# Digital System Report coursework

计科2202 陈诚 22723042

Table of Contents

[Digital System Report coursework 1](#_Toc148868568)

[1. Purpose 2](#_Toc148868569)

[2. Apparatus 2](#_Toc148868570)

[3. Task 2](#_Toc148868571)

[3.1. week\_1\_task 2](#_Toc148868572)

[3.2. week\_2\_task 2](#_Toc148868573)

[3.3. week\_3\_task 3](#_Toc148868574)

[3.4. week\_4\_task 3](#_Toc148868575)

[4. Implement 4](#_Toc148868576)

[4.1. Digital Circuit Experiment\_1 4](#_Toc148868577)

[4.2. Digital Circuit Experiment\_2 5](#_Toc148868578)

[4.3. Digital Circuit Experiment\_3 7](#_Toc148868579)

[4.4. Digital Circuit Experiment\_4 9](#_Toc148868580)

## Purpose

1. Gain a comprehensive understanding of the fundamental properties of various circuit gates and develop proficiency in their applications.
2. Enhance logical thinking skills, enabling the design of more intricate and customized circuits by leveraging different logic gates.
3. Acquire a hardware-level comprehension of CPU principles, including components such as the Control Unit (CU), Arithmetic Logic Unit (ALU), registers, and other relevant elements.

## 2. Apparatus

The digital circuit integrated test box includes power source, logic level source, ground, emulated microchips, Wires,

## 3. Task

### 3.1. week\_1\_task

1. Know how to use the toolbox.

2. identify various logic gates, one can employ a truth table, displays the input and output values using the tool.

### 3.2. week\_2\_task

1. Design a comparator to compare two bits, the output will be 1 if two bits are equal.

2. Design a two - out - of - three voter logic circuit using logic gates only. The circuit has 3 inputs (A, B and C) and 1 output. The output is 1 if two or more of the inputs are 1.

### 3.3. week\_3\_task

1. Full adder design by using basic logic gate

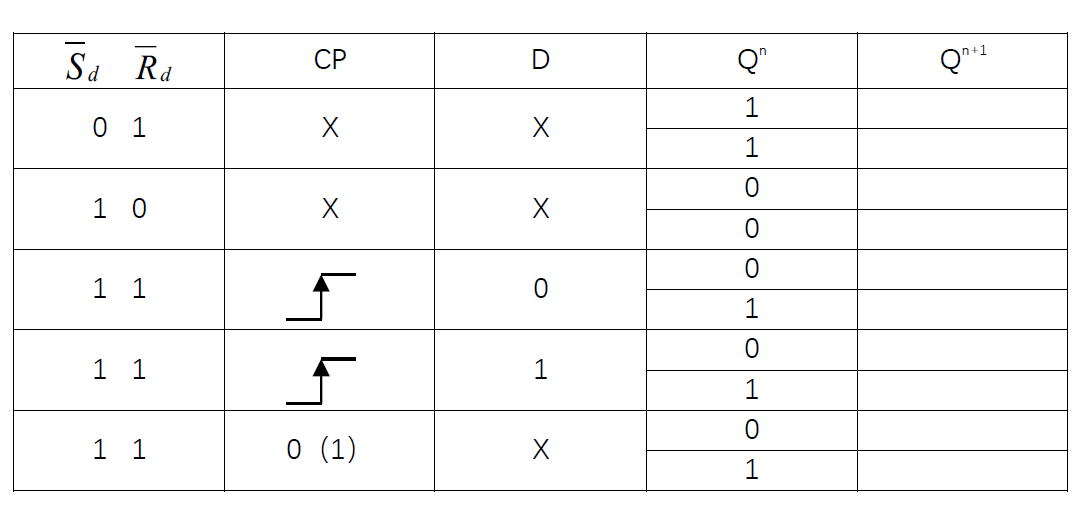
Use XOR (7486), OR(7432), AND(7408) gate to design Half adder and full adder and test logic function.

2. 74LS283 adder

Get familiar with 74283 adder, check the output and logic truth table of it

Use 74LS283 to design 4 bits adder, and then check the sum of the input.

