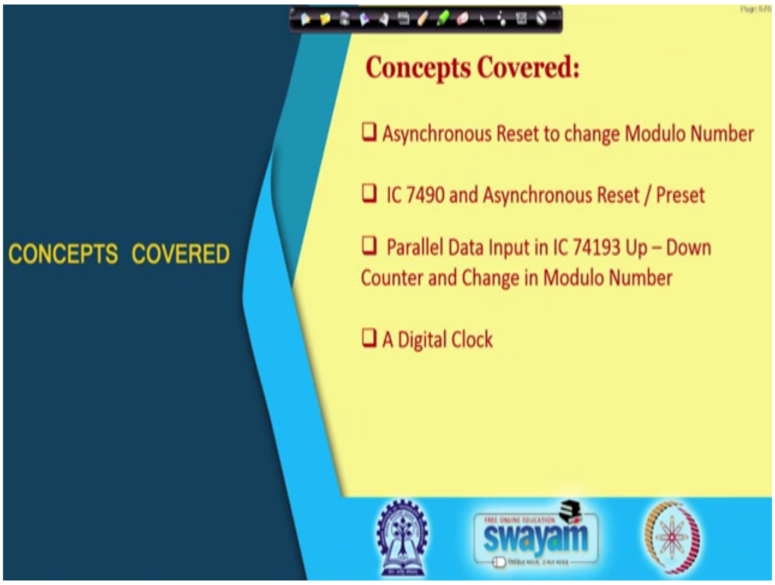
**Digital Electronic Circuits**
**Prof. Goutam Saha**
**Department of E & EC Engineering**
**Indian Institute of Technology, Kharagpur**

# Lecture – 44
# Counter Design with Asynchronous Reset and Preset

Hello everybody. In this class, we shall look at Counter Design using Asynchronous Reset and Preset.

(Refer Slide Time: 00:33)



CONCEPTS COVERED

Concepts Covered:

❑ Asynchronous Reset to change Modulo Number

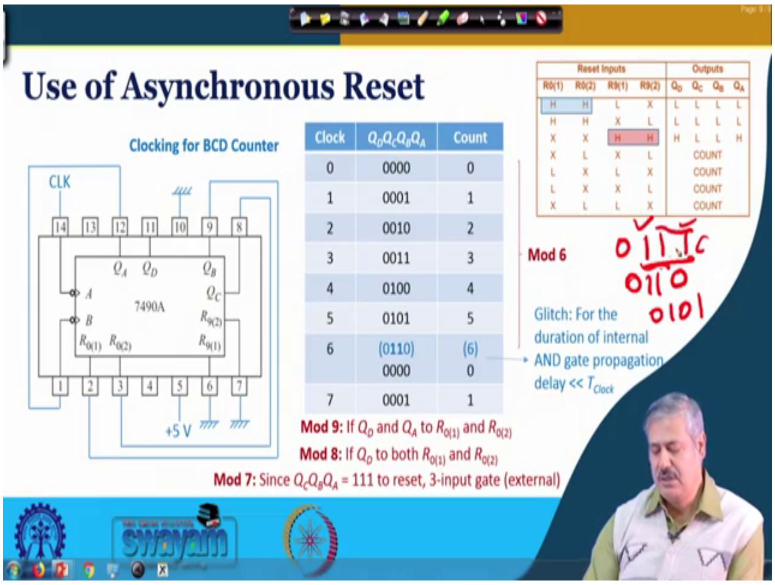
❑ IC 7490 and Asynchronous Reset / Preset

❑ Parallel Data Input in IC 74193 Up – Down Counter and Change in Modulo Number

❑ A Digital Clock

We have been seeing in our earlier classes that some of the IC's; counter IC's had a having internal AND gate logic and gate and there are two inputs to it. And we told that from the truth table they can be used for resetting purposes.

