**Introduction**

I’m tasked with designing a logic circuit for a cooking robot. I shall offer dish designs including their ingredient combinations and then create truth tables for each of them.

Furthermore, I shall design the logic circuit; to do so, I will derive and simplify the Boolean expressions of each truth table to get a simple and functional design.

I have used AI to get a overview of the steps needed to do the coursework and to find popular dishes but the mathematical parts, Boolean expressions and simplification and the logic circuits were done by me.

Dish Design:

I’ll now introduce the 8 different dishes and the ingredients necessary to compose said dishes. I decided to keep most of the ingredients in the example and look for popular dishes with the given ingredients.

Dish 0: Veggie Fried Rice – rice, soy sauce, spring onion, egg, vegetables

Dish 1: Stir-fried Noodles – noodles, soy sauce, spring onions, pepper, vegetables

Dish 2: Sweet and Sour Tofu – tofu, noodles, soy sauce, spring onions, pepper, vegetables

Dish 3: Beef and broccoli – beef, soy sauce, spring onion, pepper

Dish 4: Egg Fried Rice – rice, egg, soy sauce, spring onion, pepper

Dish 5: Spicy Pork Noodles – pork, noodles, soy sauce, spring onion, pepper

Dish 6: Fish and Rice – fish, rice, soy sauce, spring onion, pepper

Dish 7: Vegetable Stir-fry – Tofu, soy sauce, spring onion, pepper, vegetables

Truth Tables:

Now for each ingredient, I must design a truth table. The input dishes shall consist of number in binary going from 0 to 7 (A, B, C).

Truth table for Y1 (Rice)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y1 (Rice) |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Truth table for Y2 (Noodles)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y2 (Noodles) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Truth table for Y3 (Soy Sauce)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y3 (Soy Sauce) |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Truth table for Y4 (Spring Onion)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y4 (Spring Onion) |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Truth table for Y5 (Pepper)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y5 (Pepper) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Truth table for Y6 (Egg)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y6 (Egg) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Truth table for Y7 (Tofu)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y7 (Tofu) |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Truth table for Y8 (Pork)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y8 (Pork) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Truth table for Y9 (Fish)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y9 (Fish) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Truth table for Y10 (Vegetables)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Y10 (Vegetables) |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Deriving the Boolean Expressions

Now that the truth tables are done for each output, I can derive for each ingredient using Sum-of-Products, thus by finding out for which rows each ingredient = 1.

For Y1:

From the truth table for Rice (Y1), the rows where Y1 = 1 are:

* A’B’C
* A’BC
* AB’C’

Thus the Boolean expression is: Y1 = A’B’C + A’BC + AB’C’

For Y2:

The rows are:

* A’BC’
* A’BC
* AB’C’
* AB’C

Thus the Boolean expression is: Y2 = A’BC’ + A’BC + AB’C’ + AB’C

So following the same process for each of them would give:

Y3 = A’ + B

Y4 = 1 (because every output give out Y4 = 1)

Y5 = A’BC’ + A’BC + AB’C’ + AB’C + ABC

Y6 = A’BC + ABC

Y7 = A’B’C + A’BC’ + A’BC

Y8 = AB’C’ + AB’C + ABC’

Y9 = AB’C

Y10 = A’B’C + A’BC + AB’C’ + ABC’

Simplification

After simplification we get more concise answers:

Y1 = A’B + AB’C’

Y2 = A’B + AB’

Y3 = A’ + B

Y4 = 1 (no simplification needed)

Y5 = A’B + AB’

Y6 = BC

Y7 = A’B

Y8 = AB’

Y9 = AB’C

Y10 = A’B + AB’

Logic Circuits:

Now that all of this is done we can finally design the logic circuits and announce what is needed for each of them;

**1. Logic circuit for Y1 (Rice)**

Simplified Boolean expression:

Y1=A′B+AB′C′

To design the circuit :

1. **Inverters**: You need two inverters (NOT gates):
   * One for A to get A′.
   * One for C to get C′.
2. **AND gates**:
   * First AND gate for A′B (inputs: A′, B).
   * Second AND gate for AB′C′ (inputs: A, B′, C′).
3. **OR gate**:
   * The outputs of the two AND gates are fed into an OR gate to obtain Y1.

**2. Logic circuit for Y2 (noodles)**

Simplified Boolean expression :

Y2=A′B+AB′

To design the circuit :

1. **Inverters**:
   * An inverter for A to obtain A′.
2. **AND gates**:
   * First AND gate for A′B (inputs: A′, B).
   * Second AND gate for AB′ (inputs: A, B′).
3. **OR gate**:
   * The outputs of the two AND gates are fed into an OR gate to obtain Y2.

**3. Logic circuit for Y3 (soy sauce)**

Simplified Boolean expression :

Y3=A′+B

Circuit design :

1. **Inverter**:
   * An inverter for A to obtain A′.
2. **OR gate**:
   * The two inputs of the OR gate are A′ and B.

Thus, Y3 is the output of an OR gate with inputs A′ and B.

**4. Logic circuit for Y4 (spring onion)**

Simplified Boolean expression :

Y4=1

To design the circuit:

* Y4 is always equal to 1, so it can be represented by simply connecting the output directly to a constant “high” value (logic 1).
* No gates are needed, just connect the output to logic 1.

**5. Logic circuit for Y5 (Pepper)**

Simplified Boolean expression:

Y5=A′B+AB′

Design the circuit:

1. **Inverter**:
   * An inverter for A to obtain A′.
2. **AND gates**:
   * First AND gate for A′B (inputs: A′, B).
   * Second AND gate for AB′ (inputs: A, B′).
3. **OR gate**:
   * The outputs of the two AND gates are fed into an OR gate to obtain Y5.

**6. Logic circuit for Y6 (egg)**

Simplified Boolean expression :

Y6=BC

Circuit design :

1. **AND gate**:
   * One AND gate for BC (inputs: B, C).

Thus, Y6 is the output of a single AND gate.

**7. Logic circuit for Y7 (Tofu)**

Simplified Boolean expression :

Y7=A′B

Design the circuit:

1. **Inverter**:
   * An inverter for A to obtain A′.
2. **AND gate**:
   * An AND gate for A′B (inputs: A′, B).

Thus, Y7 is the output of a single AND gate.

**8. Logic circuit for Y8 (pig)**

Simplified Boolean expression :

Y8=AB′

Design the circuit:

1. **Inverter**:
   * An inverter for B to obtain B′.
2. **AND gate**:
   * An AND gate for AB′ (inputs: A, B′).

Thus, Y8 is the output of a single AND gate.

**9. Logic circuit for Y9 (fish)**

Simplified Boolean expression :

Y9=AB′C

To design the circuit :

1. **Inverter**:
   * An inverter for B to obtain B′.
2. **AND gate**:
   * An AND gate for AB′C (inputs: A, B′, C).

Thus, Y9 is the output of a single AND gate.

**10. Logic circuit for Y10 (soybean sprouts)**

Simplified Boolean expression:

Y10=A′B+AB′

Design the circuit:

1. **Inverter**:
   * An inverter for A to obtain A′.
2. **AND gates**:
   * First AND gate for A′B (inputs: A′, B).
   * Second AND gate for AB′ (inputs: A, B′).
3. **OR gate**:
   * The outputs of the two AND gates are fed into an OR gate to obtain Y10.