EEE 120
Capstone Design Project

Name:____Jin Woo Lee_____

Instructor:____Steve Millman_____

Class Time:__Tues/Thur 3:00~4:15___

Date:__11/29/2025_____

Task C-1: Planning the Synchronous Sequential Machines

(5 pts) Interview at least 3 stakeholders, but 3 is preferred. Ask questions regarding the form, function, and features needed by potential customers for this design. Make sure to capture what the customer prefers from this type of solution, as well as what environment the customer plans to use this design. Summarize your findings here and document the names of who you interviewed.

#1. Mrs. Chung

interviewed my mother as a typical non-technical tram user. She said the most important factors were safety, cost-effective, and clarity. She prefers a system that opens the doors only when the tram is fully stopped and shows the current floor clearly when opened, so there is no confusion. She emphasized that airport/shuttle systems need stability and simple, predictable behavior.

#2. Patrick (Computer engineering/College friend)

Chris focused on technical correctness. He said a tram controller must have deterministic state transitions, especially during direction changes. He liked the idea that the movement happens in discrete 1-clock steps, because this matches real synchronous behavior. He advised me to watch out for redundant states when creating state diagram.

#3. Jeremy Upsal (College professor)

Jeremy emphasized that, above all, a tram must show the exact final floor number at the moment the doors open. He explained that while movement can internally have delays or intermediate steps, the system must guarantee that when open is triggered, the displayed floor is accurate and stable.

(5 pts) Please include a comment on why your automation adds value from multiple perspectives (technological, societal, financial, environmental, etc.). (*What value does this add? What is the type of customer for whom this is designed? Where is this most needed? What couldn't you do before?*)

This automation adds values by making the overall control architecture structurally simple and highly visual. By separating the finite state machine and the ALU, the design becomes easy to understand and modify, even for someone who did not originally design it. By exploiting symmetry in the state diagram and removing redundant behavior, the controller needs only a minimal number of states and gates, which makes the hardware more economical and straightforward to implement. The compact and symmetric structure also makes the design easier to maintain or extend, providing long-term engineering and educational value for anyone who studies or reuses the system in the future.

(5 pts) It is allowable to continue to ask questions of stakeholders throughout the design process (and is preferred of a conscientious engineer). This can be done as you are designing, before you are designing if you need input and clarifications, or after you are done designing if you want feedback on improvements. Summarize any changes to your understanding or design based on the feedback you received during your initial interviews or continual interviews?

For the second design, Jeremey suggest if I reassign the binaries for states, and I reassigned two pairs of states that were not identical but shared the same transition arrows so that 1s became adjacent in K-map. This allowed me to form larger implicants and further simplify the next-state equations.

Task C-2: Document the Synchronous Sequential Machines

Design #1: (2 pts) What assumptions did you make in the design of this machine?

0. I assumed that the machine should not treat each physical gate as a sperate state. Instead, the machine keeps track of movement by storing only the previous direction and the current step motion (stop or direction). This means the states will encode directional progress rather than specific gate locations.
1. I assumed that the input 00 as stop, and it could be **reset** or **open** for the following state.
2. I assumed that the input 11 is only meaningful in movement states. (put don't car X if not exists.)
3. I assumed that the input 00 is allowed for any states.
4. If I get 00 at Open states, it goes to RESET state after clock, which means just close the door and stay for that clock.
5. I assumed the input 01/10 can't follow after 01/10. (10 ↛10 or 01↛01)

(3 pts) Create a state definition table here that describes in plain English what each state in your machine means and what binary values you have assigned to represent each state, inputs, and outputs.

STATE DEFINITION TABLE

STATE	DEFINITION	NAME	BINART
S_0	Reset/stop/closed	RST	000
S_1	Down/Open/stop	D. open	001
S_5	Up / open/stop	U. open	101
S_2	Down 1 / closed	D1	010
S_3	Down 2/closed	D2	011
S_4	Up 1/closed	U1	110
S_7	Up 2/closed	U 2	111

When $Q_1 = 0$, stop
→ $D_0 = Q_1$
(No increment when $D_0 = 0$)

$Q_2 = 0$, invert

invert = $\overline{Q_2}$

closed = $\overline{G}R$, Open = $G\overline{R}$

S_0 – Reset/Stop/ Closed

The tram will start from here; it gets back to this state only if it gets input-00 when **open** state.

Therefore, this state defined as Stop/Closed because it stays at same gate for a clock period.

S_1 -Down/Stop/Open, S_5-Up/Stop/Open :001/101

These states will stop the tram and open the door.

