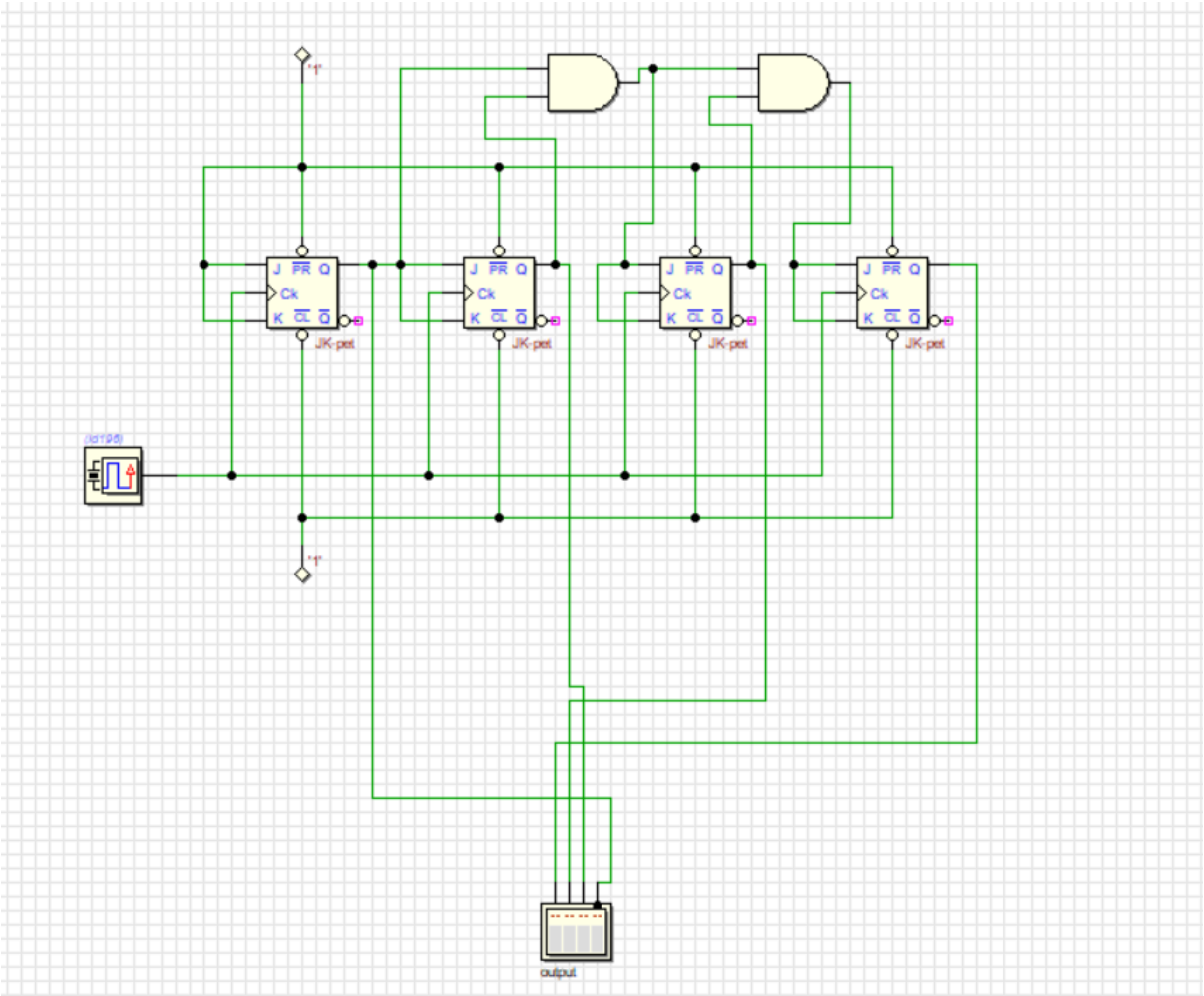
4-bit synchronous up-counter

PS:

A 4-bit synchronous up-counter (counts 0 → 1 → 2 → … → 15 → 0) using Deeds software.

Solution:



Objective:
To design and simulate a 4-bit synchronous up counter using JK flip-flops.

Circuit Overview:

- The circuit uses 4 JK flip-flops connected to a common clock, making it a synchronous counter.
- Each flip-flop toggles based on logic conditions set using AND gates.
- The output increases by 1 on each rising clock edge, from `0000` (0) to `1111` (15).

