# Practice Questions

CASEST

हैदराबाद विश्वविद्यालय
University of Hyderabad

**Lab Report No. 3**

**MV407**: IC Design Lab-1 (Digital)


**Submitted by:**

| Full Name | Enrollment No. |
|---|---|
| Abhinav M | 23PMMT11(@uohyd.ac.in) |

Lab Instructor: Dr. Bhawna Gomber

Teaching Assistant: Ms. Bhargavi



**Center for Advanced Studies in Electronics Science and Technology**
**University of Hyderabad**
**Hyderabad, India**
**October 10, 2023**

# Contents

# 1 | Sum of Dice

## 1.1 | Problem statement

Three dice are rolled simultaneously. For each die, let a binary signal represent whether the outcome is even or odd. Design a circuit whose output represents whether the sum of the outcomes of the three dice is even or odd. Verify its functionality using testbench.

## 1.2 | Truth table

The truth table can be used to derive the Boolean expression:

| $D_1$ | $D_2$ | $D_3$ | Y (Odd TRUE) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## 1.3 | K map and Reduced Boolean expression

| $D_1$ \ $D_2D_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

$$Y = D_1 \oplus D_2 \oplus D_3$$

## 1.4 | Code

### 1.4.1 | Design

```
1
2  `timescale 1ns / 1ps
3  //////////////////////////////////////////////////////////////////////////////////
4  // Company: CASEST, University of Hyderabad
5  // Author: Abhinav M
6  //
7  // Create Date: 10/05/2023 11:58:02 AM
8  // Design Name:
9  // Module Name: three_dice
10 // Project Name:
11 // Target Devices:
12 // Tool Versions:
13 // Description:
14 //
15 // Dependencies:
16 //
17 // Revision:
18 // Revision 0.01 - File Created
19 // Additional Comments:
20 //
21 //////////////////////////////////////////////////////////////////////////////////
```

```
22
23
24  module three_dice(
25      output y,
26      input d1,
27      input d2,
28      input d3
29      );
30
31      assign y=d1^d2^d3;
32
33  endmodule
```

### 1.4.2 | Testbench

```
1
2   `timescale 1ns / 1ps
3   //////////////////////////////////////////////////////////////////////////////////
4   // Company: CASEST, University of Hyderabad
5   // Author: Abhinav M
6   //
7   // Create Date: 10/05/2023 12:10:38 PM
8   // Design Name:
9   // Module Name: tb_threedice
10  // Project Name:
11  // Target Devices:
12  // Tool Versions:
13  // Description:
14  //
15  // Dependencies:
16  //
17  // Revision:
18  // Revision 0.01 - File Created
19  // Additional Comments:
20  //
21  //////////////////////////////////////////////////////////////////////////////////
22
23
24  module tb_threedice();
25
26  reg d1,d2,d3;
27  wire y;
28
29  three_dice uut(y,d1,d2,d3);
30
31
32  initial
33      begin
34
35     #10 d1=0;d2=0;d3=0;
36     #10 d1=0;d2=0;d3=1;
37     #10 d1=0;d2=1;d3=0;
38     #10 d1=1;d2=0;d3=0;
39     #10 d1=1;d2=1;d3=0;
40     #10 d1=0;d2=1;d3=1;
41     #10 d1=1;d2=0;d3=1;
42     #10 d1=1;d2=1;d3=1;
43
44      #10 $finish;
45      end
46
47  endmodule
```

## 1.5 | Elaborated design

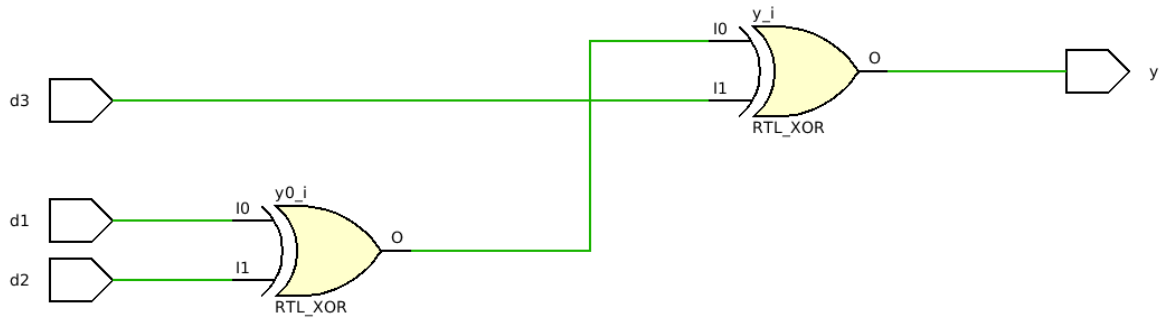


**Figure 1.1:** Elaborated design.

## 1.6 | Timing diagram

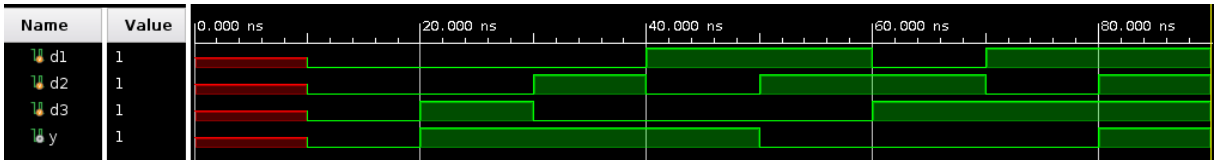


**Figure 1.2:** Timing diagram.

## 1.7 | Remarks

Circuit for the problem statement was designed and tested successfully.

