Date: **21/01/2026** **CS29204**: Switching Circuits Laboratory

Assignment 2

Realizing gate-level circuits from truth table

Group 36
Dipam Sen (24CS10059)
Mitali Laddha (24CS10111)

Part (a)

Aim

To design, minimise and implement a combinatorial logic circuit that identifies whether a 4-bit binary number is prime, and to display the output using an external LED.

Apparatus

- Breadboard
- IC 7404 × 1 (TTL Hex Inverter)
- IC 7408 × 1 (TTL 2-input Quad AND Gate)
- IC 7432 × 1 (TTL 2-input Quad OR Gate)
- LED, Resistors
- Wires, wire cutter, tweezers.

Theory

We need to realise a function F that takes a four bit binary number $N = (A, B, C, D)$ as input, where A is the MSB and D is the LSB. F is 1 if N is prime, and 0 otherwise. We can draw the truth table for F as follows:

DEC	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Table 1: Truth table for F

To obtain a switching expression from the truth table, we can find the sum of the true minterms. This gives the canonical sum-of-products representation for F.

$$F = \Sigma(2, 3, 5, 7, 11, 13)$$
$$F = A'B'CD' + A'B'CD + A'BC'D + A'BCD + AB'CD + ABC'D$$

We can minimise the expression for F by using laws of switching algebra:

$$\begin{aligned}
F &= A'B'CD' + A'B'CD + A'BC'D + A'BCD + AB'CD + ABC'D \\
&= A'B'C(D' + D) + A'BD(C' + C) + AD(B'C + BC') \\
&= A'B'C + A'BD + AB'CD + ABC'D \\
&= B'C(A' + AD) + BD(A' + AC') \\
&= B'C(A' + D) + BD(A' + C') \\
&= A'B'C + B'CD + A'BD + BC'D
\end{aligned}$$

In practice, we would like to implement F using 2-input gates and using as few gates as possible. We can do so by realising the expression $F = B'C(A' + D) + BD(A' + C')$ using 2-input gates.

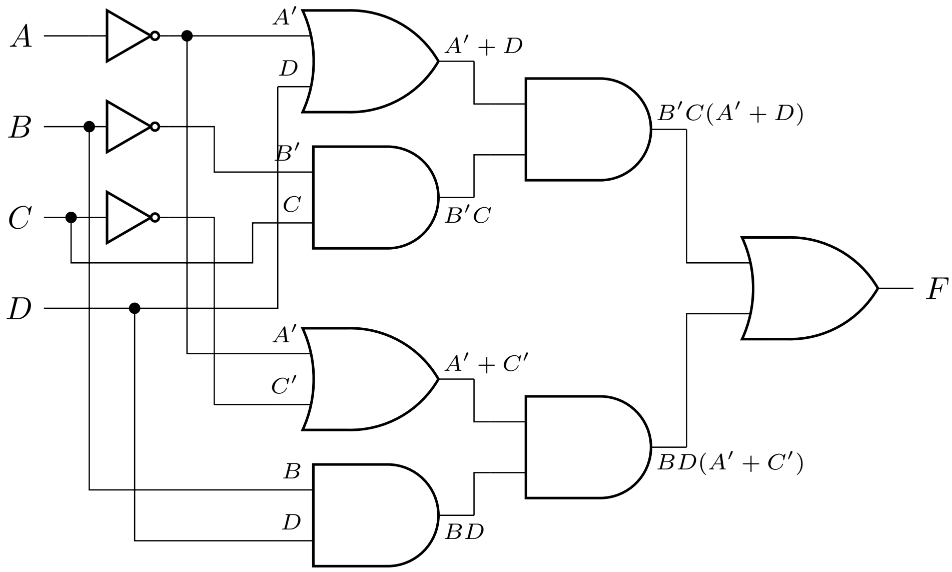

Figure 1: Gate level realisation of $F = B'C(A' + D) + BD(A' + C')$

The above realisation requires 3 NOT gates, 3 2-input OR gates and 4 2-input AND gates.

If we realise this in a circuit, we obtain the output F as a voltage level. In order to display the output, we need to connect an external LED. To limit the current flow in the LED, we have to use a current limiting resistor in series with the LED.

