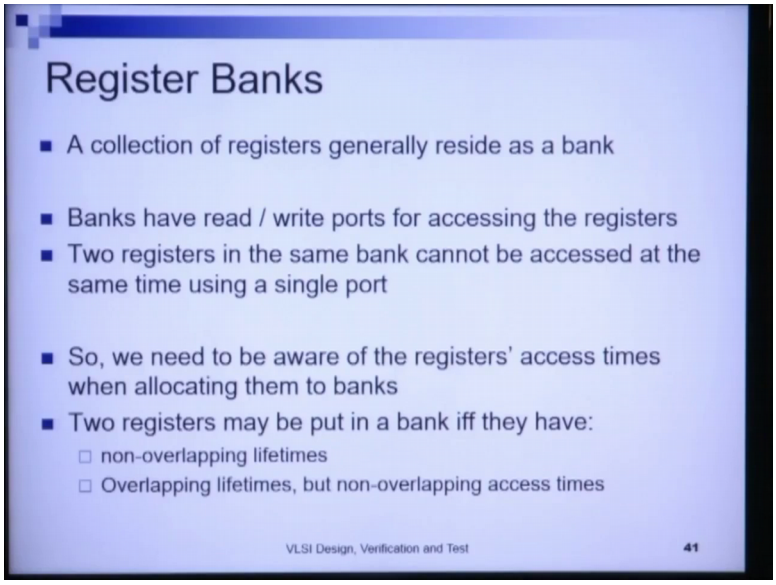
Embedded Systems – Design Verification and Test

Dr. Santosh Biswas
Prof. Jatindra Kumar Deka
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture – 12
Hardware Architectural Synthesis – 6

Welcome. In this module we will look at a few extended resource sharing and binding problems. First problem is related to register banks. A collection of registers generally reside as a bank. So, till now we have looked at individual registers.

(Refer Slide Time: 00:48)



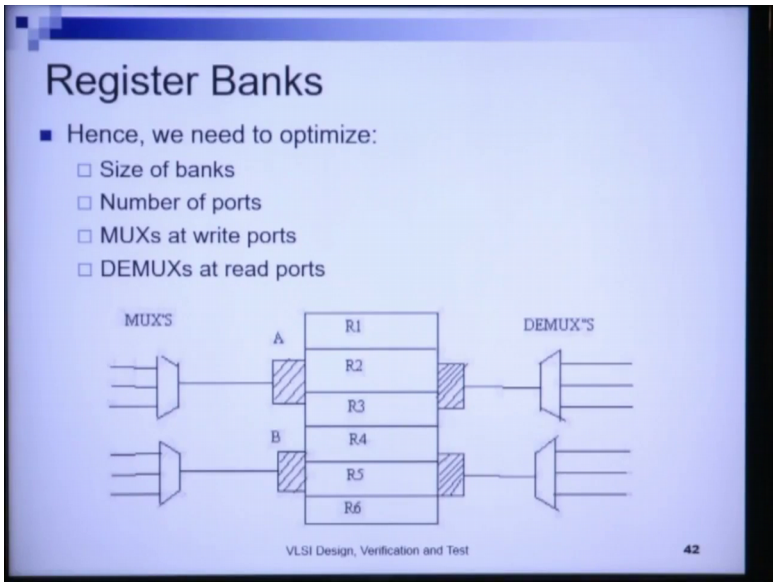
But typically registers in a reside in a large register file or a bank, say in a RISC machine. And these and these register banks so the registers inside these register banks can be accessed through read write ports. So, a collection of registers generally reside inside of bank, banks have read write ports for accessing the registers. 2 registers in the same bank cannot be accessed at the same time using a single port. So, if you have 2 registers in the same register bank, you cannot access both the registers at the same time using a single port.

So, you need to have multiple ports in the register bank for multiple accesses to registers in the register bank. So, therefore, we need to be aware of the registers access times

when allocating them to banks. So, 2 registers may be put in a bank if and only if, if the these registers have non overlapping life types; that means, that all variables assigned to register R 1 say and register R 2 say have non overlapping lifetime. So, all temporary variables all temporary variables assigned to register 1 have are in the interval say this one. And all registers in temporary with all temporary variables in register 2 adding another interval say this one and these 2 are non-overlapping.

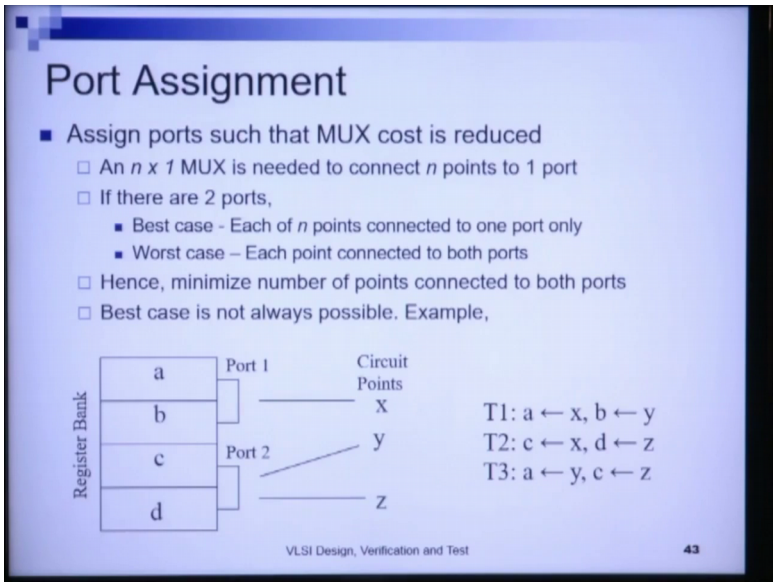
So, the registers themselves have non overlapping lifetimes the registers may have overlapping lifetimes, but non overlapping access times. So, I cannot access the registers together; that can also happen. So, this is a more constrained scenario that I have overlapping lifetimes, but they have non overlapping access times there none of the registers within the register back are accessed together using the same port.

