Model Checking
Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology – Madras

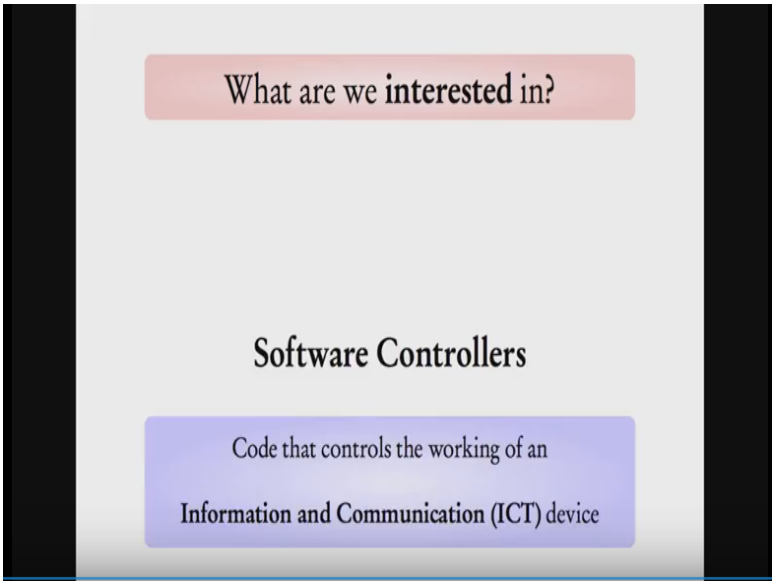
Lecture-01
Course overview

Welcome to the nptel course on model checking. The first week is meant to be an introduction to this course. We will start with a course overview and follow up with four modules each of which will introduce you to a new concept in modeling.

(Refer Side Time: 00:23)

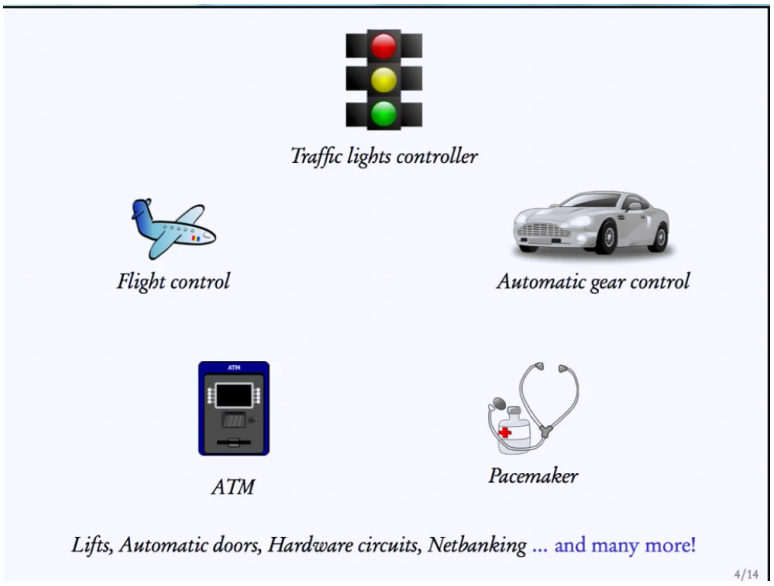
This video is meant to give an overview about this course. This is going to be more detailed than the interactive video that was posted before. What are we interested in this course?

(Refer Slide Time: 00:39)



We are interested in the code that controls the working of an information and communication device. Instead of defining what an ICT device is let me give you some examples.

(Refer Slide Time: 00:57)

