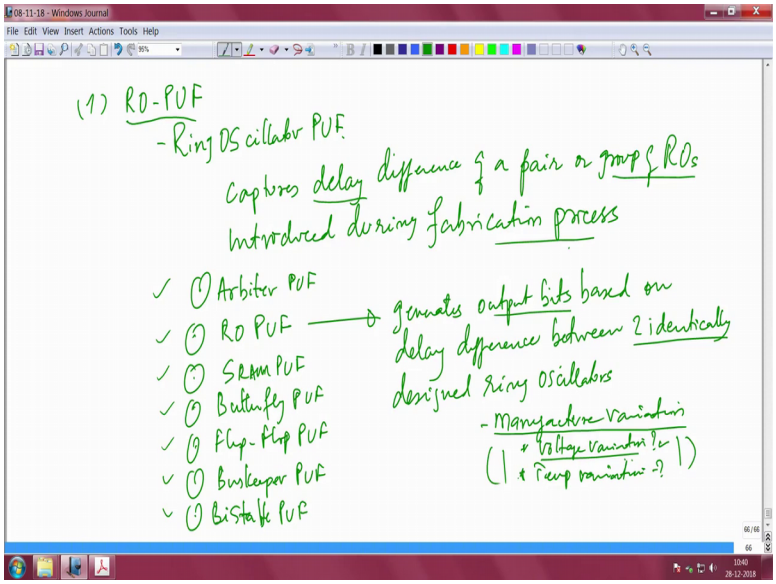
Advanced IOT Applications
Dr. T V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Lecture - 30
Device Security Part – 3

So, let us look at this ring oscillator PUF.

(Refer Slide Time: 00:32)



Why this is important is because if you have an embedded system which has an FPGA let us say on board and so, it is reconfigurable hardware right. So, you could essentially construct several digital circuits from them isn't it you can easily construct them. So, ring oscillator is just an odd number of inverters which are cascaded with a feedback right.

So, if you a very simple circuit, that simple circuit which has inverters which are cascaded odd number of inverters which are cascaded with the output connected back to the input creates a unique set of outputs for because of delay variation. This has been studied very well extensively over the last few years and I will just cap recap everything into a big framework. So, here ring oscillator PUF essentially captures the delay difference of a pair or group of ring oscillators introduced during fabrication process ok. When we talk about physically unclonable functions we talk about so, many of them arbiter PUF, ring oscillator PUF, SRAM PUF, butterfly PUF, flip flop PUF, bus keeper

PUF, bi stable PUF and so on. So, so much rich amount of literature is available on device identity using physically unclonable functions.

It is just that our imagination has to come and perhaps we have to start using them ah. So, the issue really is if you want to use them you have to measure these now these variations is not it. So, I think it is the best time now for IoT to succeed in making these measurements because, everything requires an apparatus if you want to build an equipment or an apparatus on board simple stuff. Then that is an interesting area of system building itself or electronic system building itself

So, if you just look at RO PUF it generates output bits based on delay difference between 2 identically designed ring oscillators. And why is it happening? It is because of manufacture variations I said this already many times, but the trouble really is alright if you do manufacture variation you are exploiting; what happens if these digital circuits these electronic systems are applied different voltages. What happens if you if your rail voltages are different because, most of these circuits today or most of the electronic components today work over a range of a voltages.

