كلية الهندسة - جامعة الزقازيق

الفرقة: الثالثة هندسة الحاسبات والمنظومات

المقرر: دوائر الحاسب المتكاملة

الإسم: ابراهيم احمد سعد مرسي

الرقم في السكشن: 2

رقم الجروب وإسم الموضوع: جروب 5 (Digital clock)

Digital clock

Brief:

Our project is a digital clock which displays time in hours, minutes and second and displays today's date.

Component used:

- 555 Timer
- Counter 74LS90
- 74ls48 decoder
- Common cathode 7-seg
- 74LS157 2 to 1 Multiplexer
- 74S283 4-Bit Binary Adder

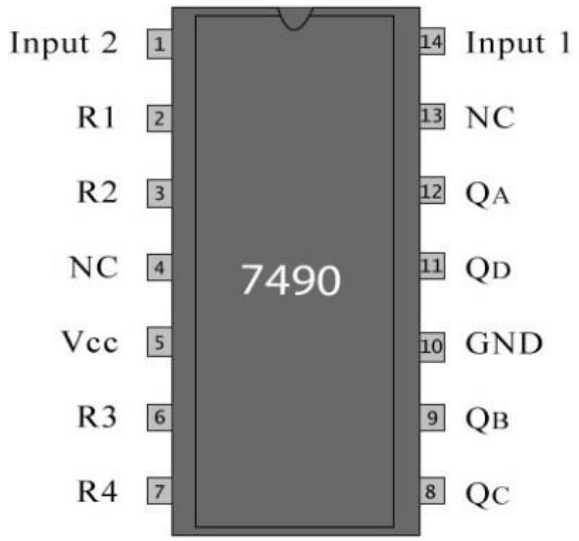
How we implement this project?

- Generating the frequency
  we used 555-timer to generate 50 HZ frequency.

- Handling time in hours, minutes and seconds **(My task)**
  We used counter 74LS90 to handle the time, and the following is a detailed discussion for how we did it.

74LS90 counter details:

- **IC ship**



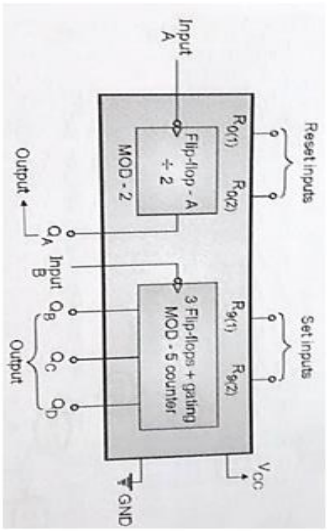


Fig10. The basic internal structure of IC 7490

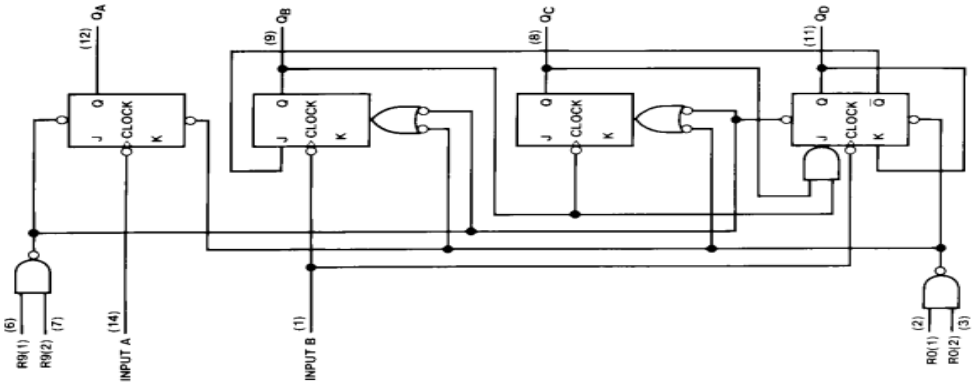
It mainly consists of [(mod-2 counter) and (mod-5 counter)]

These are the functions of the legs in details:

- Input 1: input frequency for mod-1 counter
- input 2: input frequency for mod-5 counter.
- [Qa, Qb, Qc, Qd]: are four legs for output.
- R1: reset mod-2 counter to zero.
- R2: reset mod-5 counter to zero.
- R3: reset mod-2 counter to 9.
- R4: reset mod-5 counter to 9.

