

提出日 2026 年 06 月 30 日

2026 年度 電子制御工学科 3 年

電子制御工学実験 1 レポート

課題記号 T-2

実験テーマ名 デジタル回路

提出者: 実験班 - 名列番号 15 氏名 柴田健琉

共同実験者: なし

実験実施日: 1. 2026 年 06 月 02 日
2. 2026 年 06 月 16 日
3. 2026 年 06 月 23 日

1 実験目的

今回の実験ではデジタル回路の基礎と応用について学ぶために行った。

2 理論

2.1 デジタル

デジタルとは最小単位が存在する物理量であり、連続的で最小単位が存在しないアナログと対になる。この最小単位というのは時間や電圧などの刻み幅を決めるもので分解能ともいう。分解能を高くすれば、真の値と観測値の誤差が小さくなる。デジタルには「誰が測っても同じになる」とことと「多少ノイズが混じっても大丈夫」という長所があり、逆に「最小単位より小さいものは測れない」という短所がある。デジタルにより、現実世界の様々な現象をある一定の精度で数値的に解析・再現することが出来るようになった¹⁾。

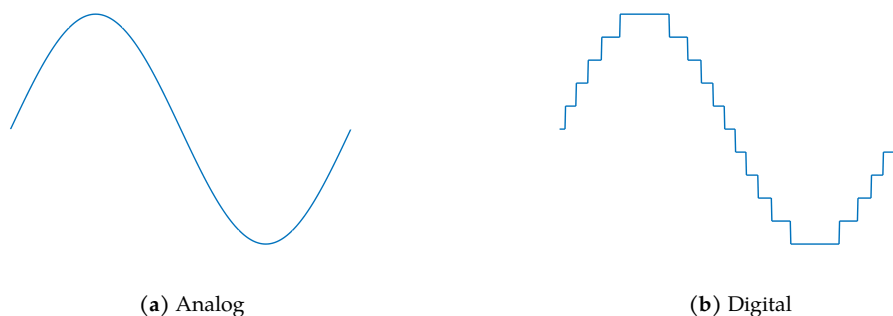


Fig. 2-1: Example of Analog and Digital Signal

2.2 二進数

二進数とは 0 と 1 などの二種類の記号を用いた数の表記法である。これにより電気のオン・オフなど回路で扱いやすい形で様々な数を表すことが出来るようになった²⁾。

2.3 ブール代数

ブール代数とは二値化された数に対し、演算子や法則を定義した数学の分野である。ブール代数では基本的な演算子は論理和、論理積、論理否定である²⁾。これらの演算子の演算結果は Table 2-1 のような真理値表でまとめることが出来る³⁾。

これら演算子に対する法則を Eq. (2-1) に示す。

Table 2-1: Truth Tables of Basic Logic Operations

(a) Logical And ($x = a \cdot b$)

a	b	x
0	0	0
1	0	0
0	1	0
1	1	1

(b) Logical Or ($x = a + b$)

a	b	x
0	0	0
1	0	1
0	1	1
1	1	1

(c) Logical Not ($x = \bar{a}$)

a	$\bar{a}$
0	1
1	0

$$\begin{aligned}
 a + a &= a && \text{(Idempotent Law)} \\
 a \cdot a &= a && \text{(Idempotent Law)} \\
 a + 1 &= 1 && \text{(Identity Law)} \\
 a \cdot 0 &= 0 && \text{(Identity Law)} \\
 a + (a \cdot b) &= a && \text{(Absorption Law)} \\
 a \cdot (a + b) &= a && \text{(Absorption Law)} \\
 (a + b) + c &= a + (b + c) && \text{(Associative Law)} \\
 (a \cdot b) \cdot c &= a \cdot (b \cdot c) && \text{(Associative Law)} \\
 \bar{\bar{a}} &= a && \text{(Double Negation)}
 \end{aligned} \tag{2-1}$$

ブール代数の代表的な定理としてド・モルガンの定理がある。

この定理は「あるブール式の変数を否定し、式の中の論理積と論理和を入れ替えた式は、全体を否定した式と同じ値をとる」というもので Eq. (2-2) のような式で表現される²⁾。

$$\overline{f(x_1, x_2, \dots, +, \cdot)} = f(\bar{x}_1, \bar{x}_2, \dots, \cdot, +) \tag{2-2}$$

2.4 論理ゲート

論理ゲートは論理演算を電気回路で表現したもので、回路記号は Fig. 2-2 で表される⁴⁾。

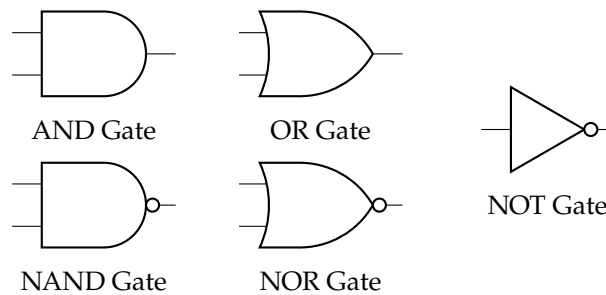


Fig. 2-2: Symbols of Logic Gates

図中の NAND ゲートや NOR ゲートはそれぞれ AND ゲートと OR ゲートの結果の論理否定を出力する。これら論理ゲートを複数個まとめてパッケージした IC が論理ゲート IC と呼ばれている。

2.5 組み合わせ回路

複雑な条件を扱う論理回路は論理ゲート 1 つだけでは実現できない。論理ゲートを複数種類・複数個組み合わせさせて作られた回路を組み合わせ回路という。

組み合わせ回路を作成するにあたり、論理式の簡略化はコストや実装空間の制限上必須である。この簡略化で用いられる方法の 1 つがカルノー図である。

カルノー図は論理式の結果を特定の条件に対応する行と列に写したもので、 $x = (\bar{a} \cdot b \cdot c) + (a \cdot b \cdot \bar{c}) + (a \cdot \bar{b} \cdot \bar{c})$ のカルノー図は Fig. 2-3 となる。

