# DIGITAL TECHNICS II

**Dr. Bálint Pődör**

*Óbuda University,*
*Microelectronics and Technology Institute*

## 3. LECTURE: COUNTERS AND RELATED

2nd (Spring) term 2017/2018

1

## 3. LECTURE

1. Counters, general concepts and properties
2. Ripple (asynchronous) counters
3. Synchronous counters
4. Counter applications

## COUNTERS (AND REGISTERS): AN INTRODUCTION

Counters and registers belong to the category of MSI sequential logic circuits. They have similar architecture, as both counters and registers comprise a cascaded arrangement of more than one flip flop with or without combinational logic devices. Both constitute very important building blocks of sequential logic, and different types of counter and register available in integrated circuit (IC) form are used in a wide range of digital systems.

While counters are mainly used in counting applications, where they either measure the time interval between two unknown time instants or measure the frequency of a given signal, registers are primarily used for the temporary storage of data present at the output of a digital circuit before they are fed to another digital circuit.

## COUNTERS (AND REGISTERS): AN INTRODUCTION

Everybody is familiar with the role of different types of register used inside a microprocessor, and also their use in microprocessor-based applications.

Because of the very nature of operation of registers, they form the basis of a very important class of counters called shift counters or shift-register counters.

## COUNTERS: GENERAL CONCEPTS

- The counter is a special case of the sequential circuits i.e. a circuit with no input (primary) variables.
- Its function is to count the input pulses (clock signal) and store the result till the arrival of the next signal.
- The counting process, therefore consists of a series of storage and addition operations.
- The counters are built from flip-flops and of gate circuits.

## CLASSIFICATION AND PROPERTIES

According to the direction of counting:

- up counter
- down counter
- up-down counter or reversible counter

According to state encoding:

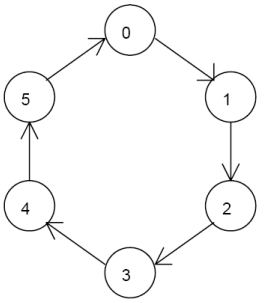
- binary
- decade (e.g. BCD)
- other

## STATES OF A COUNTER

The state transition diagram of counters is of the form of a closed ring.
Example: mod 6 counter

The number of states, the counter goes through before recycling is the modulus of the counter.

The largest possible modulus of a n-bit counter is $2^n$.

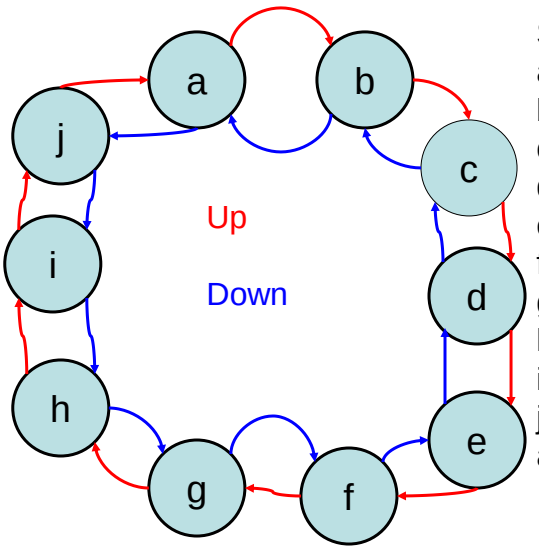



## REDUCED CYCLE COUNTERS

- The range of an n-bit binary counter is $0 – (2^n-1)$.
- Modulo counter reaching a predetermined value, returns to the initial state.
- E.g. a 4-bit counter counting from zero up to 11 (0000-1011):
  modulo 12 counter, the counting cycle repeats itself after every 12th pulse.
- The modulus is the length of the cycle.

8

## UP-DOWN MOD 10 COUNTER: STATE TRANSITION DIAGRAM




| State | encoding | decimal |
|---|---|---|
| a | 0 0 0 0 | 0 |
| b | 0 0 0 1 | 1 |
| c | 0 0 1 0 | 2 |
| d | 0 0 1 1 | 3 |
| e | 0 1 0 0 | 4 |
| f | 0 1 0 1 | 5 |
| g | 0 1 1 0 | 6 |
| h | 0 1 1 1 | 7 |
| i | 1 0 0 0 | 8 |
| j | 1 0 0 1 | 9 |
| a | 0 0 0 0 | 0 |
| … | …….. | … |

## ASYNCHRONOUS AND SYNCHRONOUS COUNTERS

Many different types of electronic counters are available. They are all either asynchronous or synchronous type and are usually constructed using JK flip-flops.

• Asynchronous (ripple) counter: The input signal is applied to the clock input of the first FF, and the output of each FF is connected directly to the clock input of the next.

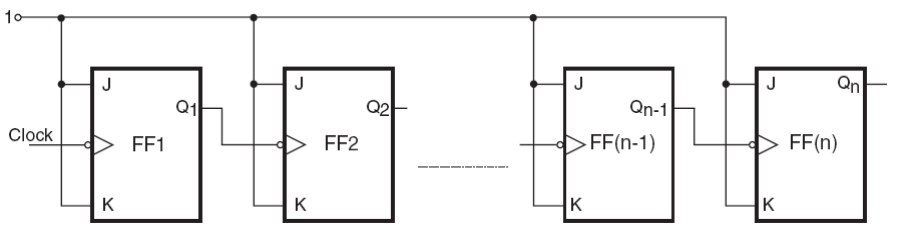
• Synchronous counter: all flip-flops are controlled by a common clock. Logic gates between each stage of the circuit control dataflow from stage to stage so that the desired count behaviour is realized.