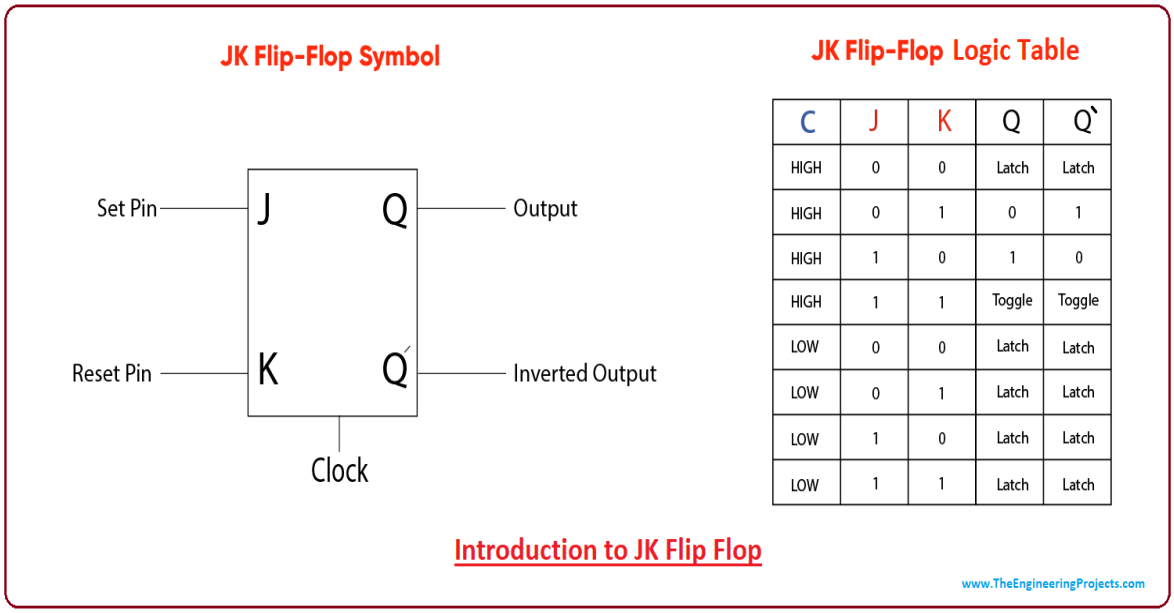
➢ Counter structure

This is the circuit design from datasheet:



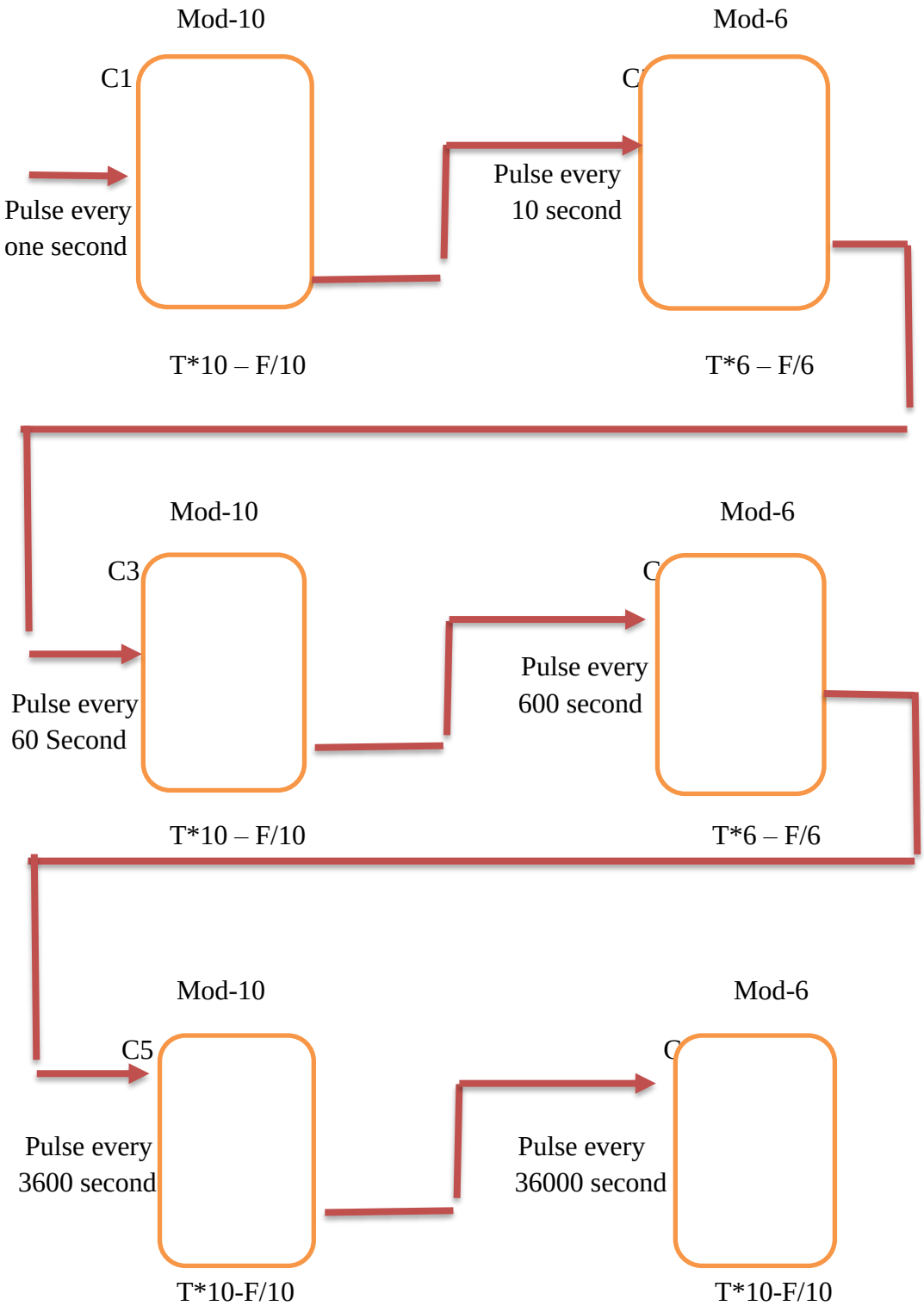
It has four jk-flip flops and some logic gates which is used for resetting the counter after certain counts.

jk-flip flop has the following behavior:



C	J	K	Q	Q`
HIGH	0	0	Latch	Latch
HIGH	0	1	0	1
HIGH	1	0	1	0
HIGH	1	1	Toggle	Toggle
LOW	0	0	Latch	Latch
LOW	0	1	Latch	Latch
LOW	1	0	Latch	Latch
LOW	1	1	Latch	Latch

This is an overview that shows the counters connections and mods to generate a signal with the desired period and frequency.



In this project we used 74LS90 counter in 3 different modes [(10-mode) – (6-mode) – (5-mode)]

1. Mod-10 counter

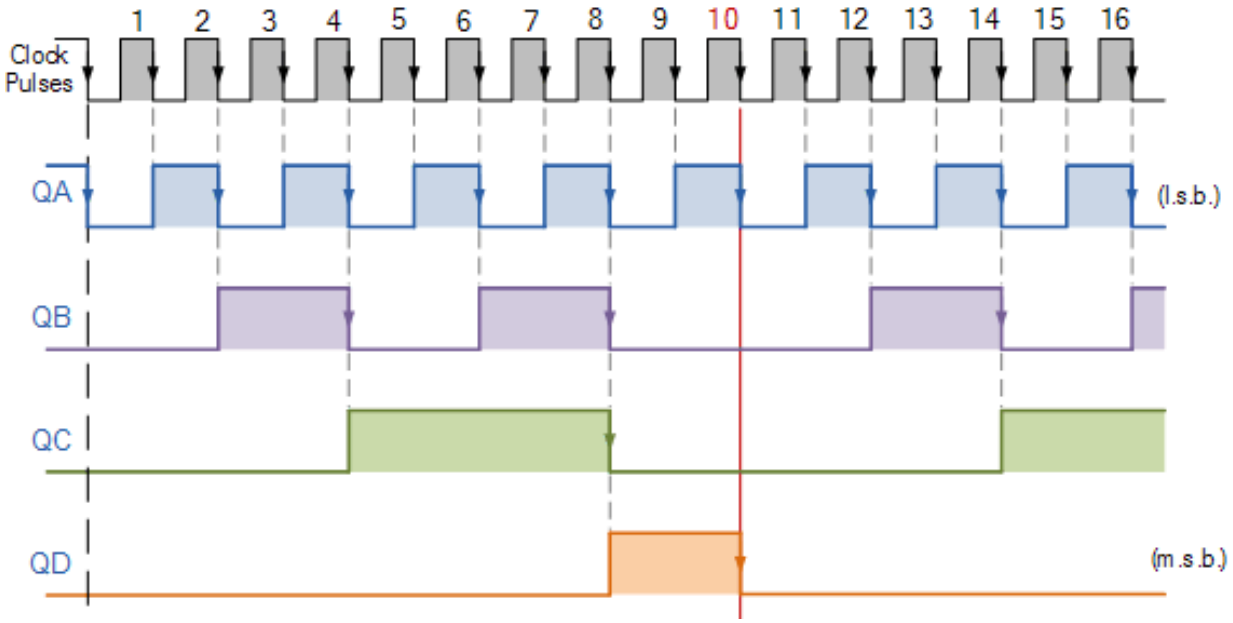
In this mod the counter reset after 10 counts and divide its input frequency by 10