	$\bar{c}$	c
$\bar{a} \bar{b}$		
$\bar{a} b$		1
$a b$	1	
$a \bar{b}$	1	

Fig. 2-3: Karnaugh Map of $x = (\bar{a} \cdot b \cdot c) + (a \cdot b \cdot \bar{c}) + (a \cdot \bar{b} \cdot \bar{c})$

カルノー図から隣接する 1 のセルを「くくる」と論理式を簡略化することが出来る⁵⁾。今回の例では $a \cdot b \cdot \bar{c}$ のセルと $a \cdot \bar{b} \cdot \bar{c}$ のセルでは b の値によらず 1 になっているので、くくって $x = (a \cdot \bar{c}) + (\bar{a} \cdot b \cdot c)$ を得る。

2.6 順序回路

論理回路には現在の状態と入力によって次の状態へ変化する回路が存在する。これらの回路は順序回路と呼ばれている。例えば、ある状態 A で入力が 0 の時、出力を 1 にし状態 B へ、入力が 1 の時、出力を 0 にし状態 C へ遷移するというような回路のことである⁶⁾。

順序回路の動作を説明するには状態遷移図や状態遷移表が有効である。状態遷移図は各状態をノード、入力/出力の順で表記された状態の遷移をエッジとする有向グラフとして表現される。状態遷移表は入力と現状態が次にどの状態と出力をするのかを表にまとめたものである⁷⁾。前述の例を状態遷移図と状態遷移表で表現するとそれぞれ Fig. 2-4 と Table 2-2 となる。

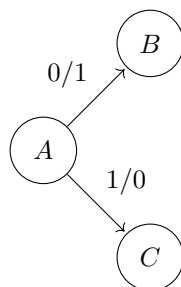


Fig. 2-4: Example of State Transition Graph

Table 2-2: Example of State Transition Table

State	Input	Next State	Output
A	0	B	1
A	1	C	0

3 実験手順・条件

3.1 実験器具

- ブレッドボード
- 直流安定化電源
- タクトスイッチ x4
- LED x4
- 7セグメント LED
- カーボン抵抗器 330Ω or 470Ω x12
- ジャンパーワイヤ
- AND ゲート IC TC74HC08
- NAND ゲート IC HD14012BP
- OR ゲート IC TC74HC32
- NOT ゲート IC TC74HC04
- 7セグメントドライバ IC TC4511
- D フリップフロップ IC TC74HC74 x2
- Nch MOSFET

3.2 課題 1-1

AND ゲートあるいは OR ゲートの動作を確認する。

2つのボタンを入力とし, 出力を7セグメント LED のドットに表示させる回路を作成する。

3.3 課題 1-2

4入力 NAND ゲートの動作を確認する。

4つのボタンを入力とし, 出力を7セグメント LED のドットに表示させる回路を作成する。

3.4 課題 1-3

Fig. 3-1 に示す組み合わせ回路を作成し動作を確認する。

4つのボタンを入力とし, 出力を7セグメント LED のドットに表示させる回路を作成する。

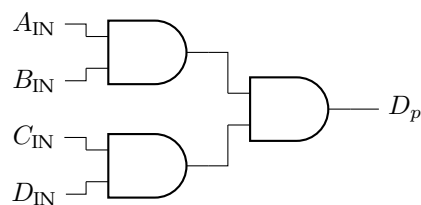


Fig. 3-1: Circuit Diagram of Assignment 1-3

3.5 課題 1-4

Fig. 3-2 に示す組み合わせ回路を作成し動作を確認する.

4つのボタンを入力とし, 出力を7セグメントLEDのドットに表示させる回路を作成する.

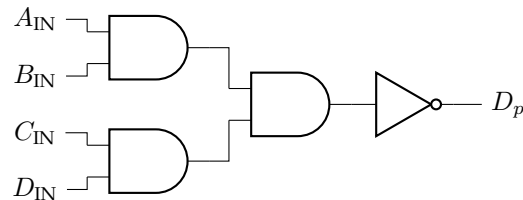


Fig. 3-2: Circuit Diagram of Assignment 1-4

3.6 課題 2-1

7セグメントドライバICを用いて1桁のBDCデコーダを作成する.

3.7 課題 2-2B

論理ゲートを複数個使用し, 0-9 以外の数の表現を実現する回路を作成する. なお, 既存の 0-9 に対応する BDC コードが入力された際にはセグメントを点灯させないこと.

3.8 課題 3-1

論理ゲートを組み合わせて Fig. 3-3 の RS フリップフロップを作成し, 動作を確認する.

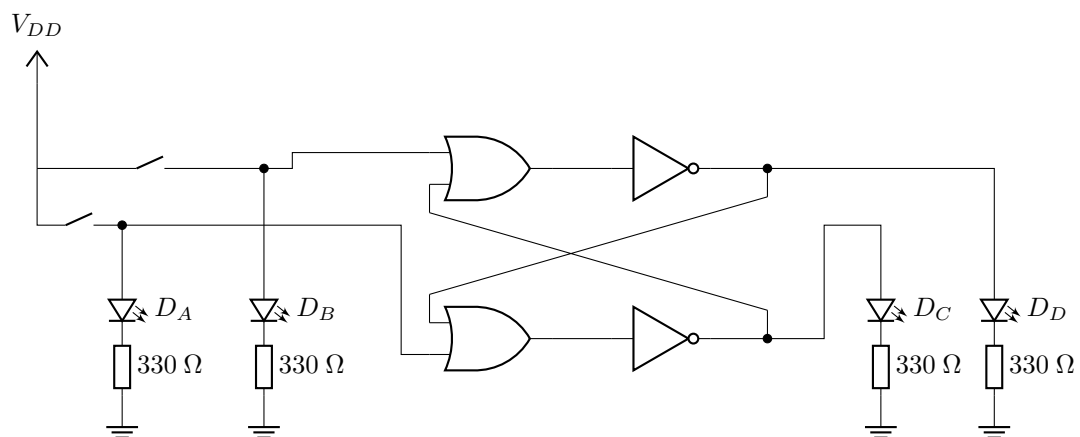


Fig. 3-3: Circuit Diagram of Assignment 3-1

3.9 課題 3-2

D フリップフロップ IC を用いて Fig. 3-4 の回路を作成し, 動作を確認する.

