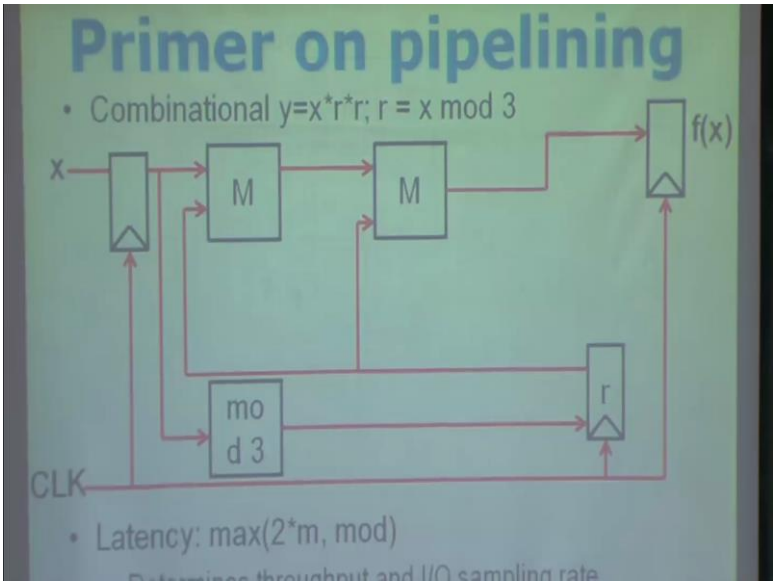
Computer Architecture
Prof. Mainak Chaudhuri
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 11
Basic Pipelining, Branch Prediction

We are talking about similar case where we go to pipeline is function, where these two computations depend on value of r, which is computed from X.

(Refer Slide Time: 00:26)



So, this was a particular combination you want to pipeline. So, as you can see for particular value of X. You still lead the previous value of r and then you update r that is what we done here. And a combinational implementation, we essentially take the value of r from the latch, whatever has in the latch by the previous one ((Refer Time: 00:59)). And new value of r, will going to latch and the same time as, you take the current output.

And so the which pipeline it was that say equal, if we put a pipeline man shield meaning that this is stage one, and this is stage two, these one does not what? Because, this particular value essentially is not in the available on that and that a problem. Because, as you can see here the forward path here, go to the latch. So particular value of comes in I

compute this, so on that clock it is staying it is here not here there actually.

So, the next value of X cannot take it from here, but should take it from here, is already there. So, that is called a bypass.

Student: ((Refer Time: 02:03))

Which one?

Student: ((Refer Time: 02:07))

Which one? This one.

Student: ((Refer Time: 02:17))

The top one.

Student: ((Refer Time: 02:24))

((Refer Time: 02:27))

Student: ((Refer Time: 02:28))

This one.

Student: ((Refer Time: 02:34))

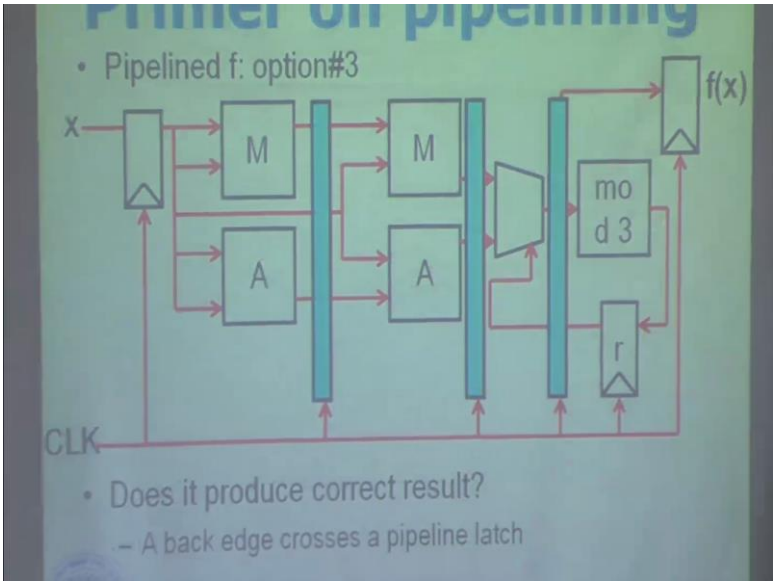
So up to this ok.