The behavior in this mod can be described by the following truth table and timing diagram:

- **truth table for mod-10 counter:**

Input Pulses	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
0	0	0	0	0 (resets)

➢ **Timing diagram for mod-10 counter:**

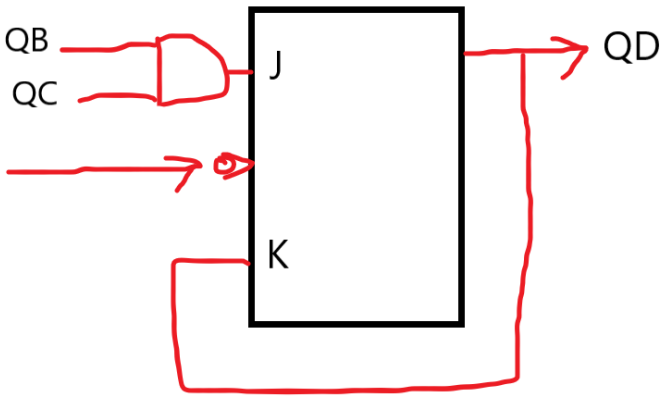


From timing diagram, we notice that QA toggle its state with every failing edge of the clock, and each of the rest legs [QB, QC, QD] toggle its state with the failing edge of the previous leg.
And this toggle operation is the role of JK-flip flop which is built-in component in 7490 counters.

❖ **Rest operation in mod-10 7490 counter**

Mod-10 counter should count from [0 : 9] then reset to Zero again and so on, on other words when the output is about to be 10 -> (1010) it must return back to 0 -> (0000).
So, exactly the legs which I want to rest to zero after 10 counts are [QB, QD].
And that exactly what 7490 counters do as a default behavior by some built-in logic gates…

For QD leg:
below is a detailed discuss for how this leg reset after 10 counts…



This JK-flipflop receives its input freq. from QA (failing edge every 2 sec.)

1. At the start of counting (0000) QD is low

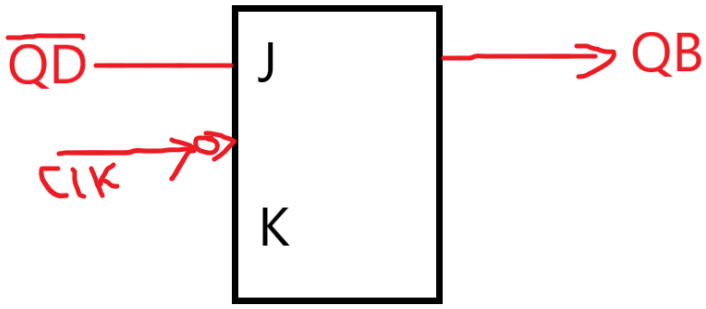
2. When the counter reaches 7 (QD -> 0, QC -> 1, QB -> 1, QA -> 1)
   So, (JD -> 1) and (KD -> 0)

3. At the next failing edge QD -> 1, and the rest legs toggle to zero
   So, we have (QD -> 1, QC -> 0, QB -> 0, QA -> 0) = 8
   And (JD -> 0) and (KD -> 1)

4. At the next failing edge (which is after 2 sec.) exactly at 10:
   (QD -> 0)

So, the leg QD reset to zero after 10 counts.