If a substantial amount of current is drawn from a TTL gate output, the voltage level is not maintained. Conversely, a TTL gate can sink a considerable amount of current (when it returns a 0 value), so we will display the output as **active-low**: The LED will glow when the output is 0, and it will not glow when the output is 1. Figure 2 shows the circuit for this configuration.

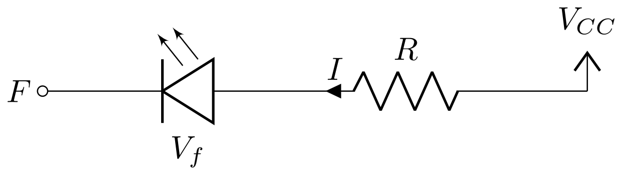

Figure 2: Displaying the output on an LED

The voltage drop across a forward biased LED is typically 1.7 to 3.3 volts. We can use an approximate value of $V_f = 2\,\text{V}$.

To limit the current I to around $5\,\text{mA}$, we can find the R required as follows:

$$I = \frac{V_{CC} - V_f}{R}$$

$$\Rightarrow \frac{V_{CC} - V_f}{R} = 5\,\text{mA}$$

$$\Rightarrow R = \frac{5\,\text{V} - 2\,\text{V}}{5\,\text{mA}} = 600\,\Omega$$

From the available resistance values, we can use 2 resistances of $330\,\Omega$ each to get a total resistance of $660\,\Omega$.

Circuit Diagram

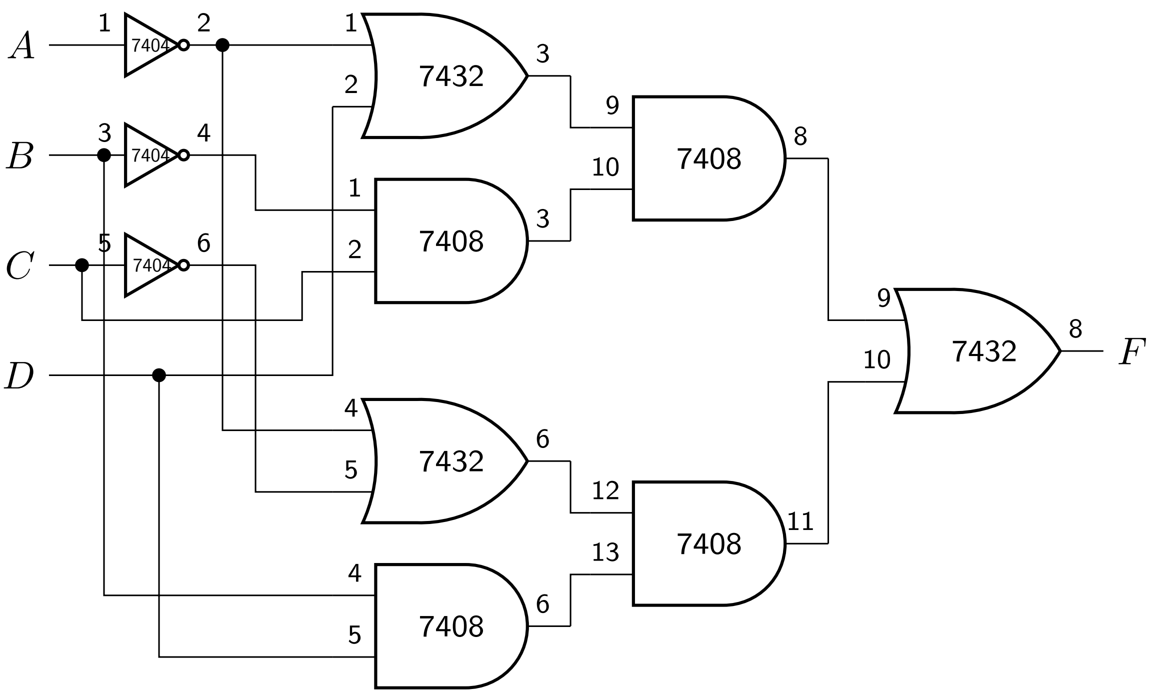

Figure 3: Combinatorial circuit for checking whether a 4 bit number is prime

Procedure

1. Draw the truth table for the function F.
2. Write the expression for F and simplify it to obtain a minimal expression.
3. Draw the circuit diagram for a gate level realisation from the minimal expression, using basic gates.
4. Connect the circuit (Figure 3) on the breadboard using corresponding TTL ICs.
5. Connect the output of the circuit to an LED, as shown in Figure 2. Use the calculated value of resistance.
6. Verify the truth table by testing all input combinations and checking the output.