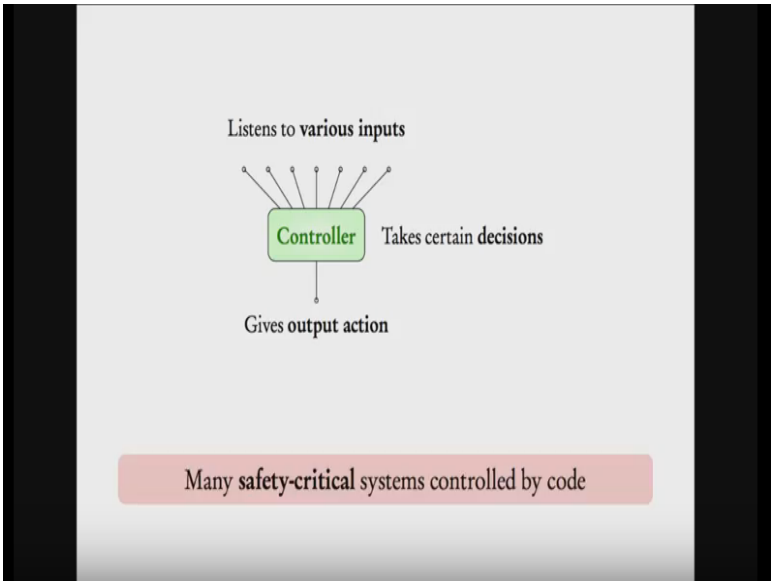


Let us start with an ATM. The working of an ATM is controlled by a program that listens to various inputs given by the user with that performs some internal manipulation and generates an output action. The output would be for instance to display the balance or give out cash etc. Now let us get in to an airplane an airplane contains a lot of code. It has programs that decide its trajectory based on inputs like location, weather etc.

Now let us come all the way to medical instruments. Let us talk about a pacemaker of course this is not the pacemakers but let me tell about. A pacemaker is a device which is implanted inside the people with the heart problems. It is used to regulate the heart beat. The pacemaker has a program that listens to electrical signal generated by the heart. If the heart misses a beat the pacemaker the pacemaker program has to output necessary electrical impulses to generate a heart beat.

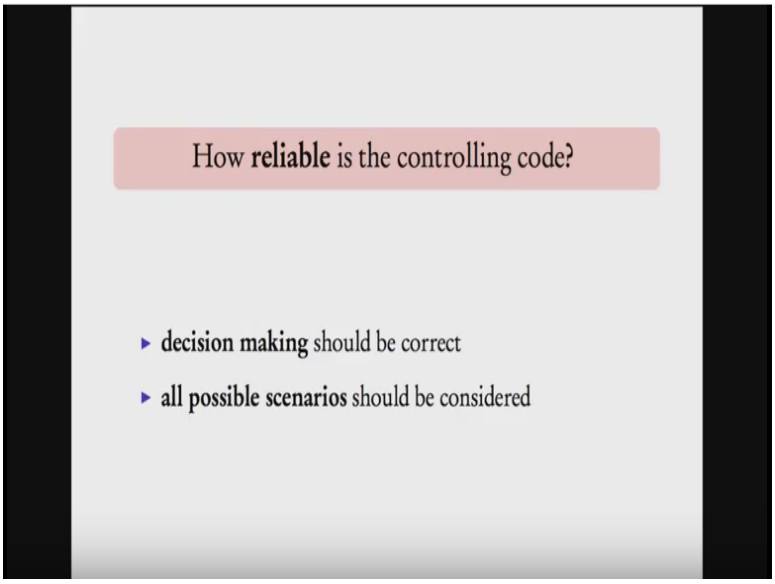
There are many more examples of such system which are crucially controlled by programs.

(Refer Slide Time: 02:23)



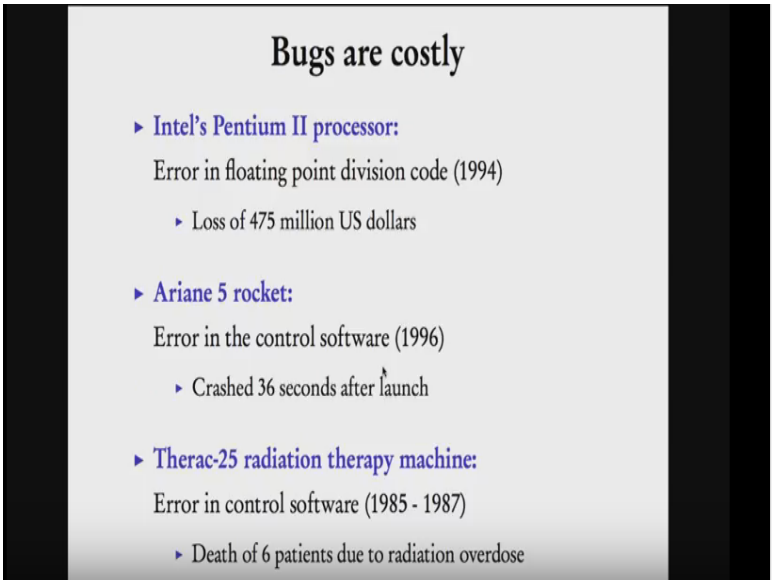
Let us called this program as the controller. A common feature of this controller is that it listens to various inputs takes certain decisions and gives an output action. Clearly many of the systems which are controlled by programs are safety critical. For example a flight, a pacemaker, car, even traffic lights all of these are safety critical systems.