Student: ((Refer Time: 02:43))

You talking about this one.

Student: ((Refer Time: 02:47))

(Refer Slide Time: 02:50)



So, let see what is says, he says that can we bypass r in this case, how many of you thinks it is you can no, why?

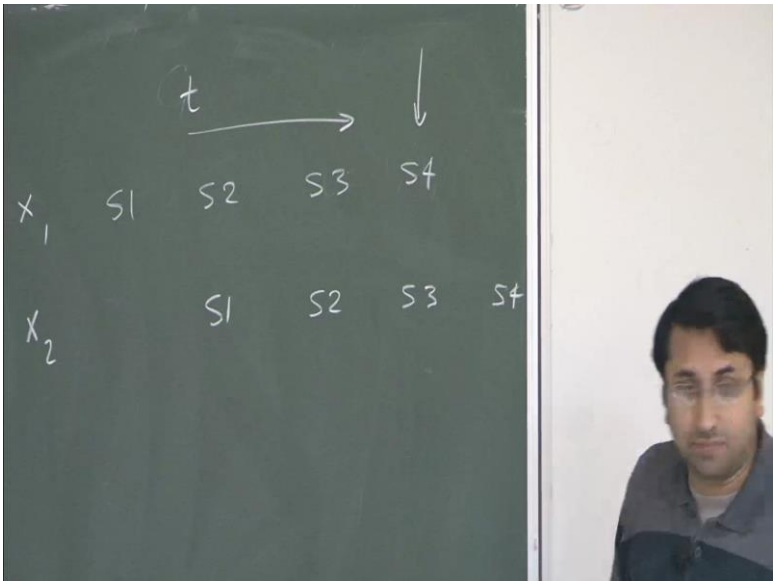
Student: ((Refer Time: 03:12))

Exactly, so it is not you will compute here, there is no point in bypass. So, bypassing comes into feature only, if the value has been computed in this kind computed after this particular stage, so this has...

Student: ((Refer Time: 03:27)) it value r is equal to next value.

So, essentially in this case, what is would happen is that? You need a value of r in a particular cycle. When it is been computed in the same cycle, this is been computed and we also need the value, that is impossible.

(Refer Side Time: 03:51)



And that is why, if you draw the pipeline value of that, we take last time. We discussed here four stages, so time goes in this direction. So, this is let us X 1, X 1 is doing this. For X 2, so in this particular cycle S 4 is computing r for X 1 and that r is needed exactly in the same side, which is one possible. So, these hazards are not easy to solve, where as you say in last time writing with formula is used promotional also.

(Refer Slide Time: 04:40)

So, source stage of data comes after the destination stage that is not easy to resolve. And the only while resolve it can somebody suggest. So, going back your example, what can I do here, resolving this particular value?

Student: ((Refer Time: 05:00))

Sorry.

Student: ((Refer Time: 05:01))

This is my pipeline ((Refer Time: 05:06)) I cannot change the pipeline. So, r is computed in the S 4 that needed in this ((Refer Time: 05:13)), by the next computation. Is any suggestion, how might you dissolve it? What would you do?

Student: ((Refer Time: 05:33))

We can use what?

Student: ((Refer Time: 05:38))

What kind of king, so remember this thing that at time X ((Refer Time: 05:44)), at time t, r is been computed, at the same time I needed. So what q are you going to use?

Student: ((Refer Time: 05:54))

Buffer what?

Student: ((Refer Time: 05:56))

Buffer, I mean is not there in mode is not there it. So, it will be available in future, so what is the buffer going do here?

