

## Design Document

Elevator mechanical and controller Model (For studying)

แบบจำลองกลไกการทำงานและการควบคุมของลิฟต์ (เพื่อการศึกษา)

### สมาชิกโครงการ

นางสาวกัญญารัตน์ ไชยยันต์บุรณ์	รหัสนักศึกษา 64010037	กลุ่ม18
นางสาวบัณฑิตา วงศ์วรรณ	รหัสนักศึกษา 64010454	กลุ่ม18
นายพิตรพิบูล พงษ์พจนธรรม	รหัสนักศึกษา 64010591	กลุ่ม19
นางสาววิมลศิริ ธรรมดา	รหัสนักศึกษา 64010813	กลุ่ม19
นายสิรภพ แสงมี	รหัสนักศึกษา 64010893	กลุ่ม20

## สารบัญ

บทนำ .....	4
1.1 ที่มาและความสำคัญของโครงการ .....	4
1.2 จุดประสงค์ .....	4
กระบวนการออกแบบ .....	5
2.1 ทฤษฎีและการประยุกต์ใช้.....	5
2.1.1 SR Latch .....	5
2.1.2 Shifter .....	5
2.1.3 Decorder .....	6
2.2 การออกแบบวงจร .....	7
2.2.1 Module CALL LOGIC .....	8
2.2.2 Module Direction.....	9
2.2.3 Module MOTOR & DOOR LOGIC .....	10
2.2.4 Module DOORSIM.....	11
2.2.5 Module POS LOGIC .....	12
2.3 การออกแบบระบบลิฟต์ .....	13
2.3.1 ตัวลิฟต์และโครงลิฟต์ .....	13
2.3.2 รอก .....	14
2.3.3 เชือก .....	15
2.3.4 มอเตอร์ .....	16
กระบวนการพัฒนา.....	17
3.1 เทคนิคที่ใช้ .....	17
3.2 รูปแบบของวงจรที่ได้พัฒนา .....	18
3.2.1 ลิฟต์เวอร์ชัน 1 (3-Floor-Elevator) .....	18

3.2.2 ลิฟต์เวอร์ชัน 2 (4-Floor-Elevator) .....	19
3.3.3 ลิฟต์เวอร์ชัน 3 (4-Floor-Elevator) Clock And Sensor .....	20
3.3 กระบวนการทดสอบ .....	21
3.3.1 ทดสอบการทำงานของวงจร .....	21
3.3.2 ทดสอบการสื่อสารระหว่าง FPGA 2 บอร์ด .....	21
3.3.3 ทดสอบการใช้ pin เป็น output .....	21
3.3.4 ทดลองเปลี่ยนค่า Clock .....	21
3.3.5 ทดลองต่อเซนเซอร์ติดแต่ละชั้น .....	22
3.4 ปัญหาและแนวทางการแก้ไข .....	23
3.4.1 ปัญหาด้าน Software .....	23
3.4.2 ปัญหาด้าน Hardware .....	23
Top-Down Design .....	24
1 <sup>st</sup> layer Elevator .....	24
2 <sup>nd</sup> layer call_logic .....	24
2 <sup>nd</sup> layer Direction .....	26
2 <sup>nd</sup> layer MOTOR & DOOR LOGIC .....	27
2 <sup>nd</sup> layer doorsim .....	28
2 <sup>nd</sup> layer Lift_Position .....	29
5. ลิ้งค์วีดิทัศน์แนะนำชิ้นงาน .....	30
อ้างอิง .....	31

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

ลิฟต์เป็นปัจจัยที่สำคัญสำหรับการขนหรือเคลื่อนย้ายสิ่งของ การพักอาศัย และการทำงานของมนุษย์ในตึกที่มีความสูงมาก ๆ รวมทั้งเป็นตัวช่วยอำนวยความสะดวกแก่ผู้พิการแทนการใช้บันไดอีกด้วย ซึ่งจะเห็นได้ว่าไม่เพียงแต่ตึกสูงระฟ้าเท่านั้นที่นำลิฟต์มาใช้ประโยชน์ภายในอาคาร แต่สำหรับอาคารที่มีจำนวนชั้นไม่มากก็ยังจำเป็นต้องใช้ประโยชน์จากลิฟต์ด้วยเช่นเดียวกัน

อีกทั้งในปัจจุบัน อุตสาหกรรมการออกแบบวงจรรวมมีการพัฒนาอย่างรวดเร็วและมีความต้องการสูง เพราะเป็นอุปกรณ์ที่มีขนาดเล็กแต่สามารถทำงานได้อย่างมีประสิทธิภาพซึ่งการออกแบบพัฒนางจรรวมเบื้องต้นสามารถทำได้โดยใช้ FPGA (field-programmable gate array) โดยข้อได้เปรียบของ FPGA คือสามารถพัฒนาและแก้ไขการทำงานของวงจรที่ได้ออกแบบไปแล้วอย่างง่ายดาย

ดังนั้นผู้จัดทำจึงอยากจะศึกษาวิธีการทำงานของลิฟต์และพัฒนางจรที่สามารถทำให้ลิฟต์เคลื่อนที่ได้โดยใช้ FPGA ที่เหมาะสำหรับการออกแบบวงจรที่ซับซ้อน และสร้างโมเดลจำลองเพื่อให้เห็นภาพการทำงานของลิฟต์ได้อย่างสมจริงและมีประสิทธิภาพ

### 1.2 จุดประสงค์

เป้าหมายของโครงการนี้เพื่อเป็นการศึกษาและเพิ่มประสบการณ์การทำงานโดยใช้บอร์ด FPGA ในการทำงานที่มีขอบเขตกว้างกว่าในห้องเรียน จึงกำหนดวัตถุประสงค์ไว้ดังนี้

1. เพื่อศึกษาวิธีการประยุกต์ใช้งานบอร์ด FPGA สำหรับควบคุมการทำงานของลิฟต์
2. สามารถนำความรู้มาใช้เพื่อออกแบบวงจรสำหรับการเคลื่อนที่ของลิฟต์
3. สร้างโมเดลที่สามารถแสดงวิธีการทำงานของลิฟต์ได้อย่างชัดเจนและมีประสิทธิภาพ

## กระบวนการออกแบบ

### 2.1 ทฤษฎีและการประยุกต์ใช้

การดำเนินโครงการมีการใช้ SR-flipflop , Shifter และ Decoder ในการต่อวงจร circuit

#### 2.1.1 SR Latch

จะมี state อยู่ 2 state คือ 0 หรือ 1 สำหรับ SR มี 2 input คือ set กับ reset เช่น set = 1 Output จะออก 1 ไปเรื่อย ๆ จนกว่าจะกด reset = 1 output ค่อยเปลี่ยนเป็น 0 ซึ่งให้ set = 1 และ reset = 1 ไม่ได้ เพราะจะทำให้ข้อขัด

Latch 1 ตัว เก็บข้อมูลได้ 1 bit = 2 state ถ้ามี n ตัวก็เก็บได้ 2 กำลัง n ตัว เช่น ถ้ามีทั้งหมด 5 state ก็ต้องใช้ latch 3 ตัวเก็บ (2 กำลัง 3) เอาให้มากกว่าหรือเท่ากับ

การประยุกต์ใช้ SR Latch นั้นจะเป็นการรับค่า input 0 หรือ 1 เข้า Latch ประยุกต์ใช้ให้เข้ากับ วงจรลิฟต์ ตัวอย่างเช่น มีหน้าที่จำชั้นลิฟต์ที่กด เป็นการ stack ค่า รับ input จากสวิทช์ที่กดเลือกชั้น ซึ่งเป็นการรับค่า 0 หรือ 1 จากนั้นส่งให้มอเตอร์ มีหน้าที่จำว่าประตูมีการเปิด-ปิดหรือไม่ โดยรับ input เป็นค่าการหยุดของลิฟต์ ว่าลิฟต์หยุดหรือไม่ โดยถ้ารับ 1 เข้า SR latch ทำให้จำว่าประตูเปิด และถ้ารับค่าเป็น 0 ให้ SR latch จำว่าประตูปิด

#### 2.1.2 Shifter

มีหน้าที่เป็น Delay ให้กับการเปิด-ปิดประตูลิฟต์เมื่อจอดลิฟต์ลงชั้นที่ต้องการ โดย Shifter มีลักษณะการทำงานที่รอบที่ข้อมูลของ clock จะเขยิบเปลี่ยนไปเรื่อยๆ latch แต่ละตัวตอนแรกมี data เก็บไว้ ต่อมาก็จะ load Data ตัวใหม่มาโดย load จาก latch ตัวก่อนหน้ามันพร้อมกับโยน data ตัวเก่าของมันให้ latch ตัวต่อไป

