Project 61: Programmable Divider

A Comprehensive Study of Advanced Digital Circuits

By: Abhishek Sharma , Ayush Jain , Gati Goyal, Nikunj Agrawal

Documentation Specialist: Dhruv Patel & Nandini Maheshwari

Contents

1 Project Overview

A Programmable Divider project focuses on designing a circuit to divide an input frequency by a selectable integer, used in applications like digital clocks and communication systems. Key aspects include counter-based design, programmable registers, clock synchronization, and low power consumption. Implemented with HDL and simulated on FPGAs, it tests functionality, accuracy, phase noise, and stability across varying frequencies.

2 Programmable Divider

2.1 Key Concept of Programmable Divider

- **Frequency Division:**
  Divides an input frequency to a lower output frequency by a user-defined ratio, crucial for creating various clock signals.

- **Programmable Ratio:**
  Allows dynamic adjustment of the division factor through a register, enabling flexible frequency output without hardware changes.

- **Counter-Based Design:**
  Uses a counter to reach the programmed division factor, resetting and outputting a divided frequency.

- **Control Logic:**
  Manages the counter, register, and synchronization for stable output timing.

- **Applications:**
  Vital in digital clocks, communication systems, and frequency synthesis.

- **Challenges:**
  Ensuring low phase noise, accurate division, and efficient performance across temperatures and power constraints.

2.2 Working of Programmable Divider

- **Input Clock Signal**
  The divider receives a high-frequency input clock signal, which will be divided down to a lower frequency according to a programmable division ratio.

- **Setting the Division Ratio**
  The division ratio, an integer value, is stored in a programmable register. This ratio defines how many input clock cycles are needed to produce one cycle of the output clock.

- **Counter Operation**
  A digital counter starts counting each pulse of the input clock. When the counter reaches the set division ratio (e.g., 8), it triggers a change in the output signal (typically toggling the output between high and low states), thus completing one output cycle. After reaching the division ratio, the counter resets and begins counting again for the next output cycle.

- **Control Logic**
  Control logic coordinates the counter and ensures it resets correctly after reaching the division ratio, maintaining timing accuracy for the output clock.This logic also manages loading the division ratio from the programmable register, enabling adjustments without disrupting the circuit.

- **Output Clock Signal**
  The output signal oscillates at a frequency determined by the input clock and division ratio, with a frequency given by
  Output Frequency = Input Frequency/DivisionRatio
  The divided clock is synchronized with the input signal but oscillates at a lower frequency.

- **Dynamic Adjustment**

  By reprogramming the division ratio in the register, the output frequency can be adjusted on-the-fly, which is valuable for applications needing variable frequencies without hardware modifications.

2.3 RTL Code

Listing 1: Programmable Divider

```
module ProgrammableDivider #(parameter WIDTH = 8) (
    input logic clk, reset,
    input logic [WIDTH-1:0] dividend, divisor,
    output logic [WIDTH-1:0] quotient, remainder,
    output logic valid
);
    always_ff @(posedge clk or posedge reset) begin
        if (reset) begin
            quotient <= 0; remainder <= 0; valid <= 0;
        end else if (divisor != 0) begin
            quotient <= dividend / divisor;
            remainder <= dividend % divisor;
            valid <= 1;
        end else begin
            quotient <= 0; remainder <= dividend; valid <= 0; //
               Handle division by zero
        end
    end
endmodule
```

2.4 Testbench

Listing 2: Programmable Divider

```
1
2  module ProgrammableDivider_tb;
3      parameter WIDTH = 8;
4      logic clk = 0, reset;
5      logic [WIDTH-1:0] dividend, divisor, quotient, remainder;
6      logic valid;
7
8      ProgrammableDivider #(.WIDTH(WIDTH)) uut (.clk(clk),
           .reset(reset), .dividend(dividend),
9                                                .divisor(divisor),
                                                     .quotient(quotient),
10                                               .remainder(remainder),
                                                     .valid(valid));
11
12     always #5 clk = ~clk; // Clock generation
13
14     initial begin
15         reset = 1; #10 reset = 0;
16         dividend = 30; divisor = 5; #10;
17         $display("Div=%0d, Divisor=%0d, Quot=%0d, Rem=%0d, Valid=%b",
               dividend, divisor, quotient, remainder, valid);
18
19         dividend = 20; divisor = 0; #10;
20         $display("Div=%0d, Divisor=%0d, Quot=%0d, Rem=%0d, Valid=%b",
               dividend, divisor, quotient, remainder, valid);
21
22         dividend = 16; divisor = 4; #10;
23         $display("Div=%0d, Divisor=%0d, Quot=%0d, Rem=%0d, Valid=%b",
               dividend, divisor, quotient, remainder, valid);
24
25         dividend = 10; divisor = 2; #10;
26         $display("Div=%0d, Divisor=%0d, Quot=%0d, Rem=%0d, Valid=%b",
               dividend, divisor, quotient, remainder, valid);
27
28         $finish;
29     end
30 endmodule
```

