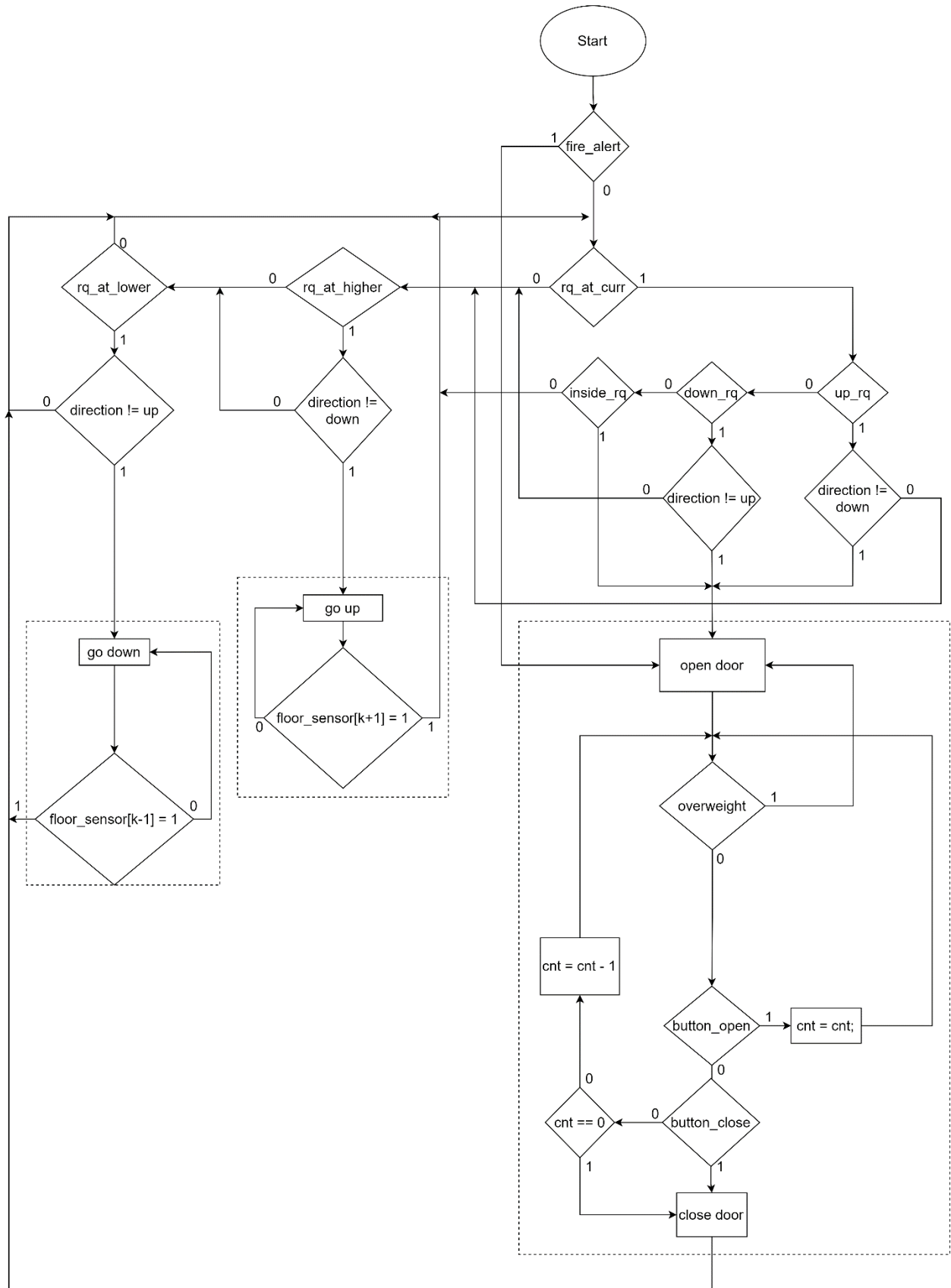
Chương này mô tả thiết kế chi tiết từng khối và toàn hệ thống Điều khiển thang máy cao ốc.

3.1 Thiết kế phần cứng

3.1.1 Lưu đồ thuật toán cho bộ điều khiển 1 thang máy

Lưu đồ thuật toán ở Hình 3.1 mô tả hoạt động của bộ điều khiển cho một thang máy. Giả sử thang máy đang ở tầng thứ k, thang máy sẽ hoạt động như sau:

- Nếu có tín hiệu cảnh báo cháy từ cảm biến báo cháy, đến tầng gần nhất và mở cửa.
- Nếu có yêu cầu đi lên tại tầng k (nút bấm ngoài thang máy), thang máy đang không đi xuống thì thang máy sẽ mở cửa.
- Nếu có yêu cầu đi xuống tại tầng k, thang máy đang không đi lên thì thang máy sẽ mở cửa.
- Nếu có yêu cầu đi đến tầng k (nút bấm trong thang máy), mở cửa thang máy.
- Nếu có yêu cầu đi lên, đi xuống hoặc đi đến tầng k tại tầng thấp hơn tầng k và thang máy đang không đi lên thì điều khiển thang máy đi xuống.
- Nếu có yêu cầu đi lên, đi xuống hoặc đi đến tầng k tại tầng cao hơn tầng k và thang máy đang không đi xuống thì điều khiển thang máy đi lên.
- Trong quá trình mở cửa, bộ điều khiển kiểm tra xem có tín hiệu overweight nhận từ cảm biến hay không, nếu overweight thì thang máy sẽ duy trì trạng thái mở cửa. Nếu không có tín hiệu overweight, thang máy sẽ kiểm tra xem có yêu cầu mở cửa hay không, nếu có yêu cầu mở cửa, thang máy sẽ duy trì trạng thái mở cửa, nếu có yêu cầu đóng cửa, thang máy sẽ đóng cửa. Nếu không có yêu cầu mở cửa hay đóng cửa nào từ người dùng, thang máy sẽ mở cửa trong 1 khoảng thời gian sau đó đóng cửa.



Hình 3.1 Lưu đồ thuật toán mô tả hoạt động thang máy

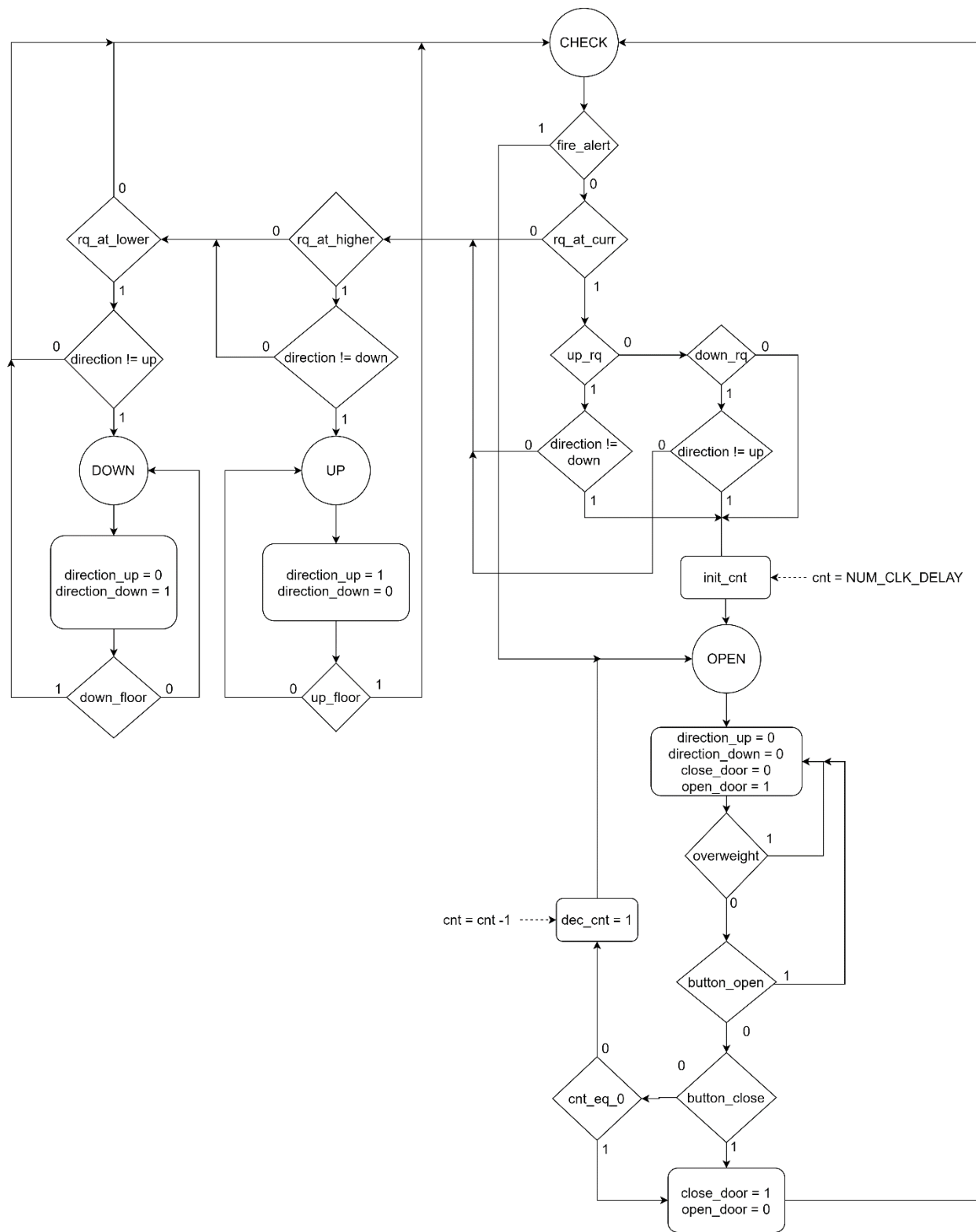
Chú thích các tín hiệu:

- $rq_at_curr = button_select_floor[k] \mid button_up[k] \mid button_down[k]$
- $rq_at_higher = button_select_floor[g] \mid button_up[g] \mid button_down[g] \mid g > k$
- $rq_at_lower = button_select_floor[g] \mid button_up[g] \mid button_down[g] \mid g < k$

3.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath)

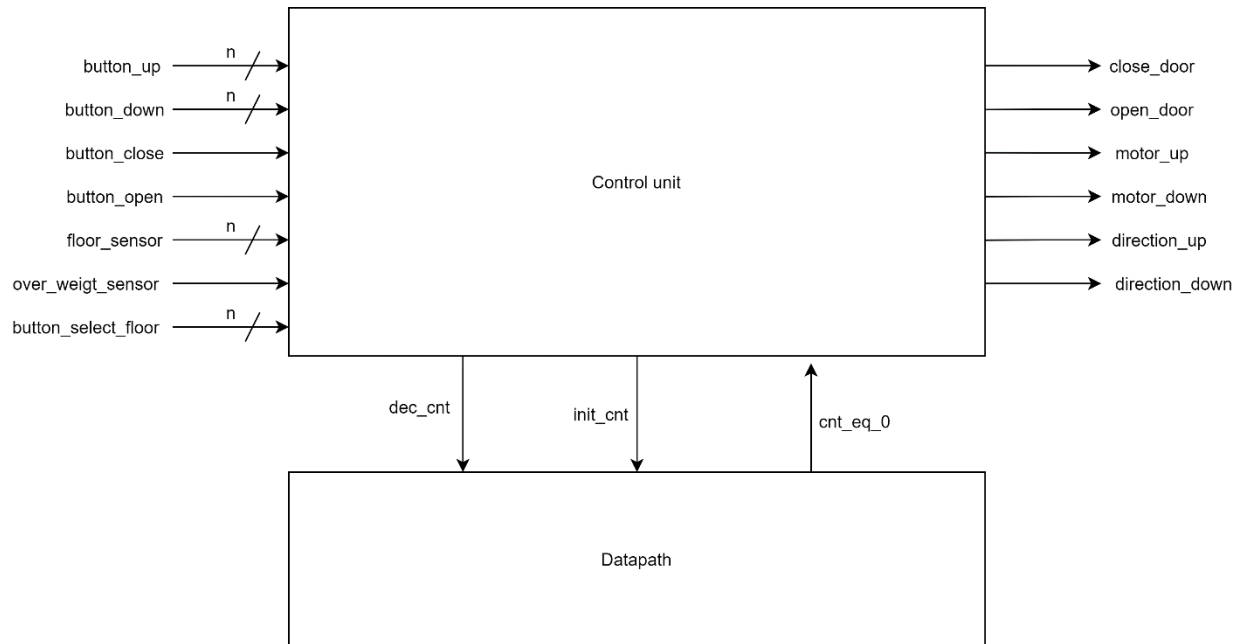
Từ sơ đồ thuật toán hoạt động của bộ điều khiển một thang máy, khoanh vùng xác định được bốn trạng thái:

- CHECK: trạng thái kiểm tra các yêu cầu được gửi tới thang máy.
- OPEN: trạng thái mở cửa thang máy.
- UP: trạng thái thang máy đi lên.
- DOWN: trạng thái thang máy đi xuống.



Hình 3.2 Sơ đồ ASMD

Từ sơ đồ ASMD ta xác định được sơ đồ khối giao tiếp giữa control unit và data path của hệ thống:



Hình 3.3 Sơ đồ kết nối khối Control unit và datapath

Bảng 3-1 Bảng mô tả thông số kỹ thuật

STT	Tên tín hiệu	Độ rộng (bit)	Chiều	Mô tả
1	button_up	n	input	Nút bấm lên tại n tầng bên ngoài thang máy
2	button_down	n	input	Nút bấm xuống tại n tầng bên ngoài thang máy
3	button_close	1	input	Nút bấm đóng cửa bên trong thang máy
4	button_open	1	input	Nút bấm mở cửa bên trong thang máy
5	floor_sensor	n	input	Cảm biến xác định vị trí của thang máy
6	overweight_sensor	1	input	Tín hiệu xác định thang máy quá tải trọng

7	button_select_floor	n	input	Nút bấm chọn n tầng trong thang máy
8	close_door	1	output	Tín hiệu đóng cửa thang máy
9	open_door	1	output	Tín hiệu mở cửa thang máy
10	motor_up	1	output	Tín hiệu điều khiển thang máy đi lên
11	motor_down	1	output	Tín hiệu điều khiển thang máy đi xuống
12	direction_up	1	output	Tín hiệu trạng thái thang máy đang đi lên
13	direction_down	1	output	Tín hiệu trạng thái thang máy đang đi xuống

Từ sơ đồ ASMD sử dụng ngôn ngữ mô tả phần cứng Verilog triển khai controller cho 1 thang máy.

3.2 Thiết kế phần mềm

Trình bày tổng quát về phần mềm thiết kế, cũng như có 1 cái nhìn tổng thể về hệ thống, cũng như những vấn đề phát sinh và hướng giải quyết khi gặp vấn đề và đưa ra giải pháp thay thế với những bài toán đã đặt ra.

3.2.1 Mục tiêu

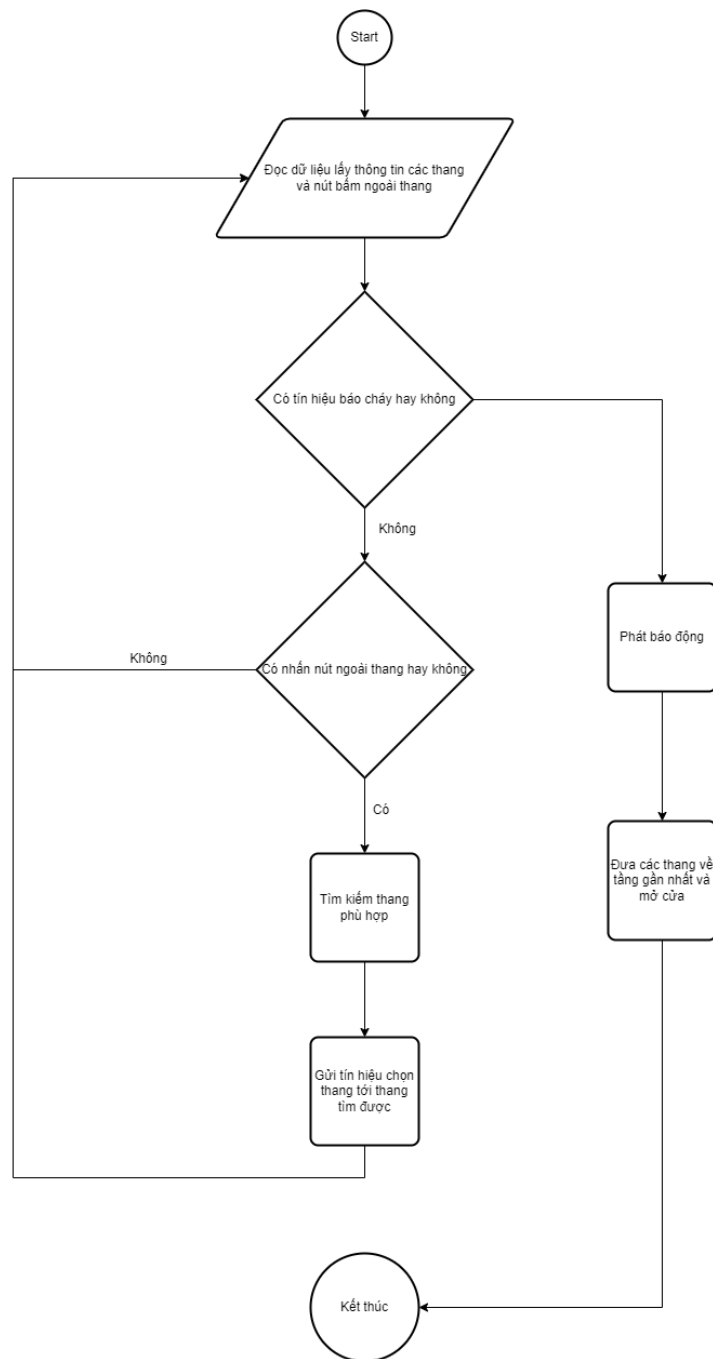
Nhóm đề xuất việc áp dụng phần mềm vào việc phục vụ việc khi hệ thống có nhiều hơn 1 thang, lúc này bài toán trở nên phức tạp hơn vì không chỉ việc chọn thang nào phục vụ mà nó là còn về bài toán tối ưu năng suất làm việc của hệ thống thang máy. Ví dụ khi có nhiều thang trong cùng một tòa nhà, và bạn đang đứng ở 1 tầng, bạn muốn đi lên thì hệ thống phần mềm phải lựa chọn ra chiếc thang ở gần ngay phía dưới của bạn thay vì việc chọn 1 chiếc thang ở phía trên bạn và đang đi lên để phục vụ bạn. Điều đó thật là ngớ ngẩn đối với người sử dụng.