## RIPPLE (ASYNCHRONOUS) COUNTER

A ripple counter is a cascaded arrangement of flip-flops where the output of one flip-flop drives the clock input of the following flip-flop. The number of flip-flops in the cascaded arrangement depends upon the number of different logic states that it goes through before it repeats the sequence, a parameter known as the modulus of the counter.

In a ripple counter, also called an asynchronous counter or a serial counter, the clock input is applied only to the first flip-flop, also called the input flip-flop, in the cascaded arrangement. The clock input to any subsequent flip-flop comes from the output of its immediately preceding flip-flop.

## RIPPLE (ASYNCHRONOUS) COUNTER

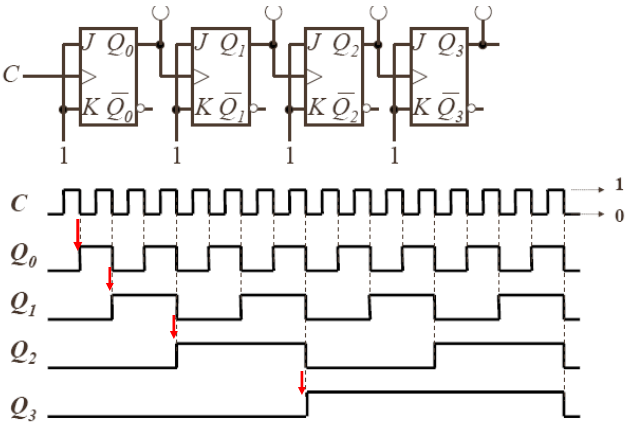



Generalized block schematic of n-bit binary ripple counter.

For instance, the output of the first flip-flop acts as the clock input to the second flip-flop, the output of the second flip-flop feeds the clock input of the third flip-flop and so on. In general, in an arrangement of n flip-flops, the clock input to the nth flip-flop comes from the output of the (n−1)th flip-flop for n>1.

# RIPPLE COUNTER (UP)

JK flip-flops operating in the toggle mode (J=K=1).




| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | DEC |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | =0 |
| 0 | 0 | 0 | 1 | =1 |
| 0 | 0 | 1 | 0 | =2 |
| 0 | 0 | 1 | 1 | =3 |
| 0 | 1 | 0 | 0 | =4 |
| 0 | 1 | 0 | 1 | =5 |
| 0 | 1 | 1 | 0 | =6 |
| 0 | 1 | 1 | 1 | =7 |
| 1 | 0 | 0 | 0 | =8 |
| 1 | 0 | 0 | 1 | =9 |
| 1 | 0 | 1 | 0 | =10 |
| 1 | 0 | 1 | 1 | =11 |
| 1 | 1 | 0 | 0 | =12 |
| 1 | 1 | 0 | 1 | =13 |
| 1 | 1 | 1 | 0 | =14 |
| 1 | 1 | 1 | 1 | =15 |

# ASYNCHRONOUS (RIPPLE) COUNTER (UP)

The four-stage ripple counter using JK flip-flops uses the maximum counting capability of the four stages and would thus be classified as mod 16. The FFs are connected to toggle and change state an the negative-going transitions of the waveform applied to the clock input.

The input is shown for convenience as being periodic even though it may be completely random.

The count is obtained by observing the sequence Q3 Q2 Q1 Q0 which starts of at 0 0 0 0 (=decimal 0).

14

# ASYNCHRONOUS (RIPPLE) COUNTER (UP)

Operation:

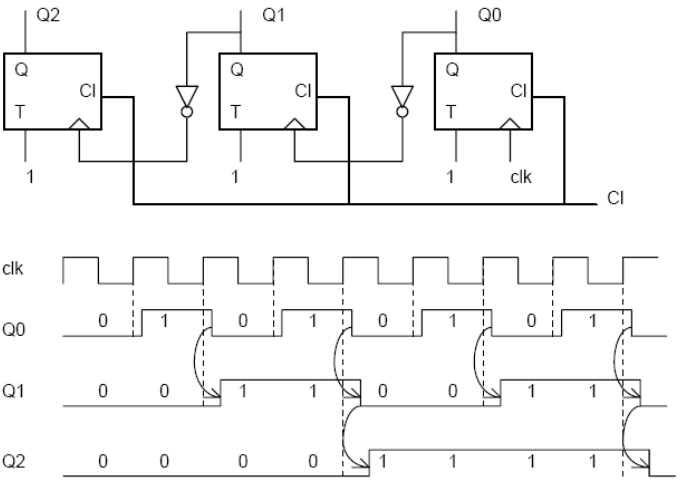
- upon arrival the clock pulses initiate the state changes of the individual flip flops. The effect of clock pulses propagates serially.

The Q outputs of the flip-flops are simultaneously control and output (state) points.

If the propagation delay of a flip-flop is $t_{pd}$, the sate change of the nth flip flop occurs only after a delay of $nt_{pd}$.