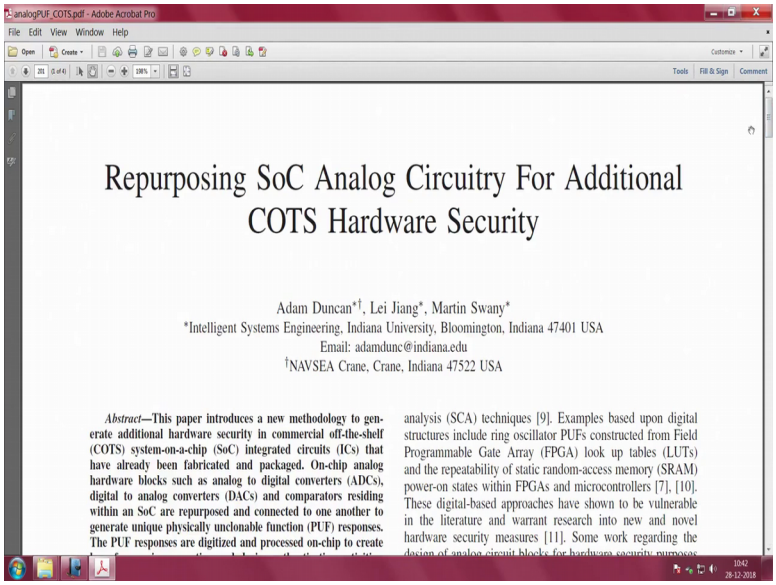
So, what happens if there is a voltage variation and what happens if there is a temperature variation? This is also something that researchers have studied in the light of you know building very reliable ring oscillators. If there is a big flip you can quantify this bit flip by applying some nice statistical methods, applying a little bit of probability, study. And, then you can essentially build an error correction system as well to introduce reliability in the measurement right.

So, just imagined that if you have an array of bits and you know that consistently this one there is one particular bit that flips right and the remaining are all ok. Then how do you remove that? Either you should remove that or bring in some amount of redundancy into the system in some not redundancy more in terms of correction of that bit and to make it consistent right. So, people have investigated erect error correction codes for building reliable p-waves as well.

So, this is the area in which there has been a lot of research plenty of research has gone in it is good, because it is a something that has been there and it is well known by the way ok. So, that is important. The next thing is you can do all these tricks in digital, but what about some tricks in analog can you do anything with ADCs, can you do anything

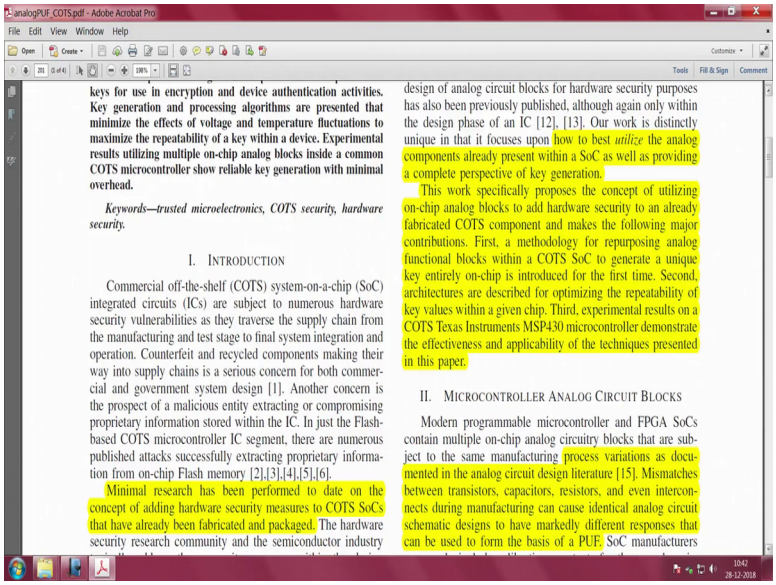
with DACs and do you can you extract any signatures from them. The big answer is yes ok.

(Refer Slide Time: 04:49)