3.2.2 Lý do lựa chọn phần mềm

Lý do nhóm lựa chọn phần mềm để thực hiện mục đích này gồm những ý chính sau:

- Thứ nhất: Việc để phần mềm xử lý logic sẽ dễ thực hiện hơn phần cứng, mặc dù về chi phí thì cũng không quá tốn kém hơn việc sử dụng phần cứng.
- Thứ hai đó là việc sử dụng phần mềm giúp chúng ta có thể linh động trong việc sử dụng những thuật toán thông minh hơn, phục vụ những mục đích mạnh hơn của hệ thống thang, thay vì chỉ đơn thuần là chọn nhiều thang.
- Cuối cùng và cũng là điều quan trọng nhất đó là tính tùy biến cho phần mềm cao hơn, thay vì việc thiết kế FPGA thì đối với việc thay đổi thuật toán trở nên vô cùng khó khăn, gần như việc đổi thuật toán nếu hệ thống thang đã đi vào vận hành thì gần như chúng ta phải thay thế con chip FPGA, nhưng đối với phần mềm, khi đưa vào vận hành, nếu thấy thuật toán đang có vấn đề, chúng ta cần tùy biến thì chúng ta chỉ cần nạp lại code, là hệ thống thang có thể thay đổi cách vận hành. Điều làm nên sức mạnh của hệ thống phần mềm.

3.2.3 Thiết kế



Hình 3.4 Sơ đồ phần mềm hệ thống

Giải thích:

Phần mềm sẽ liên tục đọc các dữ liệu từ cảm biến ở các thang để biết trạng thái hoạt động của các thang.

Đọc dữ liệu từ các cảm biến cháy, để nhận biết tín hiệu báo cháy.

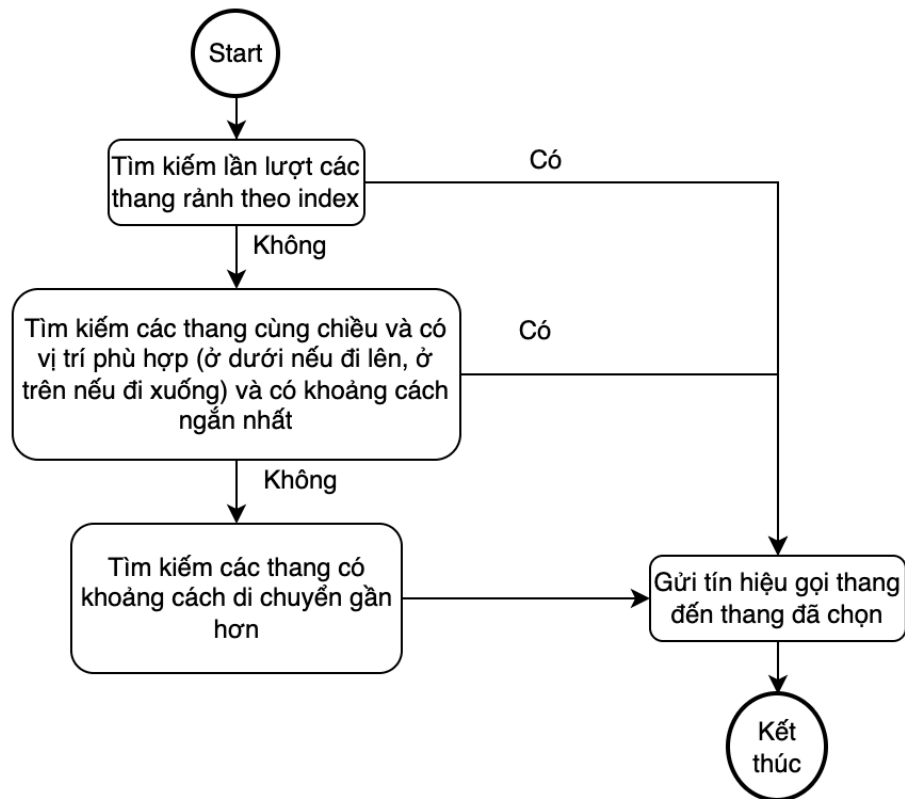
Đọc dữ liệu từ các tầng để nhận biết tín hiệu nhấn nút lên hay xuống ở các tầng.

Việc tìm kiếm thang sẽ được miêu tả chi tiết ở Hình 3.5.

Gửi tín hiệu điều khiển cho thang bao gồm:

- Thông tin các tầng chọn
- Thông tin tín hiệu cháy
- Thông tin điều khiển đóng, mở cửa
- Điều hướng đi lên, đi xuống

Sau khi gửi các tín hiệu cần thiết cho thang thì thang sẽ vận hành theo sơ đồ phần cứng FPGA.



Hình 3.5 Sơ đồ hoạt động tìm kiếm

3.2.4 Hướng triển khai

Việc triển khai được sử dụng ngôn ngữ lập trình C, lập trình trên phần mềm Vivado sử dụng board ZYNQ-7 ZC702 Evaluation Board.

Các thư viện sử dụng: <stdio.h>, "platform.h", "xil_printf.h", "xgpio.h", "xtime_l.h", <stdlib.h>.

Các hàm sử dụng:

- int XGpio_Initialize(XGpio * InstancePtr, u16 DeviceId)
- void XGpio_SetDataDirection(XGpio *InstancePtr, unsigned Channel, u32 DirectionMask)
- u32 XGpio_DiscreteRead(XGpio * InstancePtr, unsigned Channel)
- void XGpio_DiscreteWrite(XGpio * InstancePtr, unsigned Channel, u32 Data)

3.2.5 Giải quyết các vấn đề gặp phải trong quá trình triển khai

3.2.5.1 Giải quyết vấn đề về kit lập trình

Do nhóm chỉ có 1 bộ kit của một thành viên nên trong quá trình phát triển, gặp đôi chút khó khăn trong việc nạp code lên kit và chạy chương trình. Nhóm đã giải quyết vấn đề bằng cách lập trình chương trình C/C++ độc lập chạy khối logic và chương trình bên ngoài bằng “build-essential” ở trên Ubuntu.

Cách chạy thử:

1. Clone git repository
2. Install build-essential
3. Build source bằng cách sử dụng lệnh gcc
4. Chạy chương trình bằng cách chạy file đã build ở bước 3

Sau khi kiểm tra chương trình đã chạy thỏa mãn yêu cầu, chúng ta tiến hành ghép code vào SDK Vivado. Về mặt cơ bản source code sẽ không khác nhiều, chỉ có một vài thư viện cần phải thay đổi và đầu vào sẽ chuyển qua sử dụng thư viện của Xilinx.

3.2.5.2 Giải quyết vấn đề về phần cứng cảm biến thang

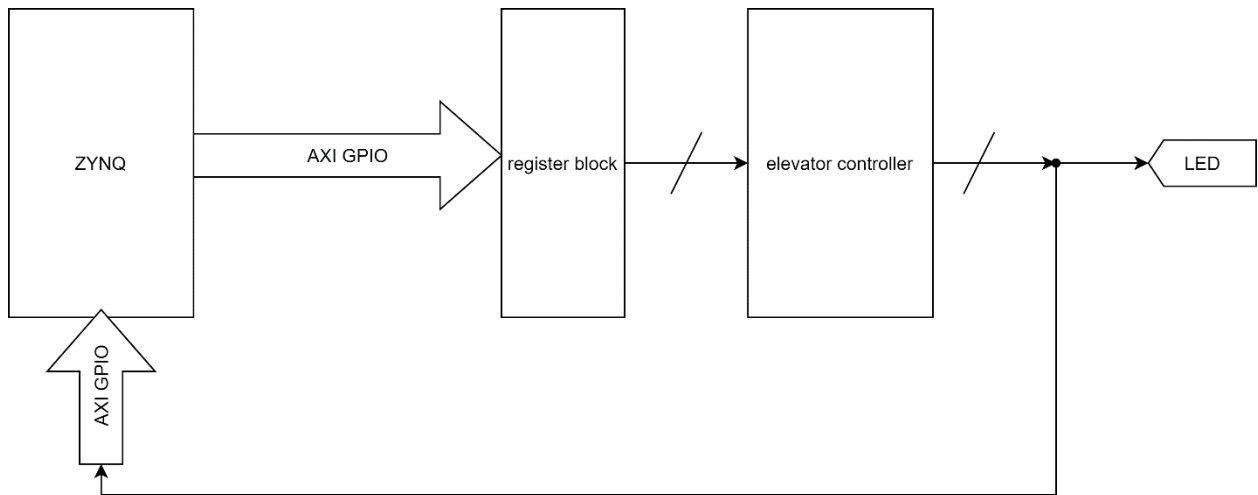
Do không có cảm biến mô tả việc chuyển động của thang (thực tế sẽ là đầu vào của cả phần cứng và phần mềm) nên nhóm đã quyết định sử dụng phần mềm để mô phỏng quỹ đạo di chuyển của các thang. Vì vậy, việc nhận tín hiệu vị trí của thang do phần mềm sinh ra chứ không phải được nhận đầu vào từ cảm biến.

Tốc độ di chuyển của thang đang được mô phỏng bởi hệ thống là 3s/tầng, thời gian chờ 1 tầng là 10s.

3.2.5.3 Giải quyết các vấn đề liên quan đến cảm biến cháy và nút bấm của thang

Các vấn đề về liên quan đến cháy nổ đang được sử dụng hàm random của C, với xác suất là một phần triệu. Tương tự như vậy với các nút nhấn của thang, cũng được sử dụng hàm random.