Student: ((Refer Time: 06:09))

((Refer Time: 06:13)) Which of course, I can delay the S 3 here. I can delay by a cycle the essentially, what you are saying is for every computation of exam we use one cycle. ((Refer Time: 06:27)). So, what we normally do, to look at the future? That is what we can do here, what you do?

Student: ((Refer Time: 06:35))

I have nothing else here, it just is a very specialized computer, which computes this particular function, what you do to look at the future?

Student: ((Refer Time: 06:47))

Sorry.

Student: ((Refer Time: 06:49))

((Refer Time: 06:54))

Student: ((Refer Time: 06:59))

Contradiction.

Student: ((Refer Time: 07:05))

But, we are so we will ((Refer Time: 07:15)). But, the earliest you can gate is here. It is impossible to get here, get it the value will be computed only at the end of this particular sign. Remember there is clock circuits, it cannot start something ((Refer Time: 07:34)) obituary, how do you see what will be tomorrows ((Refer Time: 07:45)).

Student: You predict.

You predict and I predict the value of r. Can I? So, that is possible if the values that I am computing or not completely have, if there is some patterns, if there is high enough, if there is presence of pattern I should be able to predict, what the value ((Refer Time: 08:19)).

So, I can predict the value of r here ((Refer Time: 08:23)) and use that. But, I know whether in it correct or not only one side ((Refer Time: 08:29)). So, in case of wrong prediction at undo, what I which pressurized. So, these things have going to work, only if your prediction activities, more than pressurized, which means more than half of the cases ((Refer Time: 08:45)).

Otherwise, you essentially ((Refer Time: 08:49)) hart in. But, often in the case is that, if is the concentrative everybody this going to prediction thing ((Refer Time: 08:57)). But, after the case is that the usually should error given in the wrong detect that fear that something wrong, undoing the redoing the right thing. May be solve expensive, that you might require much higher prediction in case, it out way lose.

So, for example, you may be 90 percent prediction accuracy, to out way the loss of 10 percent wastage. So, we look at this particular problem in a different contest, but I just to bring this up here, this thing is a concept clear to everybody?

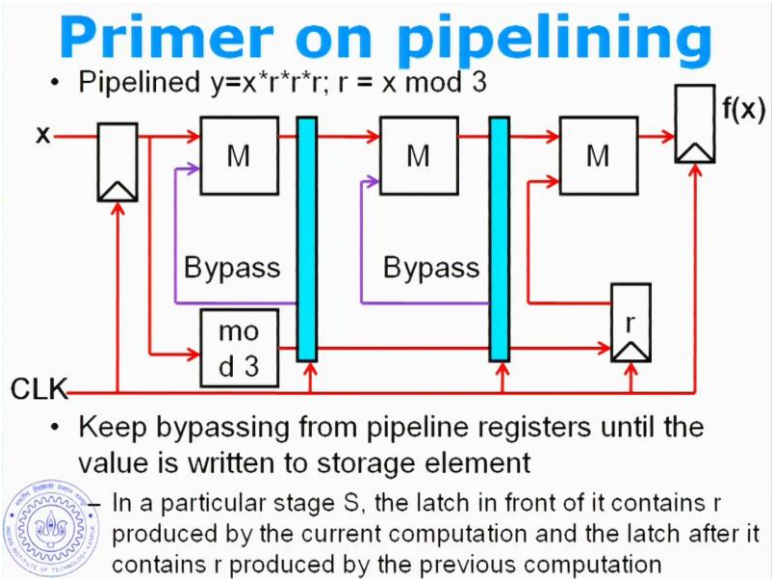
Student: ((Refer Time: 09:32))

Sorry.

Student: ((Refer Time: 09:33))

There will be some piece of hard work sitting here. You should be one print the pattern of value of r, look at several in the past 10 stands that seen a 0 ((Refer Time: 09:48)). So, it is a pattern minded that would sitting here, and giving me the next value of part, which you verify here. So, we will back to where we started.