(Refer Slide Time: 00:50)



Use of Asynchronous Reset

Clocking for BCD Counter

| Clock | $Q_DQ_CQ_BQ_A$ | Count |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | (0110)<br>0000 | (6)<br>0 |
| 7 | 0001 | 1 |

Mod 6

Glitch: For the duration of internal AND gate propagation delay << $T_{Clock}$

| Reset Inputs | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|
| R0(1) | R0(2) | R9(1) | R9(2) | $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ |
| H | H | L | X | L | L | L | L |
| H | H | X | L | L | L | L | L |
| X | X | H | H | H | L | L | H |
| X | L | X | L | | COUNT | | |
| L | X | L | X | | COUNT | | |
| L | X | X | L | | COUNT | | |
| X | L | L | X | | COUNT | | |

0111 C
0110
0101

**Mod 9:** If $Q_D$ and $Q_A$ to $R_{0(1)}$ and $R_{0(2)}$

**Mod 8:** If $Q_D$ to both $R_{0(1)}$ and $R_{0(2)}$

**Mod 7:** Since $Q_CQ_BQ_A$ = 111 to reset, 3-input gate (external)

And in IC7490 we had seen that there can be a preset of 1001 as well again there is an internal AND gate.

So, we actually were surprised why such thing is there, why that internal and get is there why not only one pin. So, it will be clear from today's discussion that how a synchronous reset can be used to get different kind of modular number and those internal those inputs and the internal AND gate can come useful in this particular context. So, that will require that will eliminate the requirement of additional external AND gate and placing another IC taking more space, wearing and other things. So, those can be prevented.

So, let us see IC7090 A with that we are looking at. So, it is we are already familiar with this circuit this IC. It is a mock 10 counter and if we connect it appropriately; that means, mod 2 followed by mod 5, then it acts like a BCD counter. So, BCD counter means it goes from 0000 0001 up to 1001 that is 9 decimal equivalent 9, then again it goes to 0000 so that we that is what we know we have seen in the previous class right. So, the IC 7490 this particular BCD counter has been connected in this manner ok. So, clock is fed to mod to counter and QA output is fed to s clock to mod 5 counter and restart the ABCD or the QA QB QC QD, these are the outputs ok

So, the count in that particular counter with clocking as given here will go on as I have mentioned here 0 0 0 0 to 1 0 0 1 and then again 0 0 0 0 ok. This we are already familiar with. Now if you are connecting QB and QC output 8 and 9, output 1 of them is going to

2 another is going to 3 right. So, if you do that, what will happen? So, these are R 01 and R 02 ok. So, these asynchronous reset input internally there is an AND gate ok. So, when both of them are high when both of them are high, then what happens? This particular outputs are asynchronously reset this counter is a synchronously reset; that means, immediately after a short time of that propagation delay of the internal AND gate all the values gate 0 this QA QB QC QD all of them acquired the value 0 ok.

So, the counter is having a short very short duration some number associated with this and then it goes to 0 0 0 0 ok. So, with the count happening 0 0 0 0 0 0 0 1 that way what is the first instance when this two become 1 1? So, that it it goes to 0 0 0 0 again it goes on counting, similarly I mean the way the clock gate gets triggered see. If we follow the counting sequence we can see that 0 0 0 0, then 0 0 0 1 0 0 1 0 0 0 1 1 you will note that QC and QB are fed ok.

So, this is your QC and this is your QB. After that 0 1 0 0 0 1 0 1 and then comes 0 1 1 0. So, this is the first time in this counting sequence as you see that QC and QB both of them are becoming 1 ok. And when both of them are becoming 1 immediately; that means, after that short propagation delay which is much you know less than the time period of the clock ok. So, what is happening? It is immediately going to 0 0 0 0. So, in that particular clock cycle, you are effectively having for larger duration effectively you are having 0 0 0 0 so; that means, you are back to this particular number this particular state. Is it fine?

So, what are the then unique states for which the clock period? During the clock period the counter holds a specific value. So, these are the 1 0 0 0 0 to 0 1 0 1 ok. After that it again goes to 0 0 0 for very small duration staying at 0 1 1 0 ok. So, effectively you are having 0 1 2 3 4 5 and again back to 0 right; so, 6 unique states ok. So, this is called this is behaving like a mod 6 counter and if you had required to do this by external connection, then we would have required a two input AND gate and then that output would have been fed to one of the input.

