Computer Aided Design
Assignment №1

Pooya Kabiri, Alireza Moradi, Ali M. Movahedian

Department of Computer Science

Iran University of Science and Technology

November 2020

0 Team

Alireza Moradi	96521479
Pooya Kabiri	96521434
AliMohammad Movahedian	96521497

1 Logic Analysis and Design: K-map

First we standardize the equation:

- $f_1 = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + \bar{x}y\bar{z}$
- $f_2 = (w + x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{w} + y + \bar{z})$
- $F = f_1.f_2 = wx\bar{y}(w + x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{w} + y + \bar{z}) + \bar{y}z(w + x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{w} + y + \bar{z}) + \bar{w}y\bar{z}(w + x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{w} + y + \bar{z}) + \bar{x}y\bar{z}(w + x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{w} + y + \bar{z})$

1.1

	w	x	y	z	F
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	1
4	0	0	1	1	0
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	0
11	1	0	1	0	1
12	1	0	1	1	0
13	1	1	0	0	1
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	0	1	0	1
$\bar{w}x$	0	1	0	0
wx	1	0	0	0
$w\bar{x}$	0	0	0	1

So F will be:

$$F = w\bar{y}\bar{z} + \bar{x}y\bar{z} + wx\bar{y}\bar{z}$$

1.2

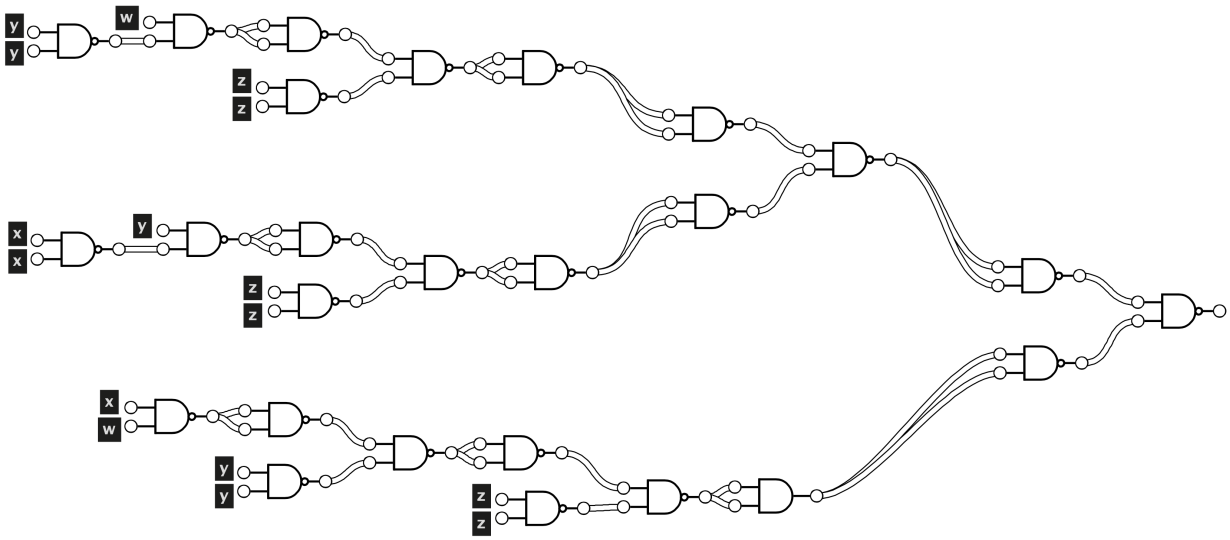

Figure 1: Logic Design, Part 1.2

2 Logic Analysis

2.1 4-Bit shifter

The circuit gets a 4-bit binary number as input.

The first set of multiplexer logic gates can shift the number zero or one unit to the left (based on first control input), and the second set of multiplexers can shift the number zero or two units (based on second control input).

When combining two sets of gates together, the whole circuit can shift the number any value between zero (00 control input) and three (11 control input).

for control inputs of 01 and 10, the circuit shifts the number two and one unit respectively.

2.2 Sequence Generator

Every iteration, the token with a value of 1 will be forked, creating two tokens with a value of 1 each. Then these two tokens are added up together creating a token of value 2, then this token is forked and one of the forked tokens is stored in snk actor, and the other one loops through the next iteration.

Based on procedure mentioned above, the snk actor record a value of $2^{iteration}$ at the end of each iteration, so the output of the model is a sequence of all powers of 2 : $2, 4, 8, 16, 32, ...$