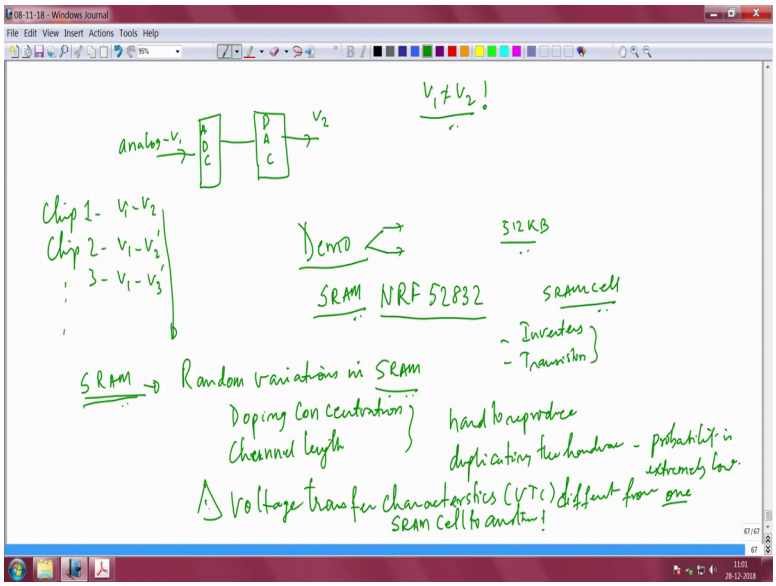
There was a recent paper which I will point you to which appeared in the year 2018 and this paper you can look up Repurposing SoC Analog Circuitry For Additional COTS Hardware Security.

(Refer Slide Time: 05:05)



So, this is something very interesting ok. So, I can quickly tell you before we move on to some demo, I just want to tell you the following.

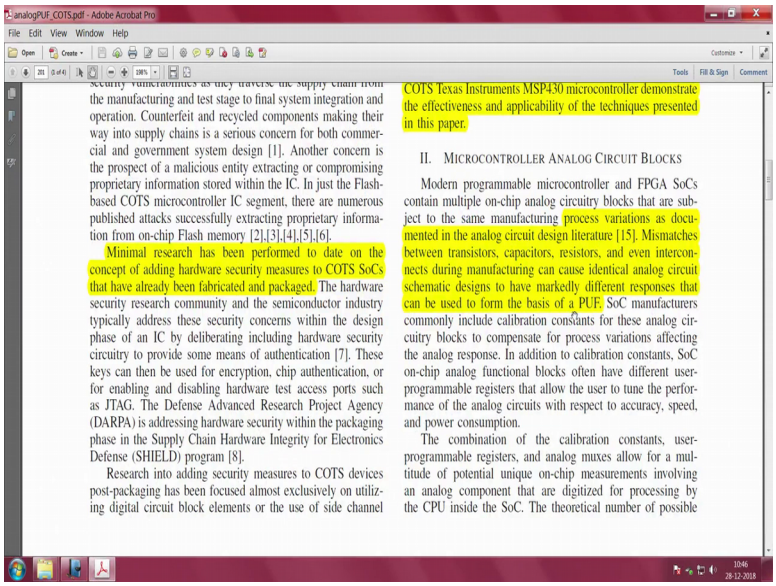
(Refer Slide Time: 05:16)



Supposing you have an ADC ok, ADC you feed in here. So, you have analog so, this is an ADC alright. You feed some analog voltage, some analog voltage you feed you call this let us say V 1 and I get a digital output this digital output I will give it to a DAC. I will give it to a DAC you must get back V 1, but what you get will be V 2. Now, it is consistent that V 1 is never equal to V 2 not equal to V 2 that is the real idea. So, if you can come if you can have and this V 1 not equal to V 2 is essentially across several chip. Chip 1 has some V 1 not equal to V 2 ok, some variation between V 1 and V 2 and chip 2 same thing V 1 you V something some other V 2 you get.

So, you get V 1 V 2 here and here you say V 1 V 2 prime you get something else you get here again chip 3 V 1 you apply you get something else. So, you can see that as you keep changing the SoC you get you start getting different variations in output voltages. This is something that one can exploit again and sort of uniquely identify a device. In essence this is the idea of what this paper has been has been doing and again it is trying to exploit the variations in silicon and so on. So, let us quickly look up that paper and then you know move on to some simple experiment.