(Refer Slide Time: 10:24)



So, you will look at disposing computation, so what kind of bypass ((Refer Time: 10:20)) correct it. And we said that, if source stage of data comes before or equal to the test is on stage, wearing the data ((Refer Time: 10:30)) is later stage, we can now such a absolute bypass. So, this case, source tell us this one, were I produce r and the dentition stills also this one, by the bypassing I can dissolve it.

And, if I want to change the computation further supply X times r q. So, I want to keep on bypassing until ((Refer Time: 11:02)). So, if you look at the particular stage S let us a this one. So, it is demarcated by two latches ((Refer Time: 11:14)), for the remaining bypass. So, latch in front of it this is one that is are produced by the current computation, current value of X in produce r, which we get letch here.

And, the letch after it contains r producer by the previous computation. This one contains the r that is previously, so this is on norm that is, why I am taking the bypass from this latch ((Refer Time: 11:42)) this by pass concept.

Student: ((Refer Time: 11:58))

Sorry.

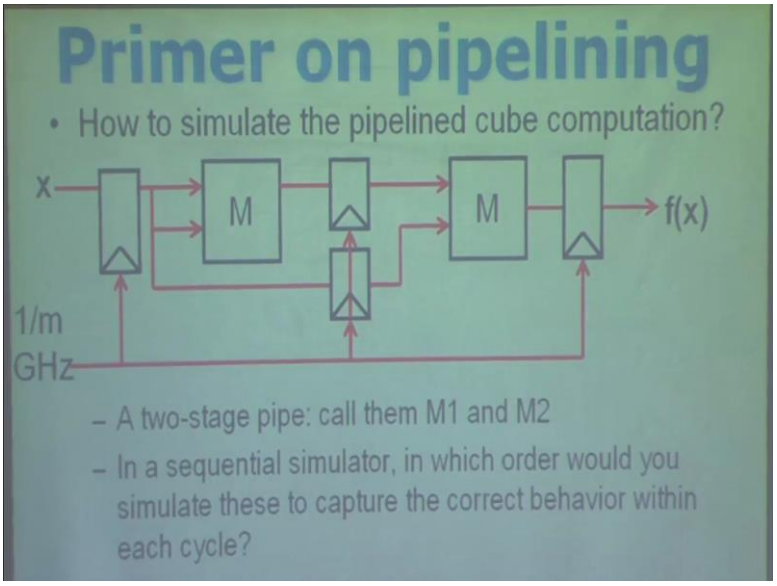
Student: ((Refer Time: 12:02))

This one?

Student: ((Refer Time: 12:05))

This one currently contains, so I mean, so this one will contain the final value of r. But, of course it will take several cycles to propoently this latches and r. Any question about this? Is a bypassing here from, which latch should be bypassed. ((Refer Time: 12:37)) So, bypassing always is wants with, when you take the value from the latch behind. Otherwise, this is not a bypass you just computing the value that will be produced ((Refer Time: 12:57)).

(Refer Slide Time: 12:57)



So, at the say last time usually the computed designed cycle may about 5, 6 ((Refer Time: 13:09)). For example, the microprocessor that would be available for the ((Refer Time: 13:16)) design suddenly to ((Refer Time: 13:21)). So, it goes to several faces and usually the research starts, which simulation. If cannot the hydro ((Refer Time: 13:33)). Because, and that has balls you do the long to simulate, your design make sure that is correct, that is a most important ((Refer Time: 13:50)).

So, simulation is very important, so I am going to simulate this computer is represents ((Refer Time: 14:06)). So, let us essentially by stimulating I mean that you a piece of software, which mimics the behavior of this particular hardware. So, then without building the hardware, you can feed the software including X and produce of X. And measure the performance of the cyclic polis of X, and consume polis of X.