### 3.4. week\_4\_task

1. test and fill in the truth table using 74LS74, get familiar with the principle of building and function of D flip flop device

2. Use D flip-flop build counter and Use logic analyzer to observe the base4 counter waveform



## 4. Implement

### 4.1. Digital Circuit Experiment\_1

#### Task 1

First , you need to connect it with a power source, and select a developing model, choose appropriate chip. second, you need to confirm the source, input, output port. Use wire to connect it.

And-gate-7408

I start by using the input and output results from the table. This gate is a and gate because only all of the inputs are high voltage and all of the outputs are high voltage.

Or gate - 7432

The output of it is high voltage when either a or b is High voltage, so the gate is or gate. All false will be false when the gate is or gate.

Not gate – 7400

The not gate only have a input, it is easiest gate to identify. The gate will give upside result.

Not and gate – 7400

A group of electrical components

Description automatically generated with medium confidenceWhen both of a and b are High voltage, the output is low voltage. But another all result is high voltage, so this one is not and gate. This is the combination of not gate and and gate.

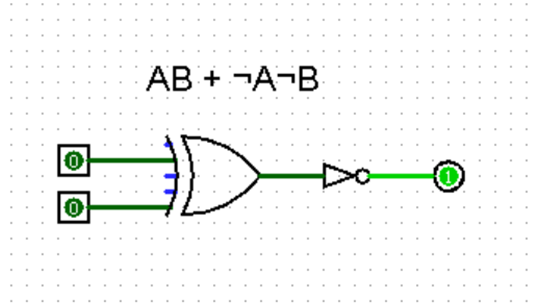
### 4.2. Digital Circuit Experiment\_2

#### Task 1

the True Table is

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 1 | 1 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

The equation (AB)∪(¬A¬B) = ¬(A⊕B) holds true, and we can utilize combinations of 7486 and 7404 logic gates to create a new circuit.

A machine with many wires

Description automatically generatedNext, we connect physical components to obtain the final result as follow:

In summary, the truth table of this circuit exhibits the exact opposite output behavior compared to an XOR logic gate. When there is a difference in the input values, the output is False. Only when the input values are the same, the output is True, representing a high voltage level, which results in the light being illuminated.

#### Task 2

The Truth table like below:

|  |  |  |  |
| --- | --- | --- | --- |
| Input A | Input B | Input C | Output |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

So, the final truth table results in the expression ¬ABC + A¬BC + AB¬C + ABC. To simplify this expression, we use a Karnaugh map, as shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AB | ¬AB | ¬A¬B | A¬B |
| C | 1 | 1 | 0 | 1 |
| ¬C | 1 | 0 | 0 | 0 |

A machine with wires and buttons

Description automatically generatedA diagram of a circuit

Description automatically generatedThe final simplified result is BC + AC + AB. We then create a circuit diagram, which is shown below:

We will be using three 7408 AND logic gates and two 7432 OR logic gates according to the circuit diagram. When physically connected, the real circuit behaves as follows:

We will be using three 7408 AND logic gates and two 7432 OR logic gates according to the circuit diagram. When physically connected, the real circuit behaves as follows:

Through testing, we have observed that when any two inputs are set to a high voltage, the output results in the illumination of the light, or alternatively, when all three inputs are high, the output is high voltage, which also causes the light to illuminate. In all other cases, the light remains off. This behavior aligns with our expectations, and the testing has been successful.

### 4.3. Digital Circuit Experiment\_3

#### Task 1

A half-adder is a fundamental building block in digital logic, and it has two inputs and two outputs. The inputs are typically labeled as A and B, and the outputs are labeled as the sum (S) and the carry (C)

Input A represents one of the binary digits to be added.

Input B represents the other binary digit to be added.

Output S stands for the sum, which is the result of adding the two input digits.