Bit Description:

- Q0 is the Least Significant Bit (LSB) toggles on every clock pulse.
- Q1 toggles when Q0 = 1.
- Q2 toggles when Q0 & Q1 = 1.
- Q3 is the Most Significant Bit (MSB) toggles when Q0, Q1 & Q2 = 1.

Working Summary:

- JK inputs are set to 1 using logic:
  - Q0 → always toggles
  - Q1 → toggles when Q0 = 1
  - Q2 → toggles when Q0·Q1 = 1
  - Q3 → toggles when Q0·Q1·Q2 = 1

- The result is a binary count: `0000` → `0001` → `0010` → `...` → `1111`.

Conclusion:

The simulation accurately demonstrates a synchronous 4-bit up counter. All flip-flops respond to the same clock, ensuring fast and reliable counting.

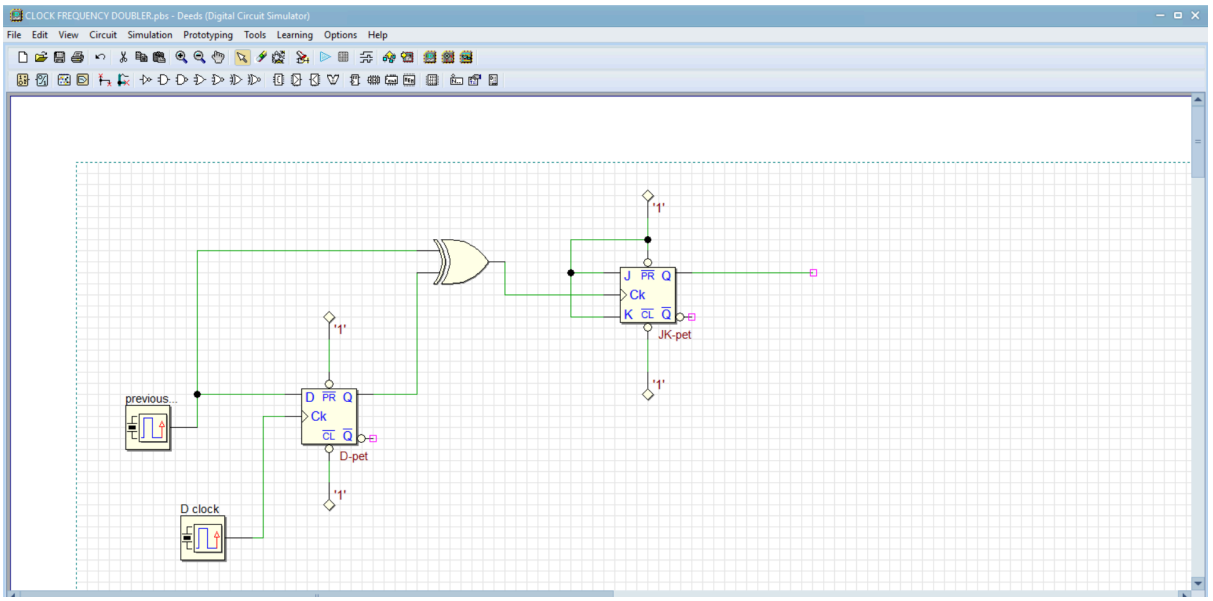
PART-2

PS:

At times, the Chrono-Module must “run twice as fast” without changing the main clock frequency (eg.223 MHz, unchangeable because of quantum synchronization). Think of all the different ways you can achieve this(may be more than one) Design—using as few flip-flops and gates as possible—a way to make a 4-bit counter that tick at 2× the normal rate.

SOLUTIONS:

SOLUTION 1:



Solution 1: Clock Frequency Doubler Using Edge Detector

Concept:

Use a clock doubler circuit that generates a pulse on both the rising and falling edge of the clock effectively doubling the frequency.

Circuit Explanation:

- A D Flip-Flop stores a delayed version of the clock signal.
- An XOR gate compares the original and delayed clock.
- The output toggles whenever there's a transition in the clock (rising or falling).
  This derived signal is used as the clock for the 4-bit counter (JK FF-based).