So, let us take a look at this simple example just of one this start up with yesterday the cubic search f x going to S cube it is a two stage pipe that is called them M 1 and M 2. I am latching between them. So, you can see that, I can easily right down a piece of software describing this particular block it is a ((Refer Time: 14:59)) whatever, I got it ((Refer Time: 15:01)). The question arises about this particular latch here, this latch here, this latch here. How to do it is there we going on? What stability should be you see?

So, let us think about it. The first question that arises is sequential simulator in, which order would you these to by the stages to capture the correct behavior got it. The two stages involved into by simulate involve first and then the cycle, because you remember in a cycle both the stages must be body is a pipeline X, M 1is what is want of X, and M2 should be what previous X is same cycle.

So, I am simulating a particular cycle that be siding, which order I simulate them what do you think, of course, I have two possibilities ((Refer Time: 16:04)) first involve the attempt to ((Refer Time: 16:06)), which one is correct.

Student: ((Refer Time: 16:09))

You, wanted to...

Student: ((Refer Time: 16:12))

You, wanted to remember that the simulator is the equation some software you heard the simulate the M 1and latched the value produced by whole here, and then you we go that is not a correct behavior computing that in the next ((Refer Time:16:44)) you has any

problem, so it has to be the ((Refer Time: 16:48))

Student: ((Refer Time: 16:55))

End of the execution means?

Student: ((Refer Time: 17:05))

((Refer Time: 17:11))

Student: ((Refer Time: 17:12))

((Refer Time: 17:14))

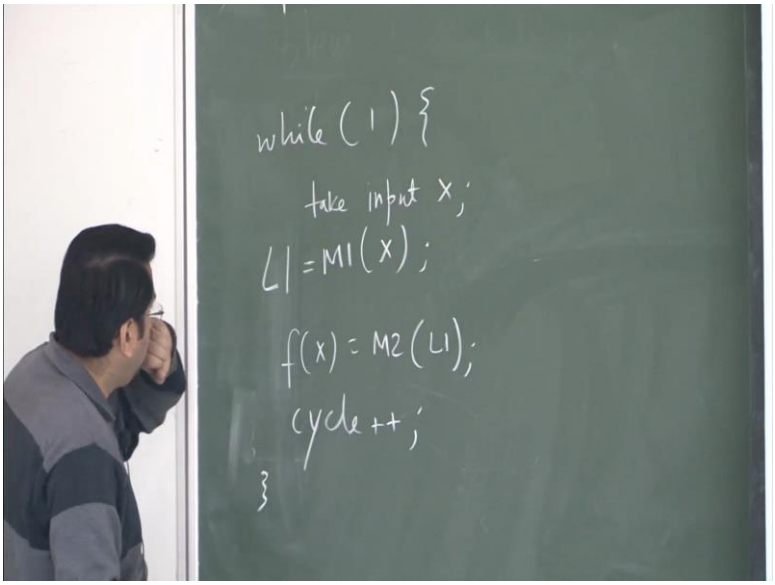
Student: ((Refer Time: 17:16))

The latches, so what computed tools and then so what who, so he has suggest a solution that do not update the latches. So, which essentially mean internally storage to store the computed by him I involved with computes is that ((Refer Time: 17:48)). So, ((Refer time: 18:09)) keep this in mind in a pipeline simulation is always go from bank to ((Refer Time: 18:14)). Similarly, in the pipe stages look in and this is the reason, while have to do this is it clear to everybody? Why cannot anyone ((Refer Time: 18:27)).