So, that additional circuitry varying and you know putting another IC 7408 powering it and all those things are not required and that need is eliminated. Now having noted this that 0 0 0 0 to you know 0 1 0 1 and the value where with which wherever it goes, then you know that and 0 0 stays together in this particular mode ok. So, if QD and QA right;

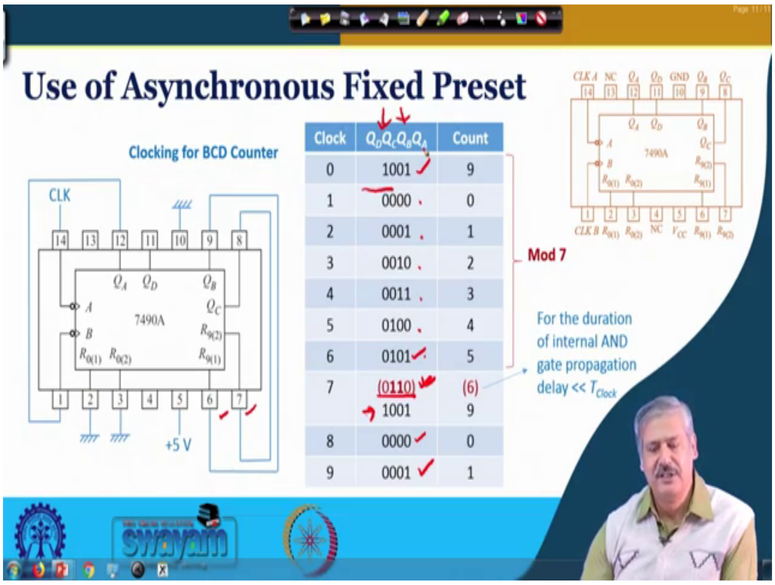
this is QD and this is QA right if they are sent here instead of QC and QB right QC and QB are sent. Now we are sending QD and QA to this a synchronous reset. What will happen then ok?

So, it will go from 0 0 0 0 to first time whenever it goes to both of them QD and QA becomes 1 that is 1 0 0 0 1, immediately it get gets reset. So; that means, for a very short duration it will be there and then 0 0 0 0 will be there right. So, effectively you are having 0 to 8; 1 0 0 0 and in this particular clock cycle 0 0 0 0 is again getting again appearing because of this asynchronous reset. So, effectively you are having 0 to 8; that means 9 unique states. So, it behaves like a mod 9 counter ok. If you want a mod 8 counter so, what is the first location? When you know this is how this is the count so, 0 0 0 0. So, 1 0 0 0 this is your this is the time when you want to reset it ok.

So, before that you had 0 1 1 1 right. You want 1 0 0 0 and 0 0 0 0 at 1 particular clock cycle. This is the one with which you want to reset it ok. So, 1 has not come before 1 is coming for the first time right. So, 1 QD you can feed to just take a wire and connect to both of them; that is fine internally it is getting ended both the inputted over. So, that way you can get mod 6, that way you can get mod 8, that way you can get mod 9 right. But how if you want to get a mod 7 right, then the first time it is happening it is 0 1 1 1 ok.

Now, with two inputs ok, it is not possible because if you give just this two, then it is mod 6 ok. First time it will occur when 0 1 1 0 and if you want to give these two inputs right that will also occur when it is 3 and if you want to give QC and QA that occurs when it is 5 isn’t it. So, you need all the 3 inputs 0 I mean this QC QB QA for which you will be needing an external gate in this particular arrangement, is it fine ok. But otherwise you can see that for the other cases, we are having this you know other modulo number obtained by just connecting weights in this manner.

(Refer Slide Time: 09:38)



## Use of Asynchronous Fixed Preset

Clocking for BCD Counter

| Clock | $Q_DQ_CQ_BQ_A$ | Count |
|---|---|---|
| 0 | 1001 ✓ | 9 |
| 1 | 0000 | 0 |
| 2 | 0001 | 1 |
| 3 | 0010 | 2 |
| 4 | 0011 | 3 |
| 5 | 0100 | 4 |
| 6 | 0101 ✓ | 5 |
| 7 | (0110) ✓<br>1001 | (6)<br>9 |
| 8 | 0000 ✓ | 0 |
| 9 | 0001 ✓ | 1 |

Mod 7

For the duration of internal AND gate propagation delay << $T_{Clock}$

Now, we shall look at something which is the where the use of fix pre set which is there with 7490A that we shall exploit for this modulo number design.

