MICS 実験第一 J1 課題レポート

学籍番号 2210342, 鈴木謙太郎

2024 年 8 月 7 日

1　問題 1

1.1　目的

本問題では,FPGA において入出力を接続する方法について理解することを目指す.

1.2　原理・理論

本問に限らずこの実験において用いる FPGA とは,Field Programmable Gate Array の略であり, プログラムによって自由に書き換えられる論理素子である.

本実験では「Quartus」と呼ばれるソフトウェアを用いて GUI 上で回路を設計し, その回路を FPGA に書き込むことで, 実際にハードウェア上で回路を動作させることができる.

実際に回路を構成する際は, 入力ピン (ボタンやスイッチ, クロック信号など) と出力ピン (LED など) の間に組み合わせ回路などを配置し, それらを接続することで入力に対して出力を得ることができる.

1.3　回路のアルゴリズム・設計

本問では, 実験資料で示されたように,FPGA ボード上のトグルスイッチや押しボタンスイッチに対して LED を接続することで, 入力が与えられた際に LED が点灯するような回路を設計する.

1.4　実際に設計した回路

図 1 のような回路を設計した.

単純に入力スイッチのピンと出力 LED のピンを接続することで, スイッチの入力が LED の出力に反映されるようになっている.

1.5　実験・測定結果

実際に FPGA ボードに書き込んだ結果, スイッチを操作することで LED が点灯することが確認できた.

図 1 からも分かる通り, この回路では各スイッチの入力は独立しているので, 同時に複数のスイッチを操作しても出力が変化することはなかった.

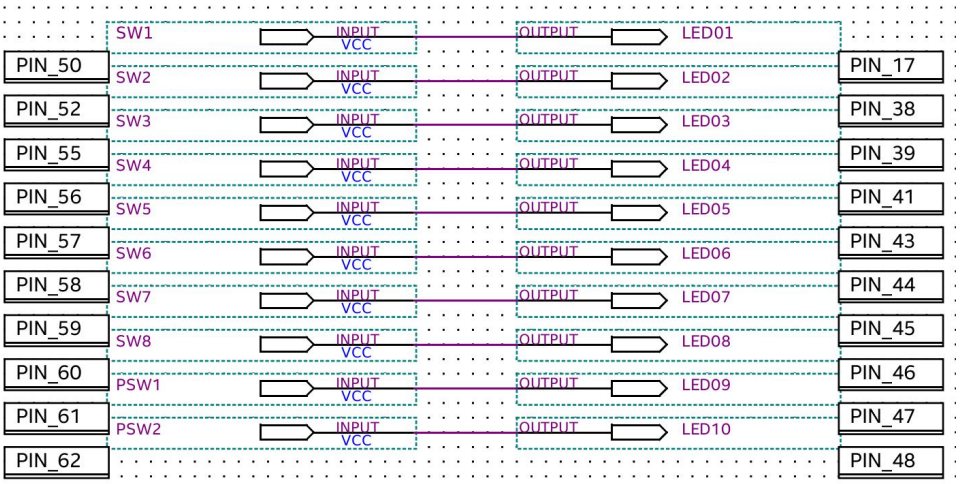


図 1: 問題 1 で設計した回路

2　問題 2

2.1　目的

本問では,AND ゲート,OR ゲート,NOT ゲートをはじめとするゲート回路を FPGA 上で動作させることを通じて, 組み合わせ回路の設計方法を理解することを目指す.

2.2　原理・理論

組み合わせ回路とは, 入力に対して出力が一意に定まる回路のことである.　この回路は, ゲート (AND,OR,NOT など) のみを組み合わせることで構成される.

2.3　回路のアルゴリズム・設計

本問では,2 ビットの入力 $x = x_1x_0$ と $y = y_1y_0$ に対して, $x > y$ のとき出力 $z = 1$, それ以外のとき $z = 0$ となる回路を設計する. なお, これらの入力は符号なしの 2 進数として扱うものとする.

このような回路を作るにあたって, 真理値表やカルノー図を用いることで簡略化された効率のよい論理回路を作ることができる.