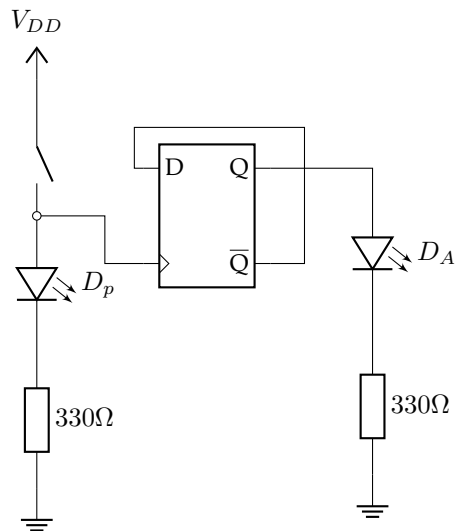


Fig. 3-4: Circuit Diagram of Assignment 3-2

3.10 課題 3-3

D フリップフロップ IC を 2 個用いて Fig. 3-5 の順列回路を作成し, 動作を確認する.

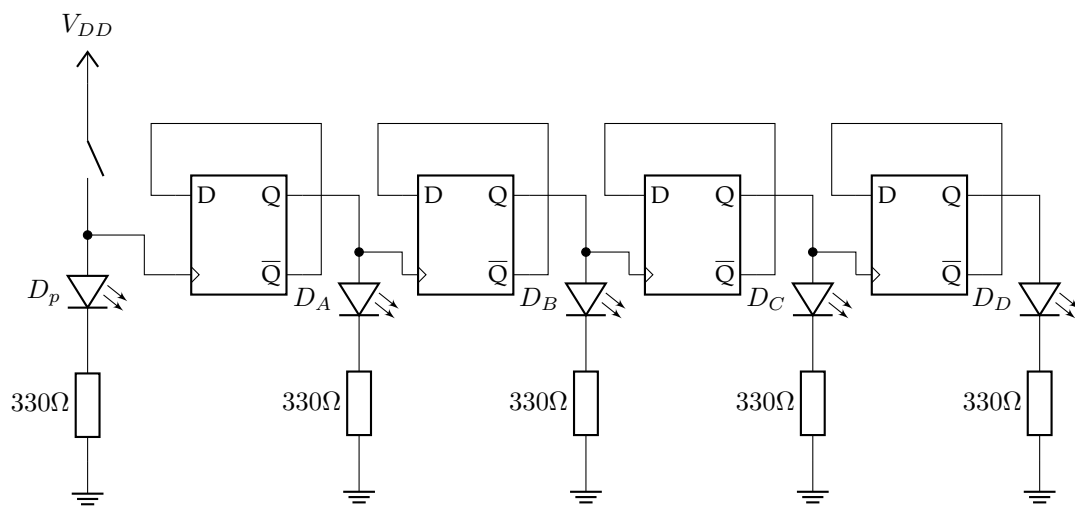


Fig. 3-5: Circuit Diagram of Assignment 3-3

3.11 課題 3-4

Fig. 3-6 の順序回路を組み, 動作を確認する. ? に適切な論理ゲートを使用する.

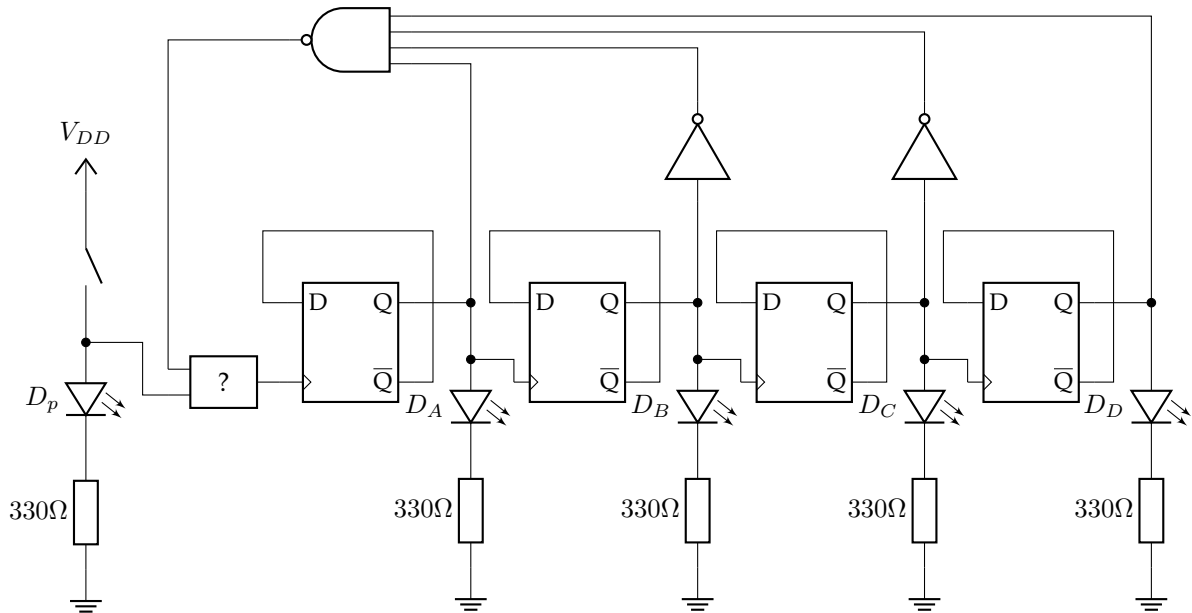


Fig. 3-6: Circuit Diagram of Assignment 3-4

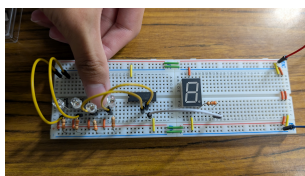
3.12 応用課題 A

Section 3.10 で作成した 4 ビットカウンタにおいて, 最下位ビットの D フリップフロップの CK にパルスを印加した瞬間のそれぞれのビット出力の応答波形を観察する.