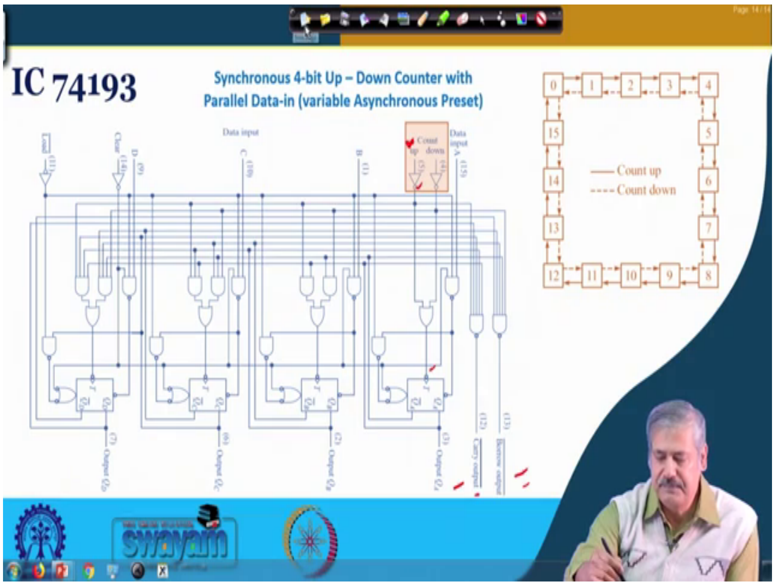
So, here we had seen that when R 91 and R 92 R 9 means it is setting a value 9 1 0 0 1 ok. So, when this 6 and 7 earlier we have get at ground because this is the 1 that we are using it right. So, if we keep 2 and 3 ground and 6 and this 6 and 7 where R 91 and R 92 are there ok. So, if both of them are high, then what happens? 1 0 0 1 this is 1 0 0 1 gets loaded isn't it? 1 0 0 1 gets loaded instead of 0 0 0 0 it is now will be having a value 1 0 0 1. So, we take note of this from the truth table and then we go forward.

So, what has been done here? What you can see is that the it is again a BCD counter right 2 and 3 now is kept at ground. So, there is no resetting happening; asynchronous resetting happening. Asynchronous pre set can occur if write condition appears otherwise it will be normal count as it was in the previous case. So, if you are having in the previous case 0 1 1 0 is the same thing right; so, 0 1 1 0 means QB and QC. So, first time it is occurring in the counting sequence say if you have you have started from 0 0 0 0 the if you have started from there it goes on and then when first time it is occurring here and then it is getting what? 6 and 7.

So, R 91 and R 92 internally it is AND gate is there. So, both of them are high. So, 1 0 0 1 will be appearing. Very short duration it will be there after that 1 0 0 1 will be there. So, primarily this particular state is 1 0 0 1 having the value counted value 1 0 0 1 except for that small propagation delay associated with the internal AND gate ok. So, this 1 0 0 1 right when it gets loaded after that what will happen? It is normal count usual count. So, after 1 0 0 1, 0 0 0 0 will come then 0 0 0 1 will come and that way it will continue after again when it comes 0 1 1 0 it will get again get reset again it will get pre set that fix pre set it will occur. So, the counting sequence counting states unique states are 1 0 0 1 0 0 0 0 and then it goes on up to 0 1 0 1 and then again it goes back to 1 0 0 1.

So, you have got now 6 unique states sorry 7 unique states 1 2 3 4 5 6 and 1 0 0 1 got added right; so, 7 units states. So, what is it? It is a mod 7 counter right. So, mod 7 counter also you can get it without any additional gate and all if you use the prefix preset part of it is it fine and there can be other uses as well right. Now we look at another IC where some such preset is there, but that is variable preset.

(Refer Slide Time: 13:12)



IC 74193

Synchronous 4-bit Up – Down Counter with Parallel Data-in (variable Asynchronous Preset)

Here there was a fixed preset. So, here we have got variable preset; that means, we can load numbers other than 1 0 0 1 1 0 0 1 is possibly any other number that we shall see how it works. So, this is IC 74193 ok. So, IC 74193 interestingly has got various features.

So, first of all it is a synchronous 4 bit up down counter. So, it can count both in the up direction; that means, 0 1 2 3 4 and also in the down direction; that means, 15 14 13 12 etcetera and for this you have got two clock input right. So, when one of them is active, the other one should be inactive. So, if it is other one need to be held at logic high, why?

You can see there is an OR gate, then it is going to the input. So, if it is at high, then this output is low. So, then other input of the OR gate to which the other clock is been get. So, that clock will take over the output and accordingly the clocking will take place is it fine.