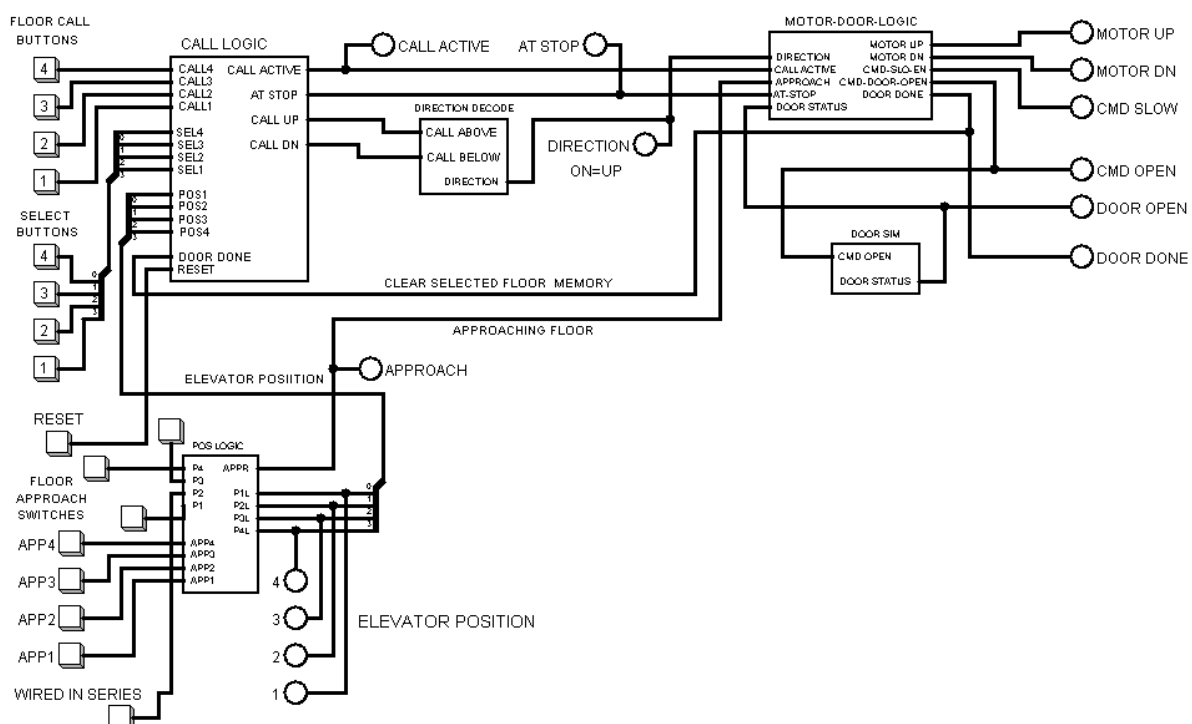
การประยุกต์ใช้ Shifter กับวงจรลิฟต์นั้น ตัว Shifter จะรับค่า 1 เป็น MSB (บิตที่อยู่ซ้ายสุด) มาจาก SR latch แล้ว right shift ไปให้ค่า 1 ตรงกับหลักที่ 2 จะส่งค่า 1 ไป set ของ SR latch ตัวต่อไป (set = 1 , reset = 0) ส่งผลให้ประตูเปิด(1) right shift ไปเรื่อยๆ รอให้ค่า 1 ตรงกับ LSB (บิตที่อยู่ขวาสุด) จะส่งค่า 1 ไป reset ของ SR latch ตัวต่อไป (set = 0 , reset = 1) ส่งผลให้ประตูปิด(0)

### 2.1.3 Decoder

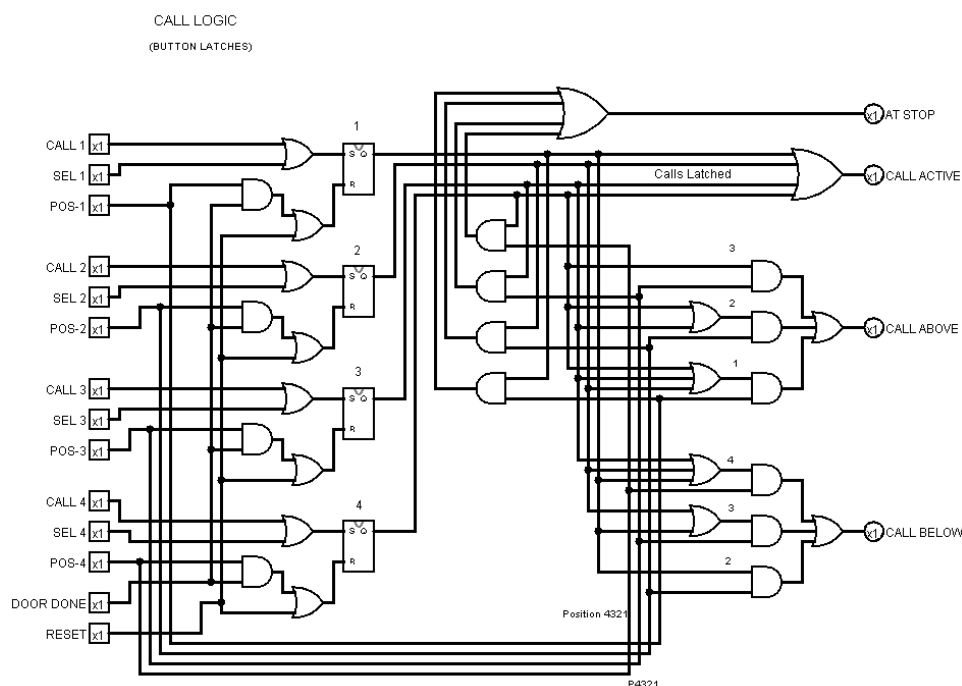
วงจรลอจิกซึ่งทำหน้าที่เปลี่ยนรหัสเลขฐานสองที่มีอินพุตจำนวน  $N$  บิต ให้เป็นรหัสใด ๆ ที่มีสายเอาต์พุตจำนวน  $M$  บิต โดยแต่ละสายจะได้รับผลออกมาจากการจัดหมู่ของอินพุตที่เหมาะสมเพียงกลุ่มเดียว เช่น เลขฐานสองที่มีอินพุตจำนวน 3 บิต เป็น 011 จะต้อง output ออกมาเป็น  $M$  เส้นที่ 4(M3) จากทั้งหมด 8 เส้นเพราะมี output 8 ตัว

การประยุกต์ใช้ Decoder กับวงจรลิฟต์นั้น เราจะต้องรู้สถานะของลิฟต์ว่าอยู่ชั้นอะไร จากนั้นก็เลือกชั้นลิฟต์ที่ต้องการจะไป ตัว Decoder ก็จะได้รับ input ที่ได้เป็น 1 หรือ 0 ประมวลผลกับวงจร ถ้า state ลิฟต์เป็นชั้นที่ต้องการจะขึ้น DC motor ก็จะหมุนขึ้น ถ้า state ลิฟต์เป็นชั้นที่ต้องการจะลง DC motor ก็จะหมุนลง

## 2.2 การออกแบบวงจร



## 2.2.1 Module CALL LOGIC



โมดูล CALL LOGIC ทำหน้าที่รับ Input มาจาก switch 2 ชุด คือ CALL n และ SEL n ซึ่งรับมาจากบอร์ด FPGA 2 บอร์ด โดย CALL n ทำหน้าที่สำหรับเรียกลิฟต์ และ SEL n ทำหน้าที่สำหรับเลือกชั้นที่ต้องการจะไป สามารถทำการ stack เก็บค่าได้ว่าการเรียกลิฟต์ไปที่ชั้นใดบ้าง และมีการลำดับความสำคัญ (สั่งการให้ลิฟต์ไปจอดชั้นที่อยู่ใกล้ก่อน)

Output **AT\_STOP** แสดงผลว่าลิฟต์หยุดการทำงานแล้ว ไม่ได้มีการเคลื่อนที่อยู่ โดยรับค่ามาจาก Output ของโมดูล Lift\_Position ที่ทำหน้าที่ตรวจสอบว่าลิฟต์อยู่ที่ชั้นใดในขณะนั้น และมาจากการดำเนินการ(SR-flipflop) ของลิฟต์แต่ละชั้น