Student: ((Refer Time: 18:28))

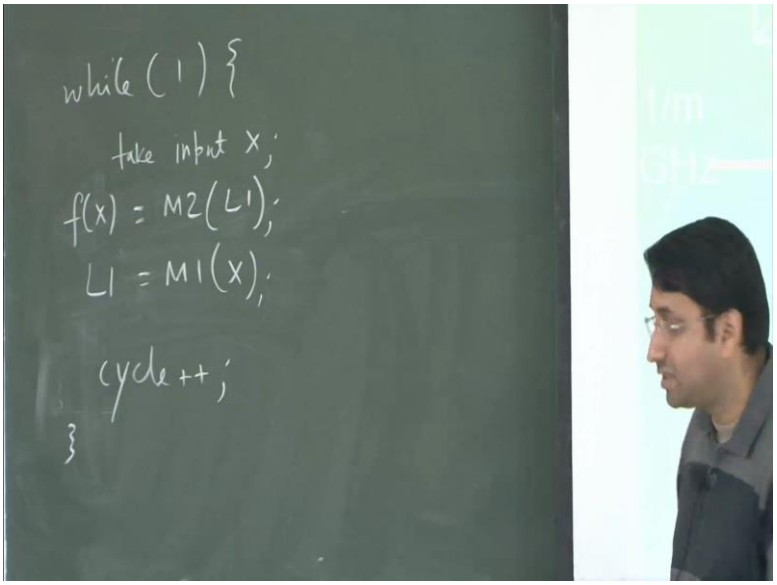
So, is a sequential piece of software, so if you first two M 1 involving update the latches and then M2 immediately see the latches in compute on that. So, essentially the method for a same cycle ((Refer Time: 18:49)) may be I will show you.

(Refer Slide Time: 18:58)



So, suppose you have some, you will have some playing tool while 1 take input X, M 1 X, so that will put in through into a latch L 1. And f x equal to M 2, L 1 that is my simulation, so I will take X, so essentially cycle plus, plus, this one cycle. I take a new input X compute put to L 1 and M 2 fix data computes f x the computation petition in the cycle. So, I taken input x produce y to 1 cycle that is wrong, we should I taken this 2 ((Refer Time: 19:54)) cycles.

(Refer Slide Time: 20:07)



So, what speech the orders see what will happens, what cycle is ((Refer Time: 20:20)) that is not nothing there in latch. The ((Refer Time: 20:24)) of x puts it in L 1 cycle, next cycle ((Refer Time: 20:31)) and then every cycle there one output coming out f of X.

Student: ((Refer Time: 20:47))

It would yes you get some garbage, which is why this particular sample latch is important for the first time if you, the first block ((Refer Time: 21:06)) clear to everybody working that if you putting simulation in the pipeline.

Student: ((Refer Time: 21:18))

While, then you taken input M1 and M 2?

Student: ((Refer Time: 21:39))

So, that case, you would pick up X and producing f x in one side. That is a problem. Because the reason L 1 is just a variable, anything else? So, if you switch the orders, it does not work. So, if you put f x M 2, M l, you update L 1, this function takes a L 1, to

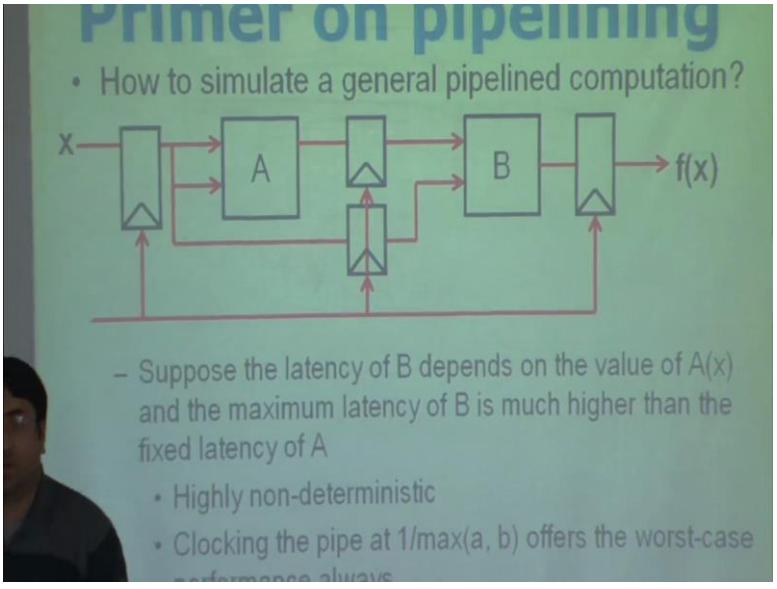
update L 1, this functions produced f x in the same side. That is not correct. That is a wrong behavior.