So, that clock input the one that is not used will be kept at high right. And other thing that is important here we know that this QB QC QD QA the uncomplimented outputs will take role when we are going for up counting and complemented outputs, they will be participating when we are going for down counting ok. So, basically the number of they need to be ended. So, and depending on count up or countdown whether ABC for here or A prime B prime C prime that will be fed here and similarly A C or A A prime B prime a or A prime that is the thing that will come here. So, these are the things that we already know from synchronous up counter down counter design. So, those things are over here. See the other thing important in this particular IC is that there is a load input, there is a clear input that is reset. So, all of them get resets 0 0 0 0.

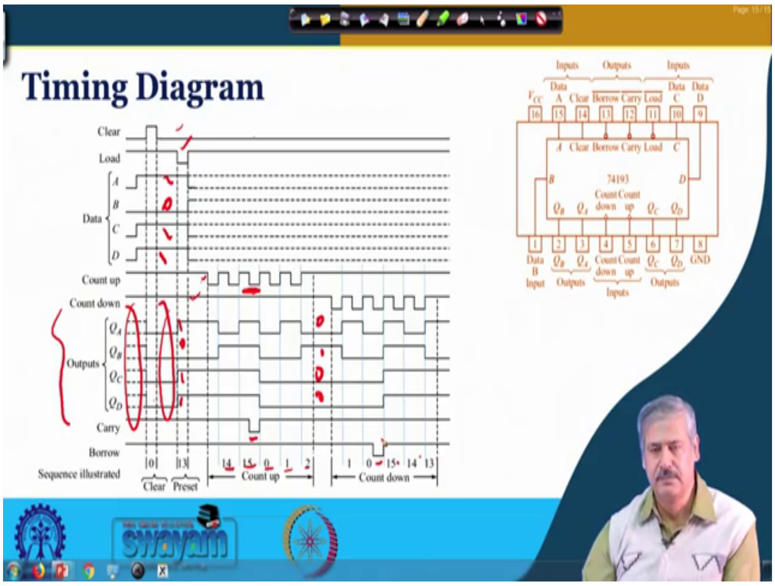
So, this is a single input ok. So, there is no such facility available that two inputs are there and all and you will see that it is not required because of this variable preset that is there. So, we can get a different kind of modulo number easily. So, this is a single input clear all right and then there is a load input ok. So, whenever it is activated, asynchronously whatever number is there DCBA that will come to QD QC QB and QA respectively without the need of the clock. So, immediately it will come after the this gate delays and all. So, it will get transferred to the output ok

So, this is a parallelly getting loaded asynchronously getting preset. So, this is variable because DC BA value can be different right. So, the idea is if you are basically putting a 1, then you are presetting it; if you putting a 0, you are resetting it is clearing it the corresponding flip flop so, similarly for other bits right. So, other than this you have got a carry output and borrow output here. So, whenever the number is increasing say, 0 1 1 3 4 going to say 15 in count up mod then it has reached 15 and it is going to say 0; next cycle ok, next clock cycle.

So, it has reached 15, the clock has gone low that it is a positive base trigger ok. So, you can see here is a inverter and here is a inverter. So, effectively it is a positive edge triggered right so, with two inverter from here from this side from the input side ok. So, whenever the counter clock input goes low in the second half of it right and the value is 15, then at that time the carry output will be active ok; only for that small duration as long as this clock input is remaining low. And borrowed output will be happening in the other direction when it is getting in the count is from 4 3 2 1 0.

So, when it goes to 0; that means, all of them are 0 and countdown clock is at logic low in the second half of the clock cycle at that time it will be active otherwise they will not be active and this is the count up count down the sequence that we can see is it ok. So, we shall see its performance with timing diagram that will make it more clear.