3 Logic Design

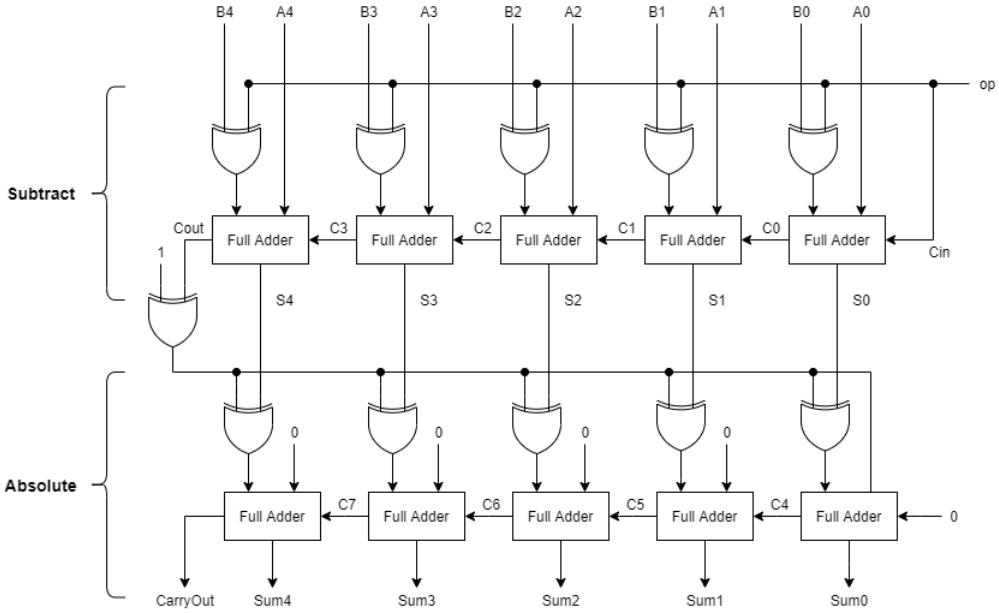

Figure 2: Logic Design, Part 3

4 Vending Machine

4.1

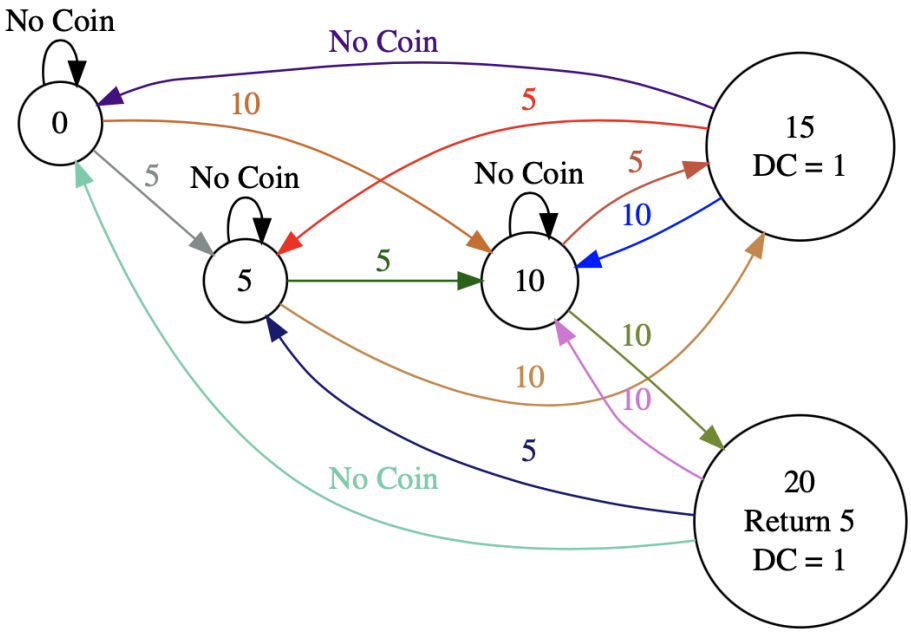

Figure 3: FSM of Improved Vending Machine

4.2 Implementation

4.2.1 One-Hot Coding

- `S`: 5 Rials
- `B`: 10 Rials
- O_1: Dispense
- O_2: Return5

	Inputs		State				
	S	B	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$
			00001	00010	00100	01000	10000
1	0	0	00001/00	00010/00	00100/00	00001/10	00001/11
2	0	1	00010/00	00100/00	01000/10	00010/00	00010/00
3	1	0	00100/00	01000/10	10000/11	00100/00	00100/00
4	1	1	x	x	x	x	x

$$Q_4 = S\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0$$

$$Q_3 = B\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0 + S\bar{Q}_4\bar{Q}_3\bar{Q}_2Q_1\bar{Q}_0$$

$$Q_2 = \bar{S}\bar{B}\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0 + B\bar{Q}_4\bar{Q}_3\bar{Q}_2Q_1\bar{Q}_0 + S\bar{Q}_4\bar{Q}_3\bar{Q}_2\bar{Q}_1Q_0 + S\bar{Q}_4Q_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + SQ_4\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0$$

$$Q_1 = \bar{S}\bar{B}\bar{Q}_4\bar{Q}_3\bar{Q}_2Q_1\bar{Q}_0 + B\bar{Q}_4\bar{Q}_3\bar{Q}_2\bar{Q}_1Q_0 + B\bar{Q}_4Q_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + BQ_4\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0$$

$$Q_0 = \bar{S}\bar{B}\bar{Q}_4\bar{Q}_3\bar{Q}_2\bar{Q}_1Q_0 + \bar{S}\bar{B}\bar{Q}_4Q_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + \bar{S}\bar{B}Q_4\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0$$

$$O_1 = \bar{S}\bar{B}\bar{Q}_4Q_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + \bar{S}\bar{B}Q_4\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + B\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0 + S\bar{Q}_4\bar{Q}_3\bar{Q}_2Q_1\bar{Q}_0 + S\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0$$

$$O_2 = \bar{S}\bar{B}Q_4\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0 + S\bar{Q}_4\bar{Q}_3Q_2\bar{Q}_1\bar{Q}_0$$