Binary shows when $Q_2 = 0$, tram opens the door.

These are the states that receive transitions on input-00 from all other states, except for S_0.

When splitting the binary in half, I wanted these two **stop** states to appear in corresponding positions across the boundary. Based on that structure, I placed all DOWN states in the upper half (1,2,3), and all UP states in the lower half (5,6,7) so that the overall encoding remained symmetric as easy to simplify.

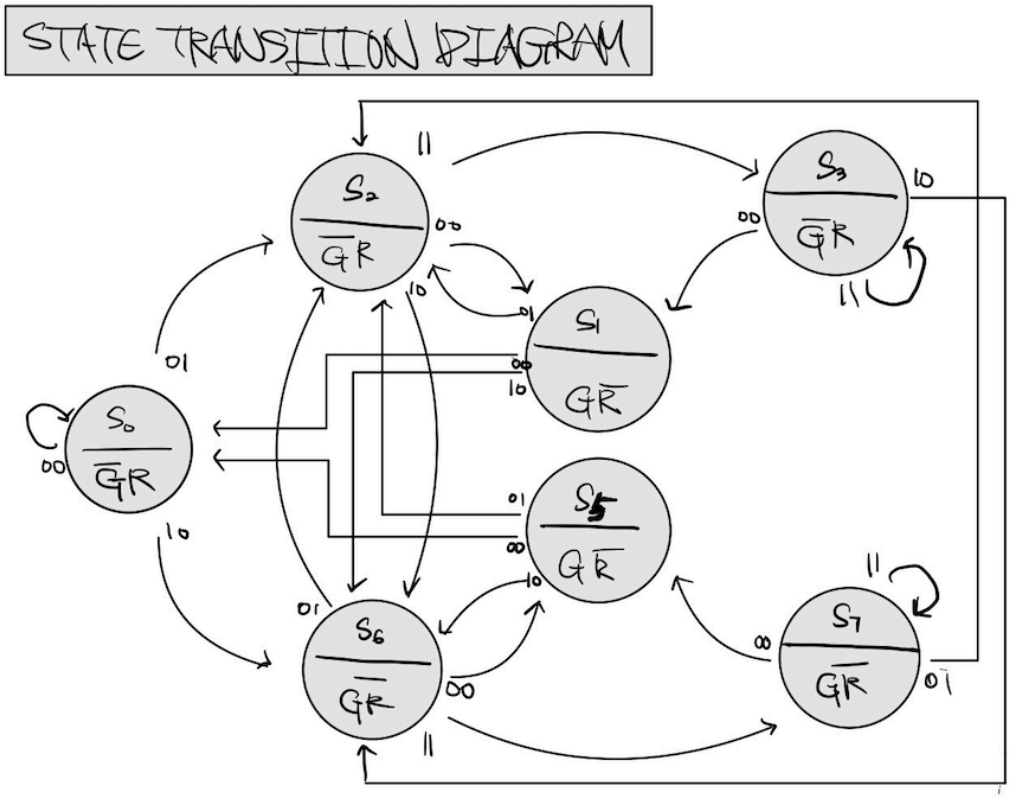
Binary shows when $Q_2 = 0$, tram opens the door.

S_2&S_3 Down/**S_6&S_7 UP**

Binary numbers 001&011/110&111 show when $Q_2 = 0$ tram moves to DOWN direction,

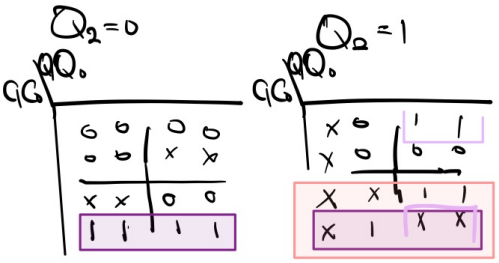
When $Q_2 = 1$, tram moves to UP direction.

(12 pts) Show your state diagrams, state transition tables and your circuit planning work (Karnaugh maps/equations/MUX/DEC/etc.) used in your design process. (You can do this by hand if you wish, do **not** show the full circuit schematic here.)