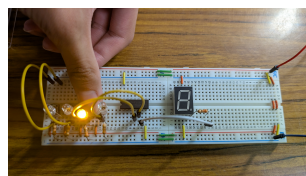
4 実験結果

4.1 課題 1-1

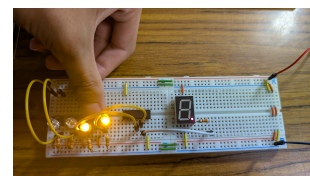
4.1.1 動作



(a) $a = 0, b = 0$



(b) $a = 1, b = 0$



(c) $a = 1, b = 1$

Fig. 4-1: 74HC08 AND Gate, Input A on Second from Right ($x = a \cdot b$)

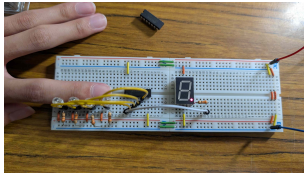
4.1.2 観測された真理値表

Table 4-1: Observed Truth Table of $x = a \cdot b$

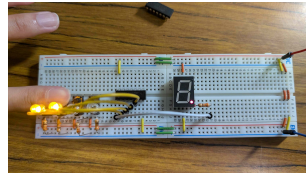
a	b	x
0	0	0
1	0	0
0	1	0
1	1	1

4.2 課題 1-2

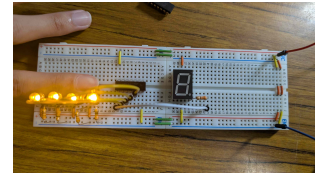
4.2.1 動作



(a) $a = 0, b = 0, c = 0, d = 0$



(b) $a = 1, b = 1, c = 0, d = 0$



(c) $a = 1, b = 1, c = 1, d = 1$

Fig. 4-2: HD14012BP 4 Inputs NAND Gate ($x = \overline{a \cdot b \cdot c \cdot d}$)

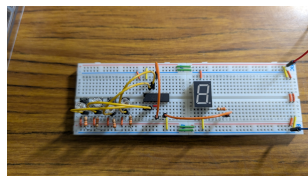
4.2.2 観測された真理値表

Table 4-2: Observed Truth Table of $x = \overline{a \cdot b \cdot c \cdot d}$

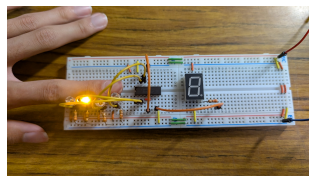
a	b	c	d	x
0	0	0	0	1
1	0	0	0	1
0	1	0	0	1
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	1
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0

4.3 課題 1-3

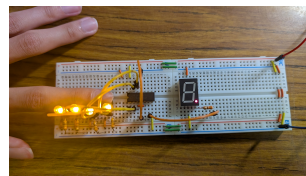
4.3.1 動作



(a) $a = 0, b = 0, c = 0, d = 0$



(b) $a = 0, b = 1, c = 0, d = 0$



(c) $a = 1, b = 1, c = 1, d = 1$

Fig. 4-3: 4 Inputs AND Gate with Three 2 Inputs AND Gates,
Input A on Most Left ($x = (a \cdot b) \cdot (c \cdot d)$)

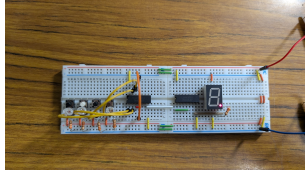
4.3.2 観測された真理値表

Table 4-3: Observed Truth Table of $x = (a \cdot b) \cdot (c \cdot d)$

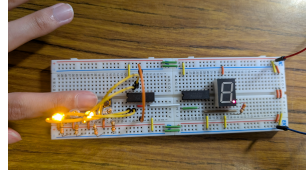
a	b	c	d	x
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	0
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

4.4 課題 1-4

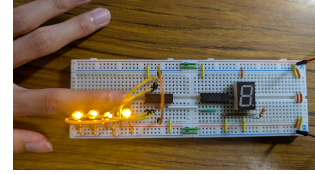
4.4.1 動作



(a) $a = 0, b = 0, c = 0, d = 0$



(b) $a = 1, b = 0, c = 1, d = 0$



(c) $a = 1, b = 1, c = 1, d = 1$

Fig. 4-4: 4 Inputs NAND Gate with Three 2 Inputs AND Gates and a NOT Gate, Input A on Most Left ($x = \overline{(a \cdot b) \cdot (c \cdot d)}$)

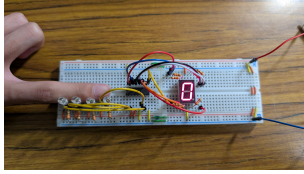
4.4.2 観測された真理値表

Table 4-4: Observed Truth Table of $x = \overline{(a \cdot b) \cdot (c \cdot d)}$

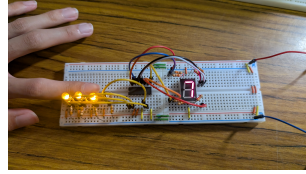
a	b	c	d	x
0	0	0	0	1
1	0	0	0	1
0	1	0	0	1
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	1
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0

4.5 課題 2-1

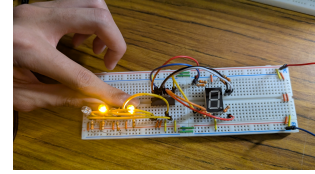
4.5.1 動作



(a) $a = 0, b = 0, c = 0, d = 0$



(b) $a = 1, b = 1, c = 1, d = 0$



(c) $a = 0, b = 1, c = 0, d = 1$

Fig. 4-5: 7 Segment Driver (4 Bits BDC Input, LSB on Left)