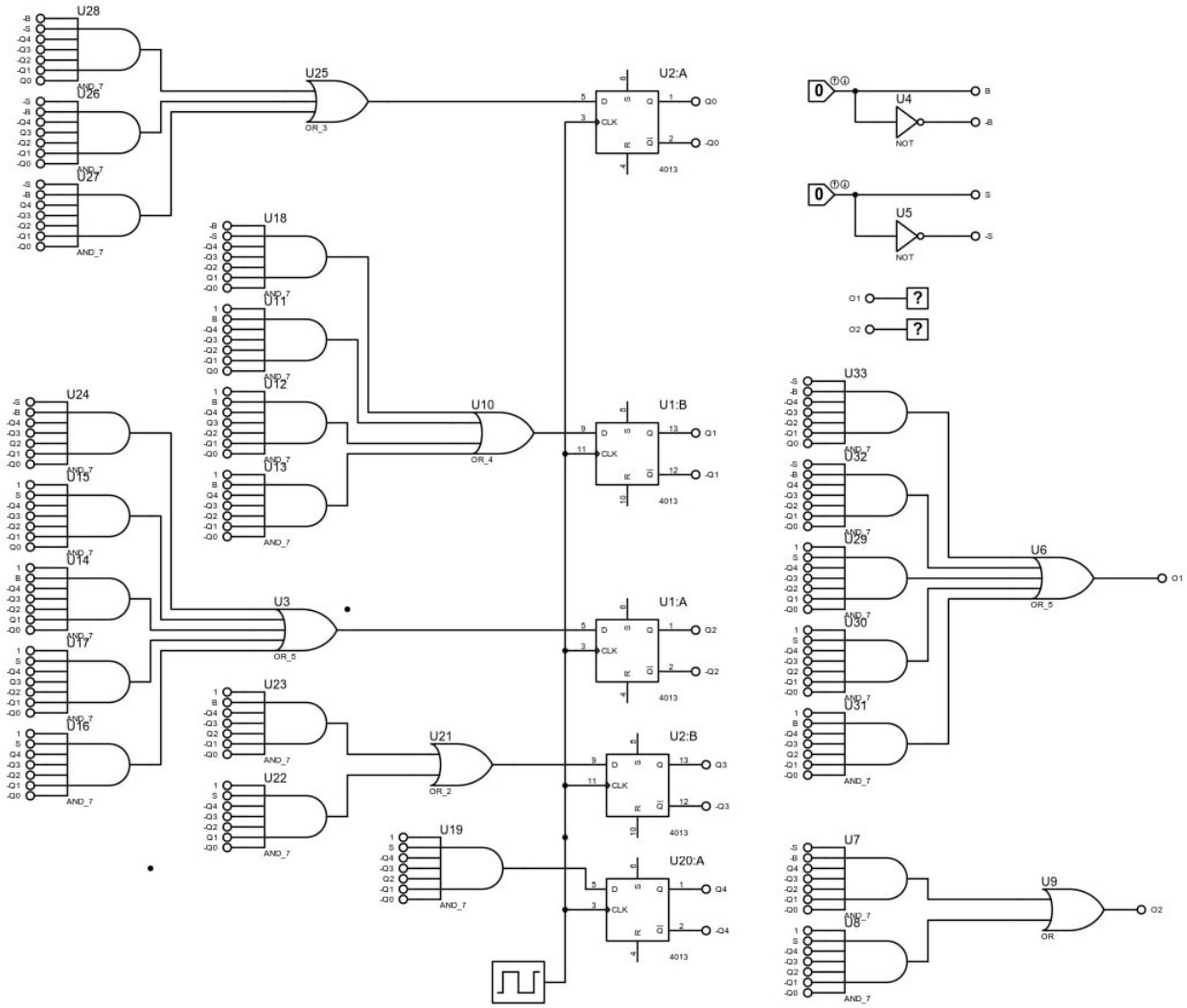

Figure 4: Logic Design, Part 4.2.1, One-Hot coding

So we need `5` flip-flops for One-Hot coding design.

4.2.2 One-Cold Coding

- `S`: 5 Rials
- `B`: 10 Rials
- O_1: Dispense
- O_2: Return5

	Inputs		State				
	S	B	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$	$Q_4Q_3Q_2Q_1Q_0/O_1O_2$
			11110	11101	11011	10111	01111
1	0	0	11110/00	11101/00	11011/00	11110/10	11110/11
2	0	1	11101/00	11011/00	10111/10	11101/00	11101/00
3	1	0	11011/00	10111/10	01111/11	11011/00	11011/00
4	1	1	x	x	x	x	x

$$Q_4 = \bar{Q}_4Q_3Q_2Q_1Q_0 + Q_4\bar{Q}_3Q_2Q_1Q_0 + \bar{S}Q_4Q_3\bar{Q}_2Q_1Q_0 + Q_4Q_3Q_2\bar{Q}_1Q_0 + Q_4Q_3Q_2Q_1\bar{Q}_0$$

$$Q_3 = \bar{Q}_4Q_3Q_2Q_1Q_0 + Q_4\bar{Q}_3Q_2Q_1Q_0 + \bar{B}Q_4Q_3\bar{Q}_2Q_1Q_0 + \bar{S}Q_4Q_3Q_2\bar{Q}_1Q_0 + Q_4Q_3Q_2Q_1\bar{Q}_0$$