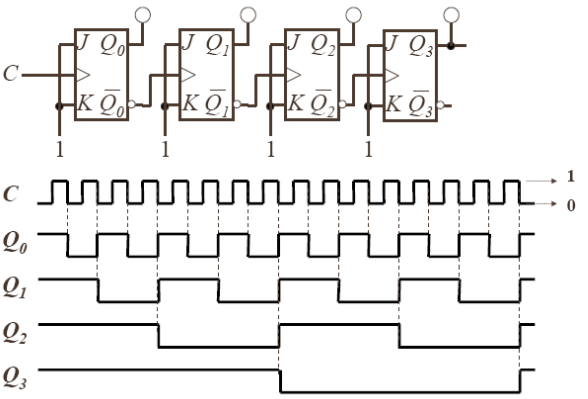
15

# RIPPLE COUNTER (UP)




Asynchronous binary up-counter with asynchronous clear. The delay of the outputs with respect to the clock input is increasing at each new stage.

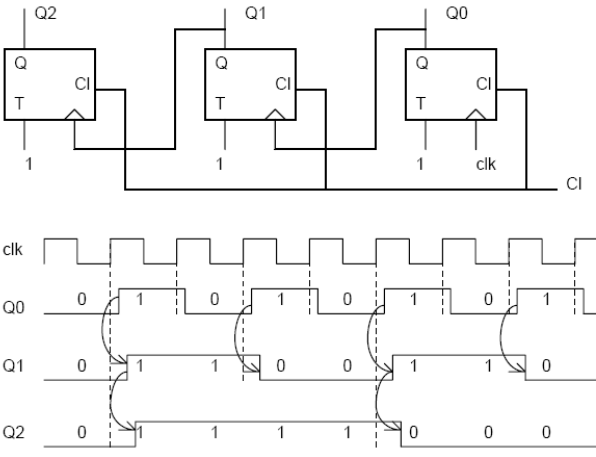
# RIPPLE COUNTER (DOWN)




| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | DEC |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | =15 |
| 1 | 1 | 1 | 0 | =14 |
| 1 | 1 | 0 | 1 | =13 |
| 1 | 1 | 0 | 0 | =12 |
| 1 | 0 | 1 | 1 | =11 |
| 1 | 0 | 1 | 0 | =10 |
| 1 | 0 | 0 | 1 | =9 |
| 1 | 0 | 0 | 0 | =8 |
| 0 | 1 | 1 | 1 | =7 |
| 0 | 1 | 1 | 0 | =6 |
| 0 | 1 | 0 | 1 | =5 |
| 0 | 1 | 0 | 0 | =4 |
| 0 | 0 | 1 | 1 | =3 |
| 0 | 0 | 1 | 0 | =2 |
| 0 | 0 | 0 | 1 | =1 |
| 0 | 0 | 0 | 0 | =0 |

The $\overline{Q}$ outputs are the control points but the states are represented by Q

# RIPPLE COUNTER (DOWN)




Asynchronous binary down-counter with asynchronous clear. The delay of the outputs with respect to the clock input is increasing at each new stage.

## REVERSIBLE (UP/DOWN) COUNTERS

• The direction of counting is controlled by an external control signal.

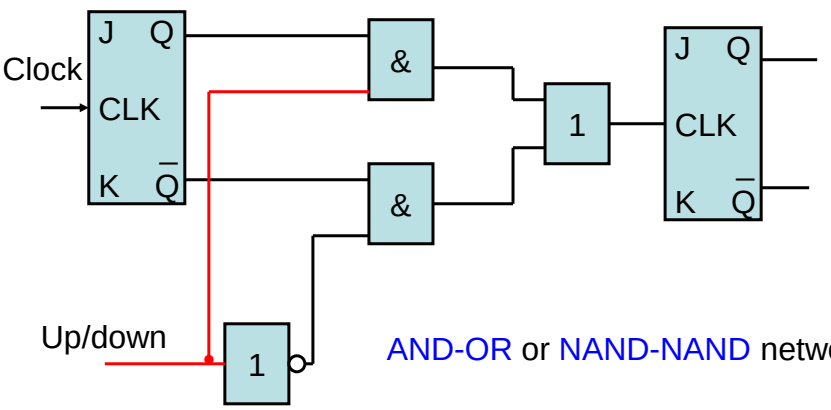
• A reversible counter can be used to add or subtract two serially coded numbers, so it can be operated as a serial mode subtractor.

• Application example: preload a number, count down, arriving to zero generate a control signal to start or stop a process.

19

## UP/DOWN BINARY RIPPLE COUNTER WITH JK FLIP-FLOPS

Scheme of the up/down control between each stages.




AND-OR or NAND-NAND network.

## PROPAGATION DELAY IN RIPPLE COUNTERS

A major problem with ripple counters arises from the propagation delay of the flip-flops constituting the counter. The effective propagation delay in a ripple counter is equal to the sum of propagation delays due to different flip-flops. The situation becomes worse with increase in the number of flip-flops used to construct the counter, which is the case in larger bit counters.

An increased propagation delay puts a limit on the maximum frequency used as clock input to the counter. We can appreciate that the clock signal time period must be equal to or greater than the total propagation delay. The maximum clock frequency therefore corresponds to a time period that equals the total propagation delay. clock frequency.