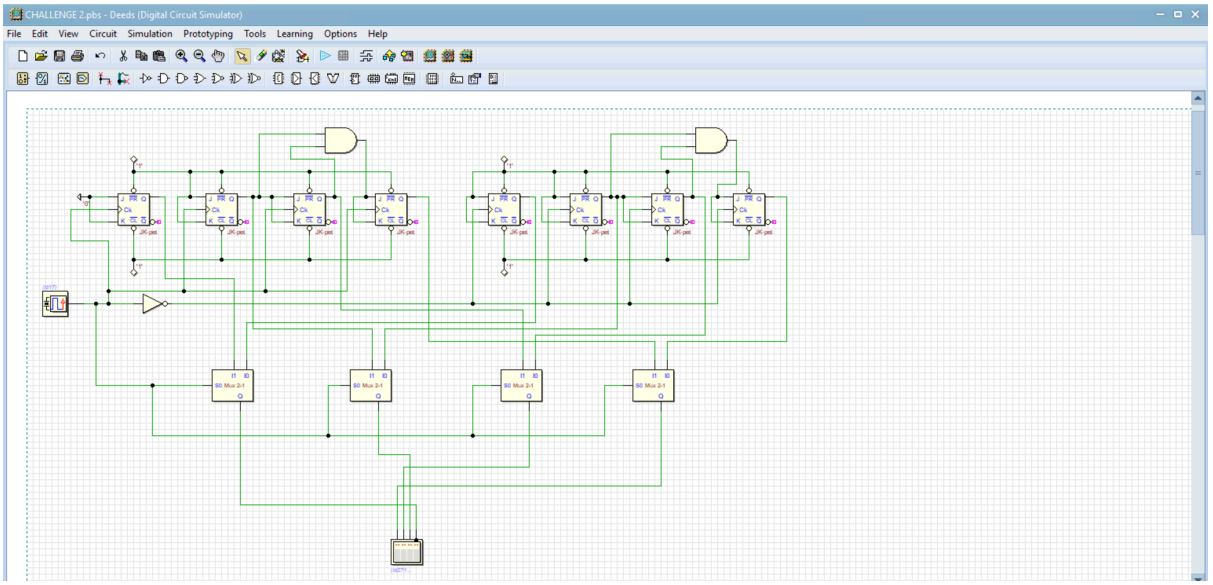
How it Works:

Component	Role
D Flip-Flop	Captures previous clock state
XOR Gate	Generates a HIGH pulse on clock transitions
JK FF	Receives the toggled signal, counting twice per cycle

Result:

Each clock edge produces a pulse to toggle the counter effectively doubling the count rate.

SOLUTION 2:



Solution 2: Dual Counters for Odd & Even Numbers with MUX Selector

Concept:

Split the counter logic into two parts:

- One counter counts even numbers on rising edges.
- The second counts odd numbers on falling edges.
- A MUX (Multiplexer) switches the output between them every half cycle.

Circuit Explanation:

Left Counter Block (Even Counter):

- Triggered on rising edges.
- Standard 4-bit synchronous JK counter.
- J = K = 1 logic used (with ANDs).

- Drives outputs for even count (0, 2, 4, ...).

Right Counter Block (Odd Counter):

- Triggered on inverted clock, i.e., falling edges.
- Same logic as the even counter.
- Drives outputs for odd count (1, 3, 5, ...).

MUXes (2:1 Selectors):

- There are four 2:1 MUXes, one for each bit.
- Select line driven by the clock.
- Selects either even counter (rising edge) or odd counter (falling edge) output.

How it Works:

Clock Phase	MUX Output	Meaning
High	Even Counter	Q from left counter
Low	Odd Counter	Q from right counter

Result:

With each full clock cycle, the MUX alternates between outputs of the even and odd counter creating a combined 4-bit counter that ticks twice per cycle.

Comparison of Two 2× Speed Counter Designs

Aspect	Solution 1: Clock Doubler	Solution 2: Dual Counters + MUX
Idea	XOR detects both edges to double clock rate	Two counters (even/odd) run on opposite edges
Design Style	Minimal and signal-driven	Modular and architecture-based
Components	1 D FF, 1 XOR, 4 JK FFs	2×4-bit counters, 4 MUXes, 1 NOT
Output	Direct from counter	MUX switches between even and odd outputs
Speed Control	Implicit via edge detection	Clock-controlled MUX ensures alternating output
Strength	Simple, efficient, easy to integrate	Creative, scalable, and timing-accurate

Both designs successfully create a **4-bit counter that operates at 2× the clock speed**, even with a fixed input clock frequency.

- The **first method** is minimal and uses edge detection to double clock rate logically.

- The **second method** is more structural: two counters run on opposite edges, and a MUX merges their outputs.