(Refer Slide Time: 18:29)



Timing Diagram

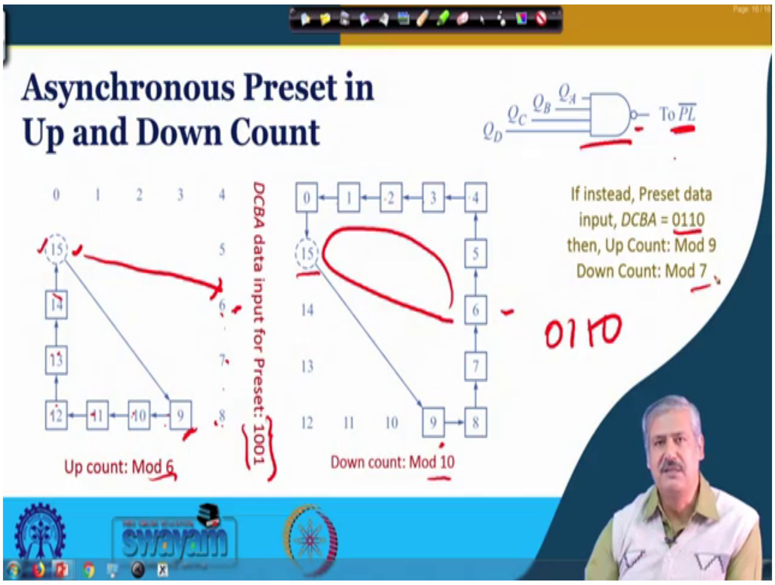
So, this is the timing diagram, I am talking about. So, you see the initially the counter may have some value QA QB QC QD over here by right. So, whenever you are giving a cleared right. So, all of them are getting cleared asynchronously clear. So, you can see that all of them are cleared right. So, after that clear input is taken away is inactive right. So, cleared is a you can see is it active high right and then in ABCD all right, these are the data that is input of it. So, at the input you are putting a value D is 1, C is 1, B is 0 and A is 1 right and whenever you give load which is active low it make it low, what will happen? Earlier it was remaining cleared ok. So, whenever you do that this QD QC QB QA, they take these values.

So, now they become 1 1 0 1 right. So, that is what has been shown here as 13 getting loaded right. So, the clock in this cases when it was happening clock varies at logic high ok. So, then the OR gate output was 0. So, basically there is no change clock was that is no clocking happening right. Now the clock countdown up that is remaining at high so; that means, it is inactive, it is not playing any role in the clocking and count up is started clocking ok. Say and when what you can see in the positive edges, it is cube E QD QC QB QA, the value start changing right I mean get triggered. So, after 13 at the positive edge, what will happen? It will go to 14, after that it will go to 15, after that it will go to 0 1 1 and so on 3. But what you are doing just to show that how count out happens to illustrate that ok.

So, after that you consider that count up has remained high ok. So, no more clocking; so, no more state change, it is remaining at 2 and countdown was also high. So, it is remaining at there is no even countdown clock also. So, it is remaining at that particular value ok. So, what you can see here 0 0 0 1 0 that is two. Then what happens? Count up remains at high and countdown starts clocking ok. So, at its positive edge what will happen? So, from two it will go to 1 then 0, then 15, then 14, then 13.

So, that is how it works. So, I understand that with this illustration you your there is more clarity about how this clocking and counting takes place and along with the parallel load and clear ok. The other thing is that you can see when it is at fifteen and clock is at logic low ok. From the logic circuit also you can see that happening. So, the carry is remaining row low for this hub clock cycle duration and similarly borrow is for when it is 0 and for this half cycle duration. Is it fine?

(Refer Slide Time: 22:12)



Asynchronous Preset in Up and Down Count

If instead, Preset data input, *DCBA* = 0110 then, Up Count: Mod 9 Down Count: Mod 7

Up count: Mod 6

Down count: Mod 10

Now, let us see how it helps in getting different modulo number ok. So, we have seen this particular state transition diagram 0 1 two 3 4 5 6 7 8 9 10 11 12 13 14 15 again 0 1 2. This is the standard mod 16 counter states are 0 to 15. Is it fine right? Now we consider that we have put an external gate ok; QA QB QC QD they are connected. So, it is a 4 input NAND gate and so its output will be low when both of all of them are 1 because for NAND gate any input is low means output is high.

So, all of them are 1, it becomes low as soon as all of them are 1, it will become low and this is set to this is sent to what this is sent to your parallel load that is asynchronous preset input which is active low ok. Now for different condition, different data input will be having different data now getting loaded whenever this parallel load occurs ok. Asynchronously within the same clock period right after the whatever this small propagation delay is there with the basic gate right. So, if you have got your DCBA data input at that time is 1 0 0 1, what will happen?