## PROPAGATION DELAY IN RIPPLE COUNTERS

If $t_{pd}$ is the propagation delay in each flip-flop, then, in a counter with N flip-flops having a modulus of less than or equal to $2^N$ , the maximum usable clock frequency is given by

$$f_{max} = 1/(N \times t_{pd}).$$

Often, two propagation delay times are specified in the case of flip-flops, one for LOW-to-HIGH transition ($t_{pLH}$)and the other for HIGH-to-LOW transition ($t_{pHL}$)at the output. In such a case, the larger of the two should be considered for computing the maximum clock frequency.

## RIPPLE COUNTER: MAXIMUM FREQUENCY

$t_{pd}$ - propagation delay,
$t_{dec}$ - decoding time,
n - number of stages

$$f_{max} = \frac{1}{n\,t_{pd} + t_{dec}}$$

The ripple counters, in case of many stages are much slower then the synchronous counters. E.g., in the case of a ripple counter IC belonging to the low-power Schottky TTL family, the propagation delay per flip-flop typically is of the order of 25 ns. This implies that a four-bit ripple counter from this logic family can not be clocked faster than 10 MHz. The upper limit on the clock frequency further decreases with increase in the number of bits to be handled by the counter.

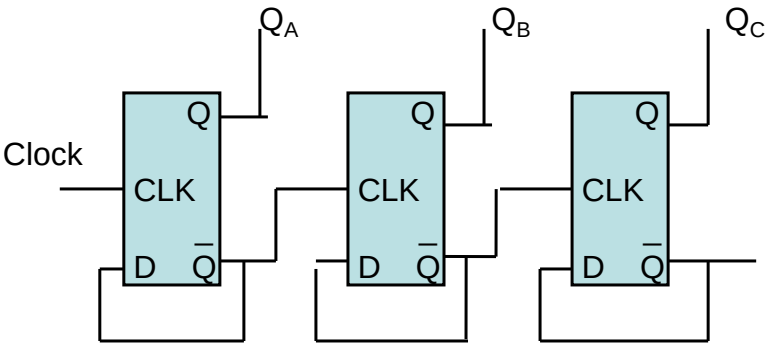


## RIPPLE COUNTER: FREQUENCY DIVISION

The waveform of the binary ripple counter indicates that the Q0 output waveform has a frequency exactly one-half that of the input, the Q1 output frequency is one-quarter of the input frequency, etc. Thus the circuit acts as a frequency divider, or frequency scaler. For an n-flip-flop circuit the output frequency is divided by $2^n$ in steps of 2.

The upper limit on the input frequency of the frequency divider is the same as the maximum rate at which the first fliip-flop in the cascade can toggle.

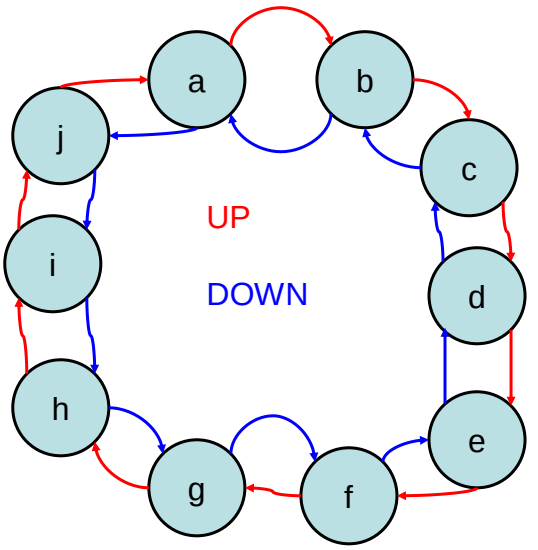
# RIPPLE COUNTER (UP) WITH D FLIP-FLOPS




D flip-flop: edge-triggered (rising edge, 0-1 transition).
Down counter: the feedback inside each stage is tha same, however the clock inputs are driven by the respective Q outputs.

25

# REVERSIBLE 10-STATE (BCD) COUNTER



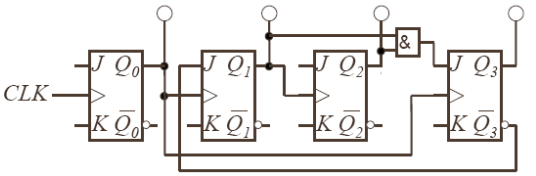

State encoding

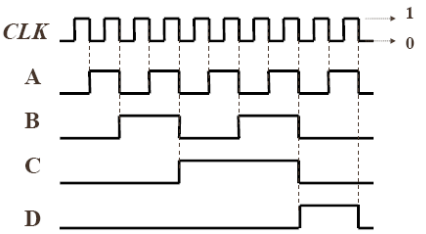
| | |
|---|---|
| a | 0 0 0 0 |
| b | 0 0 0 1 |
| c | 0 0 1 0 |
| d | 0 0 1 1 |
| e | 0 1 0 0 |
| f | 0 1 0 1 |
| g | 0 1 1 0 |
| h | 0 1 1 1 |
| i | 1 0 0 0 |
| j | 1 0 0 1 |

26

# ASYNCHRONOUS BCD COUNTER (UP)

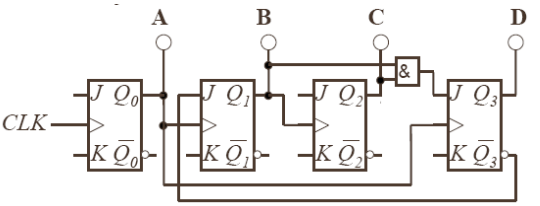
The counter should recycle after the 10th pulse. Upon arrival of the 10th pulse Q1 should not toggle to 1, but Q 3 should toggle back to 0.