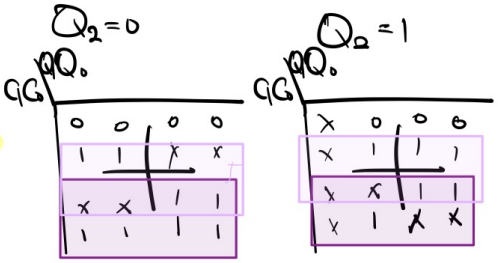


STATE TRANSITION TABLE & KARNAUGH MAP

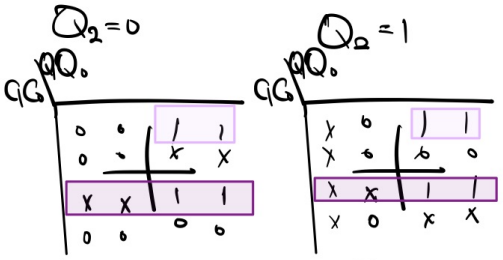
$Q_2 Q_1 Q_0$	$C_1 C_0$	$Q_2^+ Q_1^+ Q_0^+$	$G R$
0 0 0	0 0	0 0 0	0 1
0 0 0	0 1	0 1 0	0 1
0 0 0	1 0	1 1 0	0 1
0 0 0	1 1	X X X	X X
0 0 1	0 0	0 0 1	1 0
0 0 1	0 1	0 1 0	1 0
0 0 1	1 0	1 1 0	1 0
0 0 1	1 1	X X X	X X
0 1 0	0 0	0 0 1	1 0
0 1 0	0 1	0 1 0	1 0
0 1 0	1 0	1 1 0	1 0
0 1 0	1 1	X X X	X X
0 1 1	0 0	0 0 1	1 0
0 1 1	0 1	0 1 0	1 0
0 1 1	1 0	1 1 0	1 0
0 1 1	1 1	X X X	X X
1 0 0	X X	X X X	X X
1 0 1	0 0	0 0 1	1 0
1 0 1	0 1	0 1 0	1 0
1 0 1	1 0	1 1 0	1 0
1 0 1	1 1	X X X	X X
1 1 0	0 0	0 0 1	1 0
1 1 0	0 1	0 1 0	1 0
1 1 0	1 0	1 1 0	1 0
1 1 0	1 1	X X X	X X
1 1 1	0 0	1 0 1	0 1
1 1 1	0 1	0 1 0	0 1
1 1 1	1 0	X X X	X X
1 1 1	1 1	1 1 1	0 1



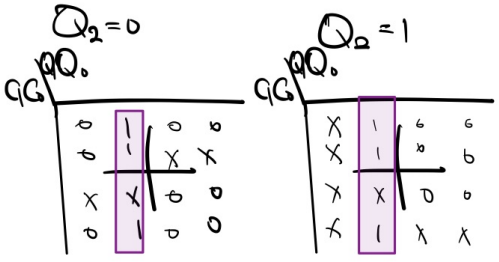
$$Q_2^+ = C\bar{C} + Q_2 Q_1 \bar{C}_0 + Q_2 C_1$$



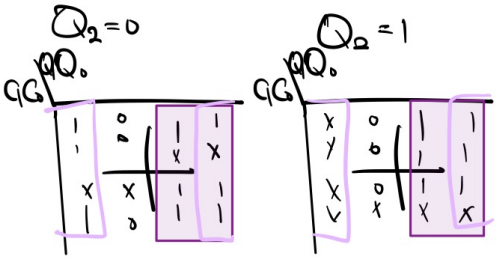
$$Q_1^+ = C_1 + C_0$$



$$Q_0^+ = C C_0 + Q_1 \bar{C} \bar{C}_0$$



$$G = \bar{Q}_1 Q_0$$



$$R = Q_1 + \bar{Q}_0 \qquad (\because G = \bar{R})$$

DESIGN EQUATIONS

$$Q_2^+ = C_1\bar{C}_0 + Q_2Q_1\bar{C}_0 + Q_0C_1$$

$$Q_1^+ = C_1 + C_0$$

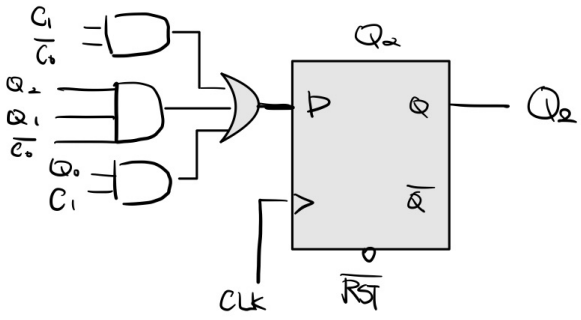
$$Q_0^+ = \bar{Q}_0 + Q_1\bar{C}_1\bar{C}_0$$