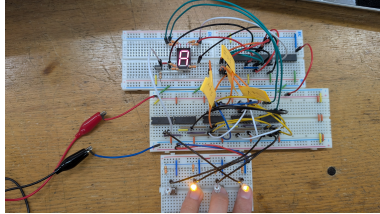
4.5.2 観測された真理値表

Table 4-5: Observed Truth Table of 7 Segment Driver

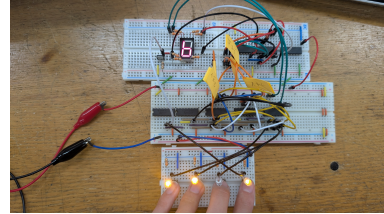
a	b	c	d	A	B	C	D	E	F	G
0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	1	1	0	0	0	0
0	1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	1	1	0	0	1
0	0	1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
1	1	1	0	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
0	1	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

4.6 課題 2-2B

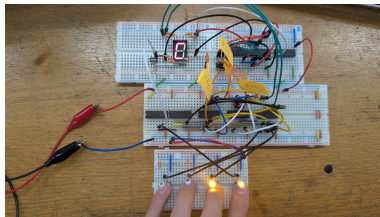
4.6.1 動作



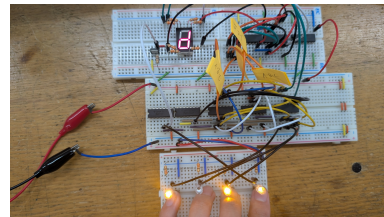
(a) $a = 0, b = 1, c = 0, d = 1$



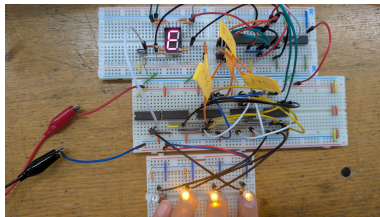
(b) $a = 1, b = 1, c = 0, d = 1$



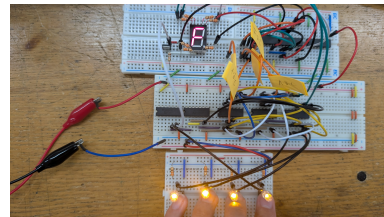
(c) $a = 0, b = 0, c = 1, d = 1$



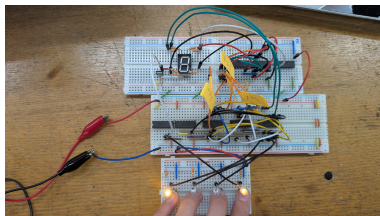
(d) $a = 1, b = 0, c = 1, d = 1$



(e) $a = 0, b = 1, c = 1, d = 1$



(f) $a = 1, b = 1, c = 1, d = 1$



(g) $a = 1, b = 0, c = 0, d = 1$

Fig. 4-6: 7 Segment Hexadecimal Decoder 10-15 Only (4 Bits BCD Input, LSB on Left)

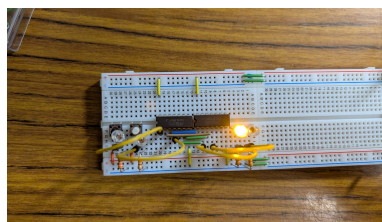
4.6.2 観測された真理値表

Table 4-6: Observed Truth Table of 7 Segment Hexadecimal Decoder (* Indicates Don't Care Term)

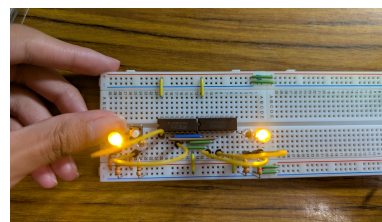
a	b	c	d	A	B	C	D	E	F	G
*	*	*	0	0	0	0	0	0	0	0
*	0	0	1	0	0	0	0	0	0	0
0	1	0	1	1	1	1	0	1	1	1
1	1	0	1	0	0	1	1	1	1	1
0	0	1	1	1	0	0	1	1	1	0
1	0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

4.7 課題 3-1

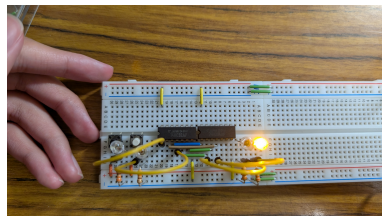
4.7.1 動作



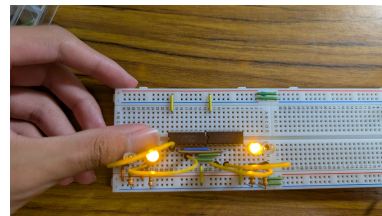
(a) Initial State



(b) Set Hold



(c) Set Release



(d) Reset Hold

Fig. 4-7: RS Latch with OR Gates and NOT Gates (LEDs Represent Set, Reset, Inverted Output, Output from Left)

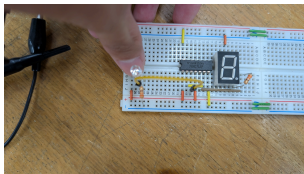
4.7.2 観測された挙動

Table 4-7: Observed State Transition Table of RS Latch

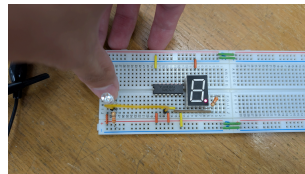
R	S	Q	$\overline{Q}$	Action
0	0	Q	$\overline{Q}$	Hold
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	X	Unstable Behaviour