3 Results

3.1 Simulation

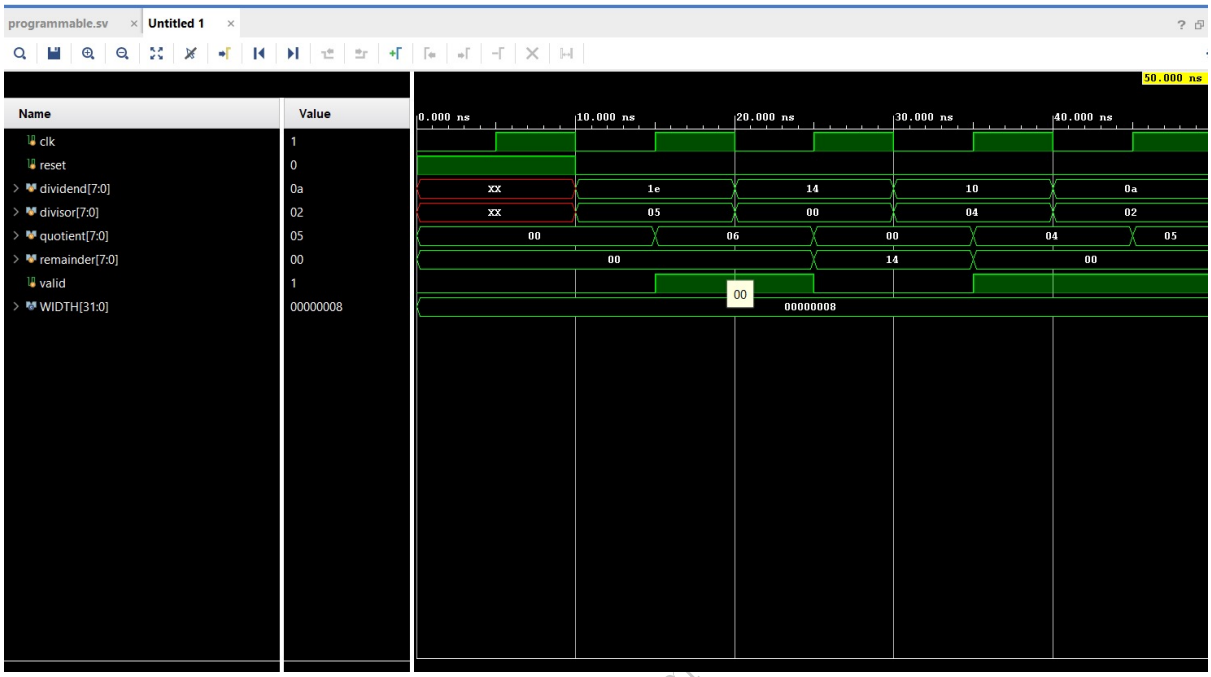

Figure 1: Simulation of Programmable Divider

3.2 Schematic

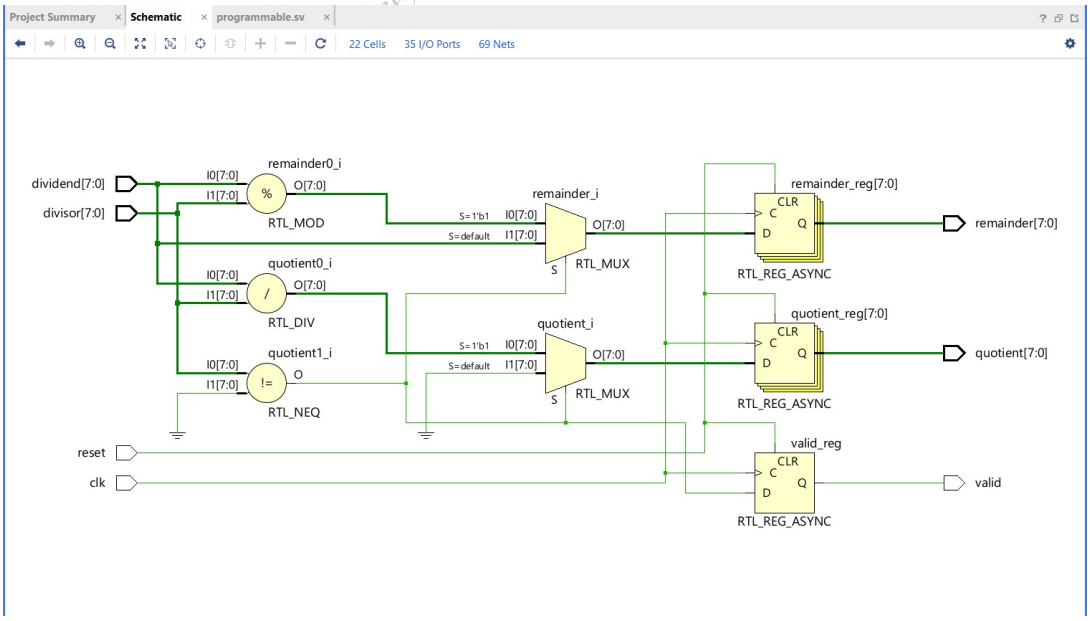

Figure 2: Schematic of Programmable Divider

3.3 Synthesis Design

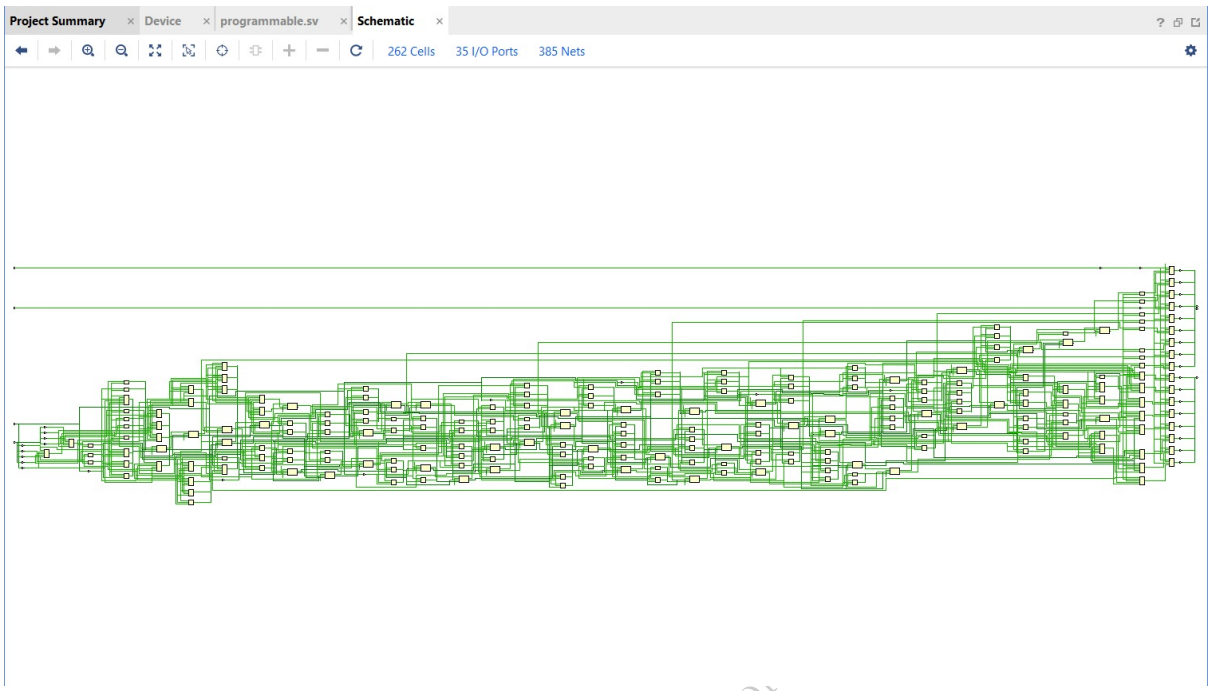

Figure 3: Synthesis Design of Programmable Divider

4 Advantages of Programmable Divider

- **Flexibility:**
  The division ratio can be adjusted programmatically, allowing the same circuit to generate multiple frequencies without hardware changes.

- **Dynamic Frequency Control:**
  Enables real-time frequency adjustments, useful in adaptive systems, such as frequency-hopping communications and multi-speed digital systems.

- **Compact Design:**
  Replaces multiple fixed dividers, saving space in hardware designs and reducing complexity.