(Refer Slide Time: 03:01)



Hence we need to up the optimized size of banks number of ports MUXs at write ports and DEMUXs at read ports. I have ports port A for reading and writing and port B for end the reading and writing. So, I have 2 ports A and B, so I have MUXs circuit points float circuit points float, their outputs on the MUXs. And the MUX chooses one of the circuit point and places it on port on the right port of A. And from A, I can choose any one of these registers. Similarly, this DEMUX, the DEMUX will read from the read port from the read port and choose any one of the circuit points. The port assignment problem is it as follows.

(Refer Slide Time: 03:49)



So, we want to assign ports such that MUX cost is reduced and n cross 1 MUX is needed to connect n points to one port. If there are if there are 2 ports and we have n points, and each of these n points is connected to one port only then we have the best possible solution. But the worst case is that; we need to connect each point to both the ports hence we want to minimize the number of points connected to both ports.

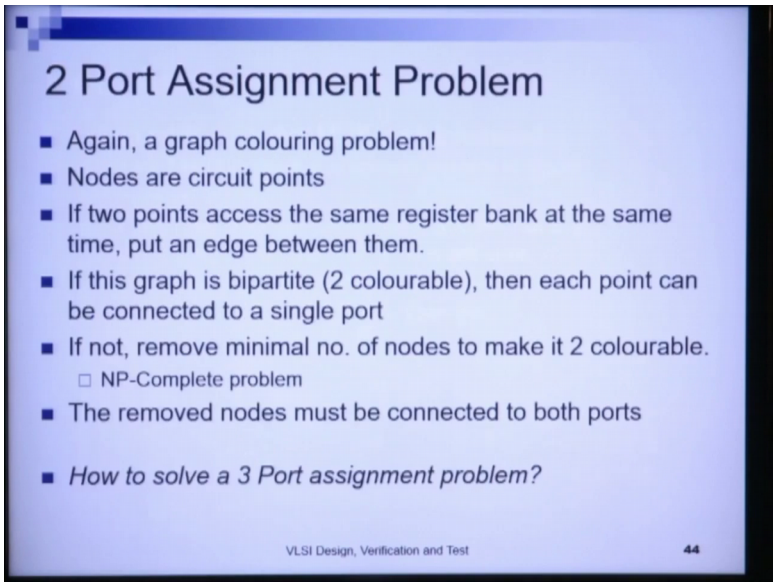
The best case is not always possible why? Let us consider 3 time points and we have 3 register assignments register transfer assignments. In time step 1, we have the assignment a equals to x and b equals to y. In time step 2 we have the assignment c equals to x and d equals to y, and in time step 3 we have the assignment a equals to y and c equals to z. Let us say we use port 1 for this assignment a to x.

So, I put the value in x to the register a through port 1, because I have used port 1 for at the first time step, for this register transfer I need another; another port. So, I use this register you I do this register transfer be equals to y through port 2. Now I come to step 2 x was already connected to port 1. I can connect the same port; I can connect through the same port and put the value of x into c in time step 2. Now z is another new circuit point let us say I choose port 2 for z and do the register transfer z to d through port 2.

Now, in time step 3 we see that both y and z has been previously allocated to port 2 and at least one of them has to be allocated to port 1 as well; to affect to appropriately affect the register transfers at time step 3, because both y and z wants to access the 2 registers

in the same register bank at the same time and they are connected to the same port, where previously connected to the same port. So, at least one of these points have to be connected to both ports.

(Refer Slide Time: 06:18)



Now, to solve this problem we again have another graph coloring problem, we again have another graph coloring problem. Here the nodes our circuit points, if 2 points access the same register bank at the same time we put an edge between them and that means, they access 2 registers in the same register bank at the same time they are they we put an edge between them.

So, when there is an edge between 2 circuit points we cannot use the same port for both these circuit points. Therefore, if the graph is bipartite that is 2 colorable if this graph is 2 colorable, I can use 2 colors to color this entire graph; that means, what do I have? I have 2 distinct colors; that means, 2 distinct ports and I have been able to give a color to all vertices while not putting the same color to adjacent vertices. That means, I have been able to put each circuit point connected to a single port only and this is the best possible solution that I have. However, if this best possible solution is not possible then our job is to remove the minimum number of nodes to make it 2 colorable.

We have if the if we have 2 ports; however, this removal of the minimal number of nodes is again an NP complete problem. The remove nodes must be connected to both ports then.

So, what we do? We first do a 2 color and try to make the graph bipartite. We try to make a 2 coloring and then if this 2 coloring is problem not possible, then we try to remove the minimum number of circuit points such that 2 coloring is possible. And for these circuit points the remaining circuit points, which I had extracted out from the graph, must be connected to both ports. Now this is for the 2 coloring problem, what do we do for the 3 port assignment problem? With this question posed to you I come to the end of this module.