4.8 課題 3-2

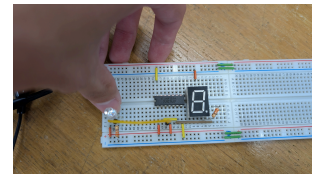
4.8.1 動作



(a) Initial State



(b) Clocked Once from Initial State



(c) Clocked Twice from Initial State

Fig. 4-8: D Flipflop Toggler

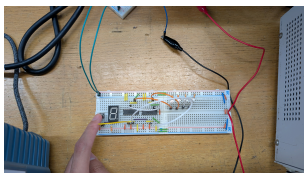
4.8.2 観測された挙動

Table 4-8: Observed State Transition Table of D Flipflop Toggler

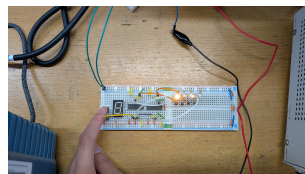
CK	Q	Q_{next}	Action
$\uparrow$	0	1	Toggle
$\uparrow$	1	0	Toggle
$\downarrow$	*	Q	Hold

4.9 課題 3-3

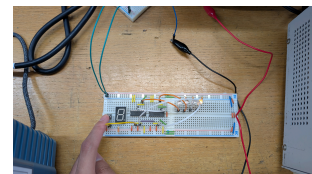
4.9.1 動作



(a) Initial State



(b) Clocked Six Times from Initial State



(c) Clocked Fifteen Times from Initial State

Fig. 4-9: 4 Bits D Flipflop Counter (LSB on Right)

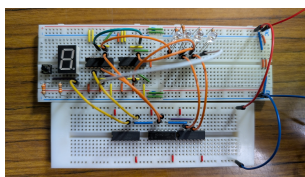
4.9.2 観測された挙動

Table 4-9: Observed State Transition Table of 4 Bits D Flipflop Counter

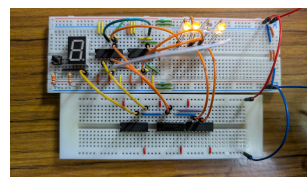
Q	CK	Q_{next}	D_A	D_B	D_C	D_D	Action
0	↑	1	0	0	0	0	Count
1	↑	2	1	1	1	1	Count
2	↑	3	0	1	1	1	Count
3	↑	4	1	0	1	1	Count
4	↑	5	0	0	1	1	Count
5	↑	6	1	1	0	1	Count
6	↑	7	0	1	0	1	Count
7	↑	8	1	0	0	1	Count
8	↑	9	0	0	0	1	Count
9	↑	10	1	1	1	0	Count
10	↑	11	0	1	1	0	Count
11	↑	12	1	0	1	0	Count
12	↑	13	0	0	1	0	Count
13	↑	14	1	1	0	0	Count
14	↑	15	0	1	0	0	Count
15	↑	0	1	0	0	0	Count
*	↓	Q	*	*	*	*	Hold

4.10 課題 3-4

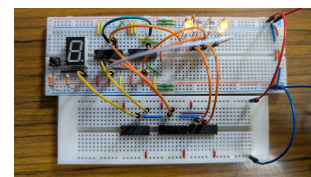
? には AND ゲートを使用した.



(a) Initial State



(b) Clocked 3 Times from Initial State



(c) Halt State

Fig. 4-10: 4 Bits D Flipflop Counter with Halt Condition (LSB on Right)

4.10.1 観測された挙動

Table 4-10: Observed State Transition Table of 4 Bits D Flipflop Counter with Halt Condition

Q	CK	Q_{next}	D_A	D_B	D_C	D_D	Action
0	↑	1	0	0	0	0	Count
1	↑	2	1	1	1	1	Count
2	↑	3	0	1	1	1	Count
3	↑	4	1	0	1	1	Count
4	↑	5	0	0	1	1	Count
5	↑	6	1	1	0	1	Count
6	↑	7	0	1	0	1	Count
7	↑	7	1	0	0	1	Halt
*	↓	Q	*	*	*	*	Hold

4.11 応用課題 A

Table 4-9 での状態 0 から状態 1 になる時の応答波形は Fig. 4-11 となった.

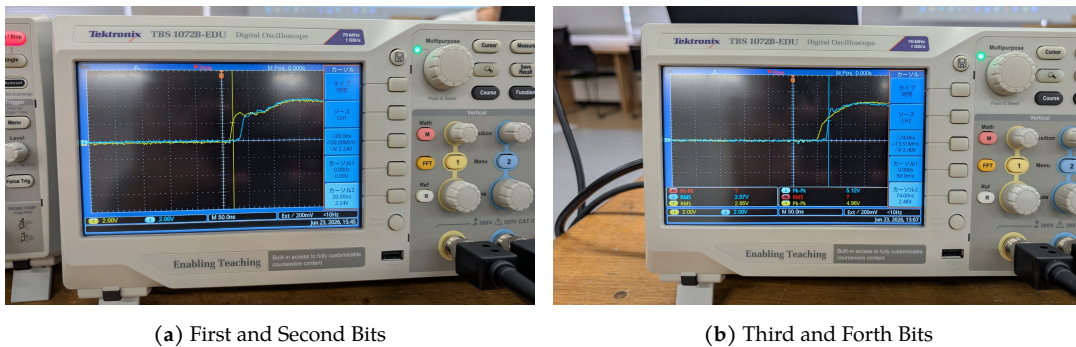


Fig. 4-11: Response Signals of Counter Output when State Transitions from 0 to 1

各出力が TC74HC74 の H レベルの入力電圧に到達するまでの時間を応答時間とし, それぞれの応答時間の概算は Table 4-11 となった.

