COE 312

DIGITAL COMPUTER SYSTEMS ASSIGNMENT

PREPARED BY

AROH ARINZECHUKWU ANTHONY

17/ENG02/012

COMPUTER ENGINEERING

AFE BABALOLA UNIVERSITY, ADO-EKITI

SUBMITTED TO

ENGR. ADIGUN SAMSON

6TH MAY 2020

TOPIC: SUMMARIZE THE PDF MATERIALS ON FLIP FLOP SYNCHRONIZATION AND COUNTERS & REGISTERS

**PART 1**

**SUMMARY OF FLIP FLOP APPLIACTIONS**

The flip flop discussed in this section is the edge triggered flip flop also known as the clocked flip flop. They are used in storing data, transfer of data, latch, registers, counters, frequency division, memory, etc. The flip flop applications mentioned above make use of sequential circuits which are a combination of combinational circuit and a storage element. The output of a sequential circuit depends on the present value of the input signal as well as the sequence of past inputs.

**FLIP FLOP SYNCHRONIZATION**

There are two types of synchronization in flip flop. They are;

1. Synchronous
2. Asynchronous

Most digital systems are synchronous because signals change with clock transitions. In Asynchronous, flip flops are triggered with different clock, not simultaneously. While in Synchronous, all flip flops are triggered with same clock simultaneously. Synchronous is also faster than asynchronous in operation. Internally, a flip-flop (the term includes everything from simple D latches to more complex edge-triggered J-K master-slave flip-flops) is an asynchronous state machine.

**DATA STORAGE AND TRANSFER**

Flip flop has its common application to be for data or information storage. These data are grouped into registers. The data could ne numerical values like binary numbers. Transfer operation is most done on data stored in a flip flop. Transfer operation is simply the transfer of data from one flip flop or register to another.

There are two types of transfer:

1. Synchronous transfer: Synchronous control and *CLK* inputs are used for transfer here.
2. Asynchronous transfer: PRESETS and CLEAR inputs of any type of flip flop are used here for transfer.

**PARALLEL DATA TRANSFER**

This is the transfer of contents of the X register into the Y register because the contents of X2, X1, and X0 are transferred simultaneously into Y2, Y1 and Y0. It is a synchronous transfer.

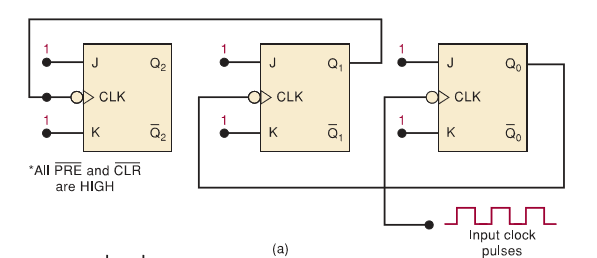
**SERIAL DATA TRANSFER: SHIFT REGISTERS**

A shift register is a group of flip flops arranged so that the binary numbers sored in the flip flops are shifted from one flip flop to the next for every clock pulse. They are used in electronic calculators.

**DIFFERENCES BETWEEN PARALLEL TRANSFER AND SERIAL TRANSFER**

|  |  |
| --- | --- |
| **PARALLEL TRANSFER** | **SERIAL TRANSFER** |
| 1. Information is transferred simultaneously upon the occurrence of a single transfer command pulse. | 1. Information worth N bits requires N clock pulses. |
| 1. This is faster than serial transfer using shift registers. | 2. This is slower than parallel transfer using  Shift registers. |
| 1. The output of each flip flop in register X is connected to a corresponding FF input in register Y. | 3. Only the last flip flop in register X is connected to register Y. |

**FREQUENCY DIVISION AND COUNTING**



From the diagram above, each Flip flop contain its J and K inputs at the level 1, for the function of toggle when the signal on its CLK input goes from HIGH to LOW.