| D | C | B | A | DEC |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | =0 |
| 0 | 0 | 0 | 1 | =1 |
| 0 | 0 | 1 | 0 | =2 |
| 0 | 0 | 1 | 1 | =3 |
| 0 | 1 | 0 | 0 | =4 |
| 0 | 1 | 0 | 1 | =5 |
| 0 | 1 | 1 | 0 | =6 |
| 0 | 1 | 1 | 1 | =7 |
| 1 | 0 | 0 | 0 | =8 |
| 1 | 0 | 0 | 1 | =9 |
| 1 | 0 | 1 | 0 | -- |
| 1 | 0 | 1 | 1 | -- |
| 1 | 1 | 0 | 0 | -- |
| 1 | 1 | 0 | 1 | -- |
| 1 | 1 | 1 | 0 | -- |
| 1 | 1 | 1 | 1 | -- |

# ASYNCHRONOUS BCD COUNTER (UP)




• The 4-bit BCD counter uses 10 states out of possible 16.
• The 1-2-4-8 weighted BCD counter counts the first 9 pulses as the binary counter. Here the 4th FF is excited directly by the 1th FF, its J input being appropriately gated.
• Upon the arrival of the 10th pulses (1001 is the state) the 1th FF toggles back to 0, the 2nd FF cannot change state, because it is inhibited by the feedback from the 4th FF. The 4th FF is toggling too in response to the signal from the output of the 1st FF, in this way the counter recycles to the 0 0 0 0 initial state.

## ASYNCHRONOUS BCD COUNTER (DOWN)

The down counting BCD ripple counter can be constructed based on similar principles. Here the critical point is the transition from decimal 0 (binary 0000) to decimal 9 (binary 1001) state.

29

## SYNCHRONOUS (OR PARALLEL) COUNTERS

Ripple counters discussed thus far are asynchronous in nature as the different flip flops comprising the counter are not clocked simultaneously and in synchronism with the clock pulses. The total propagation delay in such a counter, as explained earlier, is equal to the sum of propagation delays due to different flip-flops. The propagation delay becomes prohibitively large in a ripple counter with a large count.

On the other hand, in a synchronous counter, all flip-flops in the counter are clocked simultaneously in synchronism with the clock, and as a consequence all flip-flops change state at the same time. The propagation delay in this case is independent of the number of flip-flops used.

## SYNCHRONOUS (OR PARALLEL) COUNTERS

Since the different flip-flops in a synchronous counter are clocked at the same time, there needs to be additional logic circuitry to ensure that the various flip-flops toggle at the right time.

## SYNCHRONOUS COUNTER

In a synchronous counter, also known as a parallel counter, all the flip-flops in the counter change state at the same time in synchronism with the input clock signal. The clock signal in this case is simultaneously applied to the clock inputs of all the flip-flops.

The delay involved in this case is equal to the propagation delay of one flip-flop only, irrespective of the number of flip-flops used to construct the counter. In other words, the delay is independent of the size of the counter.

## SYNCHRONOUS COUNTERS: GENERAL

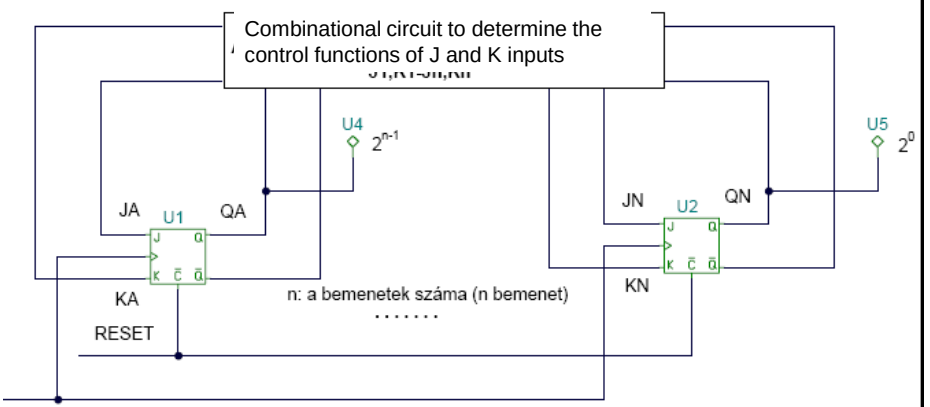
• Eliminates/reduces the delays inherent in ripple counters.

• All flip-flops receive simultaneously (parallel) the clock pulse (initiation of state transition signal), the transitions are simultaneous, i.e. synchronized.

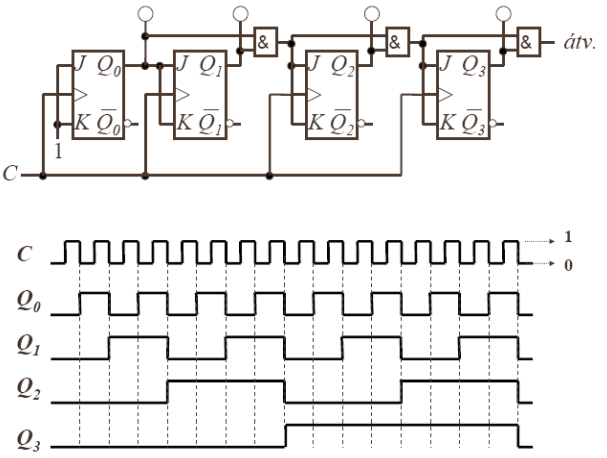
• The flip-flops are controlling each other, their output/sate variables are controlling through appropriate combinational networks the control inputs.