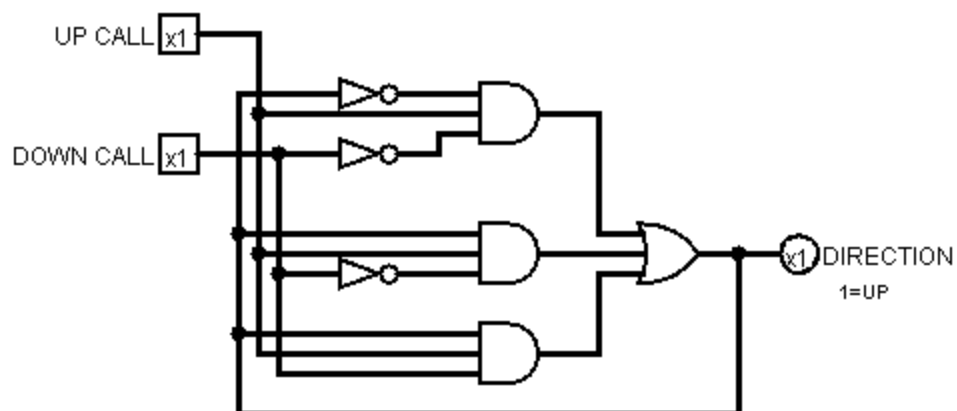
Output **CALL\_ACTIVE** แสดงผลว่าลิฟต์ยังมีการทำงานหรือเคลื่อนที่อยู่ โดยรับค่ามาจากการดำเนินการ(SR-flipflop) ของลิฟต์แต่ละชั้น

Output **CALL\_UP** แสดงผลว่าลิฟต์มีคำสั่งให้เคลื่อนที่ขึ้น

Output **CALL\_DN** แสดงผลว่าลิฟต์มีคำสั่งให้เคลื่อนที่ลง

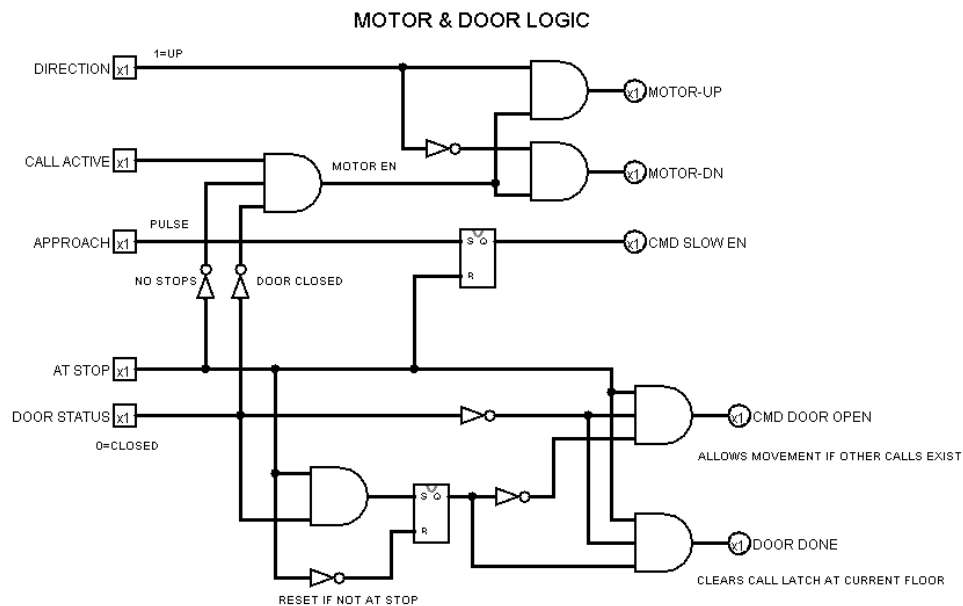


### 2.2.2 Module Direction



โมดูล Direction ทำหน้าที่สั่งการมอเตอร์ให้หมุนขึ้นหรือลง โดยรับ Input มาจาก Output ของโมดูล CALL LOGIC โดย UP CALL เป็นตัวกำหนดว่าให้ลิฟต์มีทิศทางการเคลื่อนที่ขึ้น และ DOWN CALL จะทำให้ลิฟต์มีทิศทางการเคลื่อนที่ลง เมื่อผ่าน Logic gate ต่างๆ จะแสดงผลออกมาทาง DIRECTION โดยถ้าแสดงผลออกมาเป็น 1 หมายความว่า ลิฟต์เคลื่อนที่ขึ้น

### 2.2.3 Module MOTOR & DOOR LOGIC



โมดูล MOTOR & DOOR LOGIC ทำหน้าที่สั่งการให้มอเตอร์หมุนเพื่อให้ลิฟต์เคลื่อนที่ขึ้นหรือลงได้ โดยการรับค่าจาก DIRECTION, CALL\_ACTIVE, APPROACH, AT\_STOP และ DOOR\_STATUS

Output MOTOR\_UP สั่งการให้มอเตอร์หมุนให้ลิฟต์เคลื่อนที่ขึ้น

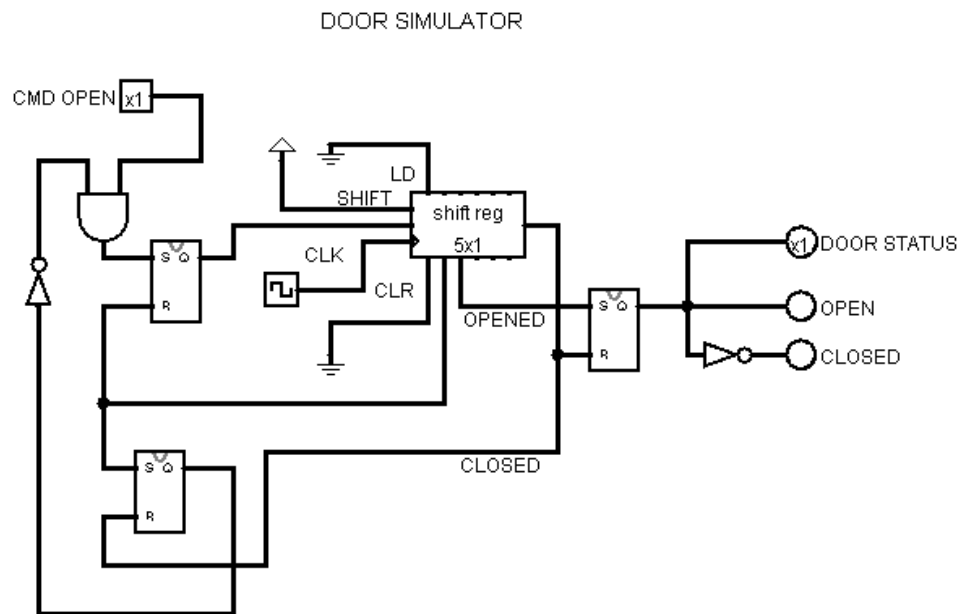
Output MOTOR\_DN สั่งการให้มอเตอร์หมุนให้ลิฟต์เคลื่อนที่ลง

Output CMD\_SLOW\_EN แสดงผลลิฟต์กำลังจะหยุด

Output CMD\_DOOR\_OPEN แสดงผลประตูลิฟต์กำลังจะเปิด

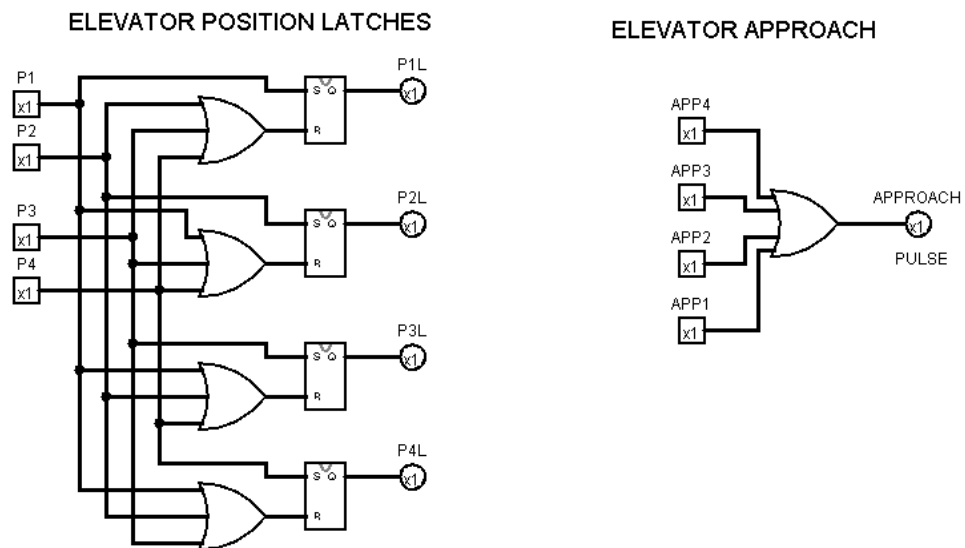
Output DOOR\_DONE แสดงผลว่าประตูลิฟต์ปิดอยู่