Result

The circuit correctly identifies whether the input number is prime or not, and displays the output on the LED.

Discussion

From the truth table we obtained the canonical sum-of-products form for F, and then simplified it by using the laws of switching algebra to get a minimal representation which can be implemented by using as few gates as possible. We also used only 2-input gates for this realisation.

Part (b)

Aim

To design and realise a combinatorial circuit that inputs a BCD number and outputs $(x + 4)$ mod 10 using only 2-input NAND gates, and to display both the input and output using 7-segment LED display units.

Apparatus

- Breadboard
- IC 7400 × 3 (TTL 2-input Quad NAND Gate)
- IC 7447 × 2 (BCD to 7 segment decoder)
- 7 Segment Display Unit × 2
- Resistors
- Wires, wire cutter, tweezers

Theory

We need to design a circuit that takes a 4-bit number $A = A_3A_2A_1A_0$ as input, and returns $B = B_3B_2B_1B_0$ as output, where $B = (A + 4) \bmod 10$. We can draw the truth table for the circuit as shown in Table 2.

A	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	B
0	0	0	0	0	0	1	0	0	4
1	0	0	0	1	0	1	0	1	5
2	0	0	1	0	0	1	1	0	6
3	0	0	1	1	0	1	1	1	7
4	0	1	0	0	1	0	0	0	8
5	0	1	0	1	1	0	0	1	9
6	0	1	1	0	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1
8	1	0	0	0	0	0	1	0	2
9	1	0	0	1	0	0	1	1	3

Table 2: Truth table for the circuit (b)

By observation, we can obtain the following expressions for B_3, B_2, B_1, B_0 which gives the required values for the above input combinations:

$$
\begin{aligned}
B_3 &= A_2A_1' \\
B_2 &= A_3'A_2' \\
B_1 &= A_2'A_1 + A_3 \\
B_0 &= A_0
\end{aligned}
$$

We need to realise a circuit for these values by only using 2-input NAND gates. We can do so, since NAND is an universal gate, and all of NOT, AND and OR can be build from NAND gates.