$$\begin{gathered} Q_2 = \bar{S}\bar{Q}_4Q_3Q_2Q_1Q_0 + \bar{S}Q_4\bar{Q}_3Q_2Q_1Q_0 + \bar{B}Q_4Q_3Q_2\bar{Q}_1Q_0 \\ + \\ \bar{S}Q_4Q_3Q_2Q_1\bar{Q}_0 + BQ_4Q_3\bar{Q}_2Q_1Q_0 + SQ_4Q_3\bar{Q}_2Q_1Q_0 \end{gathered}$$

$$\begin{gathered} Q_1 = \bar{B}\bar{Q}_4Q_3Q_2Q_1Q_0 + \bar{B}Q_4\bar{Q}_3Q_2Q_1Q_0 + Q_4Q_3\bar{Q}_2Q_1Q_0 \\ + \\ \bar{B}Q_4Q_3Q_2Q_1\bar{Q}_0 + BQ_4Q_3Q_2\bar{Q}_1Q_0 + SQ_4Q_3Q_2\bar{Q}_1Q_0 \end{gathered}$$

$$\begin{gathered} Q_0 = Q_4Q_3\bar{Q}_2Q_1Q_0 + Q_4Q_3Q_2\bar{Q}_1Q_0 + B\bar{Q}_4Q_3Q_2Q_1Q_0 + BQ_4\bar{Q}_3Q_2Q_1Q_0 \\ + \\ BQ_4Q_3Q_2Q_1\bar{Q}_0 + S\bar{Q}_4Q_3Q_2Q_1Q_0 + SQ_4\bar{Q}_3Q_2Q_1Q_0 + SQ_4Q_3Q_2Q_1\bar{Q}_0 \end{gathered}$$

$$O_1 = \bar{S}\bar{B}\bar{Q}_4Q_3Q_2Q_1Q_0 + \bar{S}\bar{B}Q_4\bar{Q}_3Q_2Q_1Q_0 + BQ_4Q_3\bar{Q}_2Q_1Q_0 + SQ_4Q_3\bar{Q}_2Q_1Q_0 + SQ_4Q_3Q_2\bar{Q}_1Q_0$$