本問でも図 2 のような真理値表とカルノー図を用いることで, 出力 z の論理式を式 (1) のように求めた.

$$z = x_0y_1'y_0' + x_1x_0y_0' + x_1y_1' \tag{1}$$

実際の回路設計では, 式 (1) の論理式をそのままゲートによる組み合わせ回路に変換することで, 期待する出力を得ることが可能である. また, 入力にはすべてトグルスイッチを用いた.

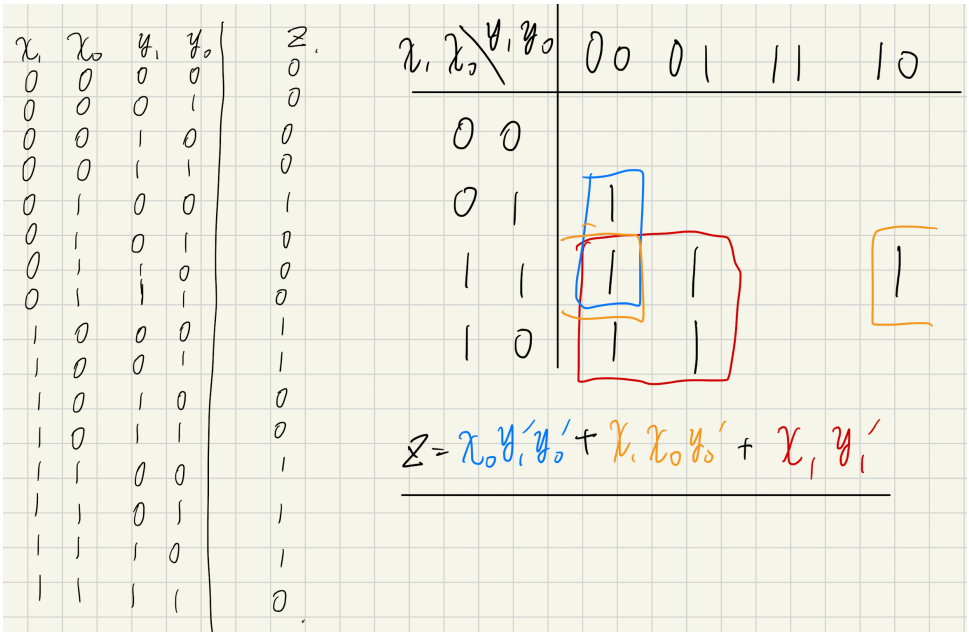


図 2: 問題 2 で用いた真理値表とカルノー図

2.4 実際に設計した回路

式 (1) をもとに, 図 3 のような回路を設計した.

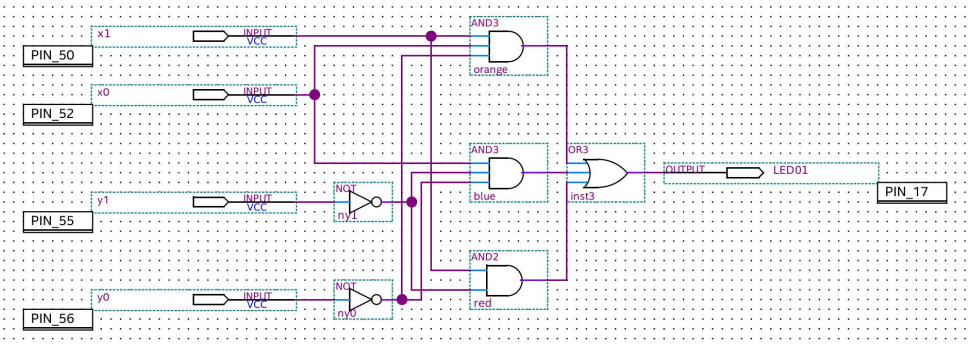


図 3: 問題 2 で設計した回路

2.5 実験・測定結果

実際に FPGA ボードに書き込んだ結果,$x > y$ のときに限り LED が点灯し,$x = y$ や $x < y$ となるときは消灯することが確認できた. よって, 期待する出力を得ることができていると考えられる.