3.3 Thiết kế giao tiếp giữa phần cứng và phần mềm



Hình 3.6 Sơ đồ toàn hệ thống giao tiếp giữa vi xử lý và phần cứng

Hình trên mô tả sơ bộ toàn hệ thống giao tiếp giữa phần vi xử lý và phần cứng tự thiết kế. Trong project này, chúng em chọn giao tiếp GPIO để truyền thông giữa phần mềm và phần cứng. Để đồng bộ giữa 2 phần, chúng em thiết kế thêm một khối trung gian là register block hoạt động cùng tần số với tần số của elevator controller, để capture đầu vào mô phỏng được gửi từ phần mềm và gửi đầu vào ổn định cho elevator controller.

3.3.1 Mô tả chi tiết các đầu vào và ra

Đầu vào của phần mềm

- Các đầu vào từ gpio dưới dạng nút nhấn
- Các đầu vào trả ra từ controller phần cứng của từng thang, bao gồm 4-bit trong đó lần lượt thể hiện các giá trị
 - Open: thang có mở cửa không
 - Close: thang có đóng cửa không
 - Direction up: thang có đang đi lên hay không
 - Direction down: thang có đang đi xuống hay không

Đầu ra của phần mềm, các đầu ra của phần mềm chính là các đầu vào cung cấp tín hiệu đầu vào cho controller phần cứng. Tất cả các dữ liệu đều bao gồm bit 1 ở cuối cùng thể hiện việc enable, tối thiểu bao gồm:

- Floor_sensor: tín hiệu báo thang đang ở đâu (00001000: thang đang ở tầng 4)
- Floor_selected: các tầng được chọn (00010010: tầng 2 và tầng 5 đang được chọn)

CHƯƠNG 4. KIỂM THỬ (VERIFICATION)

Chương này trình bày các kết quả mô phỏng cho từng khối và toàn bộ thiết kế được triển khai bằng ngôn ngữ Verilog và ngôn ngữ C trên phần mềm Vivado.

4.1 Kế hoạch kiểm thử

4.1.1 Kịch bản test

4.1.1.1 Trường hợp có 1 thang máy

Người ở ngoài thang bấm nút:

- Có người bấm thang đi lên
- Có người bấm thang đi xuống
- Vừa bấm lên vừa bấm xuống ở 1 tầng
- Có nhiều người bấm ở nhiều tầng

Người trong thang bấm nút:

- 1 người bấm chọn tầng
- Nhiều người bấm chọn tầng
- Bấm nút open door
- Bấm nút close door

Các trường hợp đặc biệt:

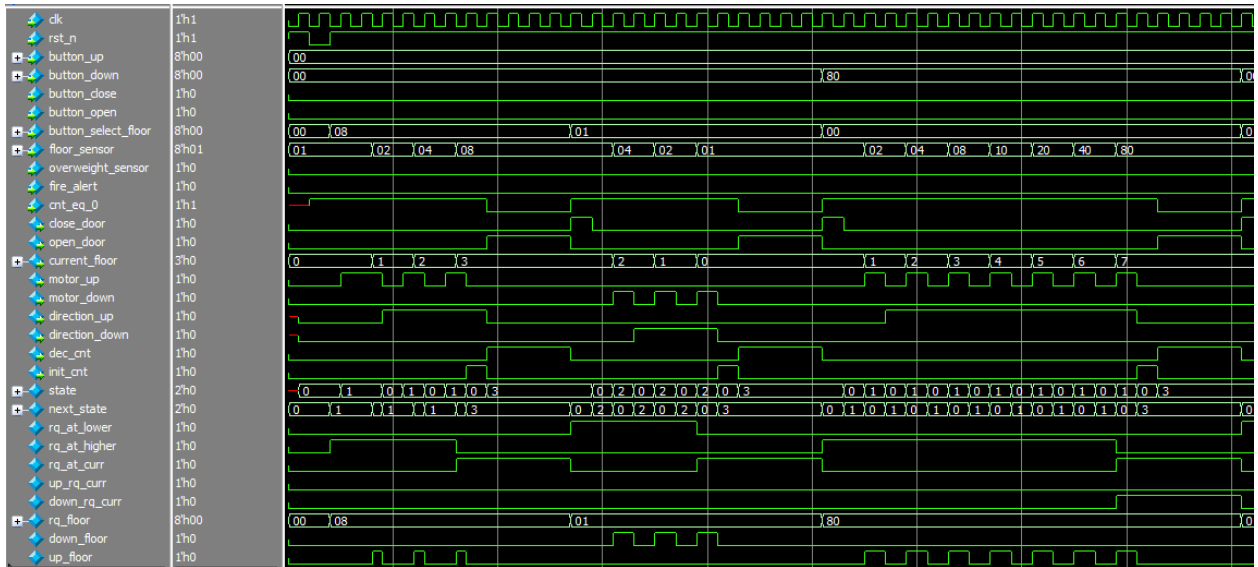
- Báo cháy
- Quá tải

4.1.1.2 Trường hợp có 2 thang máy

- Cả hai thang đang đi lên
- Một thang đi lên, một thang đi xuống
- Cả hai thang đang đi xuống

4.1.2 Kiểm thử thiết kế phần cứng bộ điều khiển cho 1 thang máy sử dụng ModelSim

Xây dựng testbench dựa trên kịch bản test và kiểm tra kết quả đầu ra của elevator controller sử dụng Verilog và ModelSim.

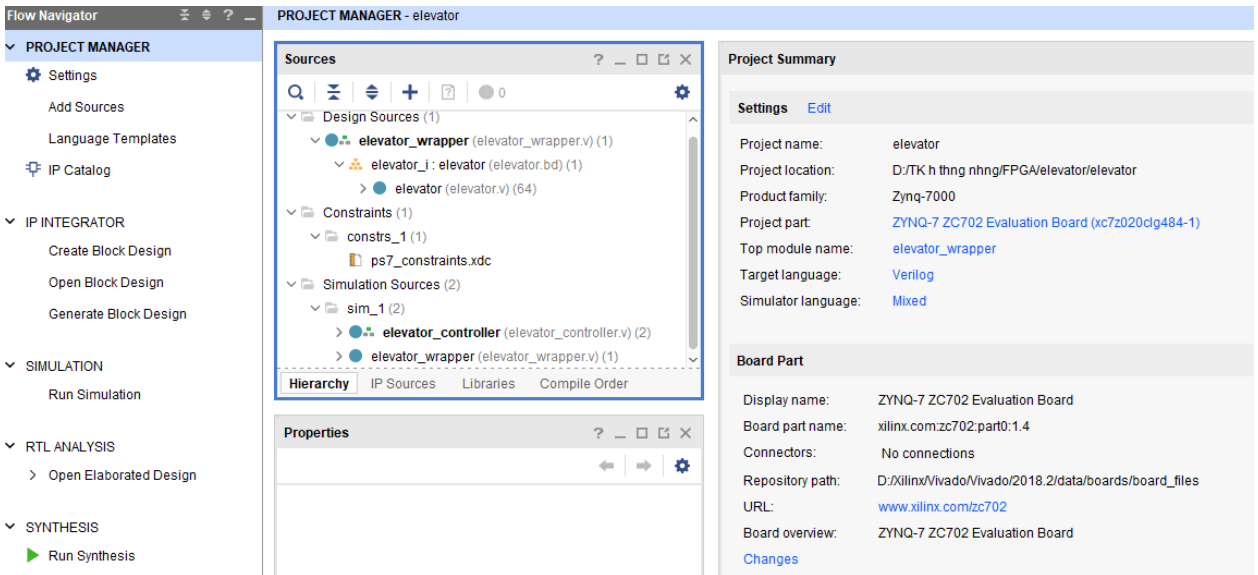


Hình 4.1 Kết quả mô phỏng

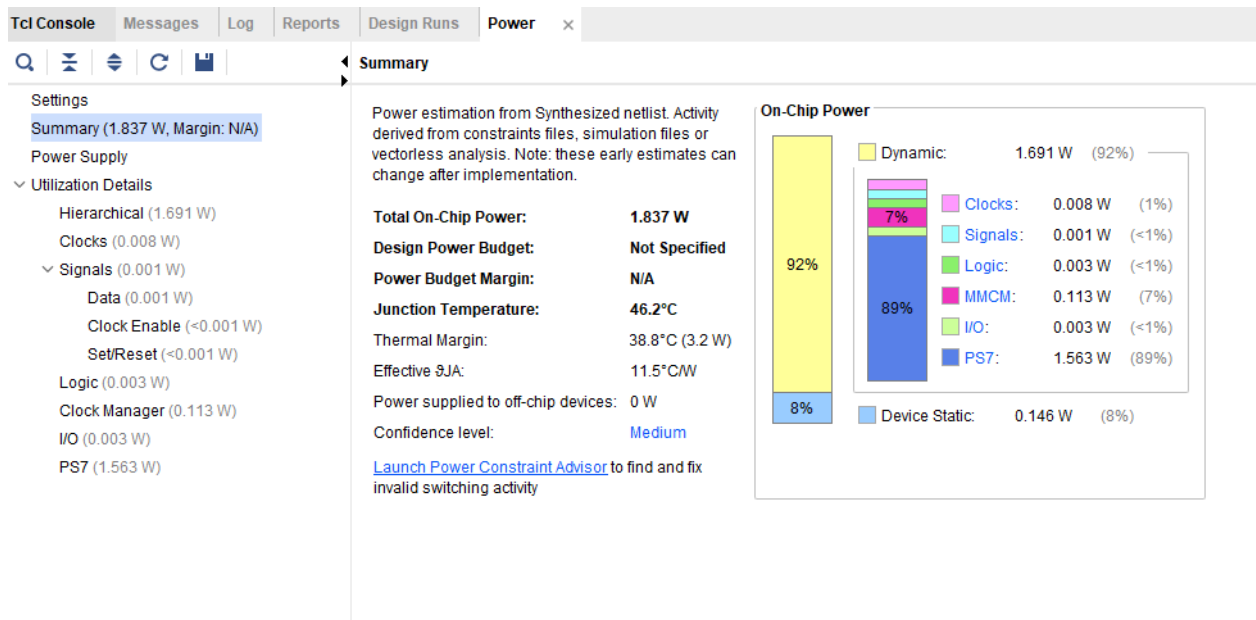
Hình 4.1 là kết quả mô phỏng đầu vào đầu ra của hệ thống phần cứng trên phần mềm ModelSim.