- **Cost-Effective:**
  Reduces the need for multiple oscillators or crystals, lowering overall system cost.

- **Precision and Stability:**
  Provides accurate frequency division across a wide range of input signals, crucial for timing-sensitive applications.

- **Power Efficiency:**
  Can optimize power usage by adjusting frequencies as needed, particularly in embedded systems.

- **Ease of Integration:**
  Works seamlessly with microcontrollers, FPGAs, and PLLs, enhancing system versatility and per-

formance.

- **Wide Range of Applications:**
  Ideal for digital clocks, communication systems, signal processing, and frequency synthesis, making it a versatile component in digital electronics.

5 Disadvantages of Programmable Divider

- **Phase Noise and Jitter:**
  Frequency division can introduce noise and timing jitter, affecting signal quality and stability, especially in sensitive RF and clock applications.

- **Limited Division Range:**
  The range of division ratios may be constrained by hardware limitations, reducing flexibility in certain applications.

- **Latency in Changing Frequencies:**
  Reprogramming the divider ratio can introduce a delay, which might be unsuitable for applications requiring instant frequency switching.

- **Power Consumption:**
  At high input frequencies, the programmable divider can consume significant power, potentially affecting battery life in portable devices.

- **Complex Control Logic:**
  The additional control circuitry for programmability increases design complexity, requiring more space and potentially slowing down processing.

- **Temperature and Voltage Sensitivity:**
  Performance may vary under different temperature and voltage conditions, requiring compensation measures for stable operation.

- **Clock Skew Issues:**
  Synchronizing the output clock with the input at high speeds can introduce skew, impacting timing accuracy in high-speed digital circuits.

6 Applications of Programmable Divider

- **Digital Clock Generation:**
  Programmable dividers generate multiple clock frequencies from a single high-frequency clock source, used in microcontrollers, SoCs, and FPGAs for synchronizing different parts of a system.

- **Communication Systems:**
  They are essential in RF communication systems for frequency synthesis, enabling frequency hopping and adjusting signal frequencies to match transmission and reception requirements.

- **Phase-Locked Loops (PLLs):**
  Programmable dividers are used in PLL circuits to produce stable, precise output frequencies, critical in signal processing and wireless communication.

- **Digital Signal Processing (DSP):**
  Used to down-sample high-frequency signals for easier processing, allowing digital systems to work efficiently with lower-speed components.

- **Frequency Synthesizers:**
  Essential in frequency synthesis for generating a range of output frequencies from a single source, useful in test equipment and radio applications.

- **Microcontroller and Embedded Systems:**
  Programmable dividers allow flexible clock generation for peripheral modules, enabling power-efficient operation and adaptive clock control.

- **Instrumentation:**
  Used in measurement and test equipment for producing reference clock signals at specific frequencies, ensuring accurate readings in time-sensitive measurements.

- **Automotive and Industrial Control Systems:**
  Used to generate multiple clock frequencies for timing control in various modules, helping manage tasks like sensor data collection and motor control.

- **Radar and GPS Systems:**
  Helps in frequency division to process received signals, synchronizing frequencies in radar and navigation systems for accurate positioning and detection.

- **Audio and Video Systems:**
  Dividers control timing in audio/video processing, ensuring smooth playback and synchronization across digital media systems.

7 Summary

Programmable dividers dynamically adjust input frequencies for applications like digital clocks, communication systems, PLLs, and DSP. They enable frequency synthesis, adaptive clock control, and efficient down-sampling, making them essential in embedded systems, radar, GPS, and audio/video processing. Their flexibility and precision support timing, synchronization, and signal processing across a wide range of electronic applications.

8 FAQs

1. **What types of devices commonly use programmable dividers?**
   Programmable dividers are used in microcontrollers, FPGAs, communication devices, test equipment, radar systems, and audio/video processing systems.

2. **How is the division ratio set in a programmable divider?**
   The division ratio is typically set using a register that can be programmed through control logic or software, allowing real-time adjustments.

3. **Can a programmable divider be used in high-frequency applications?** Yes, but high-frequency applications may require careful attention to phase noise, jitter, and signal integrity to maintain stable and accurate outputs.

4. **What is the difference between a programmable divider and a fixed divider?**
   A fixed divider has a constant division ratio, while a programmable divider allows for dynamic adjustment of the ratio, providing greater flexibility for varying applications.

5. **What is the impact of temperature or voltage on a programmable divider's performance?** Programmable dividers can be sensitive to temperature and voltage variations, which may affect timing, frequency accuracy, and stability, requiring calibration or compensation mechanisms.