$$O_2 = \bar{S}\bar{B}\bar{Q}_4Q_3Q_2Q_1Q_0 + SQ_4Q_3\bar{Q}_2Q_1Q_0$$

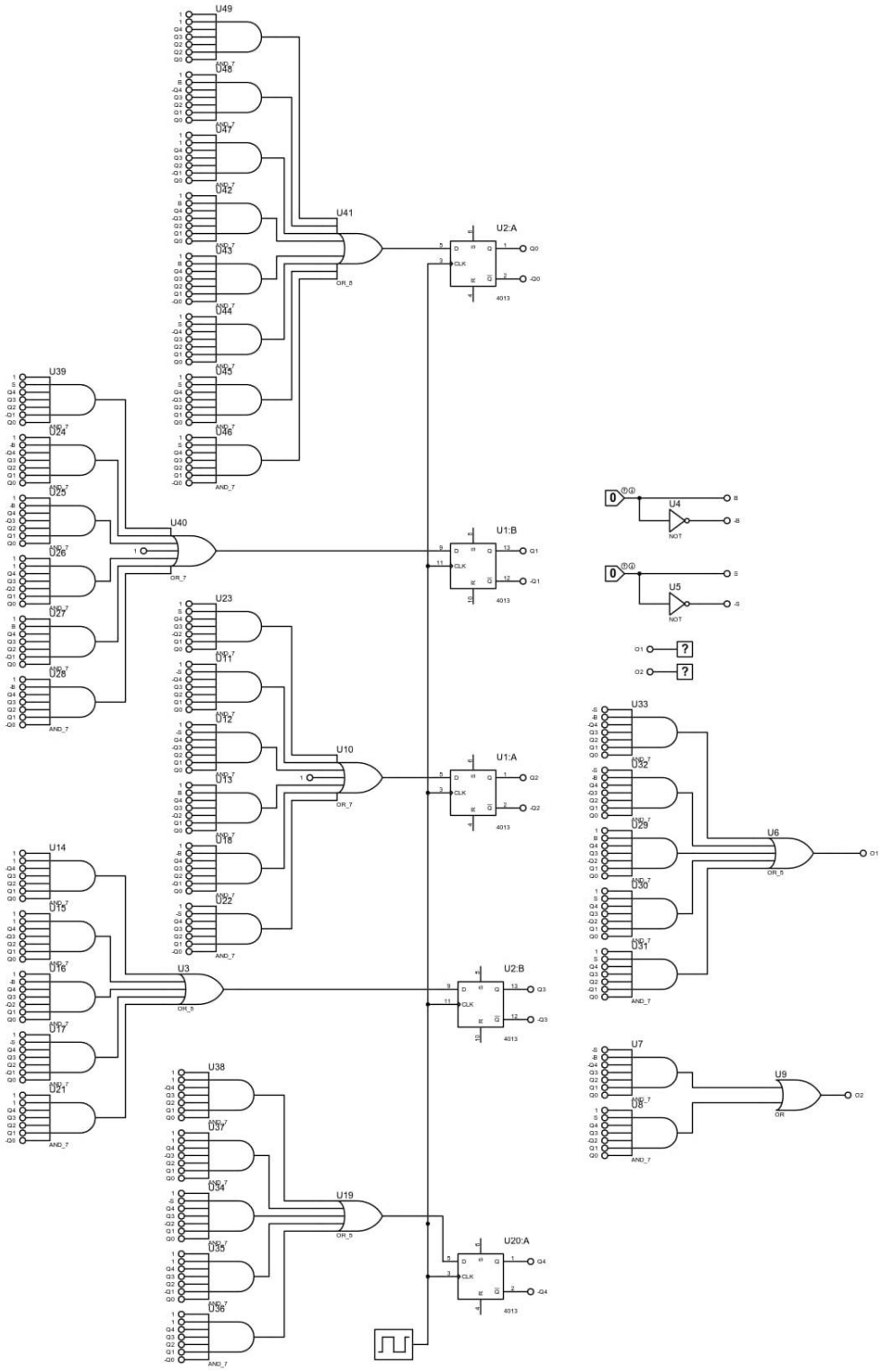

Figure 5: Logic Design, Part 4.2.2, One-Cold coding

So we need `5` flip-flops for One-Cold coding design.

4.2.3 Binary Coding

- `B`: 5 Rials
- `S`: 10 Rials
- O_1: Dispense
- O_2: Return5

	Inputs		State				
	S	B	$Q_2Q_1Q_0/O_1O_2$ 000	$Q_2Q_1Q_0/O_1O_2$ 001	$Q_2Q_1Q_0/O_1O_2$ 010	$Q_2Q_1Q_0/O_1O_2$ 011	$Q_2Q_1Q_0/O_1O_2$ 100
1	0	0	000/00	001/00	010/00	000/10	000/11
2	0	1	001/00	010/00	011/00	001/10	001/11
3	1	0	010/00	011/00	100/00	010/10	010/11
4	1	1	x	x	x	x	x

$$Q_2 = \bar{Q}_2 Q_1 \bar{Q}_0 S$$
$$Q_1 = \bar{Q}_1 \bar{Q}_0 S + \bar{Q}_2 Q_0 S + \bar{Q}_2 Q_1 \bar{Q}_0 \bar{S} + \bar{Q}_2 \bar{Q}_1 Q_0 B$$
$$Q_0 = \bar{Q}_1 \bar{Q}_0 B + \bar{Q}_2 Q_1 B + \bar{Q}_2 \bar{Q}_1 Q_0 \bar{B}$$
$$O_1 = \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 \bar{Q}_0$$
$$O_2 = Q_2 \bar{Q}_1 \bar{Q}_0$$

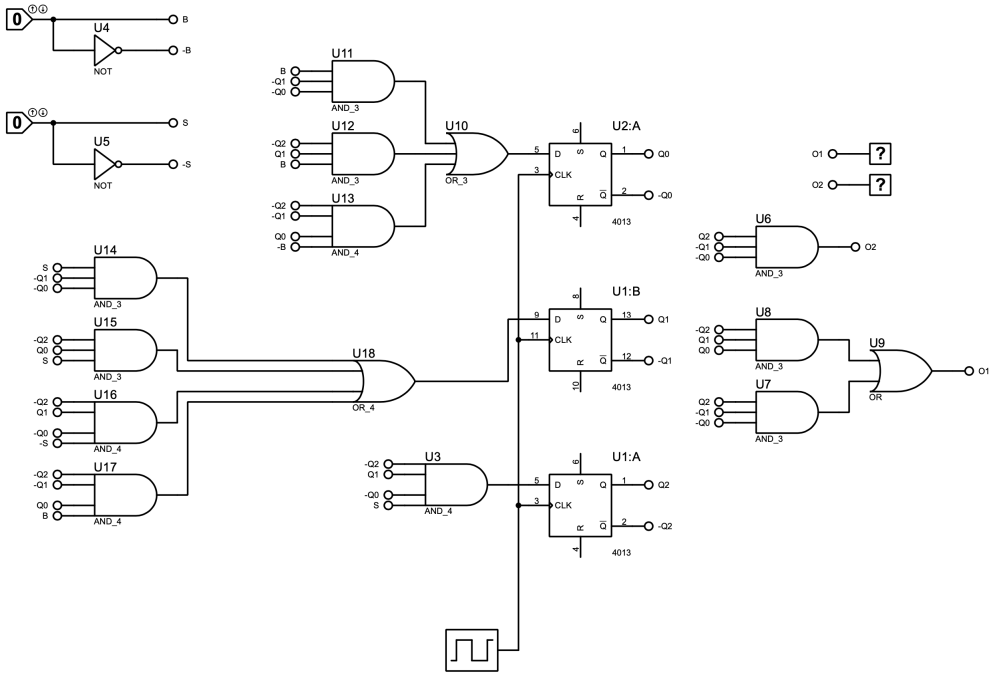

Figure 6: Logic Design, Part 4.2.3, Binary Coding

So we need `3` flip-flops for Binary Coding design.

4.3

The **Binary** coding uses only 3 flip-flops and is the least used number of flip-flops in all 3 designs.

4.4

The **Binary** coding has a much simpler circuit.