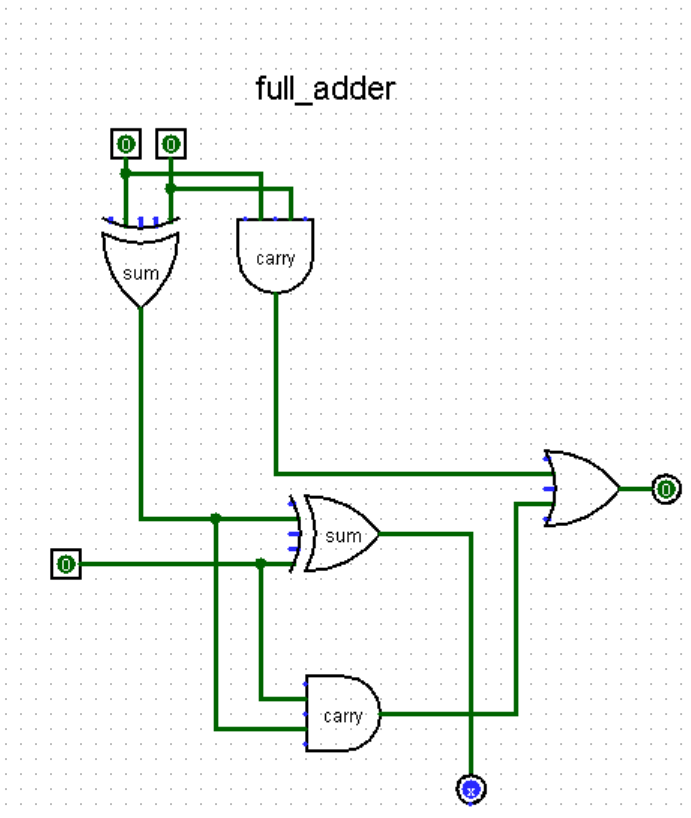
Output C represents the carry, which is generated as a result of the addition.

A white paper with black text

Description automatically generatedA diagram of a number of objects

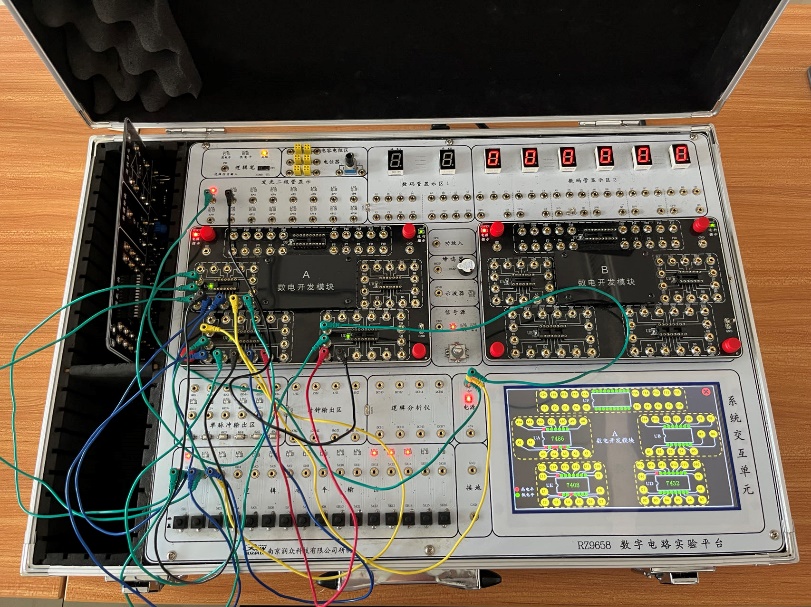
Description automatically generated with medium confidence For binary addition, the output undoubtedly includes the current bit's sum (S) and the carry (C). In addition to the two operands A and B, there may also be a carry from the previous bit, denoted as C{n-1}. However, for the first bit of a binary number, there is no carry from a previous bit, or Z=0. Thus, we can omit Z, and the inputs consist only of X and Y. This is the essence of a half-adder.

A full adder is constructed by using two half adders and an OR gate. It has three inputs and two outputs, representing the addition of two binary numbers, the carry-in from the previous bit, and the carry-out to the next bit. The circuit diagram is as follows: A group of blue squares with black lines

Description automatically generated with medium confidence

In the process of addition, the first adder is responsible for the addition of the current bit, and the carry from the previous bit does not affect the sum of this bit. Therefore, it is directly connected to the OR gate. The carry from the previous bit is combined with the sum of the current bit to determine the outcome of this bit.