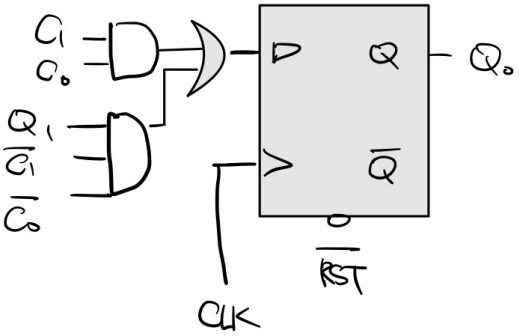
$$G = \bar{Q}_1Q_0$$

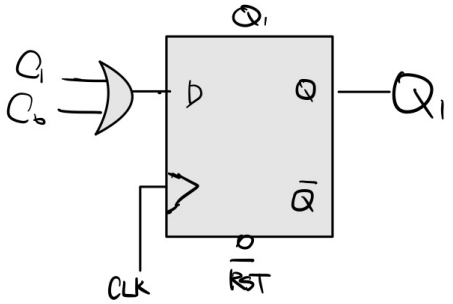
$$R = Q_1 + \bar{Q}_0$$

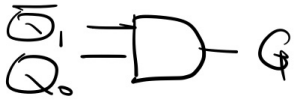
Invert $= \bar{Q}_2$ $\quad Q_0 = Q_1$

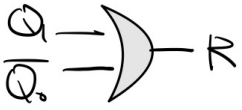
LOGIC GATE DESIGN











$\bar{Q}_2$ → Invert

Q_1 → Q_0

(3 pts) List your final design equations and required logic gates (including types of Flip Flops) needed to complete this circuit.

$Q_2^+ = C_1\overline{C_0} + Q_2Q_1\overline{C_0} + Q_0C_1$

$Q_1^+ = C_1 + C_0$

$Q_0^+ = C_1C_0 + Q_1\overline{C_1}\,\overline{C_0}$

$G = \overline{Q_1}Q_0$

$R = Q_1 + \overline{Q_0}$

$Invert = \overline{Q_2}\,, \qquad a_0 = Q_1$

(As we can see in state definition table, when connecting ALU, Invert should be connected to $\overline{Q_2}$ and a_0 will be connected to Q_1.)

3x D Flip-Flops, 6x AND gate, 4x OR gate, 2x NOT gate (for C1, C0)

Design #2: (2 pts) What assumptions did you make in the design of this machine?

0. I assumed that the machine should not treat each physical gate as a sperate state. Instead, the machine keeps track of movement by storing only the previous direction and the current step motion (stop or direction). This means the states will encode directional progress rather than specific gate locations.
1. I assumed that the input 00 as stop, and it could be **reset** or **open** for the following state.
2. I assumed that the input 11 is only meaningful in movement states. (put don't car X if not exists.)
3. I assumed that the input 00 is allowed for any states.
4. If I get 00 at Open states, it goes to RESET state after clock, which means just close the door and stay for that clock.
5. I assumed the input 01 or 10 cannot follow another 01 or 10 (10 ↛10 or 01↛01). In other words, after choosing a direction, continued motion must use 11 for input.
6. Since S_1 *and* S_5 have same transition behavior, it turns out that one of them is redundant, so the OPEN state can be represented by a single state without separating it by previous direction.

(3 pts) Create a state definition table here that describes in plain English what each state in your machine means and what binary values you have assigned to represent each state.