33

## SCHEMATICS OF SYNCHRONOUS COUNTER



# SYNCHRONOUS BINARY COUNTER (UP)




| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | DEC |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | =0 |
| 0 | 0 | 0 | 1 | =1 |
| 0 | 0 | 1 | 0 | =2 |
| 0 | 0 | 1 | 1 | =3 |
| 0 | 1 | 0 | 0 | =4 |
| 0 | 1 | 0 | 1 | =5 |
| 0 | 1 | 1 | 0 | =6 |
| 0 | 1 | 1 | 1 | =7 |
| 1 | 0 | 0 | 0 | =8 |
| 1 | 0 | 0 | 1 | =9 |
| 1 | 0 | 1 | 0 | =10 |
| 1 | 0 | 1 | 1 | =11 |
| 1 | 1 | 0 | 0 | =12 |
| 1 | 1 | 0 | 1 | =13 |
| 1 | 1 | 1 | 0 | =14 |
| 1 | 1 | 1 | 1 | =15 |

Series carry propagation, T flip-flop mode

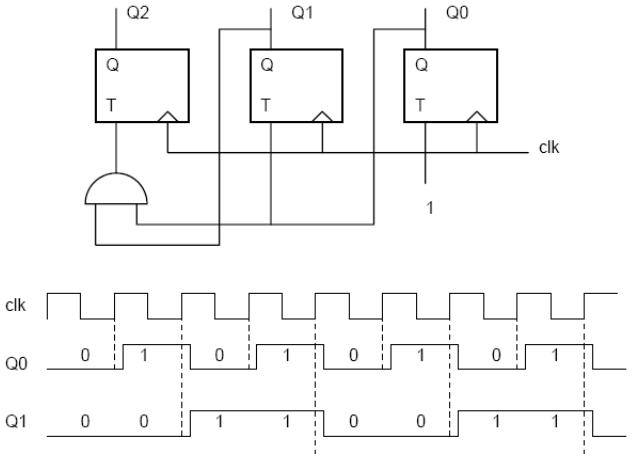
35

# SYNCHRONOUS BINARY COUNTER (UP)

Carry (between stages): level 1 appears when the stage contains the largest possible number (1111). The carry signal facilitates the chaining of subsequent stage(s).

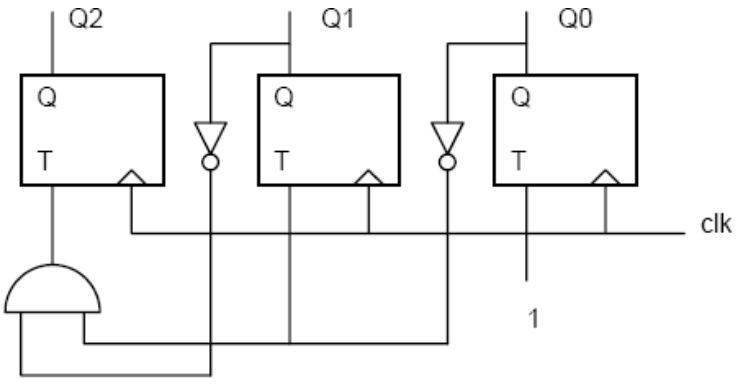
36

## SYNCHRONOUS COUNTER (UP)




Functional diagram of the synchronous binary up-counter. The outputs change their states simultaneously, their delay with respect to the clock is small.

## ASYNCHRONOUS COUNTER (DOWN)




Functional diagram of the synchronous binary down-counter.

## SYNCHRONOUS COUNTER: MAXIMUM FREQUENCY OF OPERATION

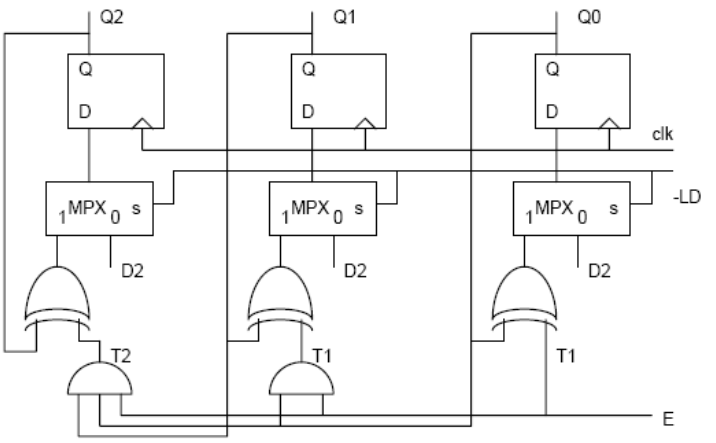
$t_{pd}$ - propagation delay,
$t_{dec}$ - decoding time,
$t_g$ - AND gate delay

$$f_{max} = \frac{1}{t_{pd} + t_g + t_{dec}}$$

Synchronous counters, in case of many stages, are significantly faster than ripple counters.