Note:

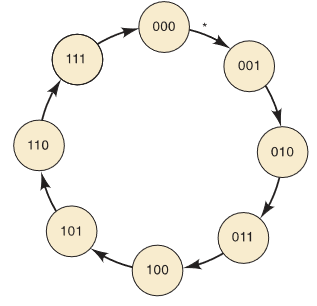
1. Q0 on flip flop toggles on the negative side of each input clock pulse. Therefore, Frequency of Q0 output waveform is one half of its clock pulse frequency.
2. Each time Q0 goes from HIGH to LOW, flip flop Q1 toggles.
3. Each time Q1 goes from HIGH to LOW, flip flop Q2 toggles.
4. Flip flop outputs are all square wave.

**Counting Operation**

Counting operation can be carried out by checking the sequence of states of the FFs after the occurrence of each clock pulse. The stable table consists of binary numbers Q2Q1Q0. Q1 is in the 21position, Q2 is in the 22position and Q0is in the 20 position. The binary counting sequence from 000 to 111 are the recognized by the first eight states which are Q2Q1Q0.

**State Transition Diagram**

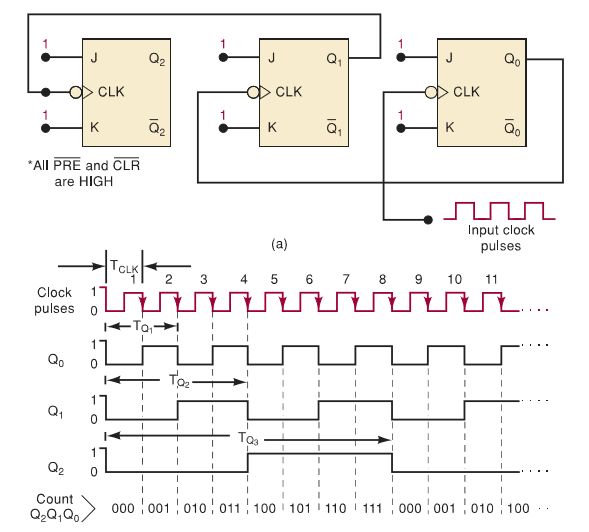
The state transition shows how the states of FFs change with each applied clock pulse as shown below;



The arrows in the state transition diagram represent the occurrence of a clock pulse. Each of the 8 circles represent a possible state. The state transition diagram can be used to analyze, describe and design counters and other sequential circuits.

**MOD NUMBER**

The MOD number indicates the number of states in the counting sequence. **MOD counters** are made using “flip-flops” and a single flip-flop can produce a **count** of 0 or 1, giving a maximum **count** of 2.The MOD-8 counter is an example of a MOD counter and can be converted to a MOD-16 counter, if a fourth FF were added, the sequence of states would count in binary from 0000 to 1111, a total of 16 states. That is a MOD-16 counter. MOD-2N are formed in N flip-flops connected in the diagram below. They have 2N different states.



**PART 2**

**SUMMARY OF COUNTERS AND REGISTERS**

**ASYNCHRONOUS (RIPPLE) COUNTERS**

**OPERATION OF AN ASYNCHRONOUS COUNTER**

1. Clock pulses are applied only to CLK input of flip flop A. So therefore, flip flop A toggles each time the clock pulse makes a negative transition.
2. Flip flop A has its normal output which acts the CLK input for flip flop B, so flip flop B toggles whenever the A output goes from 1-10.
3. Outputs D, C, B and A all represent four-bit binary number as the MSB.
4. The counter flip flops are in 1111 condition after the NGT of the fifteenth clock pulse has occurred.

**CHARACTERISTICS OF AN ASYNCHRONOUS COUNTER**

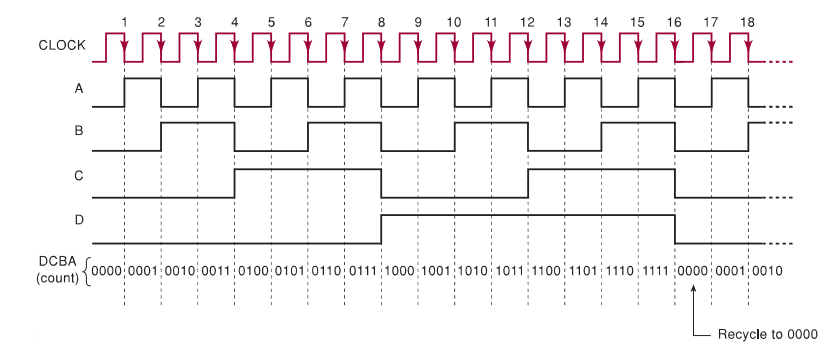
1. Each output drives the CLK input of the next flip flop.
2. The flip flops do not change states in exact synchronism with the applied clock pulses.
3. Only flip flop A responds to the clock pulses.
4. Flip flop B waits for flip flop A to change state before it can toggle.
5. There is a delay between the responses of successive flip flops of 5-20 ns per flip flop.