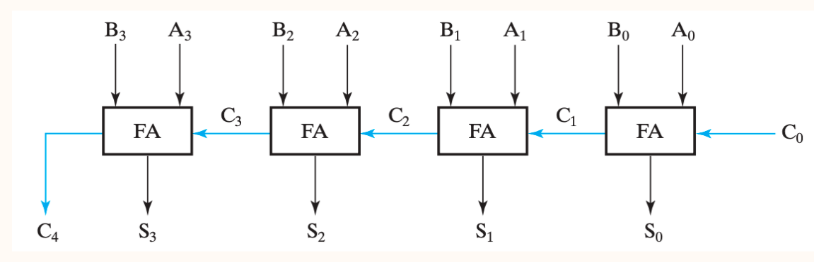
This arrangement ensures that the carry from the previous bit is considered when determining the sum of the current bit, allowing for accurate binary addition.

Finally, the actual physics circuits as follow：

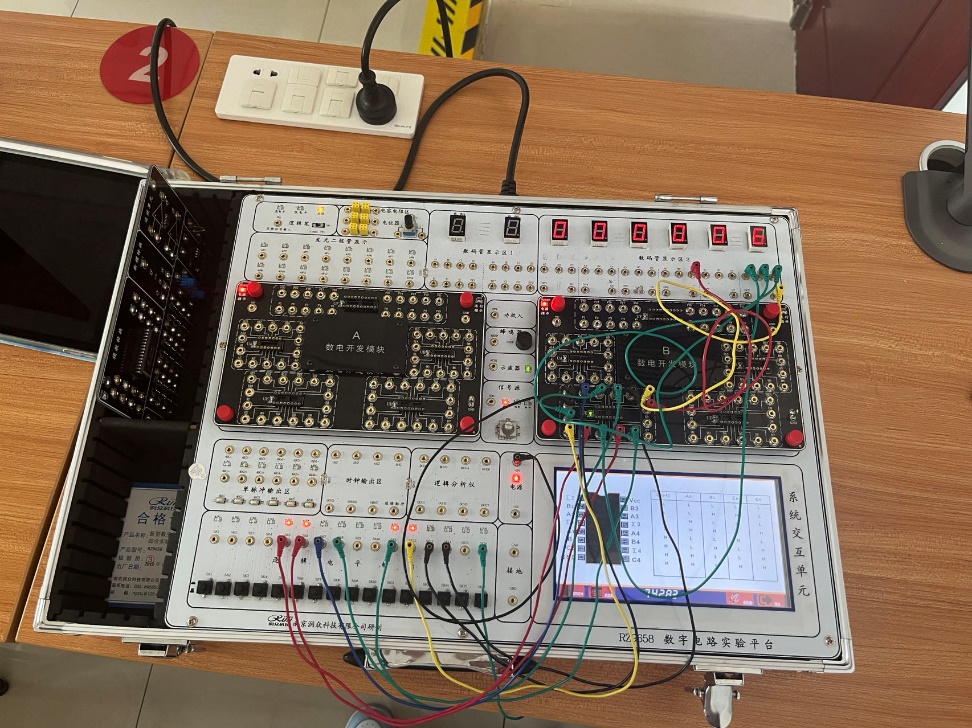
In this scenario, when the carry is zero, the sum is 1, and there's a carry of 1, and then the carry becomes zero, while the current bit is 1, the first light turns on.

#### Task 2

The 74LS283 is a 4-bit binary full adder that can be used to perform binary addition. It has two 4-bit binary inputs, A and B, and a Carry-In (C\_in) input, which allows you to chain multiple 74LS283 chips for multi-bit addition. The chip provides a 4-bit binary sum (S) output and a Carry-Out (C\_out) output.



The actual physic circuit is as follow：



As shown in the diagram, this 4-bit adder can perform four-bit binary addition and convert the result to decimal. In the current scenario, both numbers to be added are 3. After the operation of the adder, the result is 6.