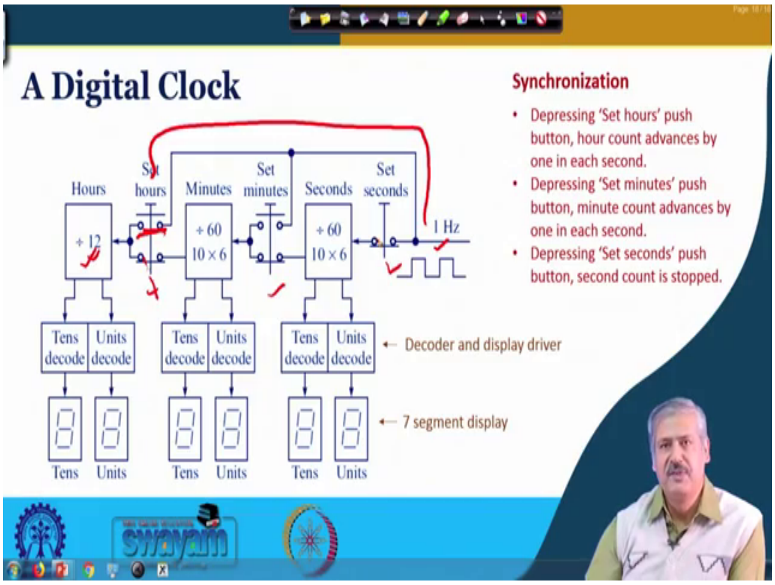
Whenever this reaches 15 immediately within that clock cycle the way we had discussed it for 7 4 9 0 right 9 gets loaded because it is now variable preset. So, now, it is 1 0 0 1 like before, but we have now put external values 1 0 0 1 separately. Earlier it was preset it, you need to do it for 7 4 9 0 1 0 0 1 was internally or getting loaded getting preset ok. So, it will go to 9 0. So, 9 10 11 it is in up count mode all right, it will if you are putting it in up count mode so, 9 10 11 12 13 14 again as soon as it reaches 15 a asynchronous reset. So, 15 and 9 are together in 1 clock cycle right and then again it will go on.

So, you have got 9 10 11 12 13 14 A mod 6 counter available with you right and had it been a down counter if you are exchanged the role of the clock one of the clock is now countdown clock is active and count up clock is held at logic high, then what would have happen ok. So, after 15 loaded is 9 ok; I mean immediately within the 15 and 9 together. So, then 9 8 7 6 5 4 3 2 1 0 again 15 right and at that time again it will get loaded immediately this 9 will get loaded. So, how many unique states are there? 9 8 7 6 5 4 3 2 1 0 and 15 and 9 are together. So, you can see that the unique states are 1 2 3 4 5 6 7 8 9 10. So, it is a mod ten counter.

Now instead of 1 0 0 1, you could have loaded any other thing and in up count mod and down count mod. There could have been different modulo numbers available depending on your requirement ok. So, if you are instead of say 1 0 0 1 if you are loading say 0 1 1 0 ok, I mean this 0 1 1 0. Then what will happen? So, instead of 9, you are loading 6 you are; instead of 9 you are loading 6. So, from 15, it is coming over here ok; so, 6 7 8 9 10 11 12 13 14 15and 6 together; so, 1 2 3 4 5 6 7 8 9 ok.

So, mod 9 count will be there and similarly over here, it will go in this direction and again it will get loaded see 15 and 6. Together it will give you mod 7, 7 unique states right. So, this is the way with variable preset, you can have different kind of such things.

(Refer Slide Time: 26:47)



A Digital Clock

Synchronization

- Depressing 'Set hours' push button, hour count advances by one in each second.
- Depressing 'Set minutes' push button, minute count advances by one in each second.
- Depressing 'Set seconds' push button, second count is stopped.

Now, to end this particular class, we shall have a look at one simple example of getting a digital clock using a multiple you know counter and an arrangement to calibrate it to set the time right. So, what you see over here is this is the 1 hertz clock which you know how to design using a triple 5 timer or if you have any other arrangement some stable, you know as table multi vibrator generator that also you can use ultimately what you need at the input is a 1 hertz.