(Refer Slide Time: 03:04)



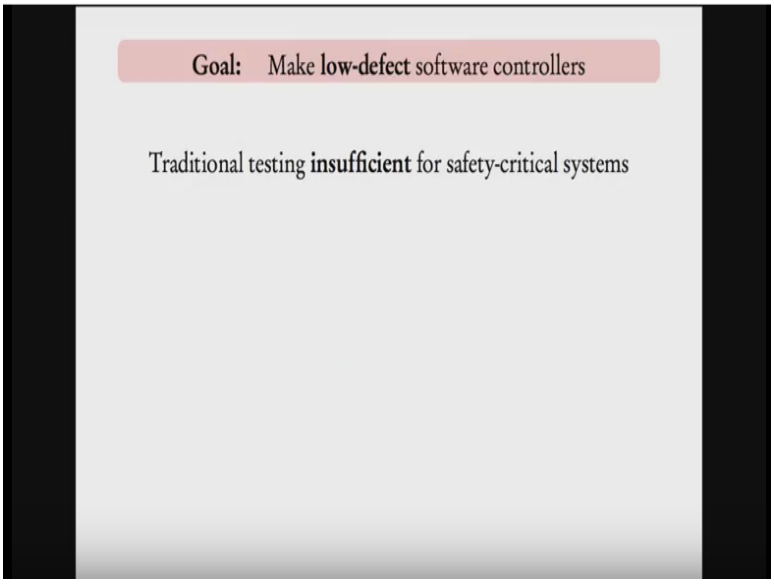
Given this scenario this question is of at most importance. How reliable is this controlling code? Essentially is the decision making of this code correct, is it complete in the sense is the code taking care of all possible scenarios?

(Refer Slide Time: 03:26)



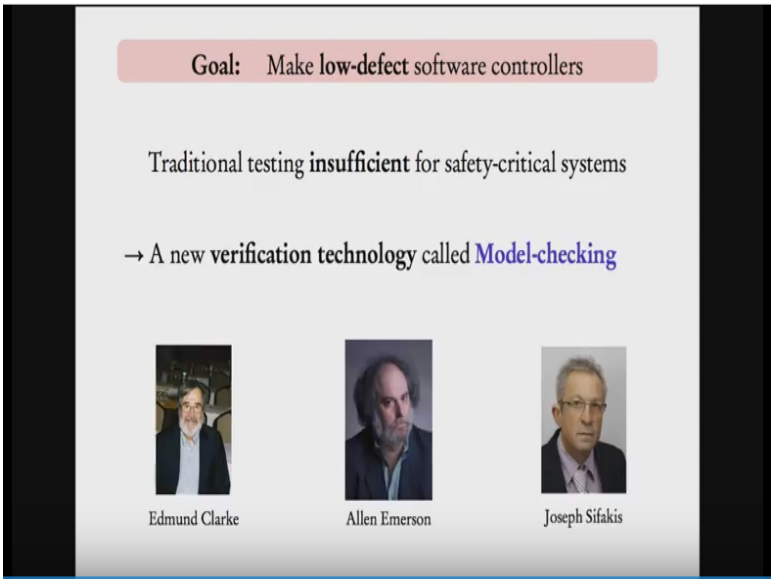
Let me give you an idea of some famous errors which have had dramatic consequences. In 1994, an error in the floating point division of Intel's Pentium II Processor cost a loss of 475 millions us dollars for the company. In 96, an error in the control software of Ariane 5 rocket made it crashed in 36 seconds after its launch. In the years 1985 to 1987, an error in the control software of a radiation therapy machine cause the death of 6 patients due to radiation overdose. Clearly errors due occur and errors due go are noticed.