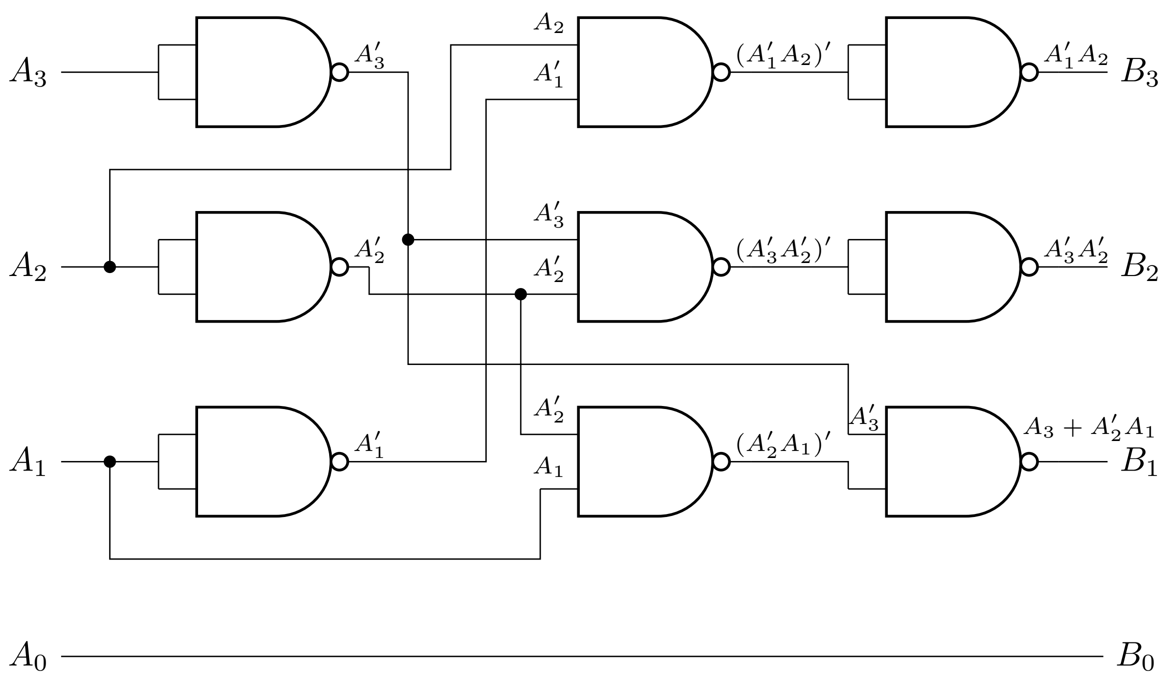

Figure 4: 2-input NAND-gate realisation of the functions

The above realisation requires 9 NAND gates. Since each IC 7400 has 4 NAND gates, we need 3 such chips.

We also need to display the input and the output on a seven segment display. We will use a common anode 7-Segment display unit, meaning that the anodes of all the LEDs in the display unit are common, and to illuminate a segment, we ground the corresponding pin. So, the inputs to the display unit need to be **active-low**, meaning that to indicate that a segment should glow, the corresponding input should be a low voltage (GND).

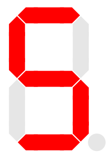

Figure 5: Seven segment display

The IC 7447 provides a BCD to 7 Segment decoder which gives active-low output, i.e. low voltage to indicate that the corresponding segment should be on, and vice versa. We can use the BCD number as an input to the decoder, and get 7 outputs, one corresponding to each segment.

We will use resistors in series connecting the outputs of the IC 7447 to the 7 Segment Display to limit the current flowing through the LEDs.

Circuit Diagrams

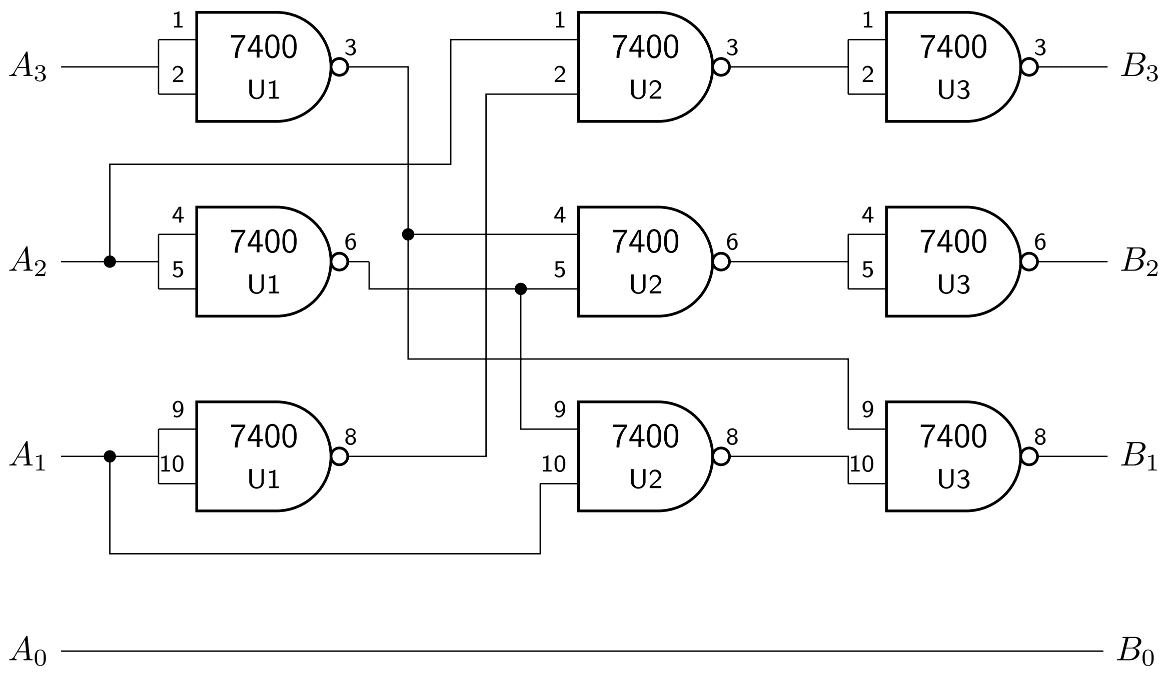

Figure 6: Combinatorial circuit for taking an BCD number A as input and finding $B = (A + 4)\%10$