STATE DEFINITION TABLE

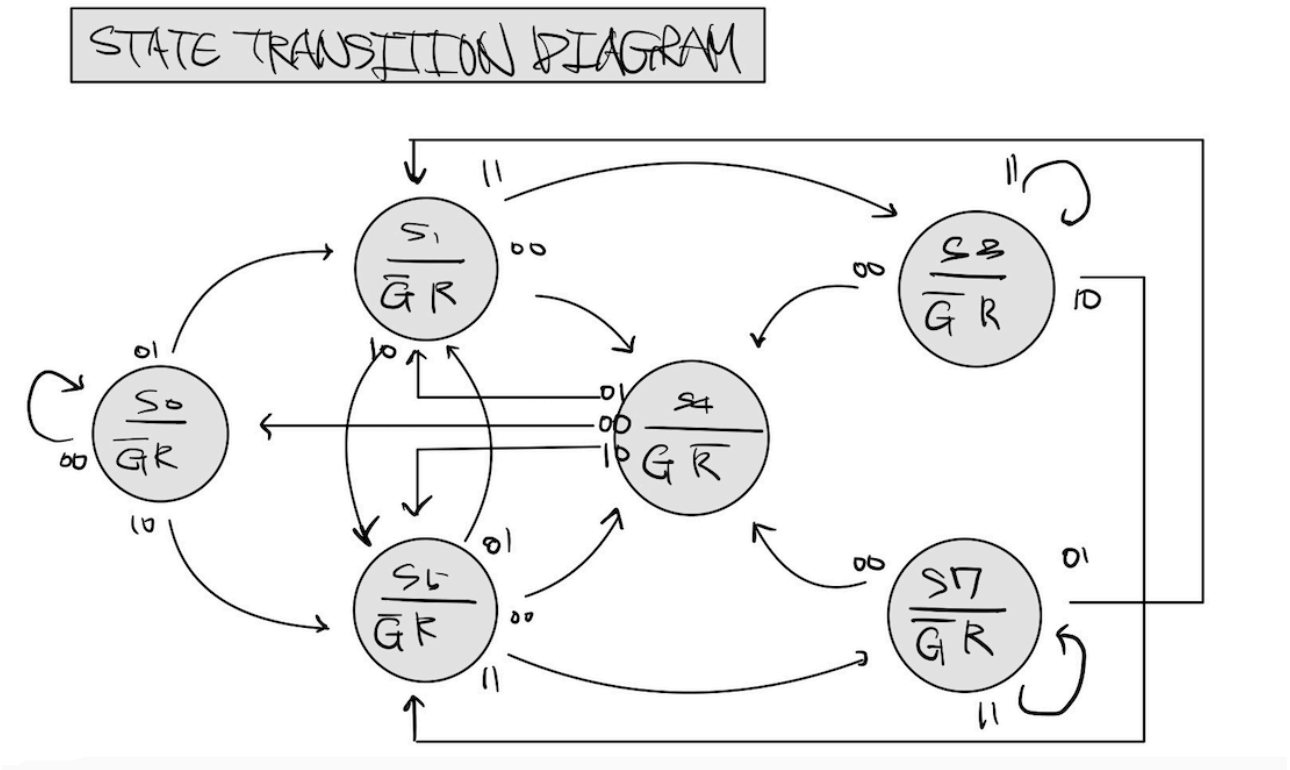
STATE	DEFINITION	NAME	BINARY
S_0	Rest/stop /closed	RST	0 0 0
S_4	stop/ open	Open	1 0 0
S_1	Down 1 /closed	D.1	0 0 1
S_3	Down 2/Closed	D.2	0 1 1
S_5	Up 1/closed	U.1	1 0 1
S_7	Up 2/closed	U.2	1 1 1

→ stop : $Q_0=0$ ⇒ $a_0=Q_0$

→ if $Q_0 \neq 0$, $Q_2 = 0$, invert = $\overline{Q_2}$

Open = $G\overline{R}$ closed = $\overline{GR}$

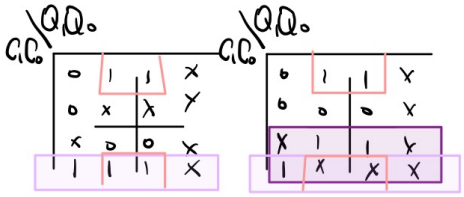
(12 pts) Show your state diagrams, state transition tables and your circuit planning work (Karnaugh maps/equations/MUX/DEC/etc.) used in your design process. (You can do this by hand if you wish, do **not** show the full circuit schematic here.)



STATE TRANSITION TABLE & KARNAUGH MAP

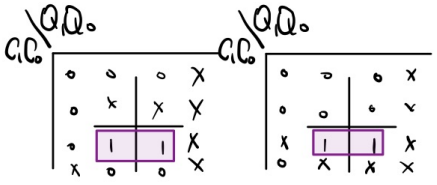
$Q_2 Q_1 Q_0$	$C_1 C_0$	$Q_2^+ Q_1^+ Q_0^+$	GR
0 0 0	0 0	0 0 0	0
0 0 0	0 1	0 0 1	0
0 0 0	1 0	0 1 0	0
0 0 0	1 1	X X X	X
0 0 1	0 0	1 0 0	0
0 0 1	0 1	X X X	X
0 0 1	1 0	1 0 1	0
0 0 1	1 1	0 1 1	0
0 1 0	X X	X X X	X
0 1 1	0 0	1 0 0	0
0 1 1	0 1	X X X	X
0 1 1	1 0	1 0 1	0
0 1 1	1 1	0 1 1	0
1 0 0	0 0	1 0 0	0
1 0 0	0 1	1 0 1	0
1 0 0	1 0	1 1 0	0
1 0 0	1 1	X X X	X
1 0 1	0 0	1 0 0	0
1 0 1	0 1	1 0 1	0
1 0 1	1 0	1 1 0	0
1 0 1	1 1	0 0 1	1
1 1 0	X X	X X X	X
1 1 1	0 0	1 0 0	0
1 1 1	0 1	1 0 1	0
1 1 1	1 0	1 1 0	0
1 1 1	1 1	0 0 0	1