For QB leg:
below is a detailed discuss for how this leg reset after 10 counts…



This JK-flipflop receives its input freq. from QA as QD (failing edge every 2 sec.)

At the 9th count (QD -> 1), ((!QD) -> 0), (J -> 0)

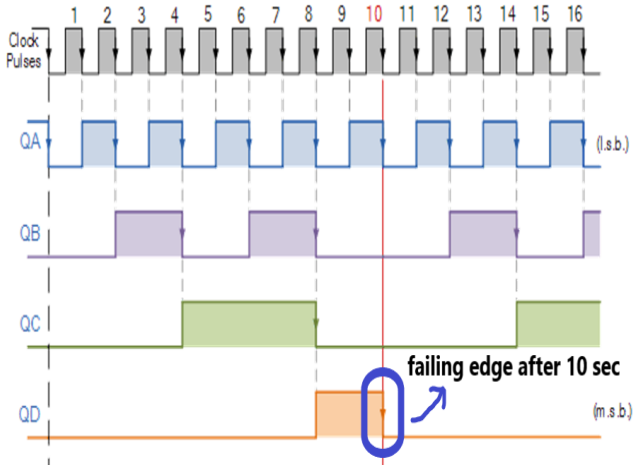
So, at the next failing edge (at 10) the QB leg rest to zero

Now we have a counter which counts form 0 up to 9 then returns back to 0 and so on.

❖ **How does mod-10 counter divide the frequency by 10?**

If we notice the truth table and timing diagram:

Input Pulses	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
0	0	0	0	0 (resets)



QD has a failing edge every 10 seconds.
So, the output frequency on QD is equal to the counter input frequency divided by 10.

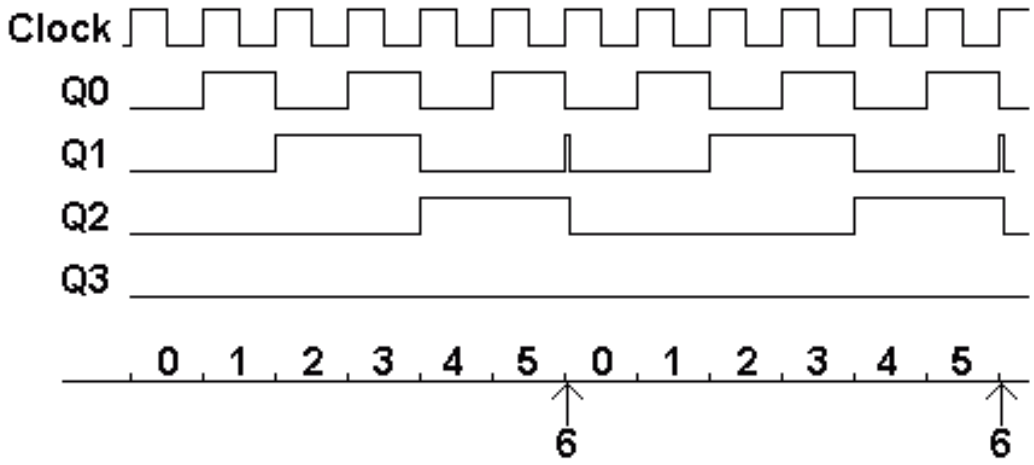
2. Mod-6 counter

The behavior in this mod can be described by the following truth table and timing diagram:

- Truth table for mod-6 counter:

Input pulse	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
0	0	0	0	0

- Timing diagram for mod-6 counter:

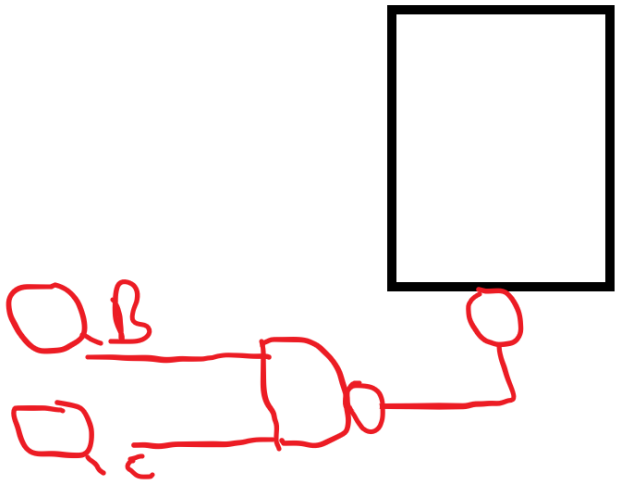


❖ **Reset operation for mod-6 counter:**

Here the reset operation is handled manually (unlike mod-10 counter which is reset at 10 by default)