**MOD NUMBER**

MOD number is equal to the number of states that the counter goes through in each complete cycle before it recycles back to its initial state. Increase of the MOD number can be done by adding more flip flops to the counter.

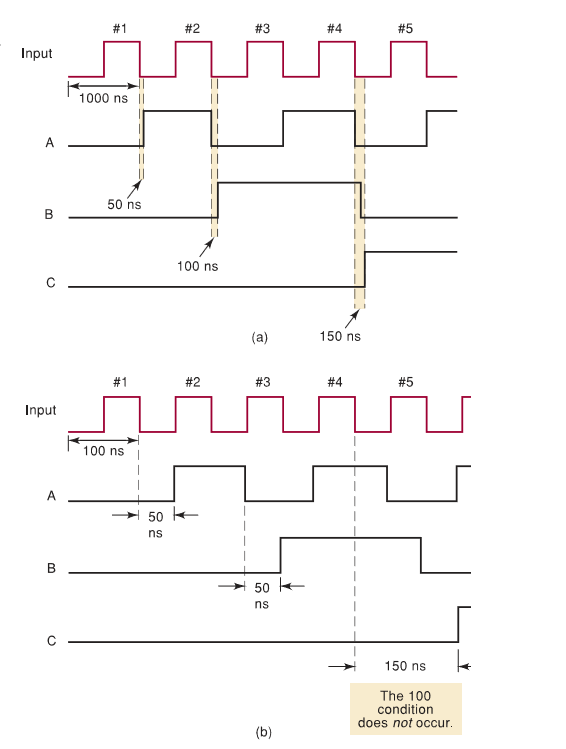
i.e. MOD number = 2N

where N is the number of Flip flops connected in the arrangement below;



**PROPAGATOIN DELAY IN RIPPLE COUNTERS**

These are referred to as the easiest binary counters as they require the fewest components to produce a given counting operation. One of the problems of a ripple counter is its basic principle which states that each flipflop is triggered by the transition at the output of the preceding flipflop. Due to the propagation delay, the second flip flop doesn’t respond until a time after the first flipflop receives an active clock transition. The third flip flop as well doesn’t act until a time equal to 2xdelaytime after that clock transition.



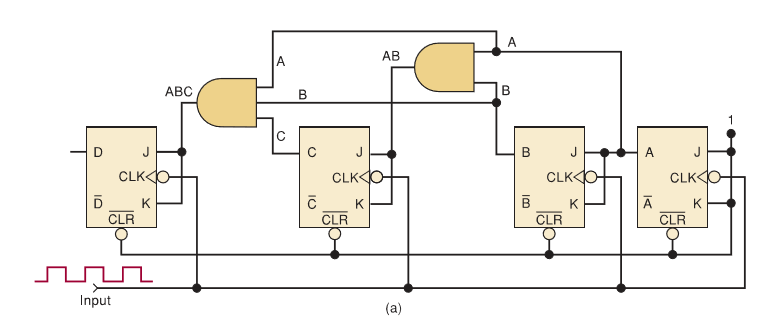
**The diagram above is Waveforms of a three-bit ripple counter showing the effects of flipflop propagation delays for different input pulse frequencies.**

It shows what happens when the input pulses occur once every 100ns.

When it comes to counters with large number of bits, asynchronous counters at very high frequencies are not useful. Problems such as glitch limits the applications of asynchronous counters in the digital world.