(Refer Slide Time: 04:20)



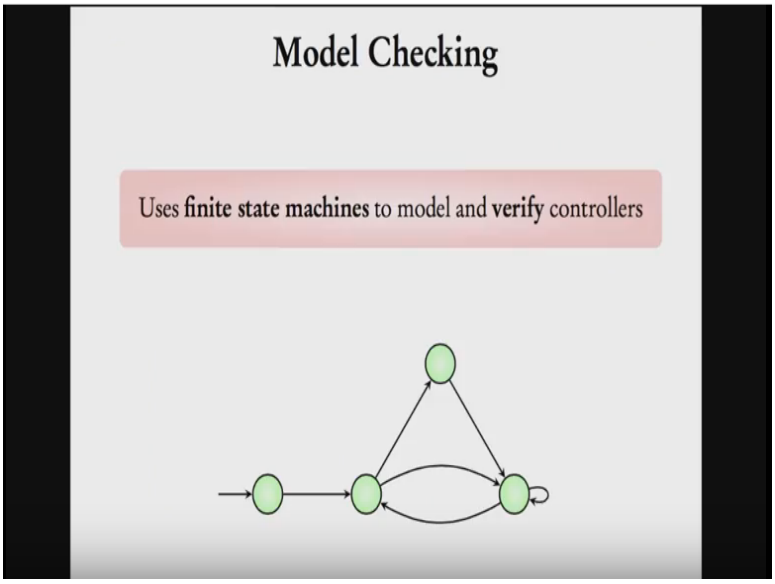
The goal is to make software controllers with as low a defect as possible and traditional testing is insufficient for safety critical systems.

(Refer Slide Time: 04:36)



This has led to a new verification technology called model checking. Clarke, Emerson and Sifakis have been the pioneers of this model checking approach.

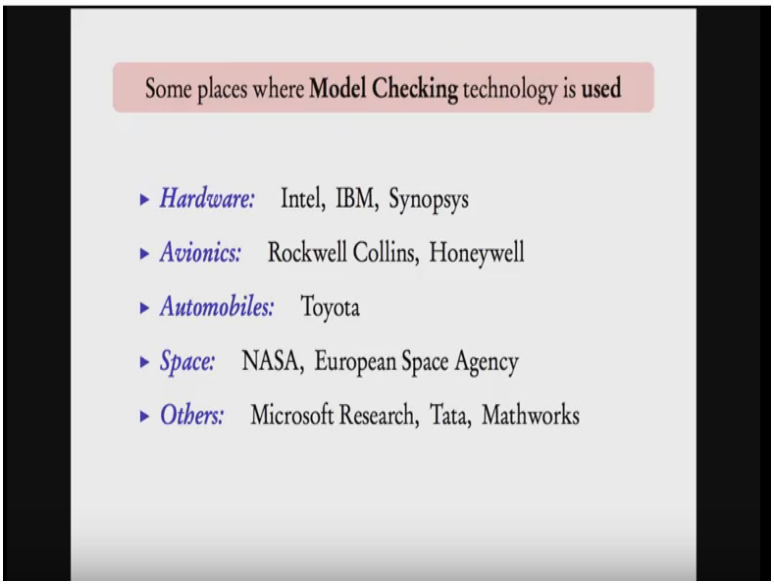
(Refer Slide Time: 04:51)



Traditional testing involves giving a lot of test cases to the code and checking if the code is correct on these test cases. However, these just not ensure that there are no errors. There could be other cases where the code might behave incorrectly. Model checking follows a different approach. A mathematical model of the code is created and requirements of the code are checked on these mathematical models.

This check can be automatically performed by existing model checkers. Typical mathematical models are extension of finite state machines. In this course we will study about this model checking technology.

(Refer Slide Time: 05:48)

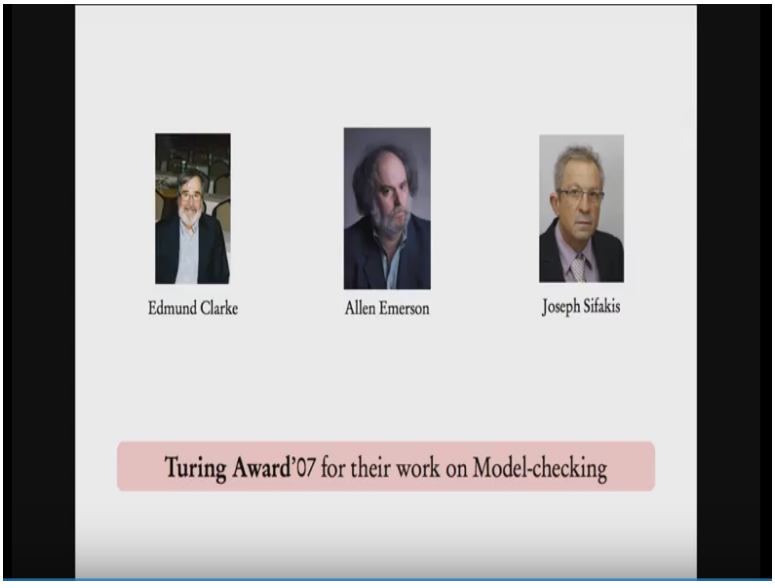


Model checking is becoming popular among many industries it has been extremely useful in hardware. Intel extensively uses model checking methods to certify its products. In avionics, there has been recent interest to incorporate this technology as well. Same is true for automobiles and space technologies.