# 2 | Snack Diplomacy

## 2.1 | Problem statement

A person “A” likes sandwich, pastry and ice-cream, person “B” likes sweetcorn and ice-cream, person “C” likes sandwich, pastry and sweetcorn. They have enough money to buy two items. Write a module which can determine, whether a choice of food items would contain at-least one food of liking for each of them. (Assign 0-1 values to each food item).

## 2.2 | Truth table

The truth table can be used to derive the Boolean expression:
(Let, $S_a$: Sandwich, $S_w$: Sweetcorn, I: Ice-cream, P: Pastry)

| $S_a$ | $S_w$ | $I$ | $P$ | Y |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

## 2.3 | K map -creamand reduced Boolean expression

| $S_aS_w$ \ $IP$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 1 |

$$Y = \overline{S_a}\,\overline{S_w}IP + S_a\overline{S_w}I\overline{P} + \overline{S_a}S_w\overline{I}P + \overline{S_a}S_wI\overline{P} + S_aS_w\overline{I}\,\overline{P}$$

## 2.4 | Code

### 2.4.1 | Design

```
1
2  `timescale 1ns / 1ps
3  //////////////////////////////////////////////////////////////////////////////////
4  // Company: CASEST, University of Hyderabad
5  // Author: Abhinav M
6  //
7  // Create Date: 10/05/2023 12:00:18 PM
8  // Design Name:
9  // Module Name: icecream
10 // Project Name:
11 // Target Devices:
12 // Tool Versions:
13 // Description:
```

```
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module icecream(output y,
    input Sa,
    input Sw,
    input I,
    input P

    );
    wire w1,w2,w3,w4,w5;

    and a1(w1,Sa,Sw,~I,~P);
    and a2(w2,~Sa,Sw,~I,P);
    and a3(w3,~Sa,~Sw,I,P);
    and a4(w4,~Sa,Sw,I,~P);
    and a5(w5,Sa,~Sw,I,~P);

    //assign y = (Sa)&(Sw)&(~I)&(~P) + (~Sa)&(Sw)&(~I)&(P) + (~Sa)&(~Sw)&(I)&(P) + (~Sa)&(Sw)&(I)&(~P);

    or o1(y,w1,w2,w3,w4,w5);

endmodule
```

### 2.4.2 | Testbench

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 10/05/2023 12:02:26 PM
// Design Name:
// Module Name: tb_icecream
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_icecream();

reg Sa,Sw,I,P;
wire y;

icecream uut(y,Sa,Sw,I,P);

initial
    begin

    #10 Sa=1;Sw=1;I=0;P=0;
    #10 Sa=1;Sw=0;I=1;P=0;
    #10 Sa=0;Sw=1;I=1;P=0;
    #10 Sa=0;Sw=0;I=1;P=1;
    #10 Sa=0;Sw=0;I=0;P=1;


    #10 $finish;
    end

endmodule
```

## 2.5 | Elaborated design

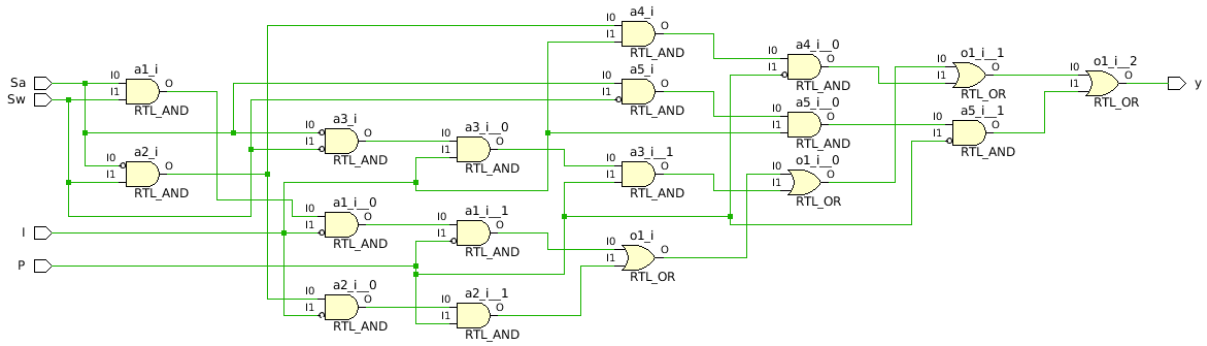


**Figure 2.1:** Elaborated design.

## 2.6 | Timing diagram

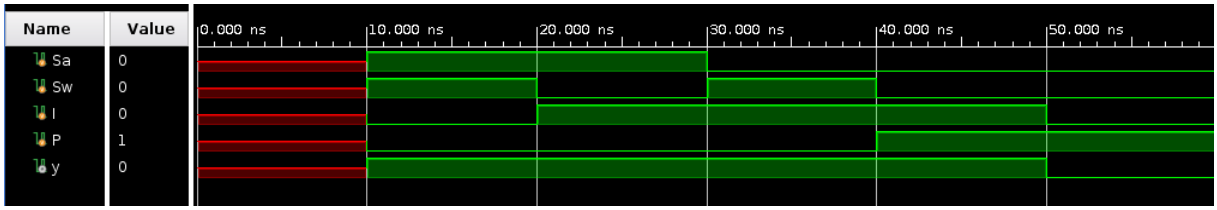


**Figure 2.2:** Timing diagram.

## 2.7 | Remarks

Circuit for the problem statement was designed and tested successfully.

# 3 | References

# A | Appendix A: GitHub Repo

The GitHub repo referenced below contains the Verilog code and testbenches mentioned in this report.
https://github.com/AbhinavM2000/MV401D/