(Refer Slide Time: 07:10)



Now, the question really is this the authors have noticed that there is minimal research that has been performed with respect to commercially of the available of the shelf available SoCs. Take the case of RO PUFs, the RO PUF is nice there is no problem at all; it is a nice way people have beaten it quite well have understood how to design reliable RO PFSs. And, you seen the number variants and different types of PUFs that in the digital world that you can build. The thing is it is not that you have an embedded system with you and you can develop an RO PUF or around it that is not going to work. Because, if you look at your own embedded system you will have an SoC which perhaps as an ADC and has GPIO pins, has a radio perhaps inside has a power supply component inside has certain RAM inside, certain flash inside.

So, on top of that if you have to build an RO oscillator ring oscillator; that means, you need additional configurable hardware which you have to put on board and then get that oscillator to be built. The this is not going to be you cannot work backwards or you cannot do a retrofit of a RO of any of the digital PUFs perhaps that is also the reason why the RO PUF world has not really taken off overall after so, much of good amount of research that has happened in that area. It is a quite a mature area actually.

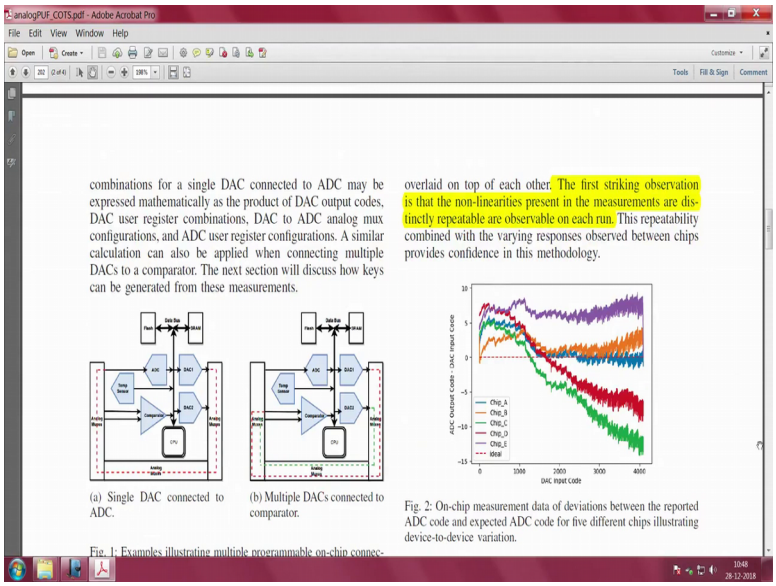
So, here they have the authors have actually found that issue and said look if you take an SoC you are bound to have an ADC because, sensors are going to be connected in the IoT world you will have a lot of sensors. So, an ADC will be there. So, can I use an ADC

if I have a slightly sophisticated controller I will also have a DAC inside for example, there is one series of MSP 430 pa popular TI microcontroller which actually has a DAC inside it. And, they use that feature in it and then they show you that you can actually construct unique signatures from the hardware using analog blocks of a controller. So, this is essentially what the paper is saying.

So, here is a very important point that you should note that it isn't that governments all over the world are not looking at it. The defence advanced research project DARPA, agency DARPA is actually addressing hardware security within the packaging phase in the supply chain hardware integrity for electronics; electronics defence it is called SHIELD. Please look up SHIELD it is an active area and DARPA itself is trying to address these very serious issues on device identification and so on and so forth.