So, the solution here to pass a pulse to the reset legs at the 6^{th} count when the legs carry the following data: (QD -> 0, QC -> 1, QB -> 1, QA -> 0)

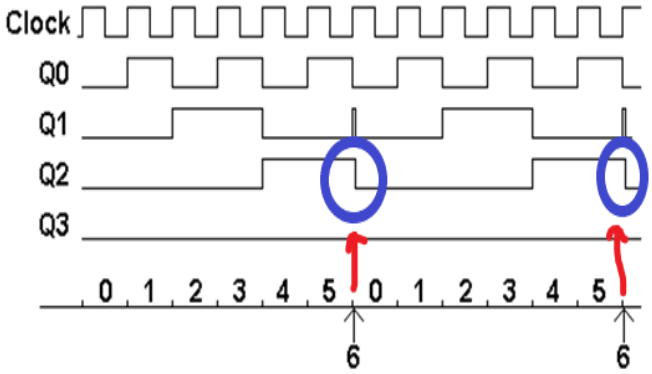
So, we can use AND gate with input QC and QB and the connect the output to reset leg as following:



❖ **How does mod-6 counter divide the frequency by 10?**

If we notice the truth table and the timing diagram:

Input pulse	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
0	0	0	0	0



QC has a failing edge every 6 sec.

So, the output frequency on QC is equal to the input frequency divided by 6

Sec, minutes, hours implementation

We used these two mods to implement sec, minutes, hours and date as following:

1. **Generating a 1 HZ signal**
   I receive from the timer a signal with frequency 50 HZ, then I make this signal pass first on (mod-10) counter then (mod-5) counter to divide the frequency by 50 and I now have a 1 HZ signal.

2. **Handling second**
   So, the next counter we used is for seconds **(specially for units' digit)** and it used in (mod 10).
   It receives a falling edge every 1 second and it counts a count with each pules (from zero to nine) and reset again Zero.
   And the output on QD is a falling edge every 10 second.
   We connect this leg (QD) with the input of the next counter which is for seconds **(specially for tens' digit)** and it used in (mod 6).
   It receives a falling edge every 10 second, and it counts a count with each pules (from zero to six) and reset again Zero.

And the output on QC is a falling edge every 60 second and pass it for the next counter.

3. **Handling minutes**
   So, the next counter we used is for minutes **(specially for units' digit)** and it used in (mod 10).
   It receives a falling edge every 60 second from QC of the previous counter, and it counts a count with each pules (from zero to nine) and reset again Zero.
   And the output on QD is a falling edge every 600 second.
   We connect this leg (QD) with the input of the next counter which is for minutes **(specially for tens' digit)** and it used in (mod 6).
   It receives a falling edge every 600 second, and it counts a count with each pules (from zero to six) and reset again Zero.
   And the output on QC is a falling edge every 3600 second.

4. **Handling hours**
   Now hours counter **(units' digit)** receives a failing edge every 3600 second, and it is mod-10.
   So, it passes a failing edge every 36000 second to the next counter which is for **(tens' digit)** for hours, and it is mod-10.

Here is the end of my Task and below are the rest steps which was done by my colleagues.

- **Displaying hours**
  We display hours in 12-hours mod and use an indication to clarify (am, pm) and this task is handled by my colleague and he made a detailed discussion for that.

- **Handling today's date**
  We uses the same concept and component we used to handle the time (7490 counter), but to make the counter of days and the counter of months reset to 1 (note to zero as default) we use Multiplexer to.
  We show months in digits and in letters using 7-Seg.

- **Displaying time and date on 7 Segments**
  we used 7 segments decoder and BCD 7 segment.

- **Simulation this project on Proteus program**

References

- **Logic Circuit Design course of the previous year.**
- **74LS90 counter datasheet.**
- **YouTube.**