3　問題 3

3.1　目的

本問では,D 型フリップフロップを用いた微分回路を設計することで, 順序回路の設計方法を理解することを目指す.

3.2　原理・理論

順序回路とは, 組み合わせ回路とは異なり, 出力が入力だけでなく過去の状態にも依存する回路のことである. このような回路は, フリップフロップなどの記憶素子を用いることで設計することができる.

フリップフロップは,1 ビットので 0 田を記憶できる素子である. 特に D 型フリップフロップは, データとクロック信号を入力として受け取り, クロックが立ち上がるときに D 入力の値を出力に反映する. このような性質上, クロックはフリップフロップの動作を制御するために必要である. この入力を人の手で正確に切り替えることは難しいため,FPGA ボードに備え付けられたクロック信号発生器を用いる.

本問で作成する微分回路とは, 入力信号の立ち上がりエッジを検出し, そのエッジが立ち上がるたびに出力信号をトグルする回路である. このような回路を作成することで, 人間がスイッチを 1 回押したときに,1 クロックの長さのパルスを 1 回だけ出力することができる. 通常人がスイッチを操作すると,1 クロック以上の信号が入力されてしまい, 論理回路を正確にテストすることが難しいため, 微分回路は論理回路のテストにおいて非常に有用である.

3.3　回路のアルゴリズム・設計

本問では, クロック信号を入力として受け取り, その立ち上がりエッジを検出する微分回路を設計する.

フリップフロップなどを含む順序回路を設計する際には, 出力が過去の状態に依存するため, 状態遷移図などを作成して設計を行うことが一般的である.

図 4 のような回路を設計するにあたって, 便宜上左側の D 型フリップフロップの入力を D_0, 出力を Q_0 とし, 右側の D 型フリップフロップの入力を D_1, 出力を Q_1 とする.

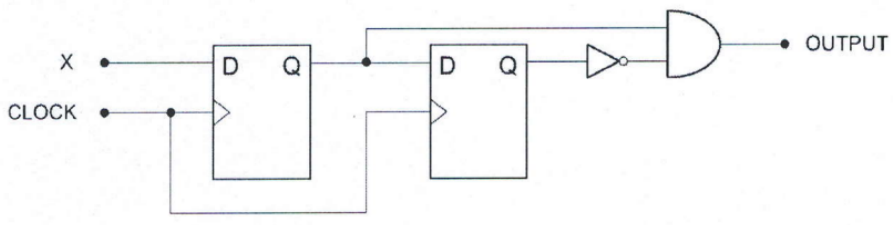
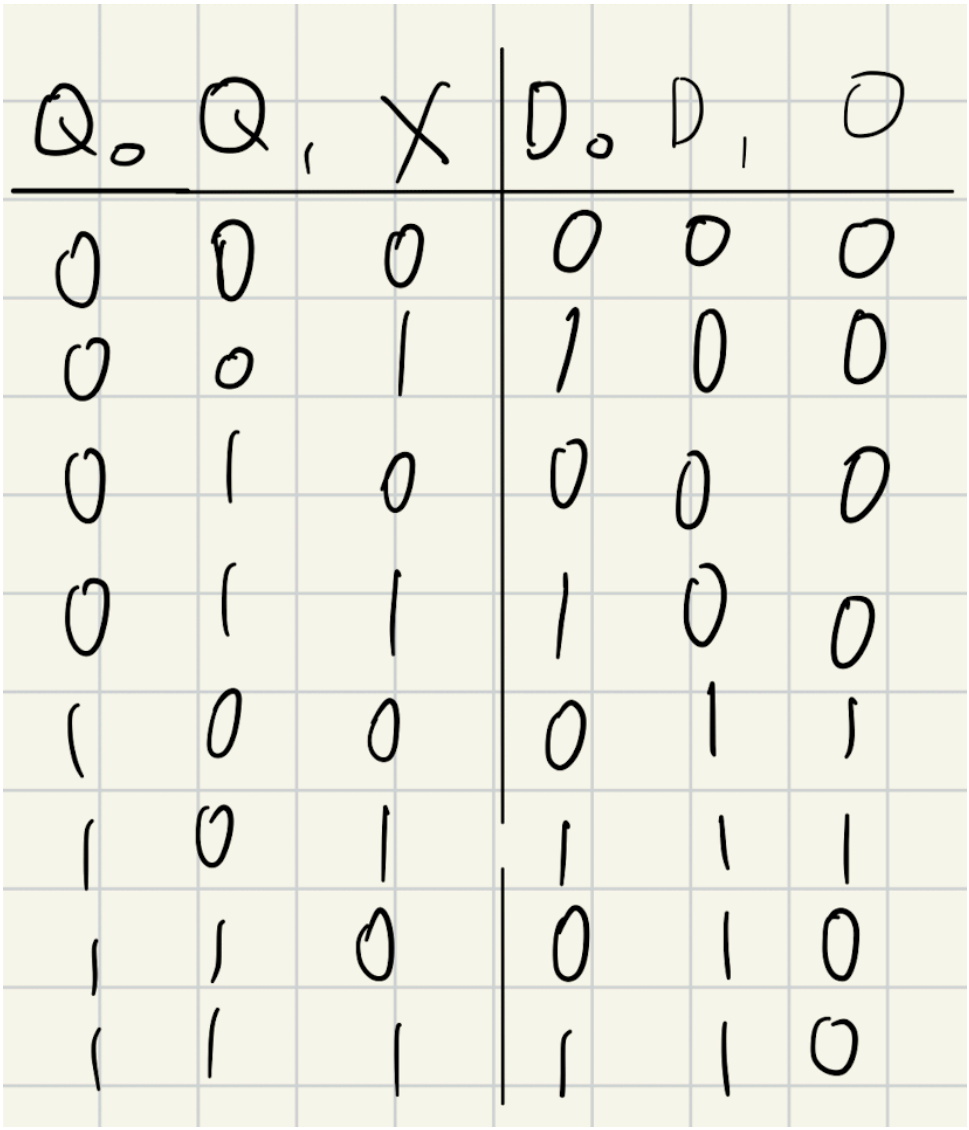