### 2.2.4 Module DOORSIM



โมดูล DOORSIM ทำหน้าที่ดำเนินการเกี่ยวกับประตูและ Output สถานะของประตู เช่น ประตูกำลังเปิด ประตูกำลังปิด

### 2.2.5 Module POS LOGIC



โมดูล POS LOGIC มีหน้าที่เพื่อระบุว่า ณ เวลานั้น ตัวลิฟต์อยู่ที่ชั้นใด โดยแสดงออก Output ผ่านทางไฟ LED

Input APP1- APP4 เป็นค่าจากเซนเซอร์ที่ตรวจสอบว่าลิฟต์อยู่ที่ชั้นใดในขณะนั้น เซนเซอร์จะตรวจจับระยะห่างของวัตถุ

## 2.3 การออกแบบระบบลิฟต์

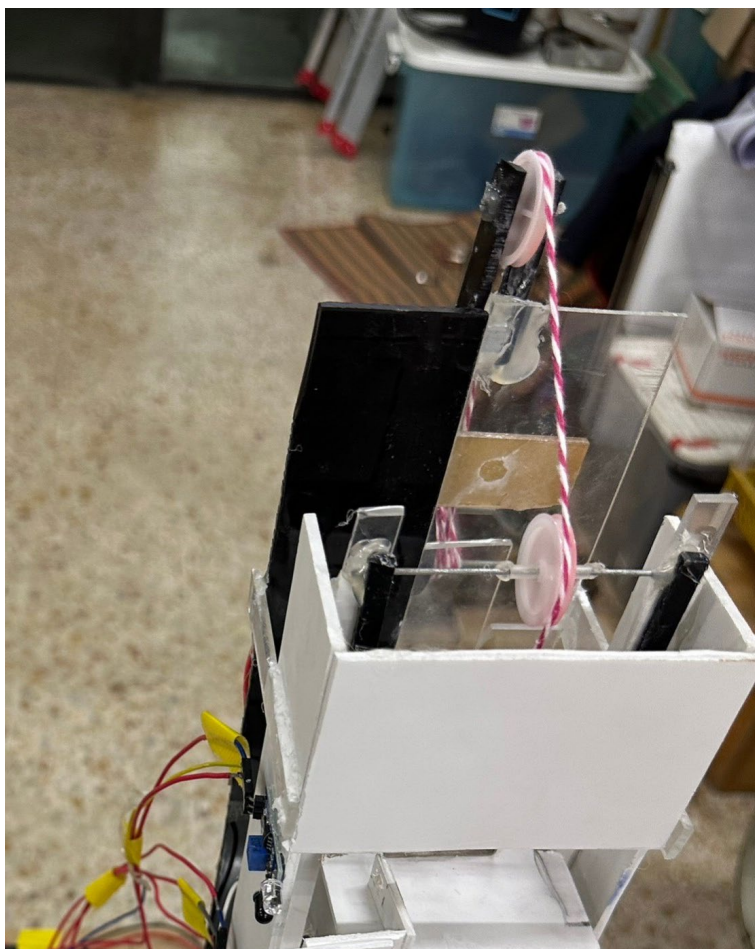
โมเดลลิฟต์ที่เราสร้างขึ้น ประกอบด้วยไม้พาสวูดเป็นหลัก อะคริลิก เชือก มอเตอร์ และมีบอร์ด FPGA ที่เป็นตัวสั่งการ

### 2.3.1 ตัวลิฟต์และโครงลิฟต์



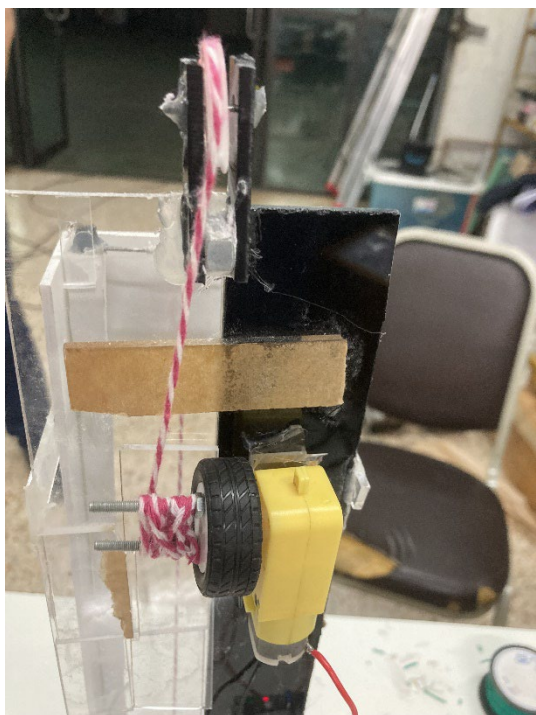
ตัวลิฟต์สามารถเคลื่อนที่ขึ้นลงได้ มีเชือกติดกับด้านบนของห้องโดยสาร และมีโครงลิฟต์เป็นรางที่ช่วยล๊อคให้ตัวลิฟต์เคลื่อนที่ได้นิ่ง

### 2.3.2 รอก



เชือกที่ติดอยู่กับด้านบนของห้องโดยสารของลิฟต์ จะนำมาพาดผ่านรอก 2 ตัว เพื่อนำไปติดกับมอเตอร์ เพื่อที่จะสามารถนำลิฟต์เคลื่อนที่ขึ้นหรือเคลื่อนที่ลงได้