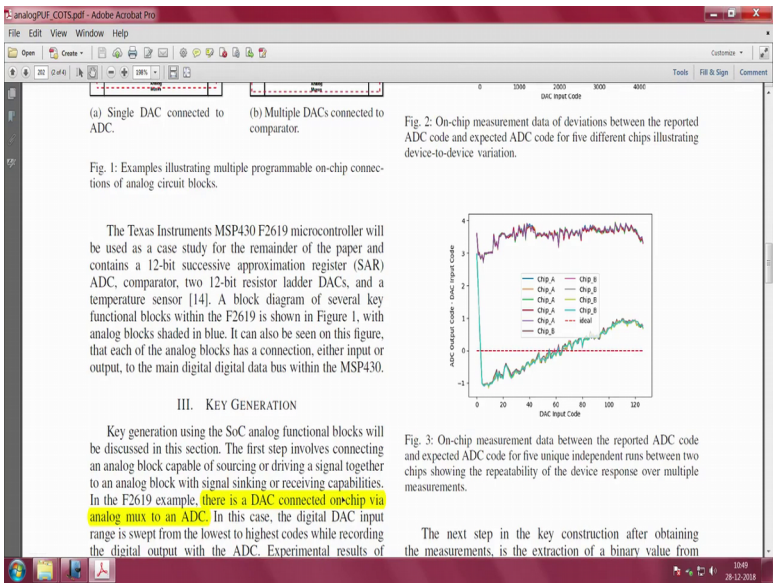
So, that is essentially what it as a paper is saying. So, as I said this is to utilize analog components present in an SoC um. So, utilizing on chip anolog blocks using commercially off the shelf components and use a MSP 430 microcontroller and see how exactly this can be exploited. Again it is the same story process variations as documented in analog circuit is what we are trying to exploit here, mismatches between transistors capacitors resistors and even interconnects. During manufacturing is what can cause identical analog circuit schematics designs to have markedly different responses that can be used to form the basis of a physically un cloneable function. So, it is a quite in detail it goes on to into that.

(Refer Slide Time: 10:49)



So, I would not go into the detail of this because really you may have to spend time understanding several aspects of this yourself. But, you can see that there is a combination of ADC and DAC which essentially will lead to this idea of measuring V 1 and V 2. And, trying to see whether V 1 is all the time equal to V 2 across all chips from the same you know same ma manufacturer or is it different.

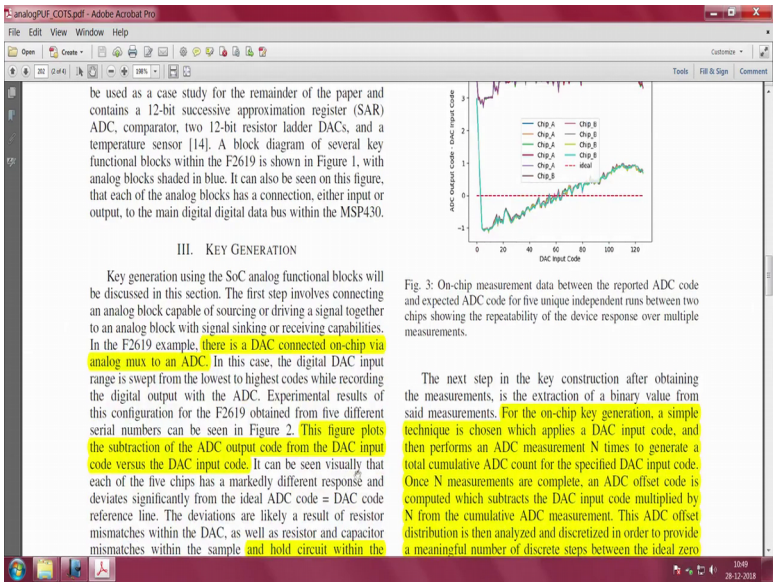
(Refer Slide Time: 11:19)



It turns out that they use this MSP 430 F2619 microcontroller and they go actually this paper is good because, it goes one step forward and even says what do you do if you

generate a unique signature; can I do a key generation ok. So, key generation using SoC analog functional blocks is also discussed a. So, in F2619 there is a DAC connected to the is chip via analog mux and they try to use this in the process of generating a signature.

(Refer Slide Time: 11:50)