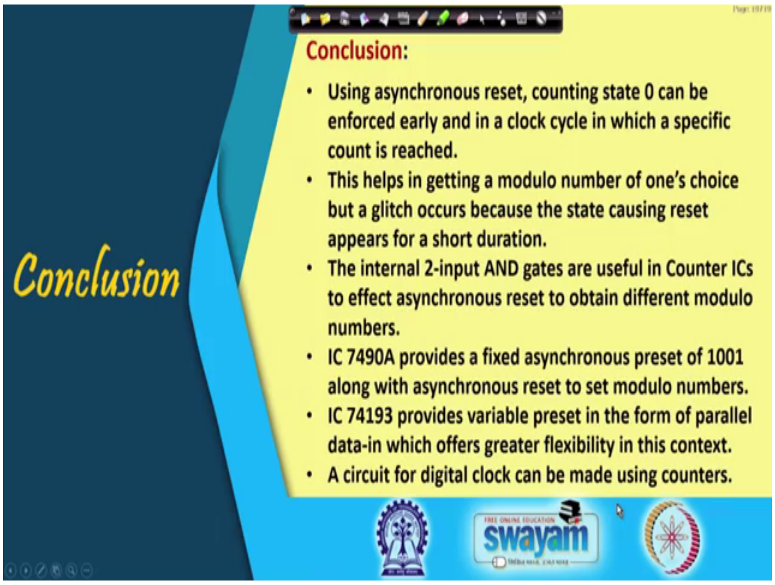
So, every 1 second there is a trigger coming ok. So, this is going to the seconds this is going to the seconds is feeding to the minutes and minutes is feeding to the hours. So, every 60 seconds so, it is a you can have a a mod 6 and mod 10 together right and another could be your what is it called for minutes also a mod 60 and for hours you need a mod 12 ok. So, this will be the unique decoding; that means, 1 0 0 to 9 and then, this is 0 to 5 the tens place ok. So, 59 then it will go to 0 0 0 to 59 0 0 to 59 then it will go to 0 0 and at that time when it reaches that 59 to 60; that means, mod 60 counter 1 clock trigger comes over here.

Similarly, for count of 60, one clock trigger comes over here. So, this is the way the cascaded counter that we have discussed before the same thing in a bigger framework ok. So, that is one part of the story, the other one is what you see the switches that are there so to calibrate or to set the initial timing right.

You first disengage this one the reset seconds right, then what you can do If you if your current time is say 350 ok? So, you can you can put this one, you just connect this one. This press this one depress this one. So, basically this will get connected. So, this now clock is feeding here. So, everyone second there will be a change here in this hour count ok, every one second because now once hertz clock is now feeding it directly. So, whenever it reaches 3, you stop there right.

So, after that you put it over here I mean remove the depression. So, it will come back it you know there is a spring load by which it is coming back it is appropriate switching mechanism all right. Then you go to minutes so, if it is 50 again you wait if it initial value was 30, then you have to wait for 20 seconds 20 such call you know things will be there and it will come coming here and then you release it. So, it is set the minute part is set and then you engage it, then the counting starts ok. Is it fine? So, this is the way the we can have a digital clock in place and it is how you initialize it, set the time correct.

(Refer Slide Time: 30:23)



Conclusion:

- Using asynchronous reset, counting state 0 can be enforced early and in a clock cycle in which a specific count is reached.
- This helps in getting a modulo number of one's choice but a glitch occurs because the state causing reset appears for a short duration.
- The internal 2-input AND gates are useful in Counter ICs to effect asynchronous reset to obtain different modulo numbers.
- IC 7490A provides a fixed asynchronous preset of 1001 along with asynchronous reset to set modulo numbers.
- IC 74193 provides variable preset in the form of parallel data-in which offers greater flexibility in this context.
- A circuit for digital clock can be made using counters.

So, with this we conclude using asynchronous reset count in state 0 can be enforced early and in a clock cycle in which a specific count is reached ok. So, 0 0 0 0 will be there whenever that particular number count value reaches. These helps in getting a modulo number of 1's choice, but a glitch occurs because this state causing reset appears for a short duration.

So, in the final class on this counter, the next class we shall see that what could be alternate arrangement by which we can get this glitch removed or we can have a different kind of counter design paradigm. The internally two input AND gates are useful in counter IC's to effect asynchronous reset to obtain different modulo numbers that we have seen IC 7490A provides a fixed asynchronous reset of 1 0 0 1 along with a asynchronous reset to set the modulo numbers 7 4 1 9 3. We have got variable preset which offers a lot of flexibility and also you have seen that a digital clock can be made using counters when we can have a mechanism to initialize it.

Thank you.