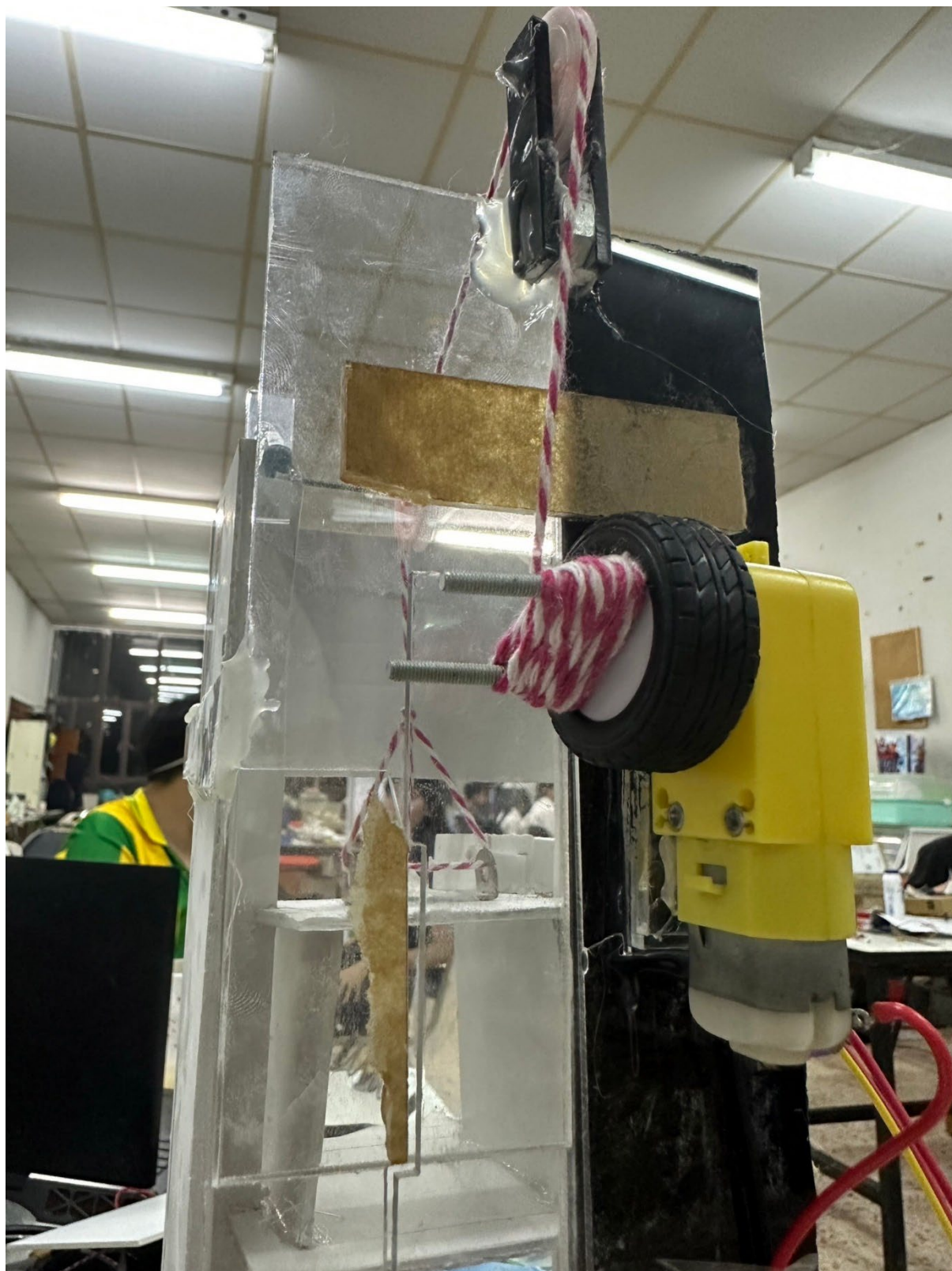
### 2.3.3 เชือก



เชือกที่พาดผ่านรอก 2 ตัว จะนำมาพันกับมอเตอร์ เพื่อดึงเก็บเชือกให้เชือกสั้นลง หรือปล่อยให้เชือกยาวขึ้น



### 2.3.4 มอเตอร์



มอเตอร์รับคำสั่งมาจากบอร์ด FPGA เพื่อทำการหมุนเก็บเชือกหรือปล่อยเชือก



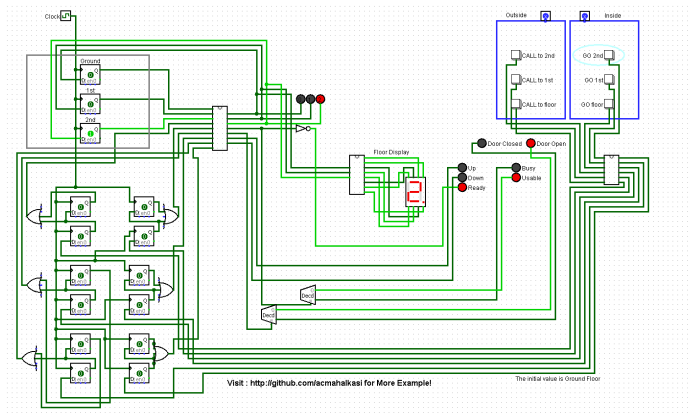
## กระบวนการพัฒนา

### 3.1 เทคนิคที่ใช้

ใช้เทคนิค Trial and Error เป็นหลักในการพัฒนาชิ้นงานแต่ละขั้นตอน จากนั้นสรุปผลลัพธ์ที่ได้ ออกเป็นข้อดีและข้อเสีย และนำปัญหาที่พบไปขยายผลสู่การพัฒนาชิ้นงานในรูปแบบต่อไป นอกจากนี้ก่อนที่จะเริ่มทำรูปแบบใหม่จะต้องมีการศึกษาข้อมูลจากแหล่งความรู้ที่เชื่อถือได้ เพื่อให้แน่ใจว่าวงจรนี้สามารถนำไปประยุกต์ใช้เพื่อพัฒนาชิ้นงานของเราได้จริง

## 3.2 รูปแบบของวงจรที่ได้พัฒนา

### 3.2.1 ลิฟต์เวอร์ชัน 1 (3-Floor-Elevator)



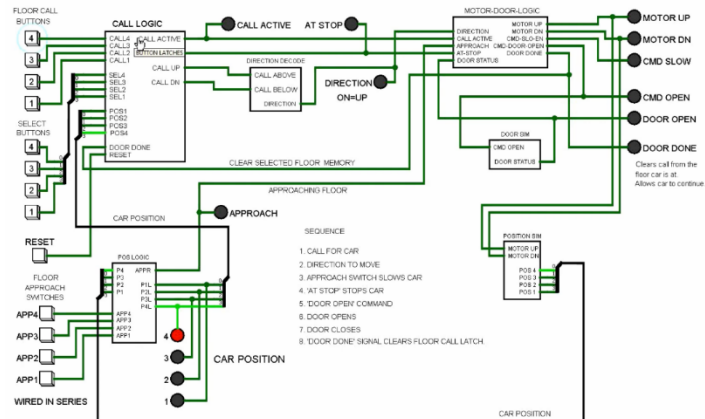
#### คุณสมบัติ

1. ลิฟต์สามารถเคลื่อนที่ได้ทั้งหมด 3 ชั้น
2. มีปุ่มกด 2 ชุด อย่างละ 3 ชั้น
3. ไม่สามารถกดไปชั้นอื่นได้พร้อมกัน
4. มี Output ดังนี้
  - 1) ไฟแสดงผลตำแหน่งของลิฟต์ทั้ง 3 ชั้น
  - 2) 7 segment แสดงผลเลขตำแหน่งของลิฟต์
  - 3) ประตูเปิดและปิด
  - 4) สถานะการทำงานทำงานของลิฟต์ (Busy, Useable)
  - 5) การเคลื่อนที่ของลิฟต์ (Up, Down, Ready)

#### ข้อสรุปผลเมื่อได้ทดลองทำลงบอร์ด FPGA

ลิฟต์มีการเคลื่อนที่ทันทีเมื่อทำการเปิดใช้งาน โดยเริ่มตั้งแต่ชั้นที่ 1 ไปถึงชั้นที่ 3 ตั้งแต่เริ่มแรก หลังจากนั้นจะสามารถกดเลือกชั้นได้ปกติ แต่วงจรไม่สามารถ stack ค่าที่เรากดลิฟต์ไปหลายๆชั้นได้ในเวลาใกล้เคียงกัน เช่น เคลื่อนที่จากชั้นที่ 1 ไป 2 ในระหว่างการเคลื่อนที่ เมื่อเรากดไปชั้นที่ 3 ลิฟต์จะยังไม่สามารถเคลื่อนที่ไปชั้นที่ 3 ได้ จนกว่าลิฟต์จะเคลื่อนที่มาชั้นที่ 2 จากชั้นที่ 1 และไฟจะแสดงผลตำแหน่งของลิฟต์ทั้งชั้น 2 และ 3 พร้อมกัน กล่าวคือ จะต้องรอให้ลิฟต์ทำคำสั่งอันแรกเสร็จก่อนจึงจะสามารถทำสั่งถัดไปได้

### 3.2.2 ลิฟต์เวอร์ชัน 2 (4-Floor-Elevator)



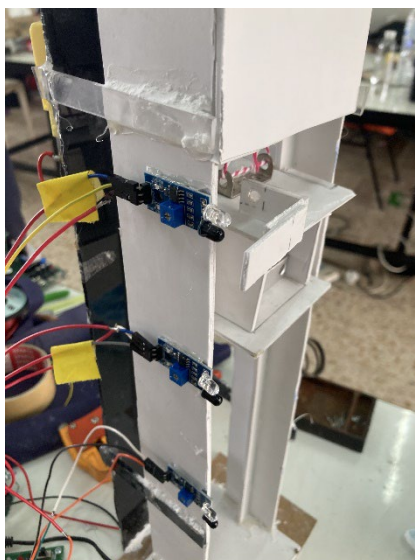
#### คุณสมบัติ

1. ลิฟต์สามารถเคลื่อนที่ได้ทั้งหมด 4 ชั้น
2. สามารถใช้ FPGA 2 บอร์ดในการควบคุมได้
3. มีปุ่มกด 2 ชุด อย่างละ 4 ชั้น
4. สามารถกดพร้อมกันได้มากกว่า 1 ชั้น (Stack)
5. มี Output หลักๆ ดังนี้
  - 1) ไฟแสดงผลตำแหน่งของลิฟต์ทั้ง 4 ชั้น
  - 2) ประตูเปิดและปิด
  - 3) มอเตอร์ขึ้นและลง
  - 4) ไฟแสดงผลสถานะการทำงานแต่ละชั้นตอน เช่น ทิศทางการเคลื่อนที่ของลิฟต์ (DIRECTION) สถานะของประตู (CMD OPEN) สถานะการเรียกใช้งานลิฟต์ (CALL ACTIVE) เป็นต้น