**SYNCHRONOUS (PARALLEL) COUNTERS**

Ripple counters often encounter problems which are caused by accumulated flip flop propagation delays. With the use of synchronous counters, all flip flops will be triggered simultaneously (parallel) by the clock input pulses which will overcome the limitations. The diagram of a synchronous counter is as shown below:



**Advantages of Synchronous Counters over Asynchronous**

1. The major advantage of a synchronous counter over an asynchronous counter is that, in the synchronous counter, all the flipflops change simultaneously while in the asynchronous counter, the propagation delays of the flipflops do not add together to produce the overall delay. Simply, for the synchronous counters, all the flipflops are all synchronized to the NGTs of the input clock pulses.
2. The synchronous counter has a more complex circuitry than that of the asynchronous counter.
3. The synchronous counter can operate at a higher input frequency than the asynchronous counter.

**Decoding a counter**

Decoding of a counter finds its application as to control the timing or sequencing of operations automatically without human intervention.

**Active-HIGH Decoding**

This decoder produces HIGH outputs to show detection. It consists of eight three-input AND gates. Each AND gate produce a HIGH output for one particular state of the counter. This decoder’s outputs can be designed to produce a HIGH or a LOW level when the detection occurs.

**Active-LOW Decoding**

This decoder produces an output of HIGH signal which goes LOW only when the number being decoded occurs if the NAND gates are used in place of AND gates.

**BCD Counter Decoding**

The BCD decoders provide 10 outputs corresponding to the decimal digits through 9 and represented by the states of the counter flipflops. They can be used to control individual indicator LEDs for visual display. The BCD Counter has 10 states that can be decoded using various techniques.

**Analyzing Synchronous Counters**

Synchronous counters can be easily designed to produce count sequence using the synchronous inputs which are applied to the individual flipflops. In analyzing counter design, a PRESENT state and a NEXT state is very useful.

Step 1: Write out the logic expression for each flipflop control input.

Step 2: Assume a PRESENT state for the counter and apply the combination of bits to the control logic expressions. The output from this logic expression allows for the prediction of commands to each flipflop and the NEXT state for the counter after clocking.

Step 3: Repeat the process over and over again until the entire count sequence is determined.

**Synchronous counter design**

There are so many different arrangements of counters and most of them are designed to count in a regular binary or BCD count sequence, although their counting sequence can be altered using several clearing methods.

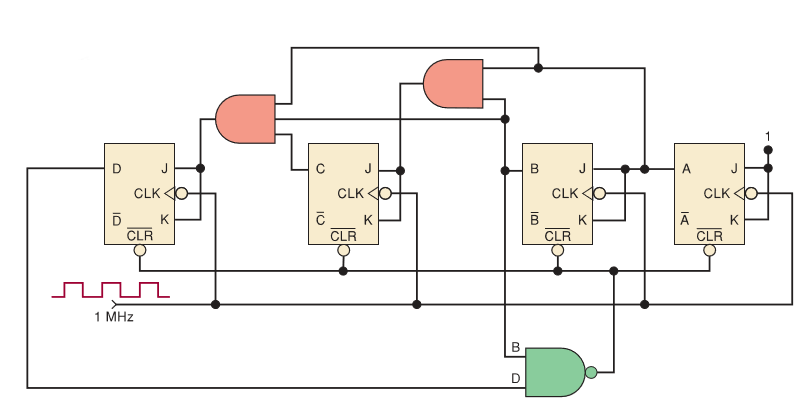
One method used for designing counters uses a technique where several design procedures are part of digital circuits known as sequential circuit design. These method follows an arbitrary sequence.

Synchronous counter can be designed using logic circuits that decode various states of the counter to supply proper logic levels to each J and K input at the correct time. The inputs of the decoder therefore, comes out from the outputs of one or more flip flop.

**Decade Counters**

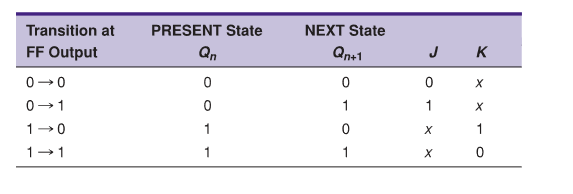
Decade counters are counters that contain 10 distinct states, no matter the sequence.