The pipeline is supposed to produce 1 output, 2 cycles later, two cycles after in except one. So, this one, actually meanings your combinational cycle

Student: ((Refer Time: 22:23))

No by mentioned a sequential simulators a just to the making component well about detail, when about the parallel simulator problem does not change. To the parallel simulator quarter is important might is given is that, I will have the third for simulate one, another thread for simulate two. And I am the pushing the problem to the schedule the schedule has consult, which one to the spoke that simulate important to that was about a cubic.

(Refer Slide Time: 23:29)



Let, take an original problem; that is how to stimulate the general pipeline. So, here again suppose the latency of B depends on the value of A x. So, whatever the value that gets latched here, will determine, how long it is grateful B equal to the value. And the maximum latency of B is much higher than the fixed latency of A. So, which means the

computational B is highly not terms. Sometimes, it takes very small on times, sometimes, it takes very large on times the depended on the value of X.

And the maximum latency of B is much higher than a. So, that means, if you the clock the pipe and moreover max a or b that often the worst case performances always. That is correct. So, that is what is, we done the two stages, you should max, but in this case, if you very poor performance. But, actually in most of the cases we might able to do better than this, because, we are essentially designed slightly flow worst cases, what if I clock ball over b, small a or small b and littlies only.

So, off course in now, the cases, what will happen is the we might be doing something higher is over, so what is the solution, so here it says that replace the latches by cubes I put A cube here, also I replace these by A cube here and I clock it A. So, what is now going to happened in some cases that q is go to drain faster, which then or may be at the same way in some cases is going to be drain slower piece.

So, let us cube will grow strait depending on how B is computed, but off course you know there will be some cases that achieve from, because B is now running slow. So, may be able to inject the anymore input. So, that automatically pushes put pressure on this and eventually, that part. But, eventually, we will finishing computing the cube will start draining pipelines will supporting.

So, these are basic process, while find them different pipe stages are proffered latencies. And off course, you will not try to clock it a moreover max of the all the pipe stages. Because, that will be the worst stages, you still till very optimistic in clocking at a pastor way. But, could cubes in between take care of this particular slide difference, is it clear to everybody, why I want to replace the latches by cubes.

Student: ((Refer Time: 26:14))

So, it will think of, it as use of latches exactly.

Student: ((Refer Time: 26:20))

Empty, why should empty, we should be computing something

Student: ((Refer Time: 26:35))

But, what I am saying is that, given that max of a come of b will be a rare case got. It most cases, you will be find much better than this one. So now, the question lets now, again go back to the same question how do we simulate this particular pipeline either you view problems or is quite simulate B first and then A. So, I can say that kind of piece of software, that, what he does and I can he does, question about interface.

Student: ((Refer Time: 27:19))

So, the fundament question I am in, which order should I simulate this A before B or B before A, I achieving between and the cube offers more functions A cube, B cube full empty.

Student: A for B first B than A I take problems, you can step one by one look at it.

So, let us supposed that, I simulate A before B I am A. So, I already told you the very good telling with the think of situation, when I not be able to...

Student: ((Refer Time: 28:16))

No, I am not asking about a specific scenario, I am saying this have been delivering the problem.

Student: ((Refer Time: 28:29))

So, that that probably, let I ensure something and B fix of that cube immediately in the same site that is wrong, what is the problem with B before in somebody see a problem.