Table 4-11: Response Time of Each Bit

Bit	Response Time (ns)
0	30
1	60
2	80
3	75

5 考察

5.1 課題 1-1

この実験では TC74HC08 の AND ゲートを使用した。

AND ゲートは等価回路は Fig. 5-1 のようにスイッチが 2 つ直列に接続された回路で, 両方のスイッチが閉じた時のみ, H レベルを出力する。

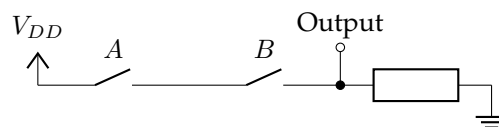


Fig. 5-1: AND Gate Equivalent Circuit

このことから, Table 4-1 で示した結果は両方の入力 H レベルの時だけ出力が H レベルとなっているため, 理論と一致することが分かる。

5.2 課題 1-2

Table 4-2 より, 全ての入力が H レベルの時のみ出力が L レベルとなっている。NAND ゲートは AND ゲートの否定なので, 結果は理屈に沿っている。

5.3 課題 1-3

Fig. 3-1 より, 各ゲートの出力は Eq. (5-1) となる。

$$\begin{aligned} O_1 &= A_{IN} \cdot B_{IN} \\ O_2 &= C_{IN} \cdot D_{IN} \\ D_p &= O_1 \cdot O_2 = (A_{IN} \cdot B_{IN}) \cdot (C_{IN} \cdot D_{IN}) \end{aligned} \tag{5-1}$$

Eq. (5-1) より, 真理値表は Table 5-1 となる。

Table 5-1: Theoretical Truth Table of $D_p = (A_{IN} \cdot B_{IN}) \cdot (C_{IN} \cdot D_{IN})$

A_{IN}	B_{IN}	C_{IN}	D_{IN}	D_p
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	0
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

Table 5-1 より, 出力 D_p が 1 となるのは全入力が 1 の時だけであり, この条件を論理式にすると Eq. (5-2) となり, Eq. (5-1) で示した D_p と等価な式になる.

$$d_p = A_{IN} \cdot B_{IN} \cdot C_{IN} \cdot D_{IN} \quad (5-2)$$

5.4 課題 1-4

Fig. 3-2 の出力は Eq. (5-3) で示され, ド・モルガンの定理で変形していくと Eq. (5-4) となる.

$$D_p = \overline{(A_{IN} \cdot B_{IN}) \cdot (C_{IN} \cdot D_{IN})} \quad (5-3)$$

$$\begin{aligned} D_p &= \overline{(A_{IN} \cdot B_{IN}) \cdot (C_{IN} \cdot D_{IN})} = \overline{A_{IN} \cdot B_{IN}} + \overline{C_{IN} \cdot D_{IN}} \\ &= \overline{A_{IN}} + \overline{B_{IN}} + \overline{C_{IN}} + \overline{D_{IN}} = \overline{A_{IN} \cdot B_{IN} \cdot C_{IN} \cdot D_{IN}} \end{aligned} \quad (5-4)$$

論理式の等価は真理値表の一致を意味するので, Eq. (5-4) で示した論理式の真理値表も全て一致する. Table 4-4 の真理値表は Table 4-2 のものと一致している, よってこれら 2 つの論理式は等価である.

5.5 課題 2-1

デコーダ回路とはある意味を持つ小さなコードを別の対応するより大きなコードへの写像を取る回路のことである. 今回の場合は 4 ビットの二進数から 7 セグメント LED の点灯パターンへの写像を論理回路で実装し

ている。

7セグメント LED デコーダの TC4511 はそれぞれの出力が論理式の出力となってる。

5.6 課題 2-2B

この回路を制作するにあたって, まずは Table 5-2 の真理値表を考える。

Table 5-2: Truth Table for Designing 7 Segment Hexadecimal Decoder

a	b	c	d	A	B	C	D	E	F	G
0	1	0	1	1	1	1	0	1	1	1
1	1	0	1	0	0	1	1	1	1	1
0	0	1	1	1	0	0	1	1	1	0
1	0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

ここで, d の値が全て H レベルであるので, 一旦省略する. この真理値表にもとづいたそれぞれのセグメントの論理式は Eq. (5-5) となる。

$$\begin{aligned}
 A &= \overline{(a \cdot b \cdot \bar{c}) + (a \cdot \bar{b} \cdot c)} \\
 B &= \overline{(\bar{a} \cdot b \cdot \bar{c}) + (a \cdot \bar{b} \cdot c)} \\
 C &= B + (a \cdot b \cdot \bar{c}) \\
 D &= \overline{(\bar{a} \cdot b \cdot \bar{c}) + (a \cdot b \cdot c)} \\
 E &= d \\
 F &= \overline{a \cdot \bar{b} \cdot c} \\
 G &= \overline{\bar{a} \cdot \bar{b} \cdot c}
 \end{aligned} \tag{5-5}$$

E セグメントには省略していた d の値を 1 との論理積として戻した. これら論理式にカルノー図を適応したが, もとが十分簡略化されていたためか, あまり有用ではなかった. かわりに, 複数式に共通する項を 1 つのユニットとして共有して使用することにした.

共通項は Eq. (5-6) とした.

$$\begin{aligned}
 X &= (a \cdot b \cdot \bar{c}) \\
 Y &= (a \cdot \bar{b} \cdot c) \\
 Z &= (\bar{a} \cdot b \cdot \bar{c})
 \end{aligned} \tag{5-6}$$

また, 禁止条件が入力された際の真理値値表は Table 5-3 となった.

Table 5-3: Truth Table for Disallowed Conditions

a	b	c	d	$\overline{EN}$
*	*	*	0	1
*	0	0	1	1

上記の真理値表から論理式を作成すると Eq. (5-7) となった.

$$EN = d \cdot (b + c) \quad (5-7)$$

この論理式の出力で Nch MOSFET を駆動させ, 7 セグメント LED のコモンカソードを制御することで一括で消灯させることにした.

これら論理式から回路を制作すると Fig. 5-2 となった.