The diagram of a type of decade counter which is the BCD counter is given below:



It counts in sequence from 0000 through 1001. It is often used to divide pulse frequency exactly by 10. It has its applications commonly where pulses are to be counted and the results displayed on some type of decimal numerical readout.

**JK Excitation Table**



The table above is the JK excitation table.

0 0 Transition: This means that the flipflop state is at 0 and it remains at 0 when a clock pulse is applied. In the JK flip flop it gives an output of 0 and *x* under J and K respectively which means a “don’t care condition”.

0 1 Transition: This means that the present flip flop state is at 0 and is to undergo a change to 1, which can either occur when either J=1 and K=0 (set condition) or J=K=1 (toggle condition). Simply, for this transition to occur J must be equal to 1 but K can be either at 0 or 1.

1 0 transition: The present state is 1 and is to change to 0, which can occur when either J=0 and K=1. There must just be K=1 while J can be on any level (0 or 1) at all.

1 1 transition: Both the present state and the changing state are 1 in this transition. This occurs when either J=K=0 or J=1 and K-0. There must be K = 0 while J can be on any level (0 or 1) at all.

**Design Procedure**

Step 1: The desired number of bits (flipflops) and the desired counting sequence are determined.

Step 2: The transition diagram is drawn showing all possible states, including those not part of the desired counting sequence.

Step 3: The transition diagram is used to set up a table that lists all present states and their next states.

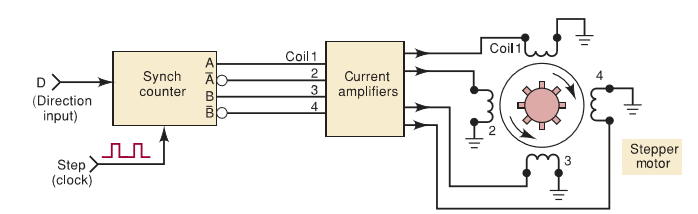
Step 4: A column is added to the table for each J and K input. For each present state, indication of levels required at each J and K inputs in order to produce the transition to the next state.

Step 5: The design of the logic circuits is needed to generate the levels required at each J and K input. A K-map of these expressions are also formed.

Step 6: Implementation of final expressions. In this stage, the J K inputs are implemented from the expressions obtained from the K map.

**Stepper Motor Control**

A stepper motor is a motor that rotates in steps, typically 150 per step, rather than in a continuous motion. Energization and de-energization of magnetic coils must be carried out windings within the motor in a specific sequence for the sole aim of producing stepping action. Currents in motor coils can be controlled by the use of digital signals. One of the major applications of a stepper motor is in situations where precise control is needed e.g. controlling robots, reading magnetic disks, etc.



**Diagram illustrating how a synchronous counter supplies the appropriate sequential output to a drive stepper motor.**

**Synchronous Counter Design with DFF**

Counter circuits can be easily designed using D flipflops than JK flipflops. In the past JK flipflop have been used to implement counters because the logic circuits needed for the J and K inputs are usually simpler than the logic circuits needed to control an equivalent synchronous counter using D flipflops. D flip flops are used in designing counters implemented in the PLDs where abundant gates are available.

**Integrated Circuit Registers**

There are 4 various types of registers and they are all classified according to the manner in which data is entered into the, for storage as well as outputted from the register. They are;

1. Parallel in/serial out (PISO): This is an eight bit Parallel in/serial out register with data entry.
2. Serial in/parallel out (SIPO): This is an eight bit Serial in/parallel out shift register with each flipflop output externally accessible.
3. Parallel in/parallel out (PIPO): This register stores multiple bits simultaneously. All the bits of the stored binary value are directly available in the register.
4. Serial in/ Serial out (SISO): This contains data loaded into it one bit at a time. This movement goes along each clock pulse through the set of flipflops toward the other end of the register.

**Ring Counter**

A ring counter has its name cause in most cases only a single 1 is in the register, and it is made to circulate around the register as long as clock pulses are applied. The simplest shift register is the circulating shift register connected so the last flipflop shifts its value into the first flipflop. Ring counters can be easily designed for any desired MOD number. Each flipflop output waveform has a frequency equal to one-fourth of the clock frequency in a MOD-4 ring counter.