Student: ((Refer Time: 28:56))

Point is that, the situational B, should be will still be a multi cycle simulation. I am still meaning the behavior A, which means the cube actually grow up gradually extend up, while we working, I know, it is a similar characteristics, of course.

Student: ((Refer Time: 29:41))

Yes, I will argue that, but, remember that, the latency depends on the value of x. So, when I put counseling I only there the I will know the vacancy all, so where I getting that sign cycles and chart only those cycle that cube may grow up. But, anyway, I do not my this details are important here. But, I am asking is scheduling B before A had the problems.

Student: ((Refer Time: 30:29))

That means; I told input coming in, that is, which cube this cube by this cube fix up A will stop accepting. So, now, off course in condition and this cube has to, then only the detail from here. So, scheduling B before A; why is that compromise?

Student: ((Refer Time: 31:28))

Why not B is missing, but A is need to go simulate to that do that do not wrote a book band exam is start here. But, I am seen what you know on the every cycle I could make, so I scheduled B or A. So, what happens, so particular B get the book particular B thus something in that cube; otherwise, B has nothing to do it. So, that is something here, this not and this is not that is also in option that also in condition of books hit.

So, no the clock frequent is same; it is same from profiting all three latches. So, suppose imagine a situation that this cube is according to the cycle this called disposed work. So, work for how should a pipeline, we lift this should not be anything in the cycle. We should pick up an item passes it become put it in the fashion that this cube has these

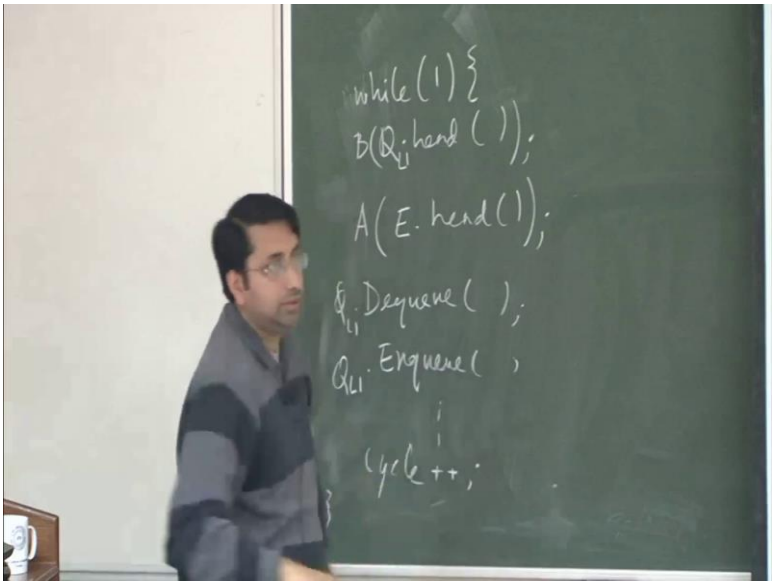
forms that should be in the current behavior.

So, let us see, now about the what a software simulated desk, if it will hopes B fix up this item goes it here, in books a is there is one slot here, it picks up a next is it clear to everybody A should not has been invoked, it is a cycle the change of this key state should not be visible in the cycle in it ((Refer Time: 34:17))

Student: ((Refer Time: 34:21))

Cycle, exactly, so in such a simulator this is again very, very difficult in an every cycles operation manually settled in the two parts the state updates. You will happen at the end of the subject.

(Refer Slide Time: 34:44)



So, what we do is while 1, so you take the ((Refer Time: 34:53)) function it says Q dot head. So, it should return be the hand of the Q, but not d Q actually. So, B will operate on that and A will operate on. So, let us call it Q L 1 operates on L 1 means l. And then you can do the Dequeue dot Q L 1 doted Q L 1 dot Enqueue and all the other things have been cycle plus two parts, one is complication; other one is redirection. Then even though, we got the head of the Q, A will still think that, the Q is here. It is actually not

wake up in that cycle ((Refer Time: 36:02)).