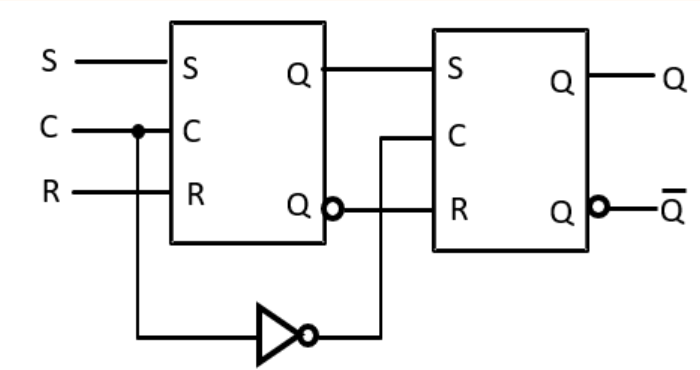
### 4.4. Digital Circuit Experiment\_4

#### Task 1

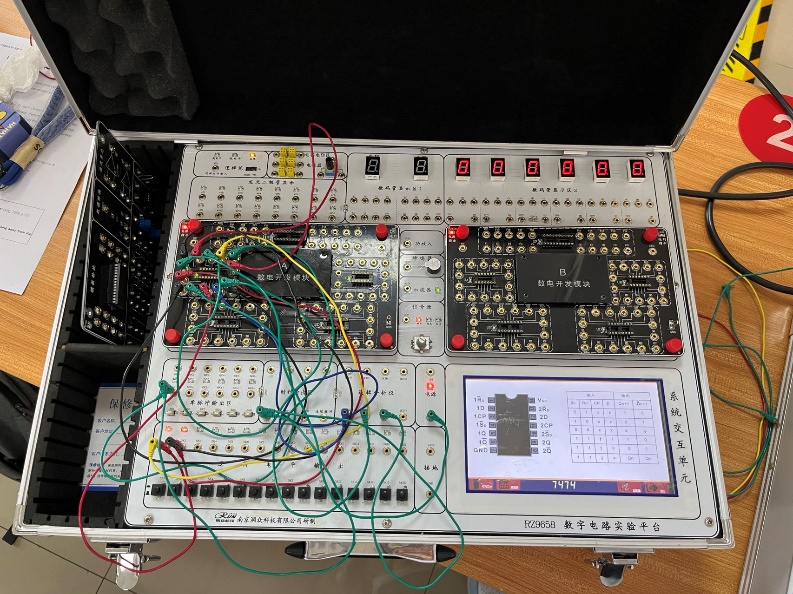
A D flip-flop, also known as a D latch, is a fundamental building block in digital circuits used for storing data. It is typically constructed from two D latches. Here's the principle and circuit diagram:

A diagram of a logic diagram

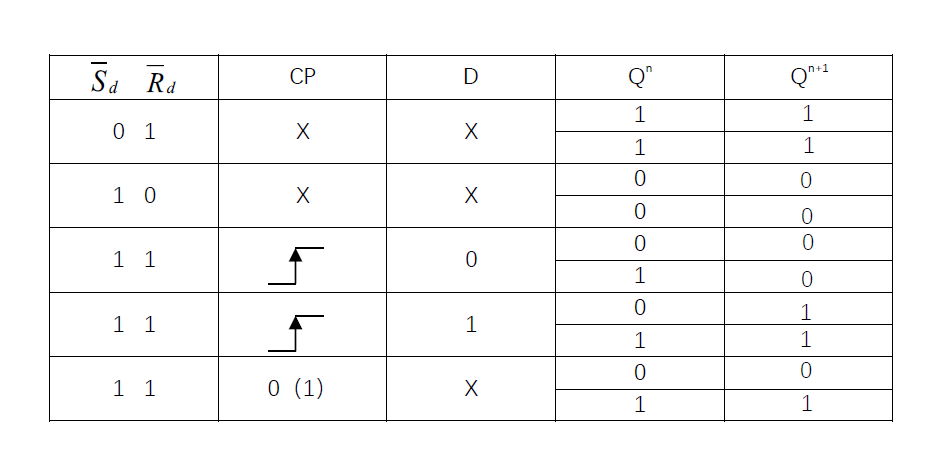
Description automatically generated We can then construct a complete D-flip-flop using the following circuit：



and then we connect the circuit according the instruction in toolbox, the final figure is as follow：



The final excel is as follow：



D Flip-Flop (D Latch) Behavior:

Upon analysis, we can determine that when the S (Set) input is low (0), regardless of the previous Q value, Q(n+1) is set to 1. Conversely, when S is high (1), again, regardless of the previous Q value, Q(n+1) is set to 0.