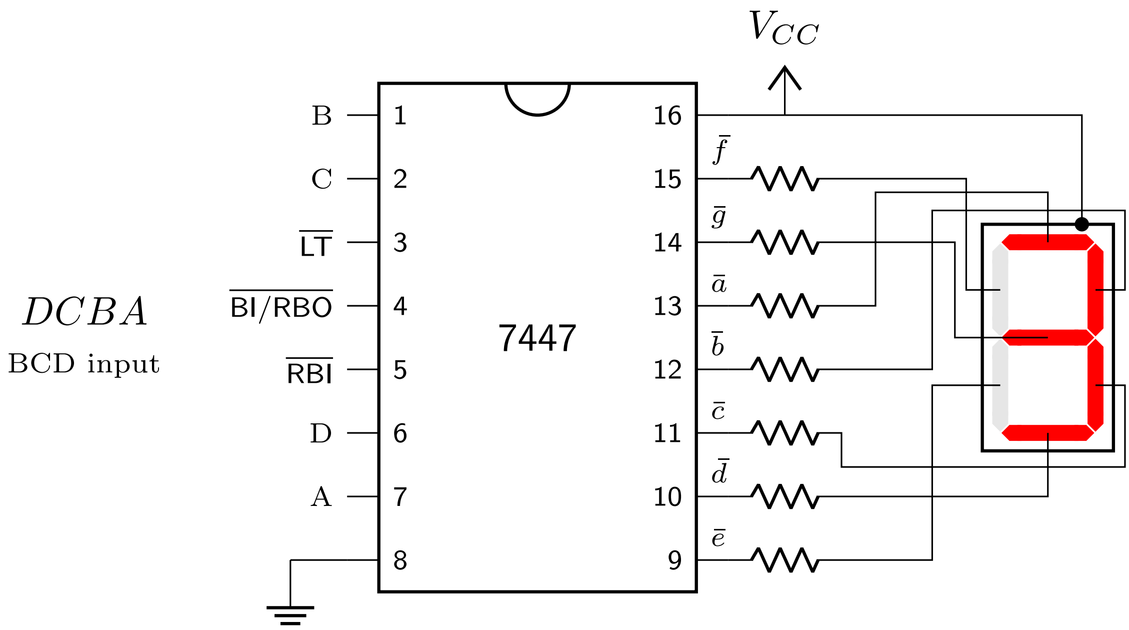

Figure 7: Connection diagram for displaying a BCD number on a seven segment display

Procedure

1. Draw the truth table for the problem.
2. For each output, obtain a simplified logic expression.
3. Draw a circuit diagram for the gate level realisations of the logic expressions, using only 2-input NAND gates.
4. Connect the circuit (Figure 6) on the breadboard, using the IC 7400 for NAND gates. Connect the 4 inputs to the circuit.
5. Connect the input and output to the BCD to 7 Segment decoder and then to the 7 segment display unit, as shown in Figure 7.
6. For each BCD input $(0 \text{ to } 9)$, verify the output shown on the display.

Result

The circuit computes the value $B = (A + 4)\%10$, and displays the input and output on the seven segment displays correctly.

Discussion

Here, since our input is 4-bit BCD, there are certain input combinations which are unused. This means, that for our circuit, it does not matter what value our circuit gives for those input combinations.

We can exploit this flexibility to come up with more simplified expressions for the functions, which leads to a more efficient gate level realisation. For example, for B_0, the minimised function which we obtain from the truth table (assuming 0 output for the unused combinations) is $B_0 = A_0(A_3' + A_2'A_1')$. But, by observation we can immediately say that $B_0 = A_0$, for the specific input combinations that are being used. (The operation $(A + 4)\%10$ does not affect the LSB.)

The IC 7447 gives the output as active-low, so that it can be used with the common cathode display unit. In the display unit, we connect the common pin to V_{CC}, and the corresponding segment pins to the decoder output. The decoder output pins draw current by pulling the voltage low, thus forward biasing the LEDs, and turning them on.