$Q_2 = 0$ $Q_2 = 1$



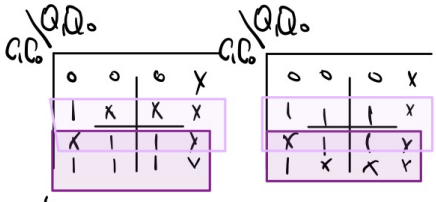
$Q_2^+ = Q_0\bar{C_0} + C_1\bar{C_0} + Q_2C_1$

$Q_2 = 0$ $Q_2 = 1$



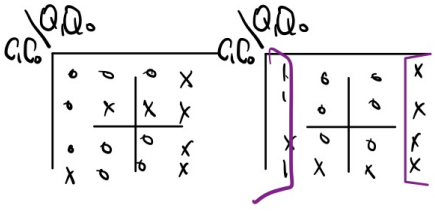
$Q_1^+ = Q_0C_1C_0$

$Q_2 = 0$ $Q_2 = 1$



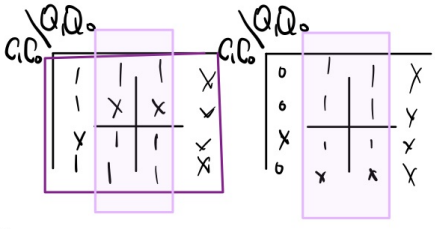
$Q_0^+ = C_1 + C_0$

$Q_2 = 0$ $Q_2 = 1$



$G = Q_2\bar{Q_0}$

$Q_2 = 0$ $Q_2 = 1$



$K = \bar{Q_2} + Q_0$ $(\because \bar{G} = K)$

DESIGN EQUATIONS

$$Q_2^+ = Q_0\bar{C}_0 + C_1\bar{C}_0 + Q_2C_1$$

$$Q_1^+ = Q_0C_1C_0$$

$$Q_0^+ = C_1 + C_0$$

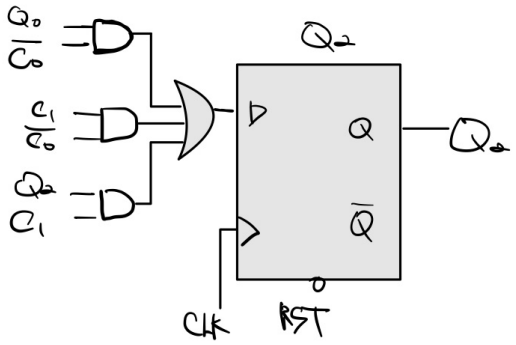
$$G = Q_2\bar{Q}_0$$

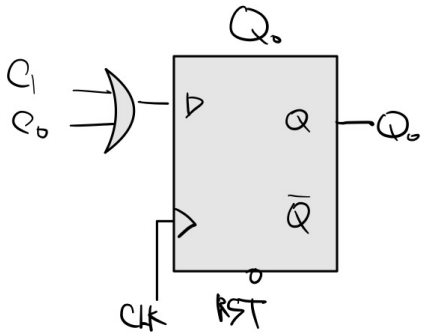
$$R = \bar{Q}_2 + Q_0$$

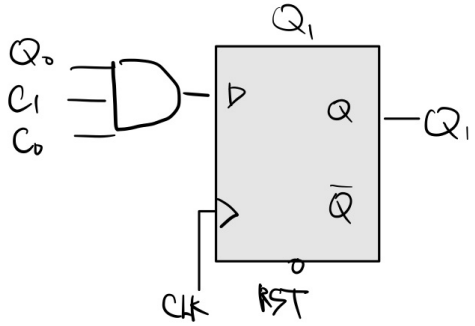
$$a_0 = Q_0$$

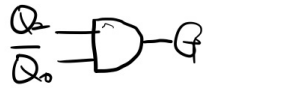
$$invert = \bar{Q}_2$$

LOGIC GATE DESIGN















(3 pts) List your final design equations and required logic gates (including types of Flip Flops) needed to complete this circuit.