4.1.3 Triển khai tích hợp hệ thống sử dụng kit FPGA ZC702 và phần mềm Vivado

4.1.3.1 Tạo project, thêm các file source code mô tả phần cứng

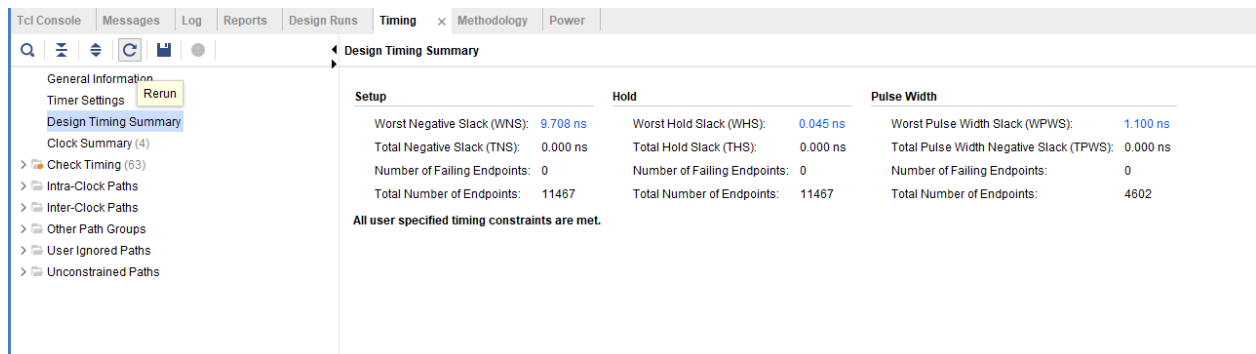


Hình 4.2 Tạo project trên phần mềm vivado



Hình 4.5 Năng lượng tiêu thụ ước tính

Hình 4.5 mô tả năng lượng tiêu thụ ước tính của bộ điều khiển thang máy khi chạy trên kit FPGA.



Hình 4.6 Báo cáo timing

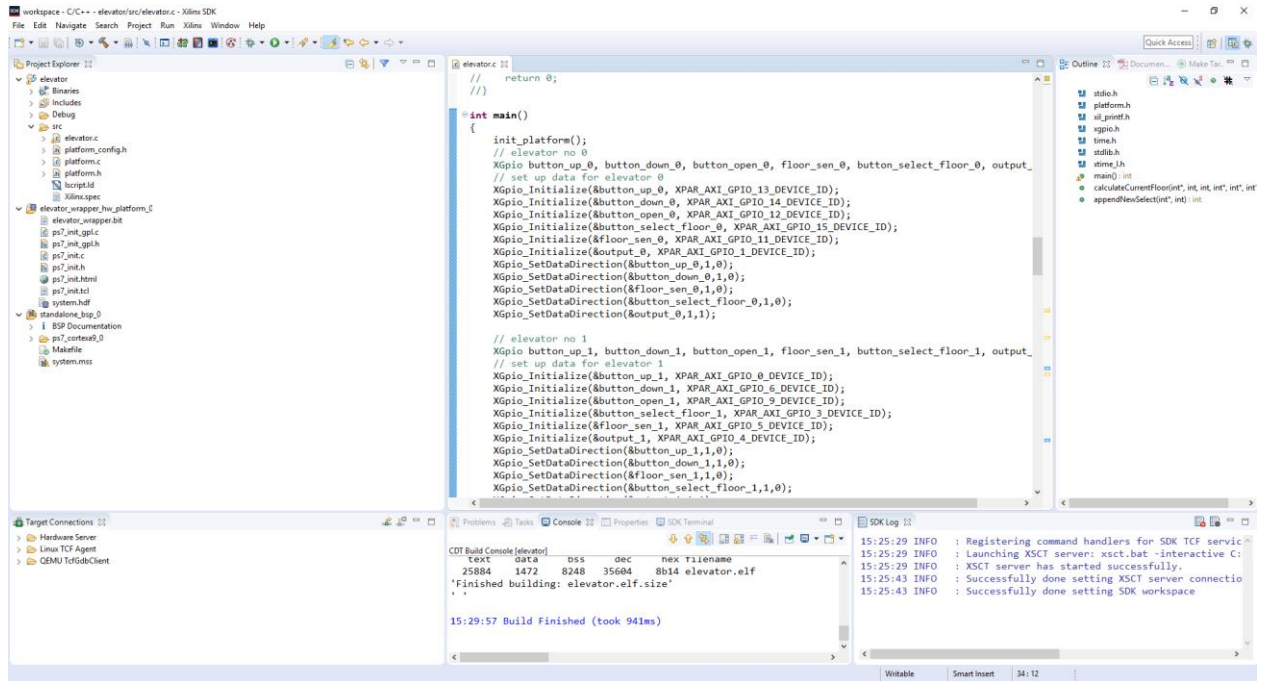
Hình 4.6 mô tả báo cáo timing của thiết kế sau khi tổng hợp trên FPGA.

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
synth_1	constrs_1	synth_design Complete!								5297	4528	0.00	0	0
impl_1	constrs_1	write_bitstream Complete!	2.157	0.000	0.046	0.000	0.000	1.832	0	4915	4328	0.00	0	0

Hình 4.7 Tài nguyên tiêu tốn

Hình 4.7 mô tả lượng tài nguyên tiêu tốn sau khi tổng hợp trên FPGA.

4.1.3.4 Gen bitstream và tạo SDK (software development kit)



Hình 4.8 Gen bitstream và tạo sdk

Sau khi gen bitstream trên Vivado, ta tiến hành tạo project trên phần mềm SDK, thiết lập môi trường biên dịch, platform hardware và biên dịch code.

4.2 Kết quả và đánh giá

Hệ thống đã chạy đúng với function đề ra ở khâu thiết kế. Độ trễ và năng lượng tiêu tốn thấp.

CHƯƠNG 5. BÁO CÁO CÁ NHÂN

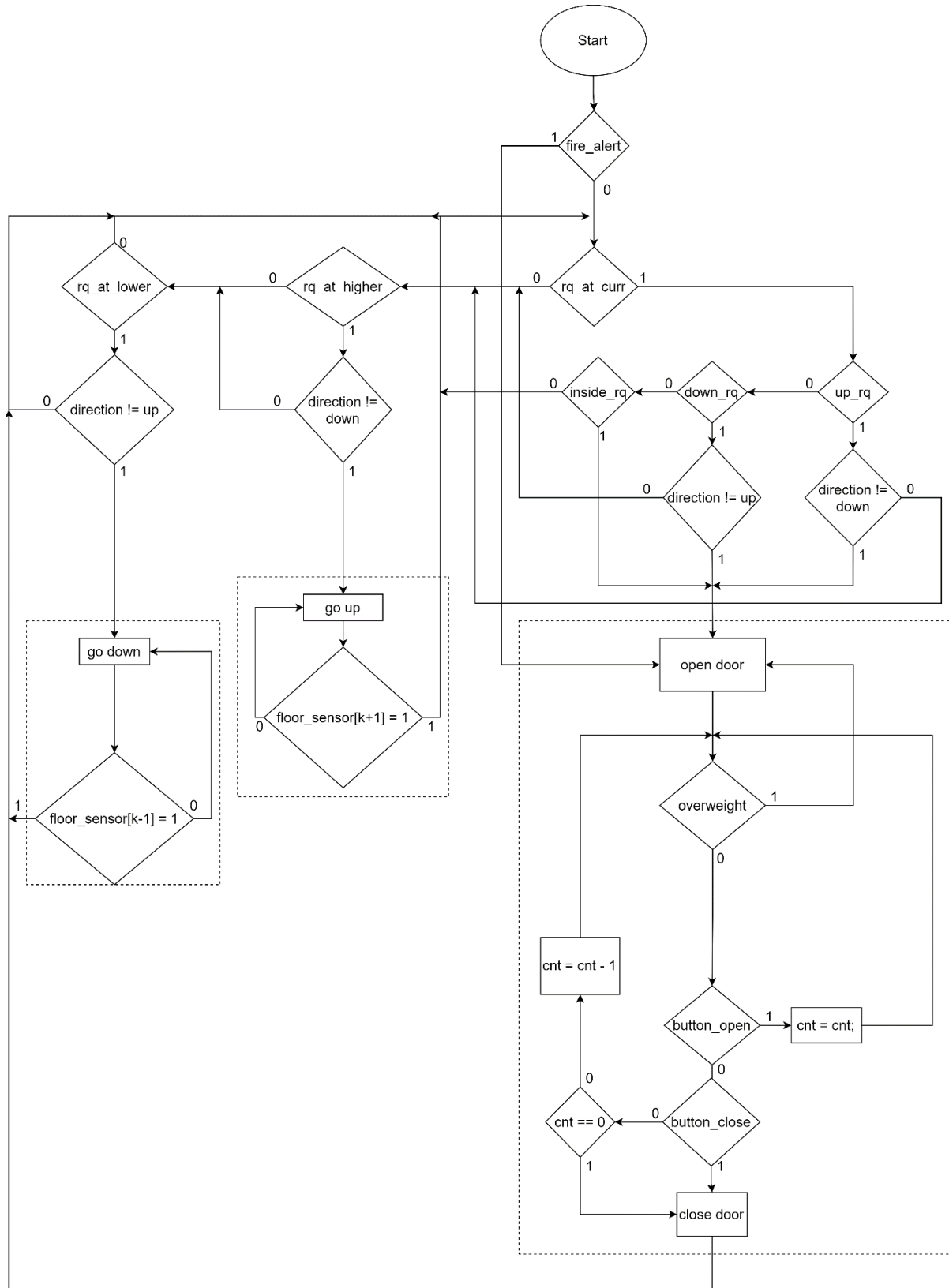
Trong chương này, em sẽ tổng hợp lại các phần mà em đã thực hiện và đóng góp vào kết quả chung của nhóm.