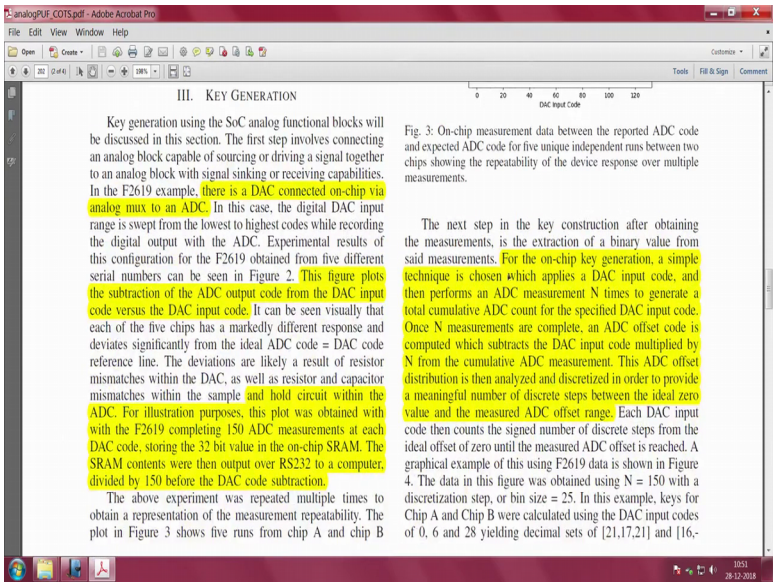
So, la let us look at a figure 2; figure 2 this is the DAC output the x axis is DAC input code. The y axis is ADC output code minus DAC input code just a nice way of representing to show a high; I would say a hard way of now representing to show that it is a very unique way of of the components. Ideally of all the chips that they have tested, ideally if there is if they are all supposed to work exactly the way they are supposed to give you an output you should get 0 right.

If you do V 1 minus V 2 you should get 0 correct, V 2 is the output V 1 is the input you have connected it from a from V 1 you connected it to a a ADC. So, you got digital output the digital output is connected back to a DAC as input and that is given back as an output as analog; you get back V 2 V 1 minus V 2 should be 0 that 0 is shown here. So, ideally your 0 line should be here, but not really true the tests I have indicated A B C D E; that means, there should be 5 lines you can see that there are clearly 5 graphs or 5 plots which show that there is a distinct variation between each one of them.

In essence this on chip measurement data of deviations between reported ADC code and expected ADC code for 5 different chips illustrates the device to device variation. So,

you can also do a DAC output DAC input code you have the same DAC input code and ADC output code minus DAC input code ok; this is another way of representing it. So, I would urge you to read this paper, but tell you that this on chip exploiting analog components is a very good possibility for creating unique signatures for devices.

(Refer Slide Time: 13:54)



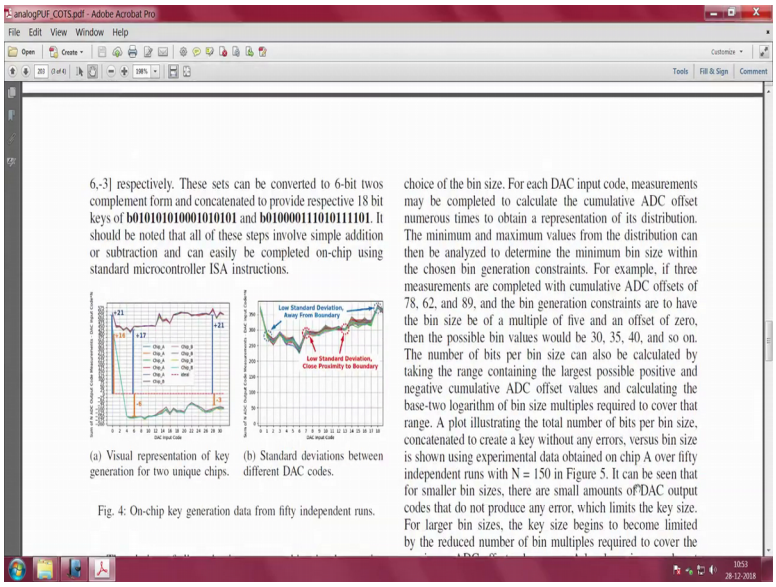
Now, the big question that remains is how do I use that unique signature, do I use it for key generation and therefore, I can use it for authentication purposes. Or, essentially what is it that I can use it for that is another part of the story. But, the fact that you can derive unique signatures from the different analog and digital blocks itself is an exciting area for device identity in a for the IoT world.