$Q_2^+ = Q_0\overline{C_0} + C_1\overline{C_0} + Q_2C_1$

$Q_1^+ = Q_0C_1C_0$

$Q_0^+ = C_1 + C_0$

$G = \overline{Q_2}Q_0$

$R = Q_2 + \overline{Q_0}$

$Invert = \overline{Q_2}\,, \qquad a_0 = Q_1$

3x D Flip-Flops, 5x AND gate, 3x OR gate, 2x NOT gate (for C1, C0)

Task C-3: Determine Criteria and Weighting for Judging Your Designs

(5 pts) Using the guidelines in the laboratory FAQ's, list your 5 criteria and associated weights here used to help decide between the two design models (weights should add to 100%):

	Criteria	Weight
1.	Flip-Flop Count/Low Gate Count	25%
2.	Symmetry/Structural simplicity	25%
3.	Redundancy Removal/Efficiency	25%
4.	Predictable & Correctness of OPEN behavior	15%
5.	Scalability Across Any Number of Floors	10%

Task C-4: Apply the Criteria to Pick the Best Design

(2 pts) Describe how you applied each of the criteria and weighting system in the above task to pick the best design. How did you choose these criteria (customer interviews, engineering preference)?

These criteria were chosen based on a combination of interviews and engineering preference. The interviews emphasized the importance of predictable OPEN behavior and simple, understandable state organization, while engineering considerations highlighted especially flip-flop and gate count, efficiency and versatility of the design (scalability to any number of floors or adaptation to different system environments).

To summarize, Criteria #1 compares the cost of the components, Criteria #2 measures simplicity of the circuit or how visually clear the state organization is especially at transition table level, Criteria #3 shows the effort to reduce unnecessary components to improve the efficiency, Criteria #4 evaluates how well the design satisfies the needs identified from customer interviews. Lastly, Criteria #5 focuses on the forward perspective of the FSM, such as scalability and adaptability for broader applications.

(3 pts) Which design is better based on your criteria and weighting system and why? Please explain how the winning design scored in <u>each</u> category and why (the winning design does not need to score the highest in every category, but it does need to score higher overall when applying the criteria weights).

	Design #1		**Design #2**	
1	23%	3F-F/12 Logic gates	25%	3F-F/11 Logic gates
2	20%	Considered symmetrical state assignment with respect to half line, dividing into up-state-portion and down -state-portion, which makes it easier to understand.	25%	Considered symmetrical state assignment with respect to half line, dividing into up-state-portion and down-state-portion, which makes it easier to understand. Considered transition behavior to get a larger implicant using symmetry. (Used S1&S3 and S5&S7 while S1&S3 are not adjacent when grouping implicants.)
3	18%	More states to interpret, requires 2 separate OPEN states.	23%	Fewer states, unified OPEN state without violating symmetry, having even number of the state makes the machine stable.
4	15%	Correct open, clarity of state separation makes the machine easier to visualize and predict the behavior.	15%	Correct open, especially unified OPEN state structure results in cleaner interpretation of the machine and simpler reasoning about transitions.
5	10%	Direction-based encoding scales. Number of the floors does not affect FSM.	10%	Direction-based encoding scales. Number of the floors does not affect FSM.

-Two deduction from 3. Efficiency criteria, because we can reduce S0 if we are not going to use reset, but it would be necessary if the tram needs to stay at current position without opening the door.

