Digital System Design Capsule 2024 Spring

PROJECT 1: Uneven Seven-Sided Dice Roller

Objective:

- To design and implement an uneven seven-sided dice roller.

In this project, you are asked to design and implement an uneven seven-sided dice roller (Fig. 1) of the following description. The system is composed of a 1-bit data generator, a 2-bit full-adder, and two registers (R_I: intermediate; R_O: output). Among two bits of the full-adder, one bit must be implemented at the *transistor* level in CMOS configuration. You will select your own full-adder circuit. The rest of the system can be implemented at the *gate* level.

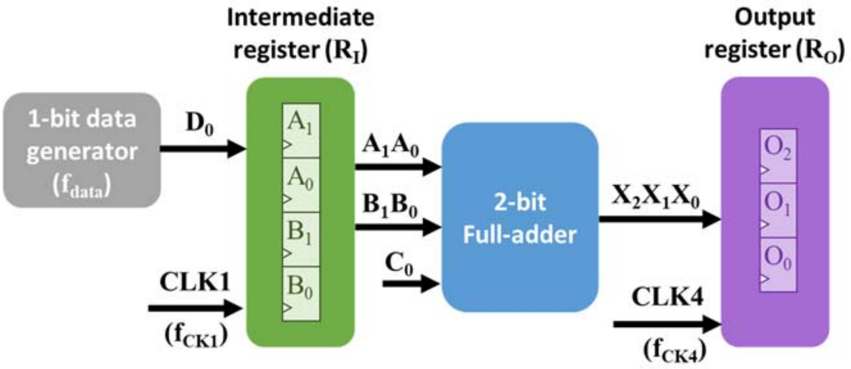
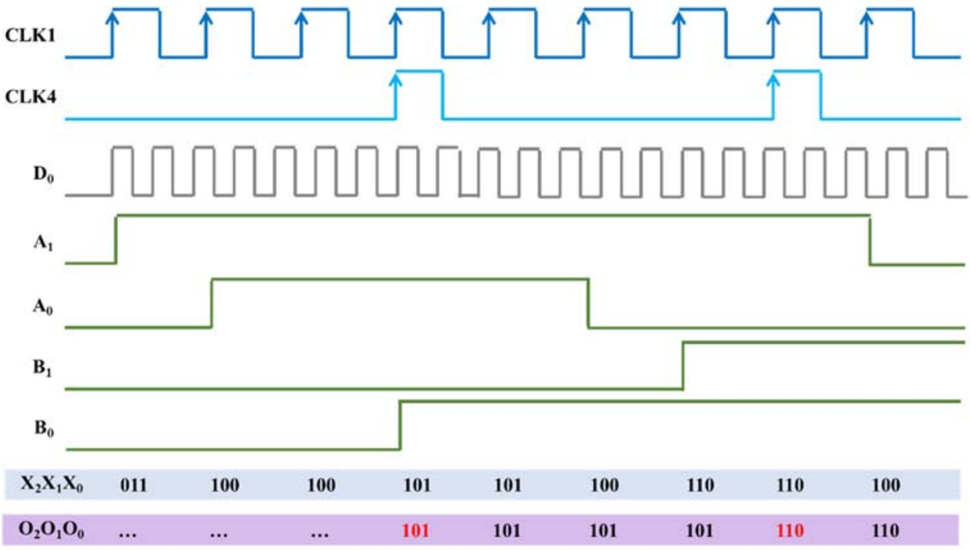


Fig. 1. Schematic diagram of an uneven seven-sided dice roller.



$X_2X_1X_0$	011	100	100	101	101	100	110	110	100
$O_2O_1O_0$	…	…	…	101	101	101	101	110	110

Fig. 2. Exemplary timing diagram.

- Function of the system
  - The circuit produces a random 3-bit number (decimal equivalent: from 1 to 7) at the frequency of f_{ck4} (= 100 Hz). This number (**O**: $O_2O_1O_0$) is stored in the output register (R_O), and refreshed at f_{ck4}. This number is to be read by a microcontroller (e.g. Arduino).
    - CLK4 is to be generated by processing CLK1 (f_{ck1} = 400 Hz) via a circuit you design. You have to come up with an idea about this design.
    - CLK1 is to be produced by a microcontroller (e.g. Arduino).
  - Input to R_O, **X** (= $X_2X_1X_0$), is produced by adding two 2-bit numbers, **A** (= A_1A_0) and **B** (= B_1B_0), and a carry (C_0). C_0 is always '1' in this system.
    - Example. **A** = 01, **B** = 10, then **X** = 100. See Fig. 2.
  - The data (D_0) is to be generated by a circuit of your design (e.g. an oscillator, a multivibrator, a timer, etc) as a pulse signal with the frequency of f_{data}. It is important that f_{data} is **not** synchronized to f_{ck4} so that a random data is generated by the system.