5.1 Thiết kế phần cứng

5.1.1 Lưu đồ thuật toán cho bộ điều khiển 1 thang máy

Lưu đồ thuật toán ở Hình 3.1 mô tả hoạt động của bộ điều khiển cho một thang máy. Giả sử thang máy đang ở tầng thứ k , thang máy sẽ hoạt động như sau:

- Nếu có tín hiệu cảnh báo cháy từ cảm biến báo cháy, đến tầng gần nhất và mở cửa.
- Nếu có yêu cầu đi lên tại tầng k (nút bấm ngoài thang máy), thang máy đang không đi xuống thì thang máy sẽ mở cửa.
- Nếu có yêu cầu đi xuống tại tầng k , thang máy đang không đi lên thì thang máy sẽ mở cửa.
- Nếu có yêu cầu đi đến tầng k (nút bấm trong thang máy), mở cửa thang máy.
- Nếu có yêu cầu đi lên, đi xuống hoặc đi đến tầng k tại tầng thấp hơn tầng k và thang máy đang không đi lên thì điều khiển thang máy đi xuống.
- Nếu có yêu cầu đi lên, đi xuống hoặc đi đến tầng k tại tầng cao hơn tầng k và thang máy đang không đi xuống thì điều khiển thang máy đi lên.
- Trong quá trình mở cửa, bộ điều khiển kiểm tra xem có tín hiệu overweight nhận từ cảm biến hay không, nếu overweight thì thang máy sẽ duy trì trạng thái mở cửa. Nếu không có tín hiệu overweight, thang máy sẽ kiểm tra xem có yêu cầu mở cửa hay không, nếu có yêu cầu mở cửa, thang máy sẽ duy trì trạng thái mở cửa, nếu có yêu cầu đóng cửa, thang máy sẽ đóng cửa. Nếu không có yêu cầu mở cửa hay đóng cửa nào từ người dùng, thang máy sẽ mở cửa trong 1 khoảng thời gian sau đó đóng cửa.



Hình 5.1 Lưu đồ thuật toán mô tả hoạt động thang máy

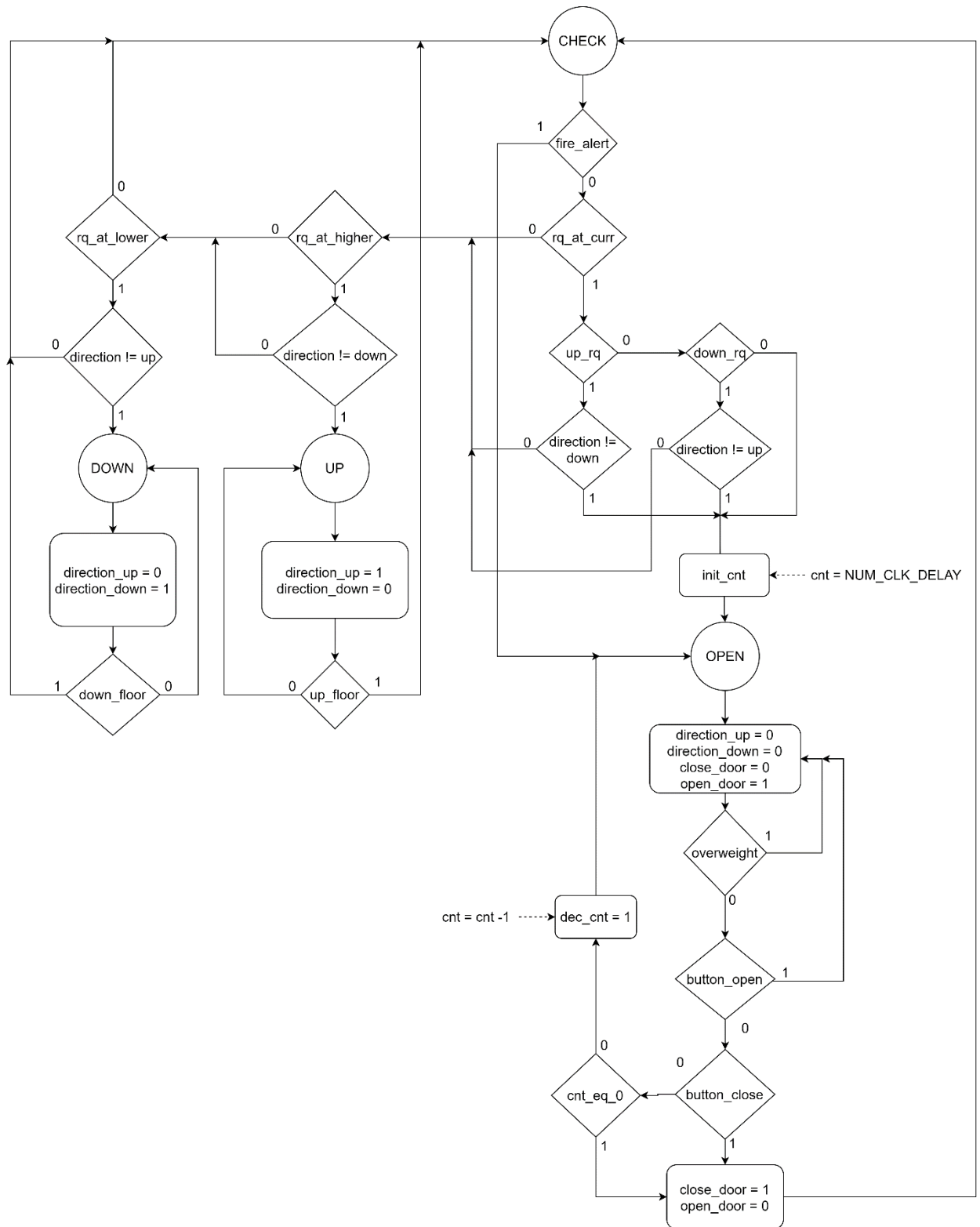
Chú thích các tín hiệu:

- $rq_at_curr = button_select_floor[k] \mid button_up[k] \mid button_down[k]$
- $rq_at_higher = button_select_floor[g] \mid button_up[g] \mid button_down[g] \mid g > k$
- $rq_at_lower = button_select_floor[g] \mid button_up[g] \mid button_down[g] \mid g < k$

5.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath)

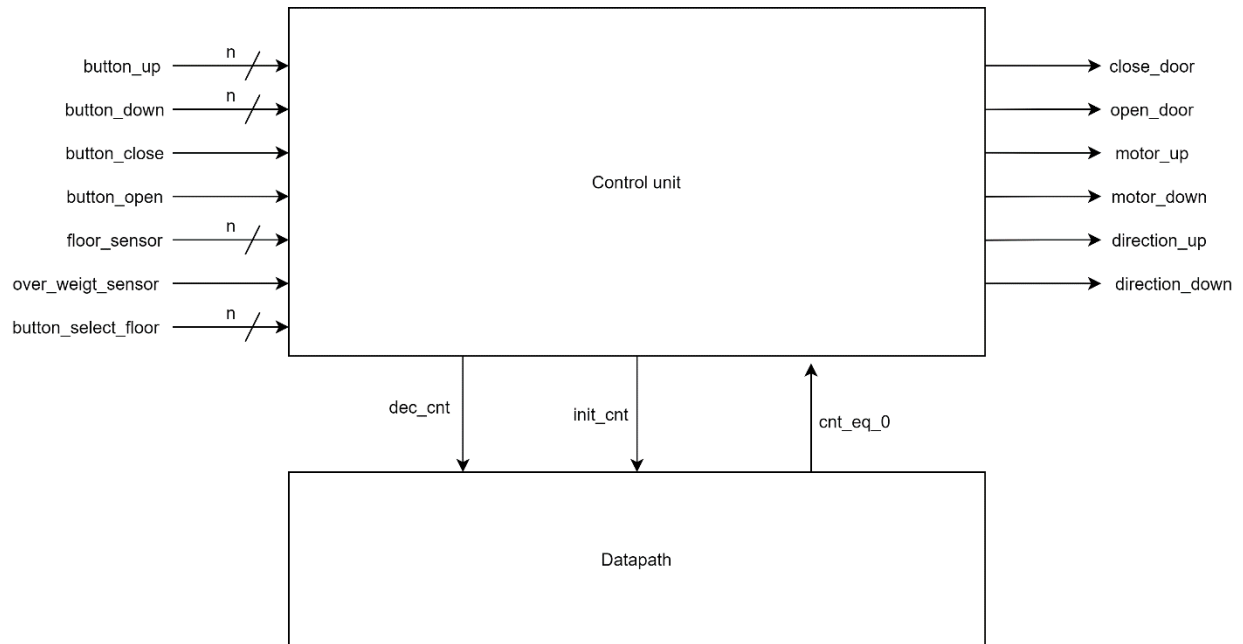
Từ sơ đồ thuật toán hoạt động của bộ điều khiển một thang máy, khoanh vùng xác định được bốn trạng thái:

- CHECK: trạng thái kiểm tra các yêu cầu được gửi tới thang máy.
- OPEN: trạng thái mở cửa thang máy.
- UP: trạng thái thang máy đi lên.
- DOWN: trạng thái thang máy đi xuống.