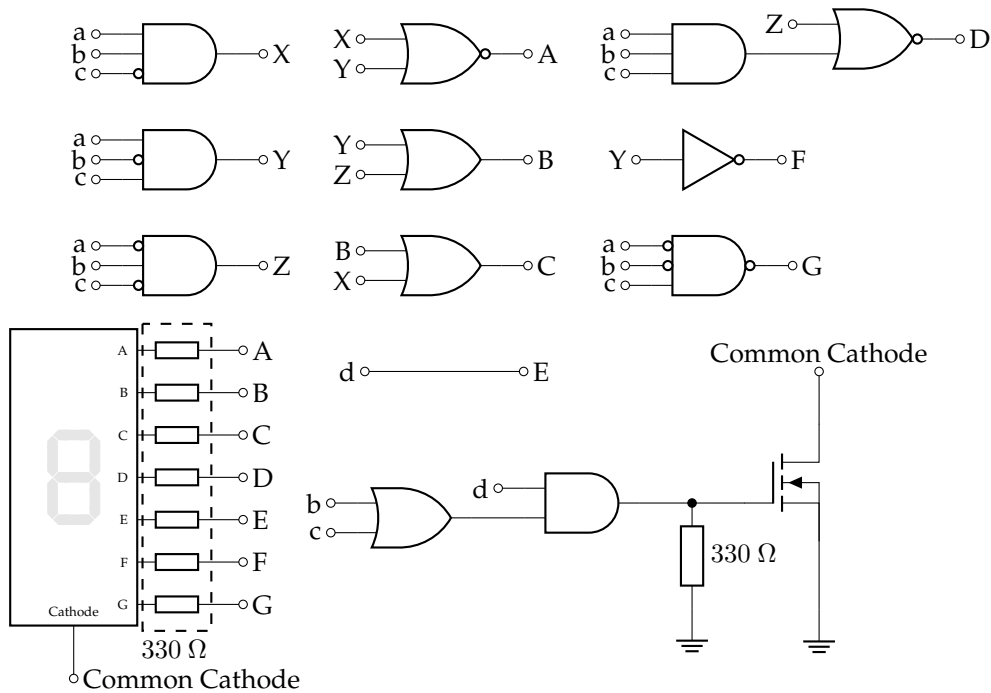


Fig. 5-2: Circuit Diagram of 7 Segment Hexadecimal Decoder

5.7 課題 3-1

今回の実験では Fig. 5-3 の状態遷移を行う NOR 構成の RS フリップフロップを作成した.

この構成では R と S 共に 1 が入力された時が禁止状態となる. 禁止状態では, 両方の NOR ゲートに 1 つ以上の H レベルの入力がかかり, 双方出力が 0 となりフリップフロップとしての機能を喪失する. どちらかの入力が L レベルになるまでこの機能は回復しない.

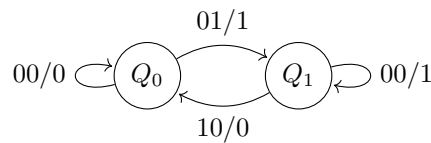


Fig. 5-3: State Transition Graph of RS Flipflop, the Edge Labels
Show Input R, Input S, and Output Q from Left

5.8 課題 3-3

今回の実験で作成したカウンタは二進数のビットごとの否定を出力するものとなった。これは初期状態では全ての出力 Q が L レベルで、次の状態で入力 D に印加されるのは全て否定出力 $\bar{Q}$ の H レベルとなるためである。

この回路は複数の D フリップフロップの出力が次のフリップフロップのクロック入力へ数珠接続した非同期のカウンタとなる。

5.9 課題 3-4

最下位の D フリップフロップのクロックをパルスさせるには? の論理ゲートの出力を NAND ゲートの入力関係無しにボタンからのオン・オフでトグルさせる必要がある。手元にあるゲートの中でこのクロックをパルスさせることが出来たのは AND ゲートであったが, NAND ゲートの出力が L レベルになるカウンタ出力の組み合わせで状態の遷移が停止してしまった。

これは AND ゲートの出力を H レベルにするには 2 つの入力が H レベルである必要があるが, NAND ゲートの出力が L レベルになってしまった以上, ボタンの出力のみで AND ゲートの出力を変えることが出来なくなってしまうことに起因する。

このカウンタを一巡させるには 1 つの H レベル入力のみで出力をパルスさせ, かつ両方が H レベルのときには出力を L レベルにすることが出来る論理ゲートが必要である。その論理ゲートこそ XOR (排他的論理和) である。この論理ゲートの真理値表は Table 5-4 で, 一方の入力が H レベルである時だけ出力を H レベルにすることが出来る⁸⁾。

Table 5-4: Truth Table of XOR Gate

a	b	x
0	0	0
1	0	1
0	1	1
1	1	0

5.10 応用課題 A

Fig. 4-11 と Table 4-11 より, 信号は最下位ビットから順番に立ち上がっている様子が分かる. しかし, 2^2 のビットは 2^3 のビットに遅れて TC74HC74 の H レベルの入力電圧である 3.15 V に到達した. これには IC 内部や, 入力端子の寄生容量が関係していると思われる⁹⁾.

これら遅延は非同期カウンタの性質上避けられない性質で, クロックとほぼ同時に状態が遷移する同期式のカウンタならこの遅延を少なくすることができる.

このことから, この回路で入力できる最大周波数の理論値は $\frac{1}{80\text{ ns}} = 12.5\text{ MHz}$ となる. 実際, ファンクションジェネレータで 16 MHz 付近の周波数をクロックに入力すると動作が不安定になった.

6 まとめ

今回の実験で以下の事実を確認した:

- 各種論理ゲートは組み合わせることで多種多様な動作をする回路を構築可能である
- ブール代数の諸法則・諸定理で論理回路の規模・複雑性を調整することができる
- 順序回路の動作の記述には状態遷移図・表が有効である

参考文献

- [1] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 7–11頁.
- [2] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 17–23頁.
- [3] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 49–54頁.
- [4] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 31–34頁.
- [5] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 55–61頁.
- [6] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 77–78頁.
- [7] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 101–106頁.
- [8] 秋田 純一. “ゼロから学ぶデジタル論理回路”. In: 株式会社講談社, 2005/06, 47–49頁.
- [9] TOSHIBA Semiconductor. “TC74HC74AP/AF”. 2014/03. URL: https://akizukidenshi.com/goodsaffix/TC74HC74AP_datasheet_ja_20140301.pdf.