図 4: 問題 3 で設計する微分回路の回路図 (実験資料より引∑∑用)

この回路の真理値表は図 5 のようになった.



Q_0	Q_1	X	D_0	D_1	O
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	1	1	1	0

図 5: 問題 3 で用いた真理値表

これの真理値表をもとに, 図 6 のような状態遷移図を作成した.

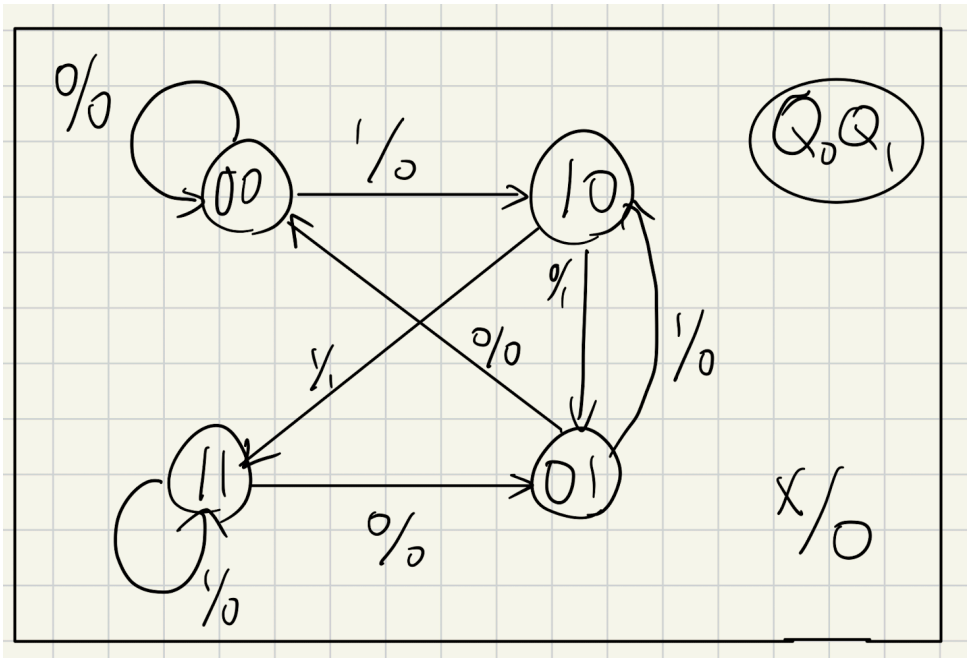


図 6: 問題 3 で用いた状態遷移図

ここで, 状態 00 と状態 01 に注目すると, ともに 0 を入力すると 0 を出力して状態 00 に遷移し, 1 を入力すると 1 を出力して状態 10 に遷移することが分かる. よって, この 2 つの状態は同じものとみなせる.

このような同じ状態が複数ある場合, その状態への遷移とその状態からの遷移をまとめることで, 状態遷移図を図 7 のように簡略化することができる.

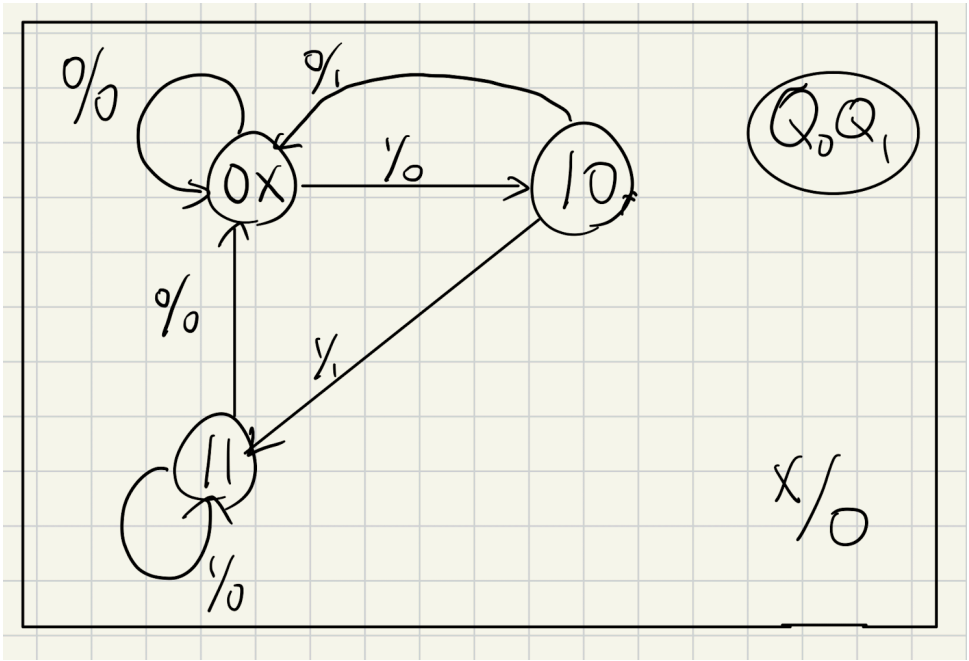


図 7: 図 6 を簡略化した後の状態遷移図

3.4　実際に設計した回路

図 4 の回路を, 実験資料で示された 2 ビットカウンタ回路と接続し, 図 8 のような回路を設計した. これを Quartus の RTL Viewer で図持すると図 9 のようになる.