Hình 5.2 Sơ đồ ASMD

Từ sơ đồ ASMD ta xác định được sơ đồ khối giao tiếp giữa control unit và data path của hệ thống:



Hình 5.3 Sơ đồ kết nối khối Control unit và datapath

Bảng 5-1 Bảng mô tả thông số kỹ thuật

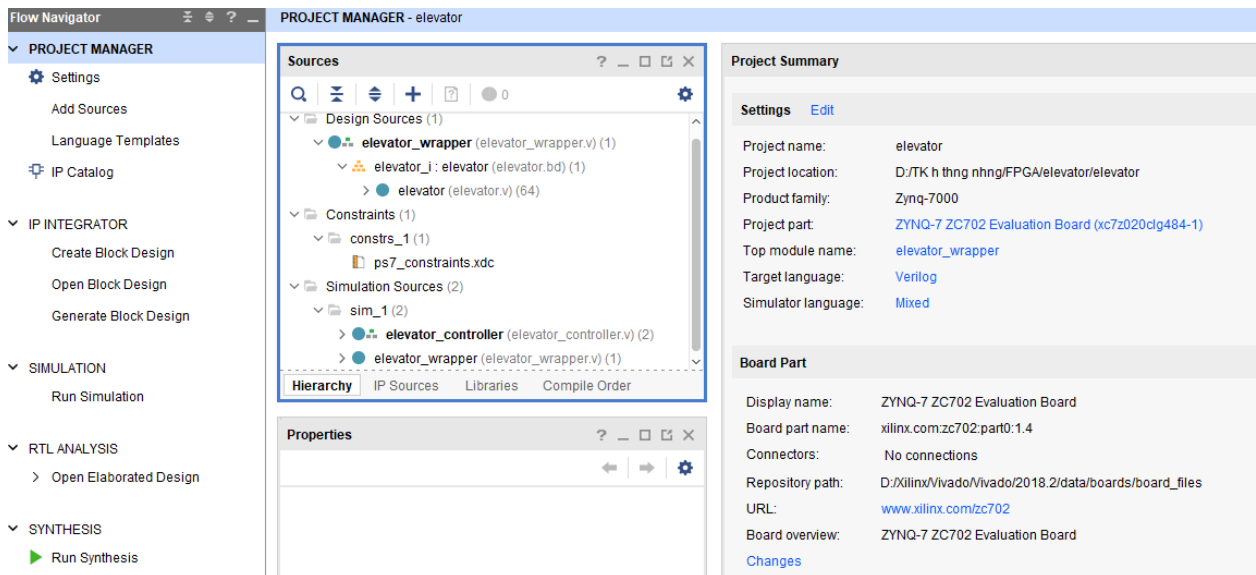
STT	Tên tín hiệu	Độ rộng (bit)	Chiều	Mô tả
1	button_up	n	input	Nút bấm lên tại n tầng bên ngoài thang máy
2	button_down	n	input	Nút bấm xuống tại n tầng bên ngoài thang máy
3	button_close	1	input	Nút bấm đóng cửa bên trong thang máy
4	button_open	1	input	Nút bấm mở cửa bên trong thang máy
5	floor_sensor	n	input	Cảm biến xác định vị trí của thang máy
6	overweight_sensor	1	input	Tín hiệu xác định thang máy quá tải trọng

7	button_select_floor	n	input	Nút bấm chọn n tầng trong thang máy
8	close_door	1	output	Tín hiệu đóng cửa thang máy
9	open_door	1	output	Tín hiệu mở cửa thang máy
10	motor_up	1	output	Tín hiệu điều khiển thang máy đi lên
11	motor_down	1	output	Tín hiệu điều khiển thang máy đi xuống
12	direction_up	1	output	Tín hiệu trạng thái thang máy đang đi lên
13	direction_down	1	output	Tín hiệu trạng thái thang máy đang đi xuống

Từ sơ đồ ASMD sử dụng ngôn ngữ mô tả phần cứng Verilog triển khai controller cho 1 thang máy.

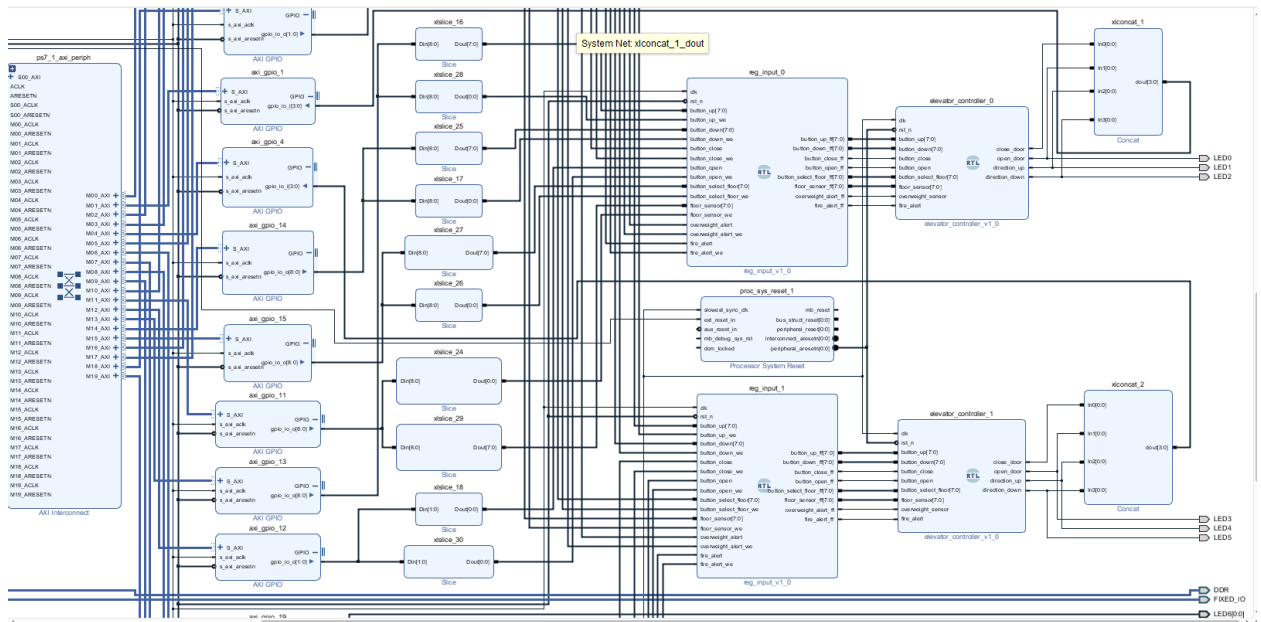
5.1.3 Triển khai tích hợp hệ thống sử dụng kit FPGA ZC702 và phần mềm Vivado