When S is low (0), the D flip-flop forces Q(n+1) to be high (1) irrespective of the previous Q value.

When S is high (1), the D flip-flop forces Q(n+1) to be low (0) regardless of the previous Q value.

Furthermore, during a rising edge of the CP (Clock Pulse) signal, the D flip-flop allows the user to change the value. Q(n+1) becomes whatever D is set to during the rising edge. When CP is at a high or low logic level, Q retains its previous state.

This behavior allows for the storage and manipulation of data in digital circuits, with Q(n+1) responding to changes in D during specific clock transitions while holding its state at other times.

#### Task 2

the principle of frequency Division Divide-by-2 Counter:

A diagram of a frequency diagram

Description automatically generated with medium confidence

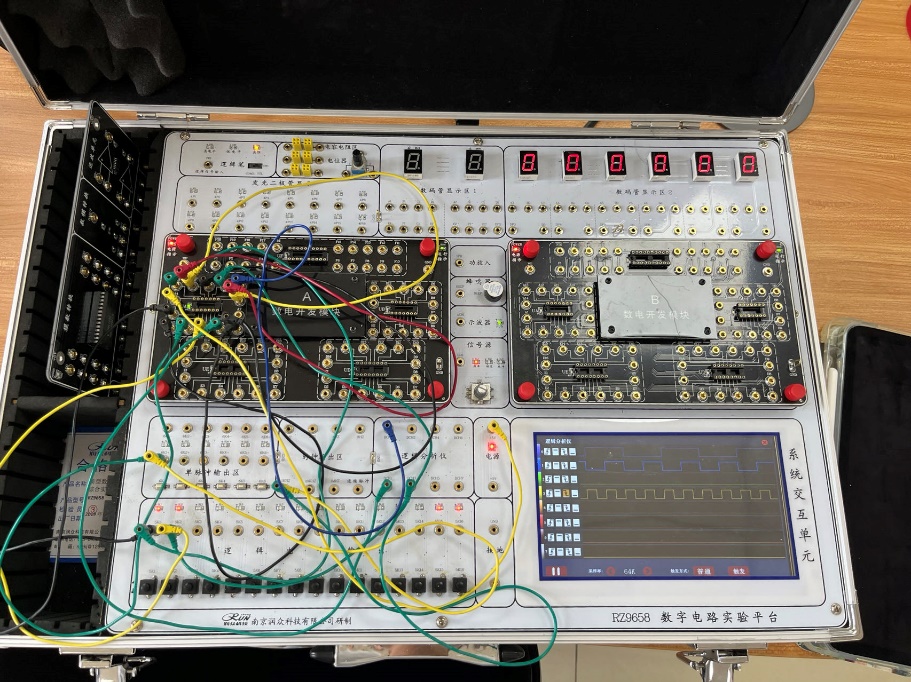
D flip-flop changes its next output value Q(n+1) only at the rising edge of a clock pulse (CP). Regardless of the current value stored in Q, it's only during the rising edge of the clock pulse that the D flip-flop will alter its output in response to the new input D. Ergo, It can be seen from the frequency waveforms above, that by “feeding back” the output from Q to the input terminal D, the output pulses at Q have a frequency that are exactly one half ( ƒ ÷ 2 ) that of the input clock frequency. In other words, the circuit produces.

For frequency division, toggle mode flip-flops are used in a chain as a divide by two counters. One flip-flop will divide the clock, ƒ(IN) by 2, two flip-flops will divide ƒ(IN) by 4 (and so on). One benefit of using toggle flip-flops for frequency division is that the output at any point has an exact 50% duty cycle.

The final output clock signal will have a frequency value equal to the input clock frequency divided by the MOD number of the counter. Such circuits are known as “divide-by-n” counters. Counters can be formed by connecting individual flip-flops together and are classified according to the way they are clocked.

4x Frequency Divider Using D Flip-Flops:

By utilizing the provided chips and connecting the circuit, you've created a 4x frequency divider using two D flip-flops. The result is as follows:



The input signal's frequency is divided by 4, meaning that for every four input clock pulses, the circuit produces a single output clock pulse. This type of frequency divider is valuable in various digital applications where you need to reduce the frequency of a clock signal while maintaining synchronization.