#### ข้อสรุปผลเมื่อได้ทดลองทำลงบอร์ด FPGA

เนื่องจาก clock ที่ใช้สำหรับโมดูลแต่ละตัวนั้นมีค่าที่ต่างกัน ส่งผลต่อการหมุนของ DC Motor ทำให้ระยะการเคลื่อนที่ของลิฟต์ในแต่ละชั้นมีความคลาดเคลื่อนจากที่ควรจะเป็น คือ การเคลื่อนที่ของลิฟต์ระหว่างชั้นที่อยู่ใกล้เคียงกัน จะใช้เวลาน้อยกว่ามากเมื่อเทียบกับสัดส่วนเวลาการเคลื่อนที่ของลิฟต์จากชั้นแรกไปชั้นสุดท้าย แต่สิ่งที่พัฒนามาจากรูปแบบแรกคือลิฟต์สามารถกดปุ่มได้มากกว่า 1 ชั้น พร้อมกัน จากนั้นจึงได้นำวงจรนี้ไปต่อยอดเป็นโมเดล

### 3.3.3 ลิฟต์เวอร์ชัน 3 (4-Floor-Elevator) Clock And Sensor



ปัญหาหลักของลิฟต์เวอร์ชัน 2 คือเรื่องของ Clock ที่ไม่สามารถทำให้ระยะการเคลื่อนที่ของลิฟต์คงที่ได้ ผู้จัดทำจึงได้ทำการพัฒนาเป็นลิฟต์เวอร์ชัน 3 โดยการนำ IR Sensor มาเพื่อเช็คตำแหน่งของลิฟต์ โดยต่อเข้ากับชุด Input ของ APPROACH SWITCHES และใช้ค่า Clock 1 Hz ทำให้การเคลื่อนที่ของลิฟต์มีความคงที่ และสามารถหยุดตามชั้นที่ต้องการได้อย่างแม่นยำ อีกทั้งยังสามารถเริ่มการทำงานใหม่ได้ทุกชั้นอีกด้วย

### 3.3 กระบวนการทดสอบ

#### 3.3.1 ทดสอบการทำงานของวงจร

ลองทดสอบรันวงจรในโปรแกรม Logisim Evolution ผลปรากฏว่าสามารถทำลิฟต์ขึ้นลงตามที่เรากดในแต่ละชั้นได้ และสามารถแสดงผลตำแหน่งของลิฟต์ได้ถูกต้อง ต่อมาจึงได้ทำโมดูลเป็น VHDL นำมาต่อวงจรใหม่ในโปรแกรม Xilinx จากนั้นจึงอัปโหลดไฟล์ลงบอร์ด FPGA

#### 3.3.2 ทดสอบการสื่อสารระหว่าง FPGA 2 บอร์ด

เริ่มจากการหาวิธีการสื่อสารกันระหว่าง fpga จำนวน 2 อัน โดยใช้ K1 connector เป็นตัวหลักในการสื่อสาร ซึ่งต้องรู้ก่อนว่าตัว K1 connector มีจำนวน 16 ขา โดยที่ ขาเลขคู่คือขา ground ส่วนขาเลขคี่คือขาที่เอาไว้เชื่อมกับ fpga อีกตัวหนึ่ง

#### 3.3.3 ทดสอบการใช้ pin เป็น output

- LED

ให้ fpga ตัวแรกกำหนด input เป็นสวิตช์ และกำหนดให้ output เป็น K1 connector ส่วน fpga ตัวที่ 2 กำหนด input เป็น K1 connector ซึ่งเชื่อมกับ K1 connector ของ fpga ตัวแรก แล้วกำหนดให้ output เป็นไฟ LED หรือ Buzzer ผลที่ได้ก็คือ ไฟ LED ติดแสงสว่าง

- DC motor

การทดสอบตัว DC motor จะเปลี่ยนตัว output ของ fpga ตัวที่ 2 จากไฟ LED หรือ Buzzer เป็น DC motor ซึ่งผลที่ได้คือ DC motor มีทิศทางการหมุนที่ถูกต้องทำให้ลิฟต์สามารถเคลื่อนที่ในแนวตั้งได้ตามปกติ

#### 3.3.4 ทดลองเปลี่ยนค่า Clock

ในการพัฒนาชิ้นงานรูปแบบที่ 2 ได้มีการใช้ Clock เป็นตัวหลักในการทำให้ลิฟต์เคลื่อนที่ การทดสอบตัว clock จะเปลี่ยนตัว input ของ fpga ตัวแรกจากสวิตช์เป็น clock (P123) ซึ่งผลปรากฏว่า มีความไม่คงที่ของเวลาในการเคลื่อนที่ลิฟต์เนื่องจาก clock ที่เข้าโมดูลแต่ละตัวมีค่าที่ไม่เท่ากัน จึงได้ทำการทดลองหาค่าของ clock ที่มีความคงที่โดยใช้ mod 1250 จะมีความเร็วที่ 16 Hz

### 3.3.5 ทดลองต่อเซนเซอร์ติดแต่ละชั้น

ในการพัฒนาชิ้นงานรูปแบบที่ 3 ได้มีการนำ IR Sensor มาเพื่อตรวจสอบตำแหน่งลิฟต์แต่ละชั้น ร่วมกับการใช้ค่า Clock 1 Hz ในการควบคุมการเคลื่อนที่ของลิฟต์ ผลปรากฏว่าสามารถทำให้ลิฟต์หยุดตามตำแหน่งที่เราต้องการได้อย่างแม่นยำ

### 3.4 ปัญหาและแนวทางการแก้ไข

#### 3.4.1 ปัญหาด้าน Software

1. จากลิฟต์เวอร์ชันที่ 2 Decode จาก Module POSITION SIM ทำให้ค่า clock มีปัญหา ส่งผลต่อการเคลื่อนที่ของลิฟต์ที่ไม่มีความคงที่

**แนวทางแก้ไข** : นำ POSITION SIM ออก และใช้ IR Sensor เข้ามาพัฒนาต่อแทน

2. Module จากไฟล์ต้นแบบที่ใช้โปรแกรม Logisim ไม่สามารถสร้างเป็น circuit ขึ้นเองได้ เนื่องจากโปรแกรมที่ต้นแบบใช้เป็นรุ่นที่เก่า ทำให้ Module บางตัวไม่มีความสมจริง

**แนวทางแก้ไข** : ทำ Module เป็นไฟล์ VHDL

3. ในโปรแกรม Xilinx ไม่สามารถต่อ Module Mod หลายๆอันในไฟล์หลักได้

**แนวทางแก้ไข** : ต่อเป็นไฟล์แยกออกมาแล้วทำเป็น Module

4. ใช้ NAND ในการต่อ SR Latch ทำให้ค่า Output มีค่าไม่ปกติอย่างที่ควรจะเป็น

**แนวทางแก้ไข** : ใช้ NOR Latch แทน

#### 3.4.2 ปัญหาด้าน Hardware

1. ต่อ Input เป็นขาลอยใน K1 connector ทำให้ค่า Output มีปัญหา

**แนวทางแก้ไข** : ตรวจสอบ Input ทุกสายว่ามีการต่อครบแล้วหรือไม่

2. จากลิฟต์เวอร์ชัน 2 ที่ใช้ clock เป็นหลักในการเคลื่อนที่ของลิฟต์ ทำให้มีปัญหาตอนทำโมเดลคือลิฟต์ไม่สามารถหยุดให้ตรงชั้นที่กำหนดไว้ได้