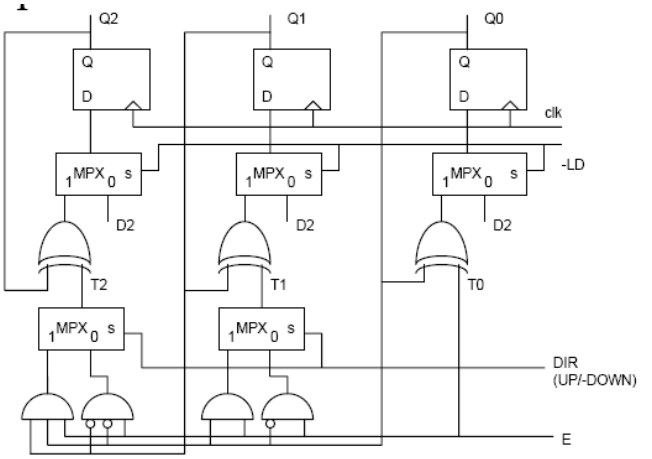
39

## SYNCHRONOUS BINARY COUNTER



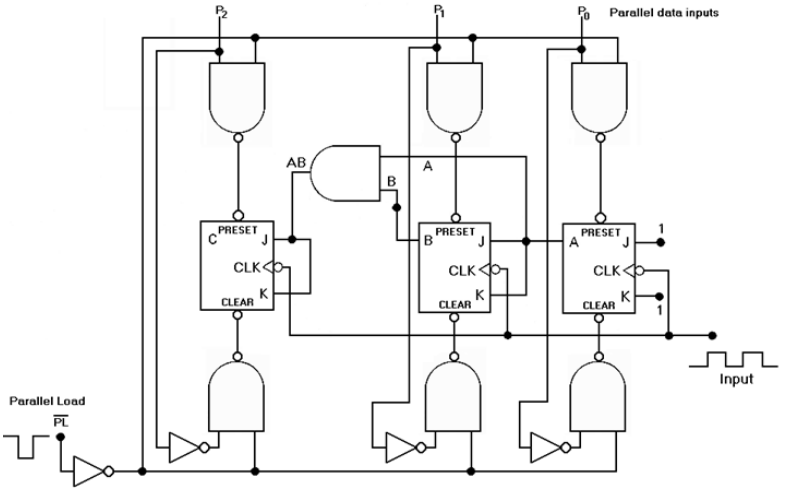

Functional diagram of the synchronous binary up-counter, with additional enable and (pre-)load functions.

## SYNCHRONOUS COUNTER




Functional diagram of the synchronous binary up/down counter, with additional enable and (pre-)load functions

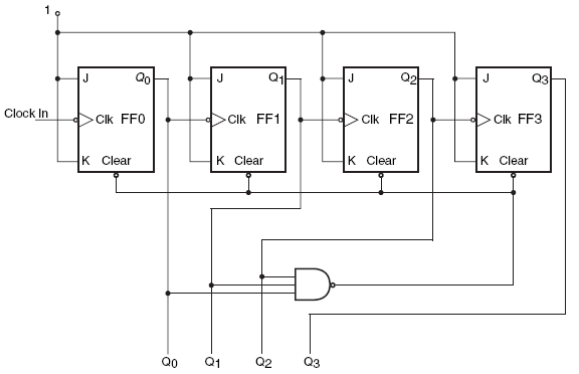
## LOADABLE/PRESETTABLE COUNTERS

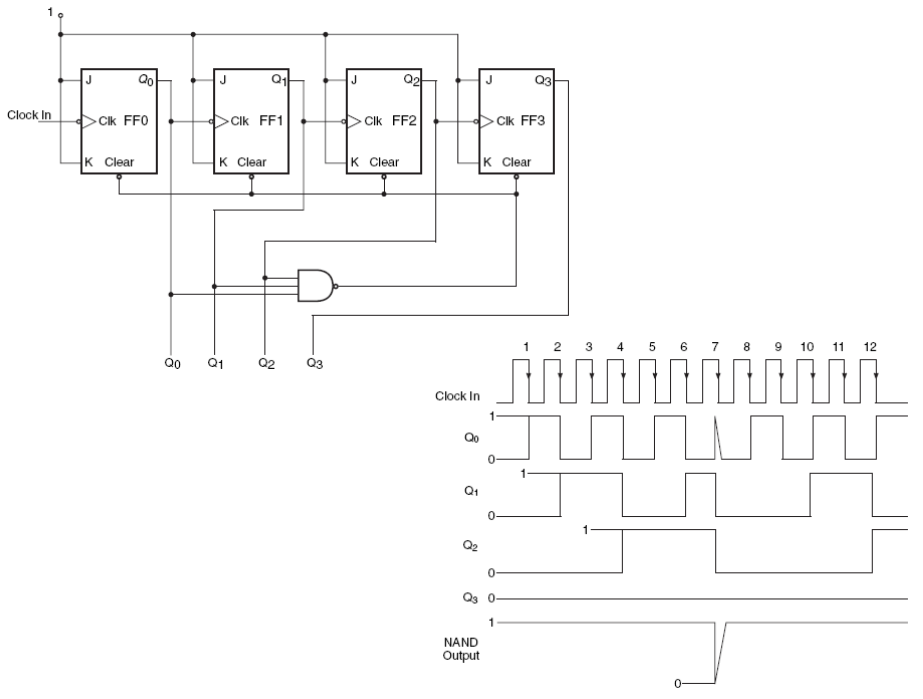



3-bit synchronous presettable counter

# BINARY RIPPLE COUNTERS WITH MODULUS LESS THAN $2^N$

An N-flip-flop binary ripple counter can be modified to have any other modulus less than $2^N$ with the help of simple externally connected combinational logic.