So, here the key construction this paper is also talking about how to construct the key, it is nice to read this paragraph to understand a little more in detail. Key construction after obtaining the measurements is the extraction of a binary value from said measurements. You after all need a set of bits right which essentially is your key. For on chip key generation a simple technique is chosen which applies a DAC input code and then performs an ADC measurement N times to generate a total cumulative ADC count for the specified DAC input code.

So, this essentially is the this result I suppose is indeed the way you do the you know cumulative count. So, you do N times to generate a total cumulative ADC count for the specified DAC input code. Once N measurements are complete and ADC offset code is

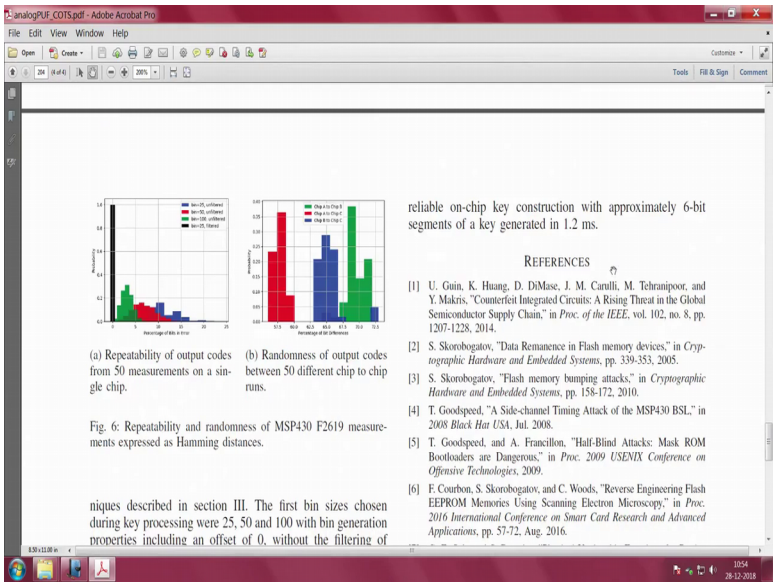
computed which subtracts the DAC input code multiplied by N from the cumulative ADC measurement. This ADC offset distribution is then analysed and discretized in order to provide a meaningful number of discrete steps between the ideal zero value and the measured ADC offset value. So, this in essence is shown in figure 4 actually.

(Refer Slide Time: 15:53)



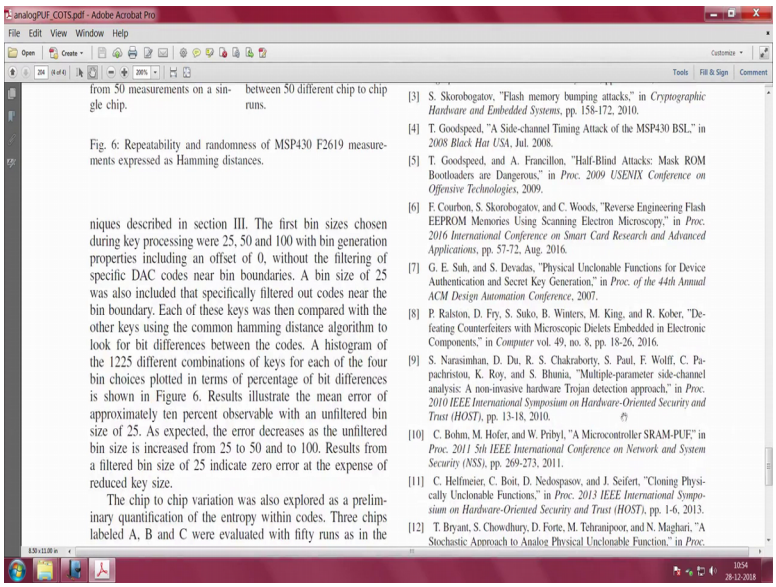
I think this is what yeah this figure 4 is showing you visual representation of key generation for two unique chips and b, is standard deviations between different DAC codes ok. So, this is x first one is DAC input code and the y axis is sum of N ADC output code measurement minus DAC input code times N. Yeah this one and this is my x axis is DAC input code and the y axis is sum of N ADC output code measurements minus DAC input code times N ok. So, if you plot you will get this nice little results.