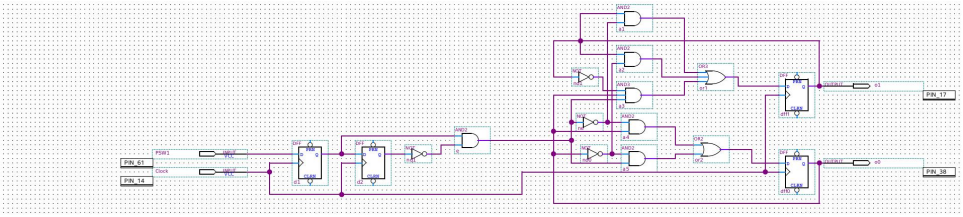


図 8: 問題 3 で設計した回路

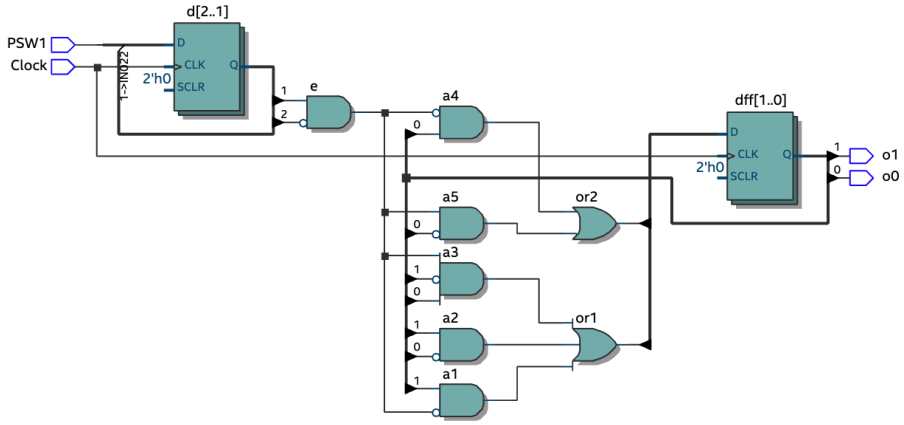


図 9: 図 8 を RTL Viewer で表示したもの

3.5　実験・測定結果

実際に FPGA ボードに書き込み, クロック信号を入力した状態で押しボタンを用いて動作を確認した. ボタンを押した長さにかかわらず, 出力 LED に表示されるカウンタの値が 1 ずつ増加したので, 微分回路が正常に動作していると考えられる.

4　問題 4

4.1　目的

本問では, 複数の D 型フリップフロップを用いた順序回路を, 効率の良い論理回路を設計し理解することを目指す.

4.2　原理・理論

本問では,100 円硬貨を 2 個受け取ると 200 円の切符を出力する図 10 のような自動販売機の制御回路 T を設計する.

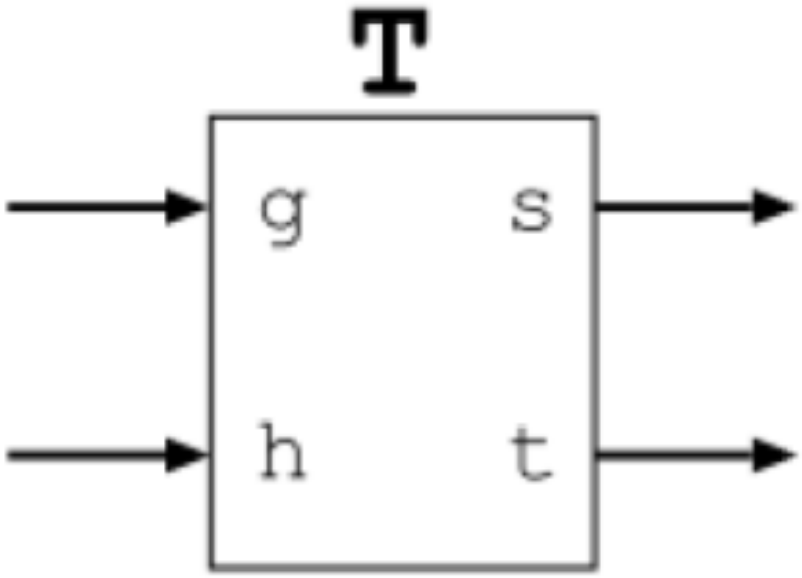


図 10: 問題 4 で設計する自動販売機の制御回路のイメージ (実験資料より引用)

4.3　回路のアルゴリズム・設計

実験資料で示された動作条件を下に, 以下のような仕様を満たす回路を設計する.

1. 100 円を入れる動作 (入力 h) を PSW1, 切符を取る動作 (入力 g) を PSW2 で与える.
2. 出力 s を LED1, t を LED2 で表示する.
3. 切符が出ていない時, 切符を取る動作 (入力 g) はドントケアとして扱う (出力に影響を与えない).
4. 切符が出ている時,100 円を入れる動作 (入力 h) はドントケアとして扱う (出力に影響を与えない).