## COUNTERS WITH ARBITRARY SEQUENCES

So far we have discussed different types of synchronous and asynchronous counters. A large variety of synchronous and asynchronous counters are available in IC form.

The counters discussed hitherto count in either the normal binary sequence with a modulus of $2^N$ or with slightly altered binary sequences where one or more of the states are skipped. The latter type of counter has a modulus of less than $2^N$ , N being the number of flip-flops used. Nevertheless, even these counters have a sequence that is either upwards or downwards and not arbitrary.

## COUNTERS WITH ARBITRARY SEQUENCES

There are applications where a counter is required to follow a sequence that is arbitrary and not binary.

As an example, an MOD-10 counter may be required to follow the sequence 0000, 0010, 0101, 0001, 0111, 0011, 0100, 1010, 1000, 1111, 0000, 0010 and so on. In such cases, the simple and seemingly obvious feedback arrangement with a single NAND gate discussed above for designing counters with a modulus of less than $2^N$ cannot be used.

## SYNCRONOUS VERSUS ASYNCHRONOUS COUNTERS

It can be seen that a ripple counter requires less circuitry than a synchronous counter. No logic gates are used at all in the example above. Although the asynchronous counter is easier to construct, it has some major disadvantages over the synchronous counter.

First of all, the asynchronous counter is slow. In a synchronous counter, all the flip-flops will change states simultaneously while for an asynchronous counter, the propagation delays of the flip-flops add together to produce the overall delay. Hence, the more bits or number of flip-flops in an asynchronous counter, the slower it will be.

## SYNCRONOUS VERSUS ASYNCHRONOUS COUNTERS

Secondly, there are certain "risks" when using an asynchronous counter. In a complex system, many state changes occur on each clock edge and some ICs respond faster than others. If an external event is allowed to affect a system whenever it occurs (unsynchronised), there is a small chance that it will occur near a clock transition, after some IC's have responded, but before others have. This intermingling of transitions often causes erroneous operations. And the worse this is that these problems are difficult to foresee and test for because of the random time difference between the events.

## REVISION QUESTIONS (FLIP-FLOPS)

Perform the following conversions:

Convert SR flip-flop to D flip-flop.
Convert SR flip-flop to JK flip-flop.
Convert SR flip-flop to T flip-flop.
Convert JK flip-flop to T flip-flop.
Convert JK flip-flop to D flip-flop.
Convert D flip-flop to T flip-flop.
Convert T flip-flop to D flip-flop.
Convert JK flip-flop to SR flip-flop.
Convert D flip-flop to SR flip-flop.

## REVIEW QUESTIONS

1. Differentiate between:
(a) asynchronous and synchronous counters;
(b) UP, DOWN and UP/DOWN counters;
(c) presettable and clearable counters;
(d) BCD and decade counters.

2. Indicate the difference between the counting sequences of:
(a) a four-bit binary UP counter and a four-bit binary DOWN counter;
(b) a four-bit ring counter and a four-bit Johnson counter.

3. Briefly explain why the maximum usable clock frequency of a ripple counter decreases as more flip-flops are added to the counter to increase its MOD-number.

4. Why is the maximum usable clock frequency in the case of a synchronous counter independent of the size of counter?

## PROBLEMS AND EXERCISES

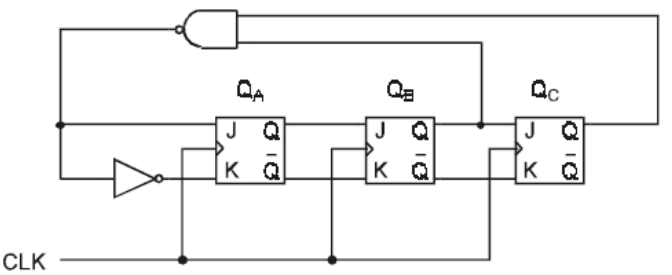
1. An eight-bit binary ripple UP counter with a modulus of 256 is holding the count 01111111. What will be the count after 135 clock pulses be?

(ANS: 00000110)

2. A certain J-K flip-flop has propagation delay of 12 ns. What is the largest MOD counter that can be constructed from these flip-flops and still operate up to 10 MHz?

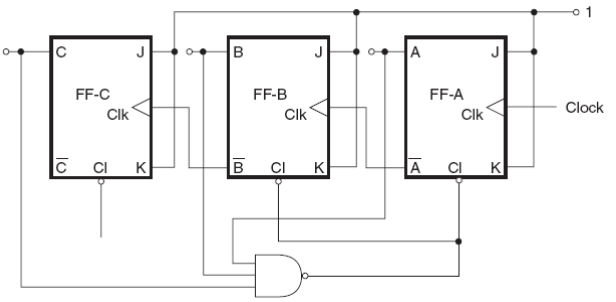
## PROBLEMS AND EXERCISES

3. Analyze the counter shown below for a “lock-up” condition in which the counter cannot escape from an invalid state or states. An invalid state is one that is not in the counter’s normal sequence.



# PROBLEMS AND EXERCISES

4. Refer to the counter schematic shown below. Determine the count sequence of this counter.




ANS: *000, 001, 010, 011, 100, 101, 110, 000,…*