(Refer Slide Time: 06:17)

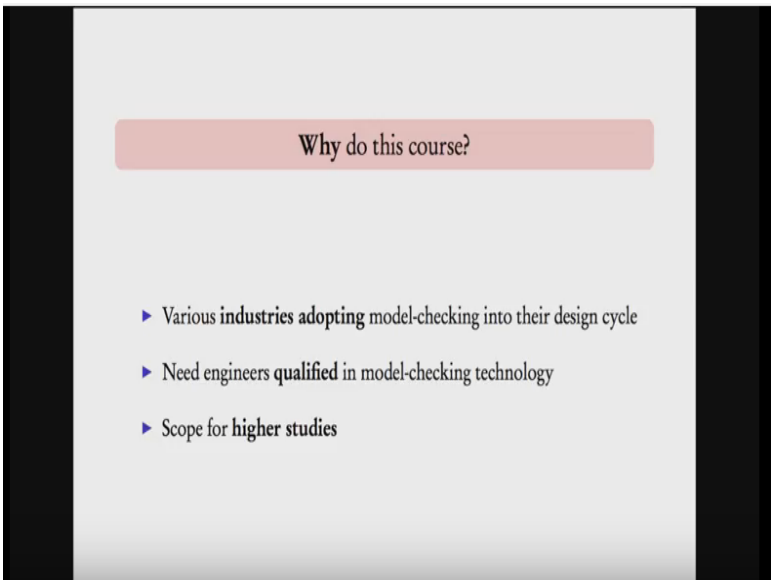
Apart from the Industries there are many universities all over the world doing research in this area.

(Refer Slide Time: 06:28)



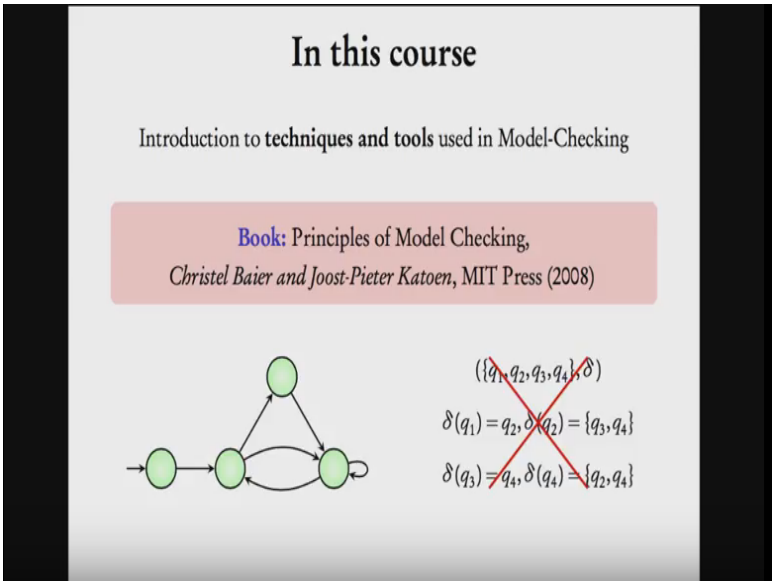
In 2007, Turing Award, the Turing Award is the equivalent of the noble prize in computer science. The Turing Award for 2007 was handed to the founders of model checking.

(Refer Slide Time: 06:45)



Why should we do this course? As mention before various industries are beginning to adopt this model checking technology into their design cycle. So there is a need for engineers who are qualified in this technology. Secondly the students there is a lot of scope for doing higher studies in this area.

(Refer Slide Time: 07:09)



So in this course we will try to give an introduction to the techniques and tools used in model checking. We will be following the book principles of model checking a Christel Baier and Joost Pieter Katoen. As for as possible I will try to avoid scary notation and use pretty pictures instead. I am not expecting much prerequisites. Bachelors or Master students in CS, IT, EEE, ECE are

welcome. Engineers in the industry who are looking forward learning new technologies for verification are welcome to this course as well. Hope you will enjoy this course