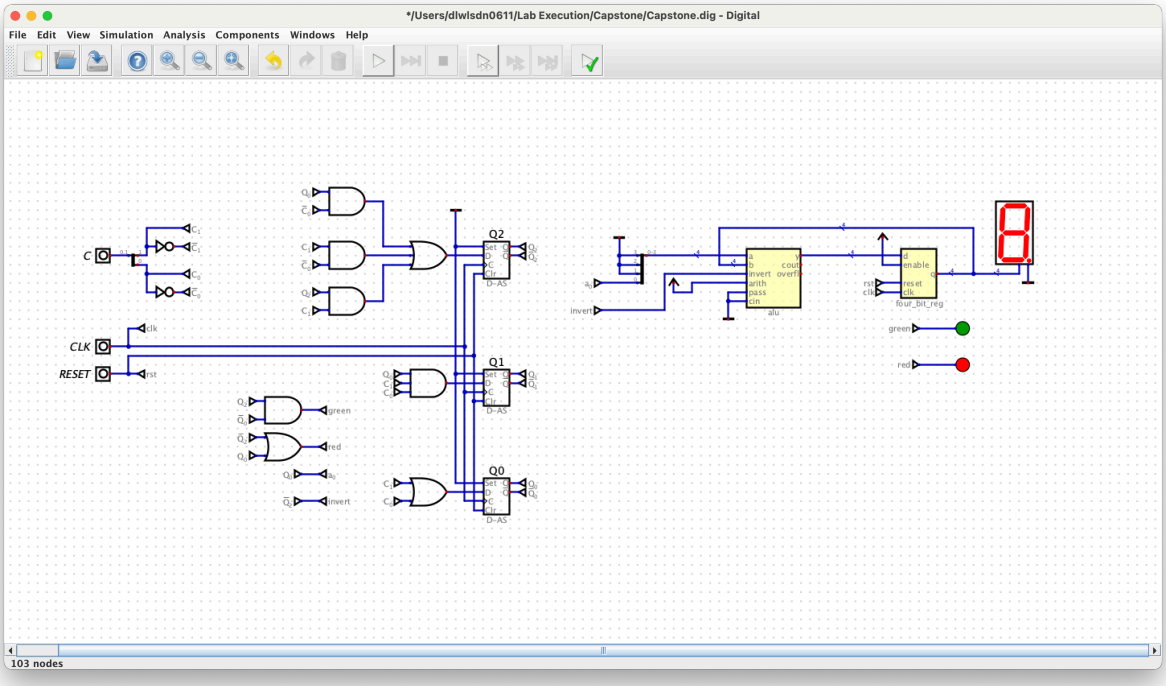
Design #1: 86%

Design #2: 98%

Design #2 WINS!

Task C-5: Build and Simulate Winning Design in Digital

(15 pts) Insert a copy of your chosen Digital Schematic here. Please make sure that you have outputs or tunnels connected to each flip flop so that you can easily monitor your states. Make sure that the logic and equations match the final equations presented in either Design 1 or Design 2.



Task C-6: Record a Video Demonstration of the Winning Design

(15 pts) Record a video demonstration showing all positions being visited and various combinations of the inputs in Digital. For every clock cycle, explain the inputs, what current state you are in, and point out any outputs that should be noted. Be sure to show what happens for different input combinations at each position. That is, your demonstration should be able to showcase all possible states and transitions required to get there. If you include any asynchronous inputs, make sure to show those features as well. Add a link to your video below. Be sure to include any required password.

Video link:
https://drive.google.com/file/d/1OA4C_222O3IzsYAYompk0Vr7fF9O7OAl/view?usp=sharing

Task C-7: Fill Out the Self-Assessment and Turn in Your Design

There are two items to submit. Turn in the zip file of your capstone project folder. Also turn in this template once it is filled out. There will be a deduction of 5 points if your template is only found inside the zip folder. The self-assessment is on the next page.

SELF-ASSESSMENT WORKSHEET

Put an 'X' in the table below indicating how strongly you agree or disagree that the outcomes of the assigned tasks were achieved. Use '5' to indicate that you 'strongly agree' and '1' to indicate that you 'strongly disagree'. Use 'NA', Not Applicable, when the tasks you performed did not elicit this outcome. Credit will be given for including this worksheet with your lab report. However, your **responses will not be graded**, they are for your instructor's information only.

Table 1: Self-Assessment of Outcomes for the Capstone Design Project Lab.

After completing the assigned tasks and report I am able to:	5	4	3	2	1	NA
Initiate a design process based on a value proposition and feedback from various stakeholders.	X					
Make assumptions to complete an incomplete functional specification.	X					
Use classical design techniques (i.e., state diagrams, state transition tables, and Karnaugh Maps), to design a synchronous sequential machine starting with a functional specification.	X					
Build, and debug a synchronous sequential machine.	X					
Develop reasonable engineering criteria for comparing different designs.	X					
Apply engineering criteria to select a 'best' design.	X					

Write below any suggestions you have for improving this laboratory exercise so that the stated learning outcomes are achieved.

CAPSTONE DESIGN PROJECT: LAB REPORT GRADE SHEET

Name: Jin Woo Lee

Grading Criteria	Max Points	Points lost
Template		
Neatness, Clarity, and Concision	5	
Description of Assigned Tasks, Work Performed & Outcomes Met		
Task C-1: Planning the Synchronous Sequential Machines	15	
Task C-2: Document the Synchronous Sequential Machines	40	
Task C-3: Determine Criteria and Weighting for Judging Your Designs	5	
Task C-4: Apply the Criteria to Pick the Best Design	5	
Task C-5: Build and Simulate Winning Design in Digital	15	
Task C-6: Record a Video Demonstration of the Winning Design	15	
Self-Assessment Worksheet (The content of the self-assessment worksheet will not be graded. Full credit is given for including the completed worksheet.)	(2 extra points)	
Lab Score	**Points Lost**	
	Late Lab	
	Lab Score	