5.1.3.1 Tạo project, thêm các file source code mô tả phần cứng



Hình 5.4 Tạo project trên phần mềm vivado

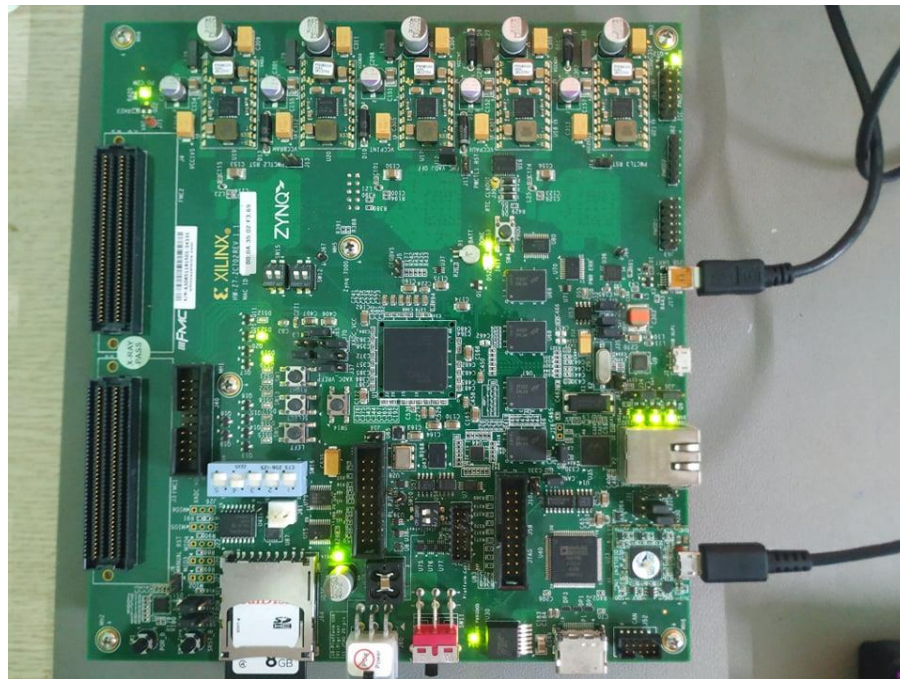
5.1.3.2 Tạo Block design



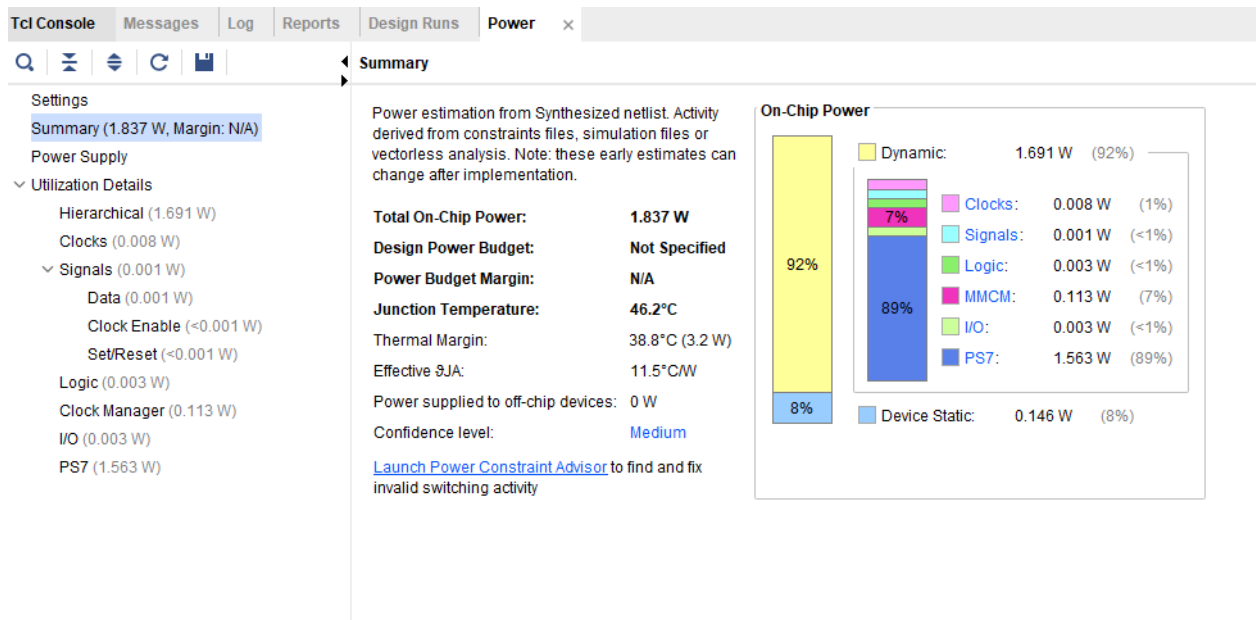
Hình 5.5 Block design

Hình 4.3 Block design Hình trên mô tả một phần của block design được thiết kế bằng phần mềm Vivado.

5.1.3.3 Tổng hợp và implement thiết kế phân cứng trên Vivado

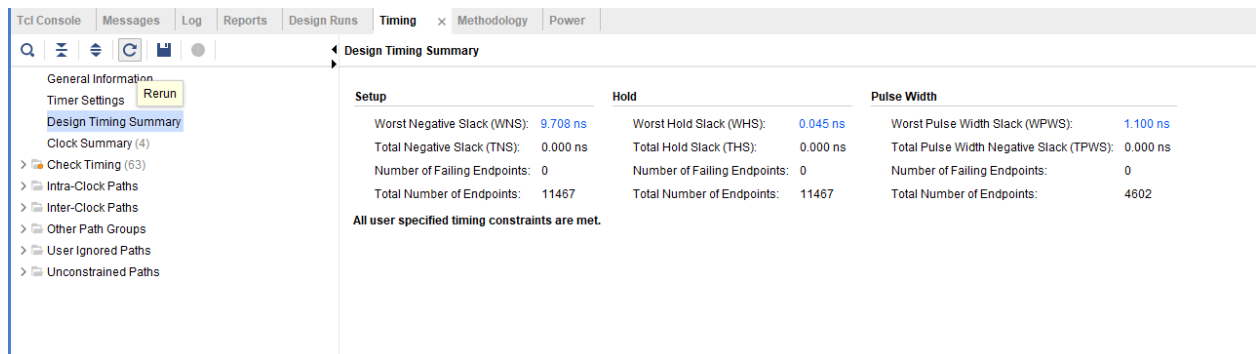


Hình 5.6 Kit FPGA ZC702



Hình 5.7 Năng lượng tiêu thụ ước tính

Hình trên mô tả năng lượng tiêu thụ ước tính của bộ điều khiển thang máy khi chạy trên kit FPGA.



Hình 5.8 Báo cáo timing

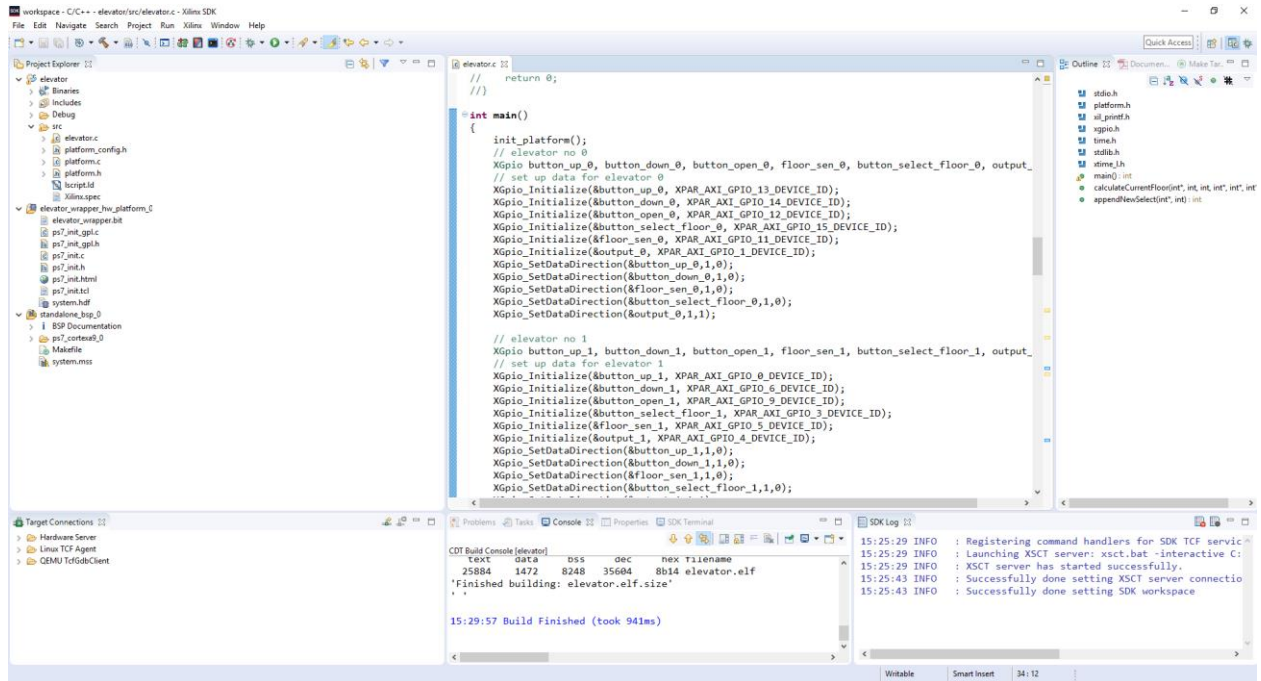
Hình trên mô tả báo cáo timing của thiết kế sau khi tổng hợp trên FPGA.

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								5297	4528	0.00	0	0
✓ impl_1	constrs_1	write_bitstream Complete!	2.157	0.000	0.046	0.000	0.000	1.832	0	4915	4328	0.00	0	0

Hình 5.9 Tài nguyên tiêu tốn

Hình trên mô tả lượng tài nguyên tiêu tốn sau khi tổng hợp trên FPGA.

5.1.3.4 Gen bitstream và tạo SDK (software development kit)



Hình 5.10 Gen bitstream và tạo sdk

Sau khi gen bitstream trên Vivado, ta tiến hành tạo project trên phần mềm SDK, thiết lập môi trường biên dịch, platform hardware và biên dịch code.

5.2 Kết quả và đánh giá

Hệ thống đã chạy đúng với function đề ra ở khâu thiết kế. Độ trễ và năng lượng tiêu tốn thấp (1.8W).

Link mã nguồn và tài liệu liên quan thiết kế phần cứng:
https://github.com/chien172431/elevator_controller_verilog.git

CHƯƠNG 6. KẾT LUẬN

Sau một thời gian nghiên cứu về hệ thống nhúng, đặc biệt là hệ thống nhúng điều khiển thang máy cao ốc, nhóm em đã nắm được các quy trình thiết kế một sản phẩm nhúng cũng như cấu tạo của thang máy và hoạt động của nó. Đồng thời nhóm đã thiết kế hệ thống điều khiển thang máy cao ốc trên Xilinx Zynq-7000 SoC ZC702. Hệ thống gồm các sensor, button sẽ nhận tín hiệu và đưa vào bộ chọn thang được viết bằng ngôn ngữ C, sau đó các tín hiệu sẽ được gửi tới bộ điều khiển được mô tả bằng ngôn ngữ Verilog để điều khiển hoạt động của thang.

Hệ thống điều khiển thang máy hoạt động đúng như những chỉ tiêu kỹ thuật đã được đề ra đảm bảo hoạt động theo yêu cầu. Trong quá trình thực hiện do thời gian và kiến thức còn nhiều sai sót, mong thầy cô và các bạn đóng góp ý kiến để nhóm em hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn TS. Phạm Văn Tiến đã giúp đỡ nhóm chúng em trong quá trình thực hiện bài tập lớn này.

TÀI LIỆU THAM KHẢO

- [1] Programming Embedded Systems in C and C++
- [2] <https://www.asic-world.com/systemc/tutorial.html>