**แนวทางแก้ไข** : เปลี่ยนมาใช้ร่วมกับ IR Sensor ในการควบคุมการเคลื่อนที่ของลิฟต์

3. IR Sensor ทำงานแบบ Active Low

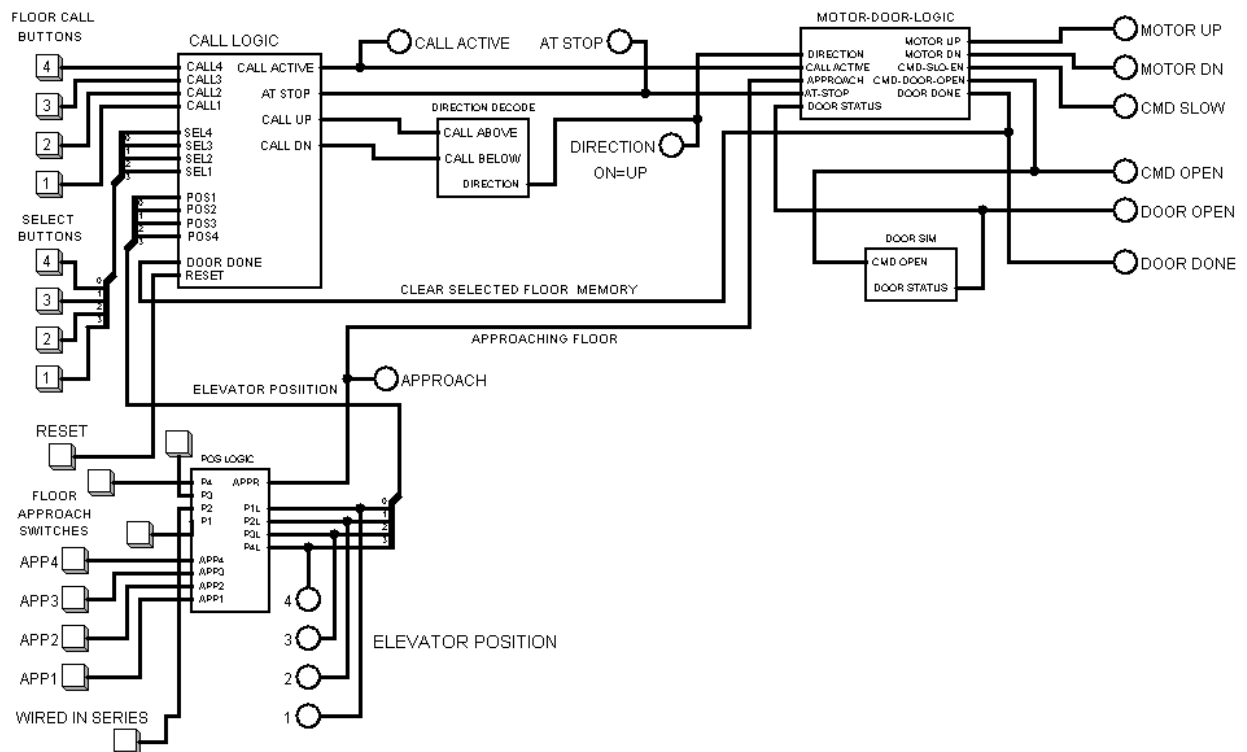
**แนวทางแก้ไข** : ต่อ NOT Gate

4. โครงสร้างที่ได้ออกแบบไว้ครั้งแรกไม่แข็งแรง เมื่อมอเตอร์ทำงานทำให้โมเดลล้มได้

**แนวทางแก้ไข** : สร้างฐานให้ใหญ่และมั่นคงกว่าเดิม

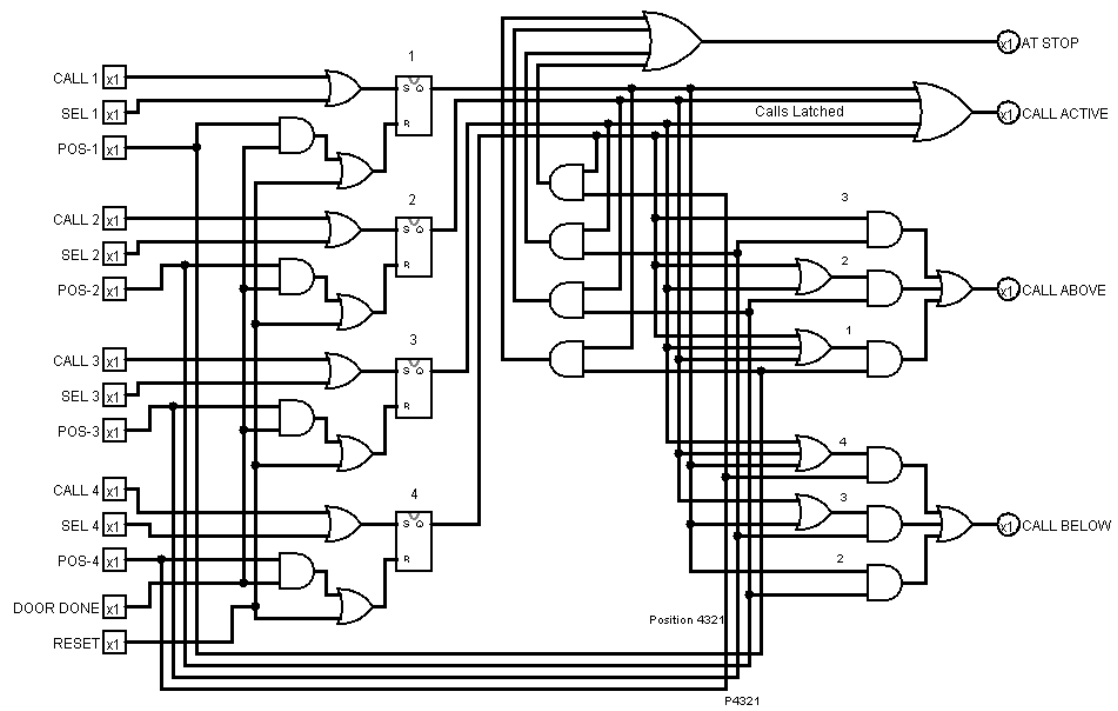
## Top-Down Design

### 1<sup>st</sup> layer Elevator

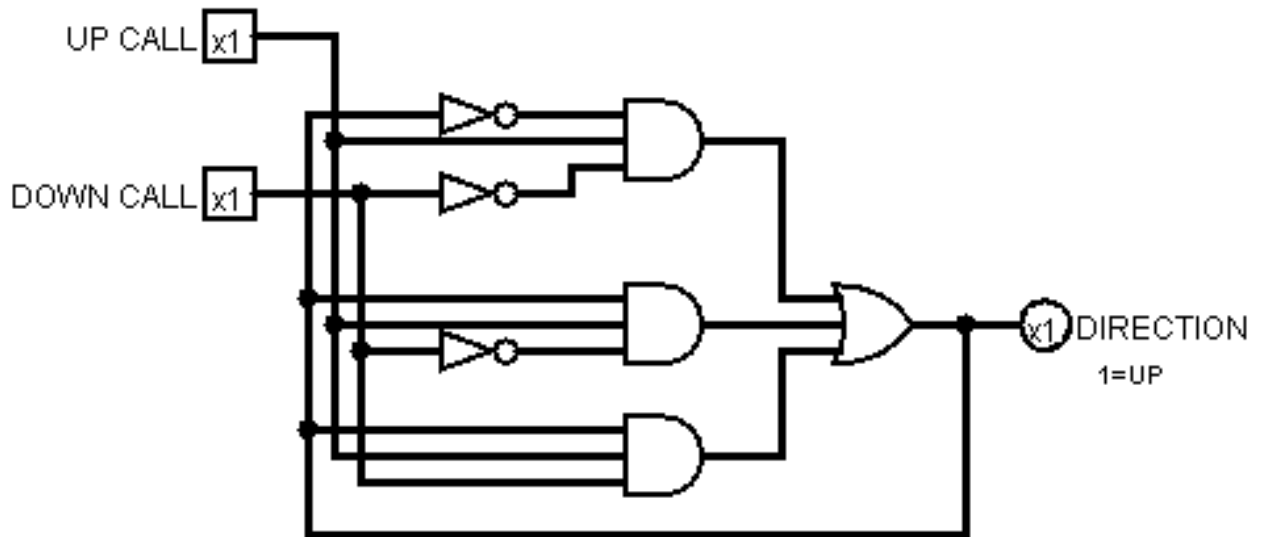


### 2<sup>nd</sup> layer call\_logic

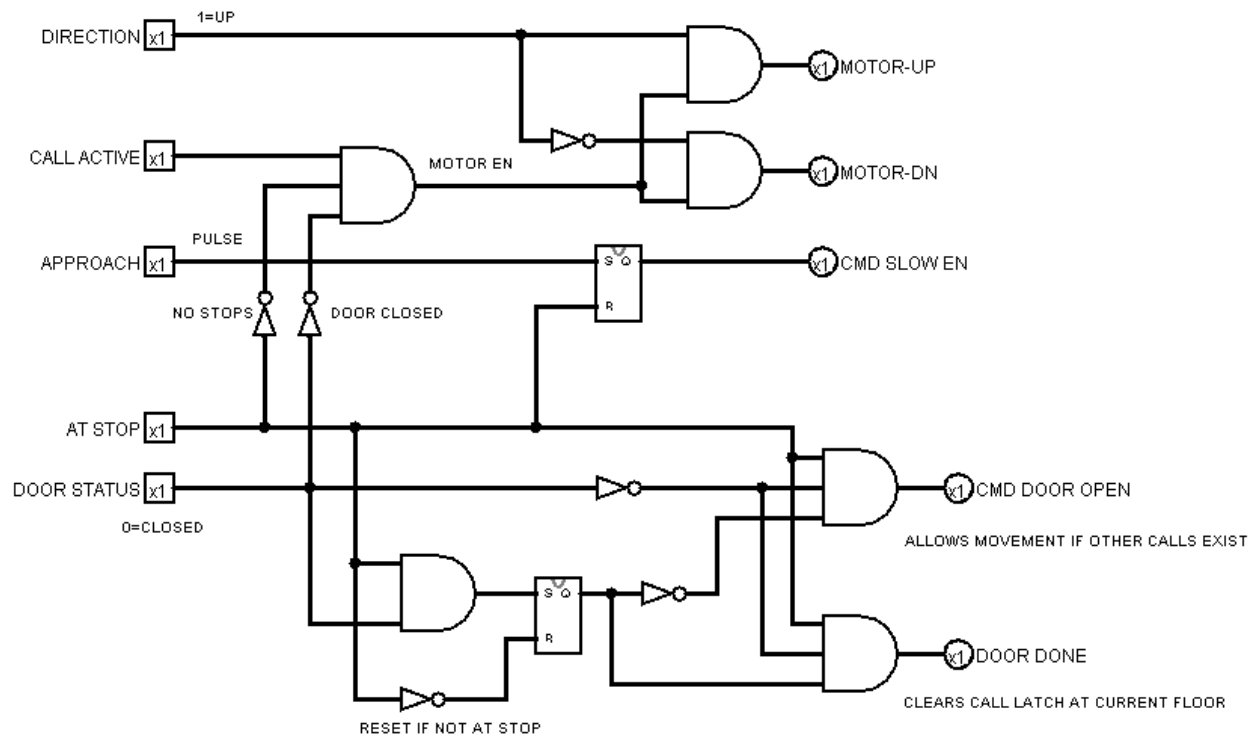


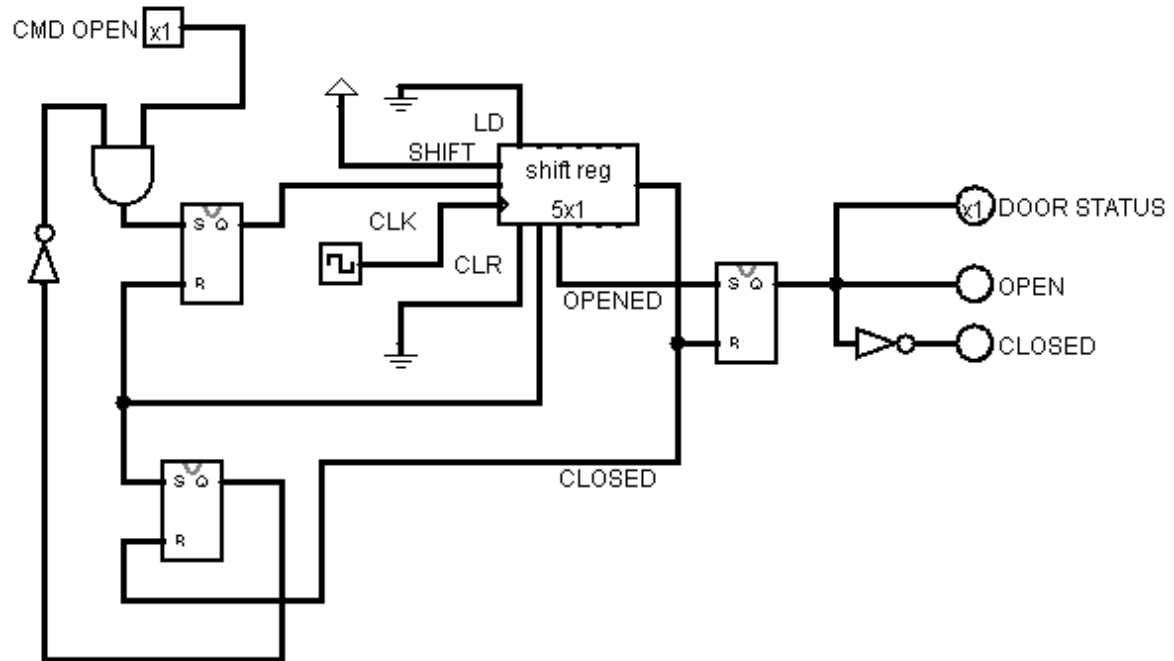


2<sup>nd</sup> layer Direction

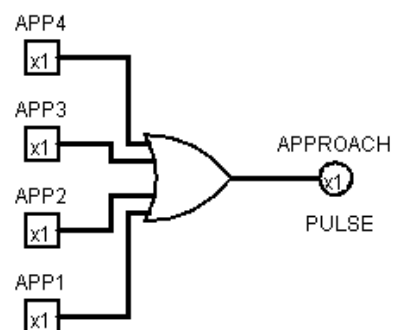
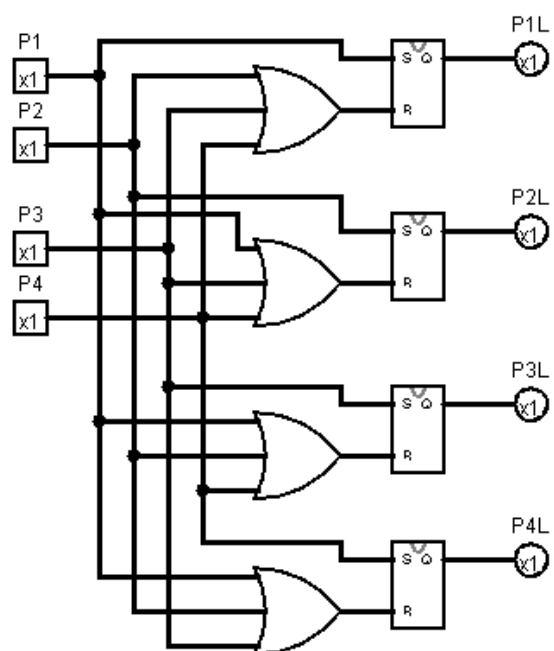


## 2<sup>nd</sup> layer MOTOR & DOOR LOGIC



2<sup>nd</sup> layer doorsim

## 2<sup>nd</sup> layer Lift\_Position



## 5. ลิ้งค์วิดีโอทัศน์แนะนำชิ้นงาน

<https://www.youtube.com/watch?v=wKbVd3CGW-o>

## อ้างอิง

- acmahalkasi. (2016). 3 Floor Elevator Logisim. Retrieved 20 November 2022. From <https://github.com/acmahalkasi/3-Floor-Elevator-Logisim>
- SuburbanDon. (2022). Logic-Controlled Elevator. Retrieved 24 November 2022. From [https://www.youtube.com/watch?v=9tDd7\\_UC89M](https://www.youtube.com/watch?v=9tDd7_UC89M)
- Allaboutcircuits. The S-R Latch. Retrieved 10 December 2022. From <https://www.allaboutcircuits.com/textbook/digital/chpt-10/s-r-latch/>
- 101computing.net. (2021). Binary Shifters using Logic Gates. Retrieved 10 December 2022. From <https://www.101computing.net/binary-shifters-using-logic-gates/>
- Tutorialspoint. Decoder. Retrieved 10 December 2022. From [https://www.tutorialspoint.com/digital\\_circuits/digital\\_circuits\\_decoders.htm](https://www.tutorialspoint.com/digital_circuits/digital_circuits_decoders.htm)