5 Intelligent House

5.1 Features

1. When the temperature goes upper than 30 °C, the cooling system is activated and wouldn't turn off until the temperature lowers to 24 °C.
2. When the temperature goes lower than 12 °C, the heating system is activated and wouldn't turn off until the temperature reaches 18 °C.
3. Every day with sunrise, the window curtains are opened and a light music is played in order to wake the residents up. After 30 minutes, the music pauses automatically.
4. Every day at late-night (23:30 PM), home is set to wind down mode: lights are dimmed, any music is paused, and curtains are closed.
5. When one of the residents says "open window", all of the windows are opened immediately.

5.2 Algorithmic State Machines

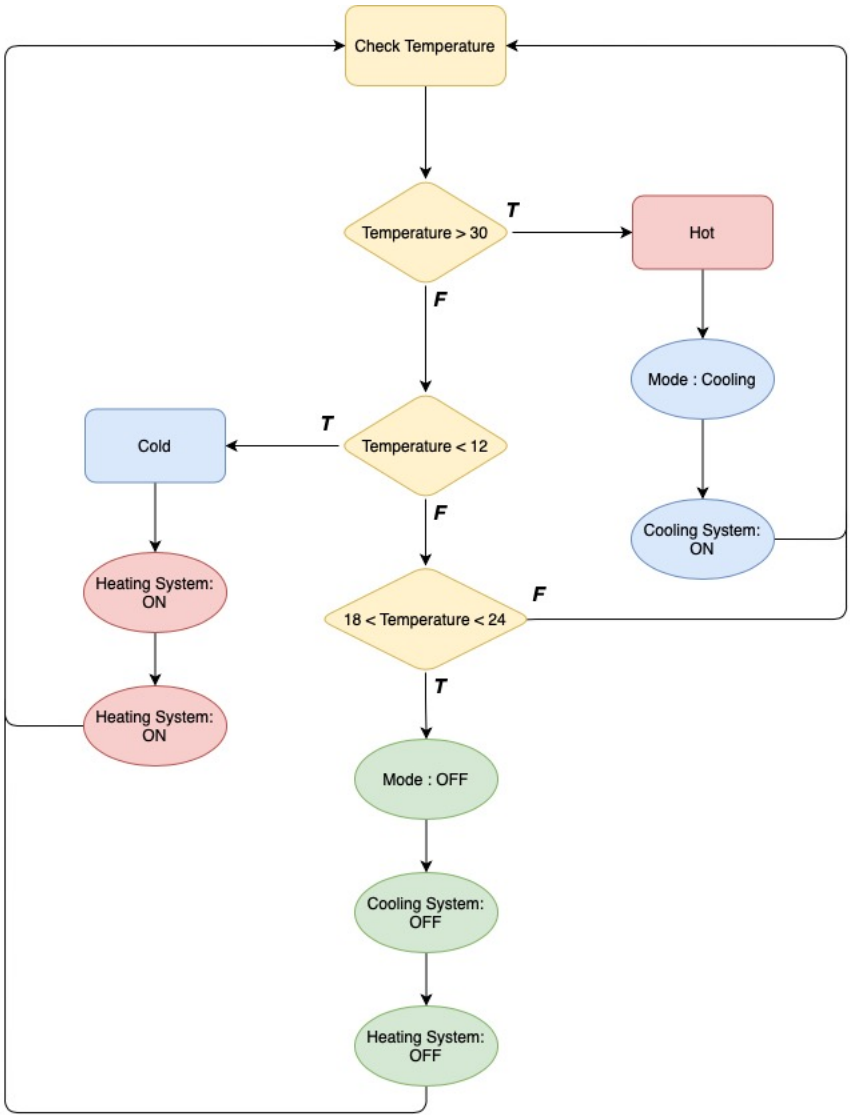

Figure 7: ASM, Heating/Cooling Control System

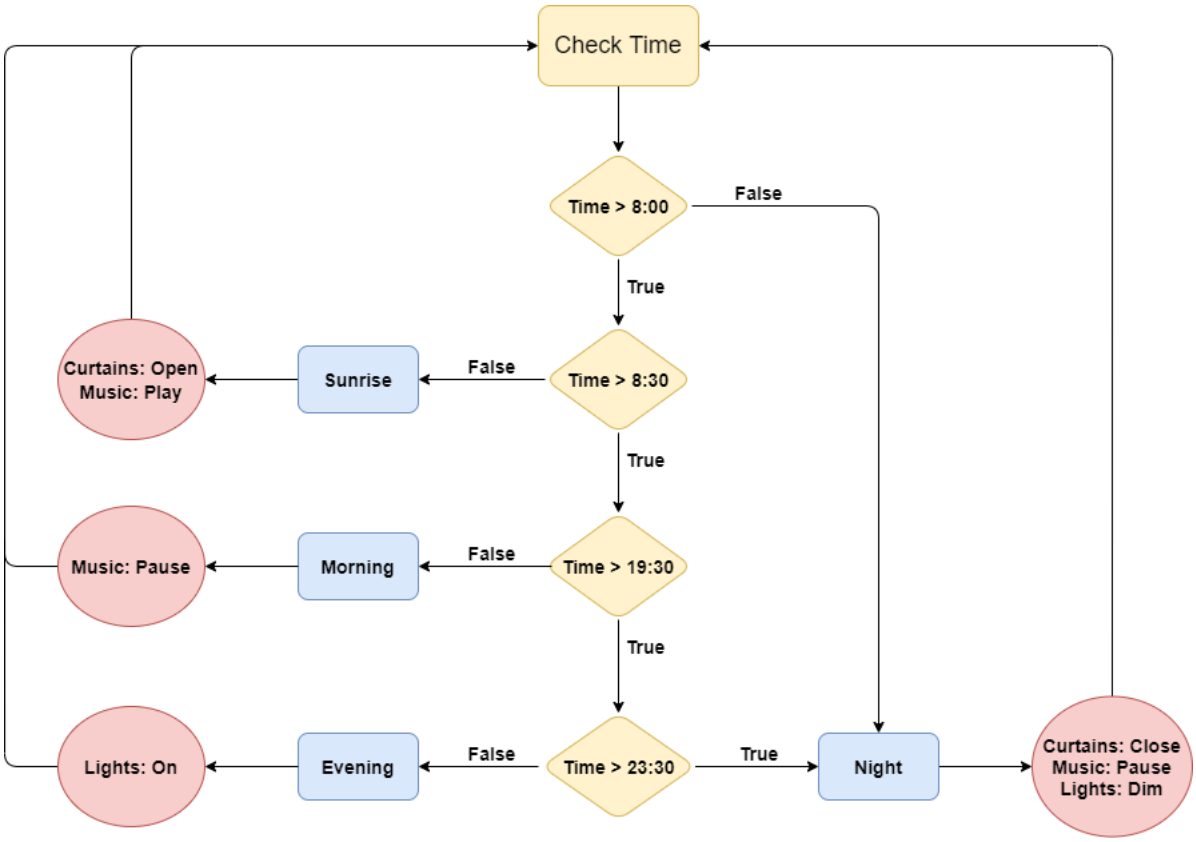