  - D_0 is distributed to and stored in the intermediate register (R_I) as shown in Fig. 2.
    - Distribution needs to be managed by the CLK1.

CLK1	1st ↑	2nd ↑	3rd ↑	4th ↑	5th ↑	…
A_1	D_0	NC	NC	NC	D_0	…
A_0	NC	D_0	NC	NC	NC	…
B_1	NC	NC	D_0	NC	NC	…
B_0	NC	NC	NC	D_0	NC	…

Note. NC: no change

    - You have to come up with a design to realize the distribution of D_0 as described above.
  - System input: CLK1
  - System output: 3-bit data **O**

- Task details
  - Task 1: Data (D_0) generator
    - Design (simulation) and build a pulse generator for D_0.
    - You may use a ring oscillator, a multivibrator, a timer, etc. However, you have to design and build a circuit. You must not realize this function just by using a single IC chip. Also, you should not produce this pulse from a microcontroller.
    - $10f_{ck1} < f_{data} < 100f_{ck1}$.
    - The output should be a proper square wave.
      - Low voltage level: ground
      - High voltage level: 5 V
    - **Report** : 3 March (S), 11:59 PM
      - Max. 3 pages, no template
      - Circuit diagram, explanation, simulation results, etc.
    - Build the circuit physically.
    - **Demonstration**: 8 March (F), Lab hour
  - Task 2: Full adder
    - Select your own full-adder circuit in a CMOS configuration.
    - Use wisely the fact that C_0 is always '1'.
    - Run SPICE simulation.
    - Use the proper device models, which will be used for the circuit implementation.
    - **Report** (max. 5 pages, no template): 24 March (S), 11:59 PM
      - Circuit diagram, explanation, simulation results, etc.
    - Build the circuit physically.

        - **Demonstration**: 29 March (F), Lab hour
            - One bit full-adder is enough for this demonstration.
            - For the other bit, you may build it at the transistor level or at the gate level. However, beware that the carry-in for this adder is not fixed as ‘1’.
    - Task 3: Generation of signals **A** and **B**
        - CLK1: produced by microcontroller
        - Intermediate register (R_I): can be implemented with flip-flops.
        - Design a circuit to distribute D_0 to R_I.
        - D_0 from Task 1 should be used as the data input.
        - Timing diagram (Fig. 2) needs to be plotted on a screen.
            - CLK1, D_0, **A** and **B**.
        - **Demonstration**: 19 April (F), Lab hour
    - Final task: Entire system
        - Combine all of the components to demonstrate the function of the system.
        - **O** signal will be collected by a microcontroller, and the distribution of the numbers will be examined.
            - You need to check the distribution of the numbers, and compare it to the theoretical one according to the (uneven) probability. Report this in the final report.
        - **Demonstration**: 26 April (F), Lab hour
        - **Final report** (max. 10 pages, *template* given below): 28 April (S), 11:59 PM

- Normally, this project will be carried out as a group of two.
- **IMPORTANT**. You will have to provide all of the electronic components and breadboards for your own project.
- **IMPORTANT**. No late submission/demonstration will be accepted except for a medical excuse or for a family emergency.

Table I. Tasks, weights, formats and deadlines

Task	**Weight**	**Format**	**Deadline**
Task 1: Data generator	5%	Report	11:59 PM, 3. 3.
	10%	Demonstration	Lab hour, 8. 3.
Task 2: Full adder	15%	Report	11:59 PM, 24. 3.
	15%	Demonstration	Lab hour, 29. 3.
Task 3: Generation of **A** and **B**	20%	Demonstration	Lab hour, 19. 4.
Final task	30%	Demonstration	Lab hour, 26. 4.
	5%	*Report	11:59 PM, 28. 4.

*You must use the following template for the final report:
https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/conference-template-a4.docx