仕様を元に, 図 11 のような真理値表を作成した. また, この真理値表をもとに図 12 のようなカルノー図と図 13 のような状態遷移図を作成した.

これらの図をもとに、回路の次状態の出力 s', t' を式 (2) のように求めた.

$$\begin{aligned} s' &= \bar{s}\bar{t}h + s\bar{t}\bar{h} \\ t' &= \bar{s}t\bar{g} + s\bar{t}h \end{aligned} \tag{2}$$

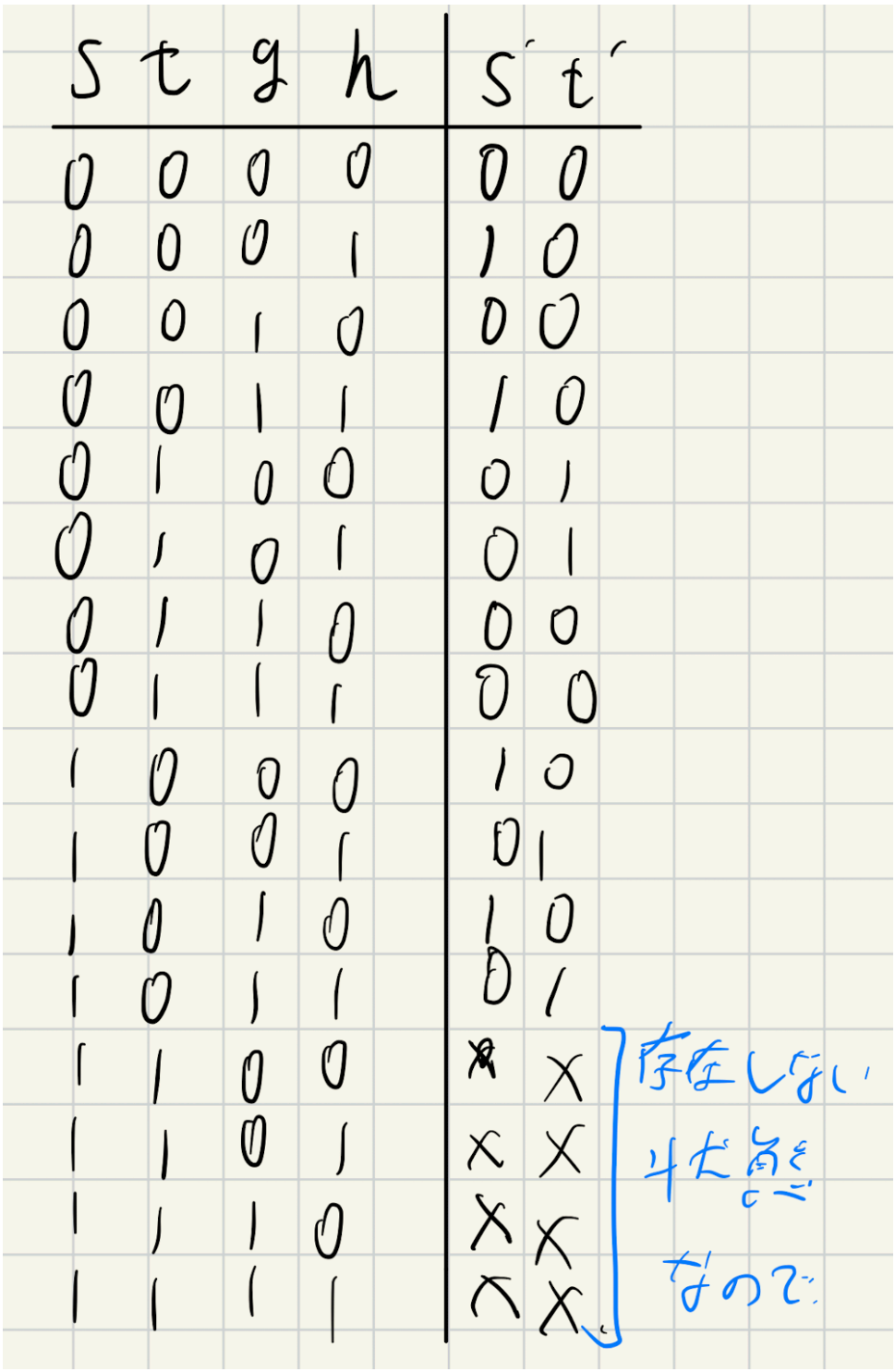


図 11: 問題 4 で用いた真理値表

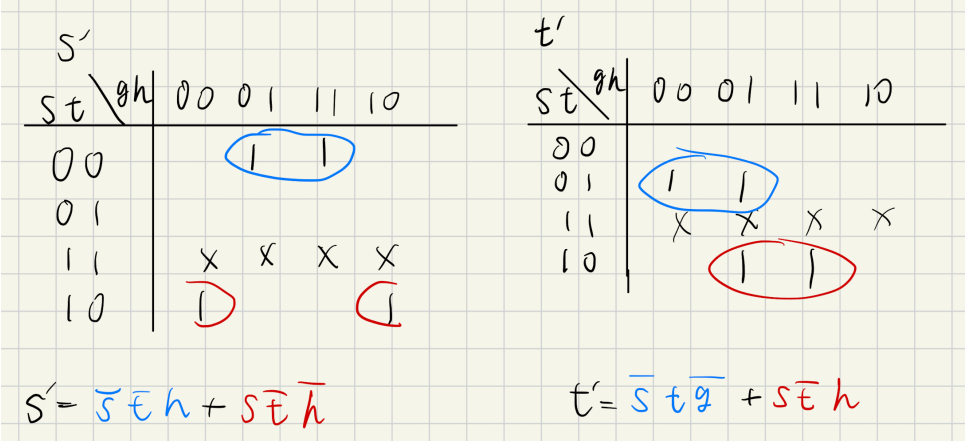


図 12: 問題 4 で用いたカルノー図

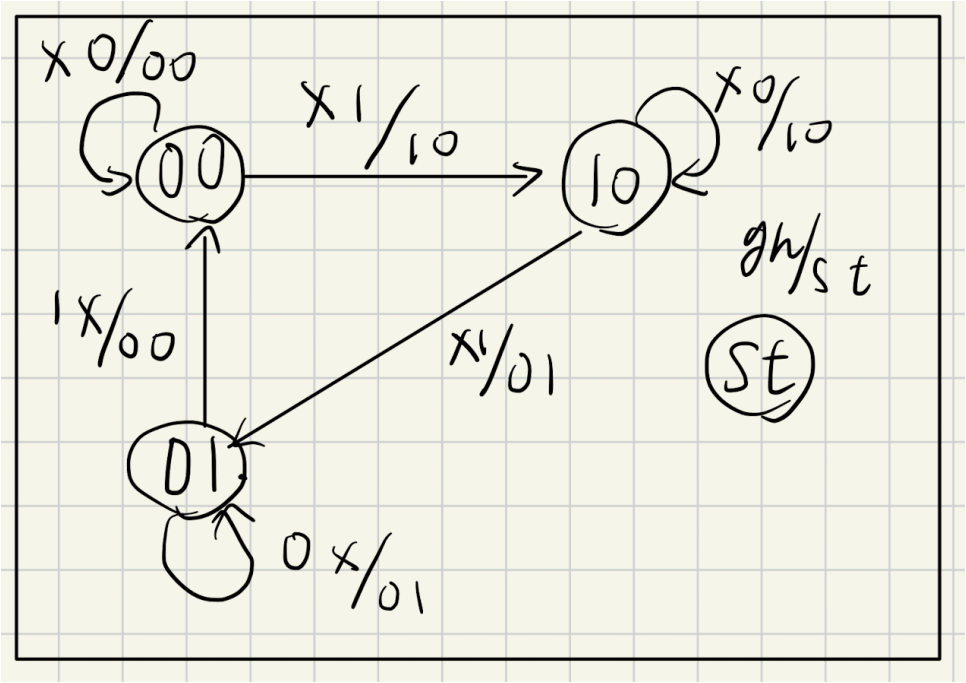


図 13: 問題 4 で用いた状態遷移図