Figure 8: ASM, Sunset/Sunrise Control System

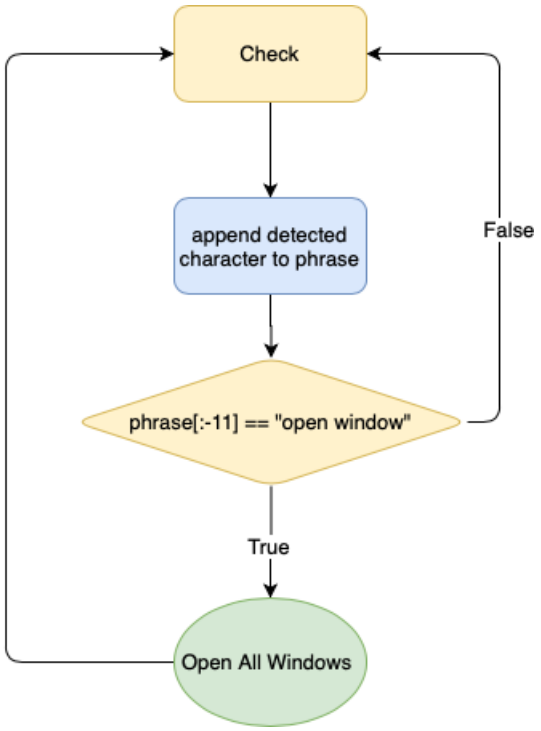

Figure 9: ASM, Window Control System

5.3 Finite State Machines

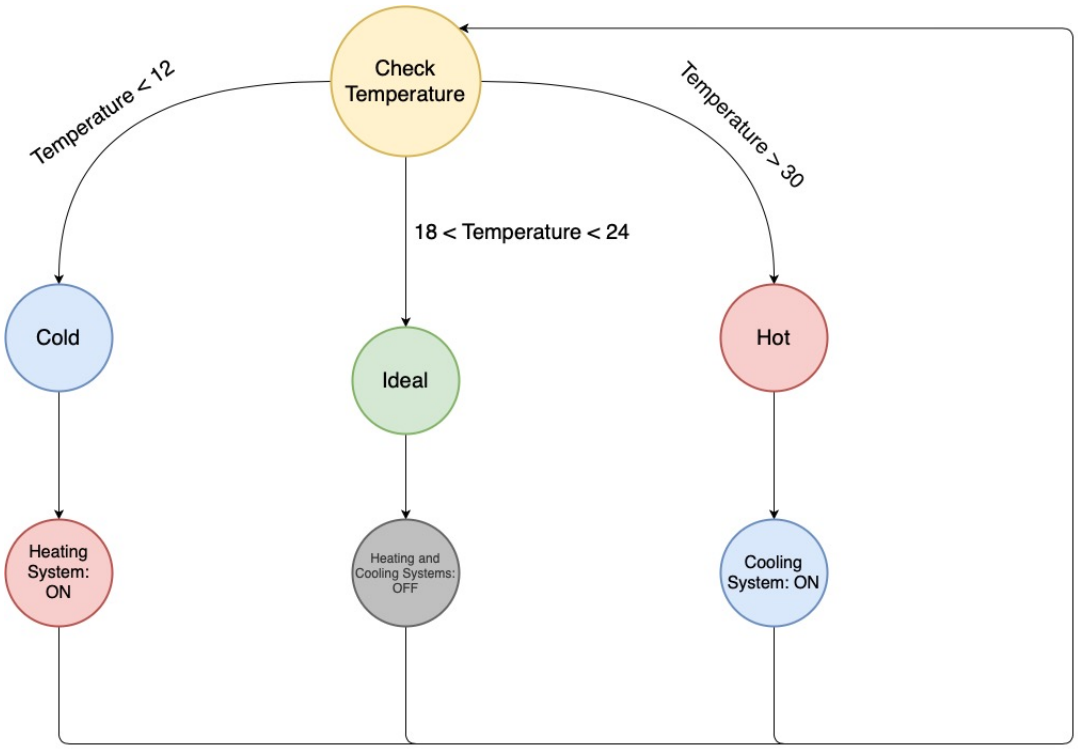

Figure 10: FSM, Sunset/Sunrise Control System

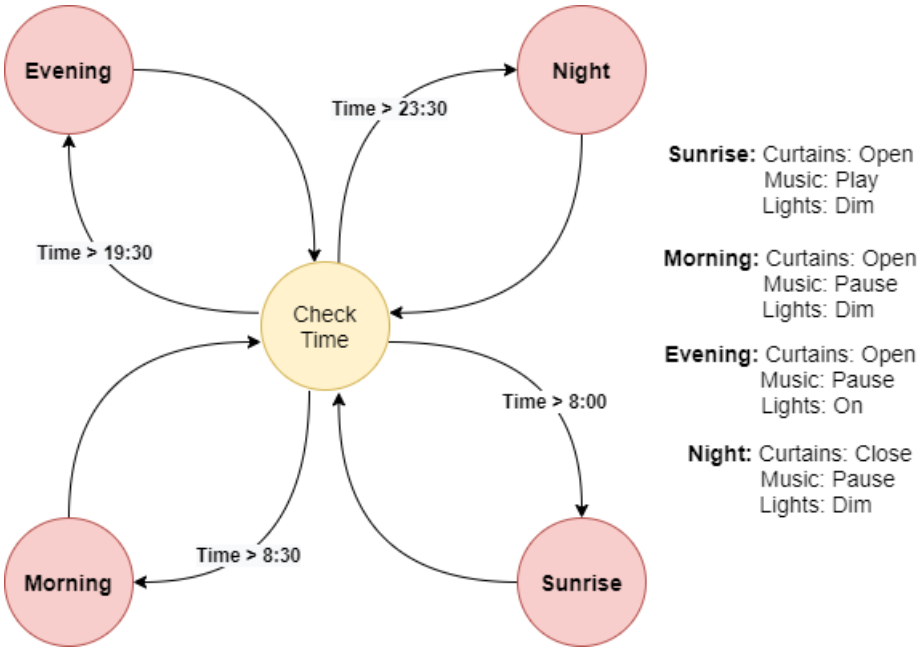

Figure 11: FSM, Heating/Cooling Control System

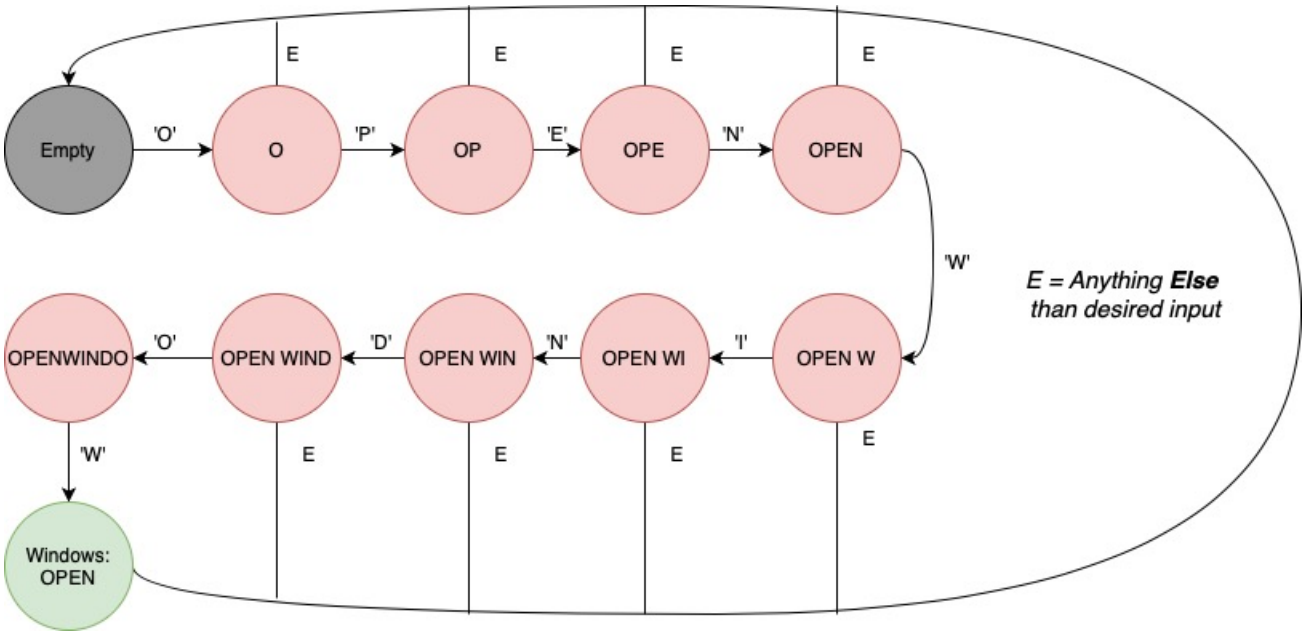

Figure 12: FSM, Window Control System