(Refer Slide Time: 16:38)



And essentially they do an experimental value you know verification of the key properties and all that. And, then they show that yeah look you can exploit the analog blocks and they also do repeatability of output codes from 50 chip measurements on a single chip. And, they find that it is a pretty nice way to to generate a signature from analog blocks.

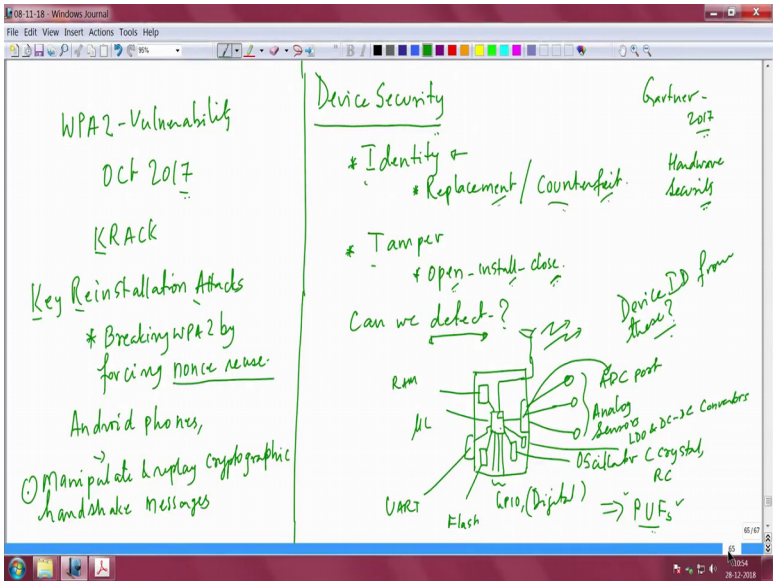
(Refer Slide Time: 17:06)



So, the summary is you can do lot of nice things even with analog blocks that as essentially you takes you to what I was trying to get at; when I started this discussion

here where, I was showing you that UART, flash, GPIO, microcontroller RAM, ADC port, analog, sensors, oscillator so, many things are all potential for signatures.

(Refer Slide Time: 17:14)



And somehow you should get to device identity security particularly identity, if you want to find device identity then you know replacement counterfeiting or something that you could perhaps easily catch. And tamper of course, will have to be somehow ha handled separately alright. So, let us now go move on to a demo ok. In the demo part this is work done by primarily 3 people: one is a PhD student working here by name Girish Vidya, he was assisted by one Suhas Sudhakara who is a project associate and then another person by name Vasanth who is a project assistant involved in this. So, they are team of 3 people who were trying to understand can we do something with respect to SRAM ok.

They took a standard microcontroller like a Nordic n NRF 52832 which is a very popular microcontroller SoC; has a Bluetooth low energy technology built into the system. It has a RAM of about 512 kilobytes and they said let us see what we can do with this chip. So, before we go into seeing the demo let us look at this SRAM itself ok; little bit of detail into the SRAM and then we will move on. See the SRAM why there are variations actually come; random variations random variations in SRAM. It is very interesting to for you to look at SRAM and design a PUF around SRAM because, if you take the smallest possible IoT device which has an 8 bit microcontroller, has a sensor which is

just giving you some binary output like a let us say you have a a sensor which says presence or absence of humans ok.

So, typically you would use a let us say a comparator based PIR sensor Pyro electric Infrared Sensor which actually gives you an analog output ah, but you can put a comparator and say presence of human or absence of human. And just give you 2 levels ok, 0 means no human 1 means human is present; you could just simplify the whole circuit and most often that is how that is the that is it is that is how it is used. So, if you take a very simple system like that which is you do not need any sophistication there then there also you will have some RAM is not it. So, that RAM can be exploited.

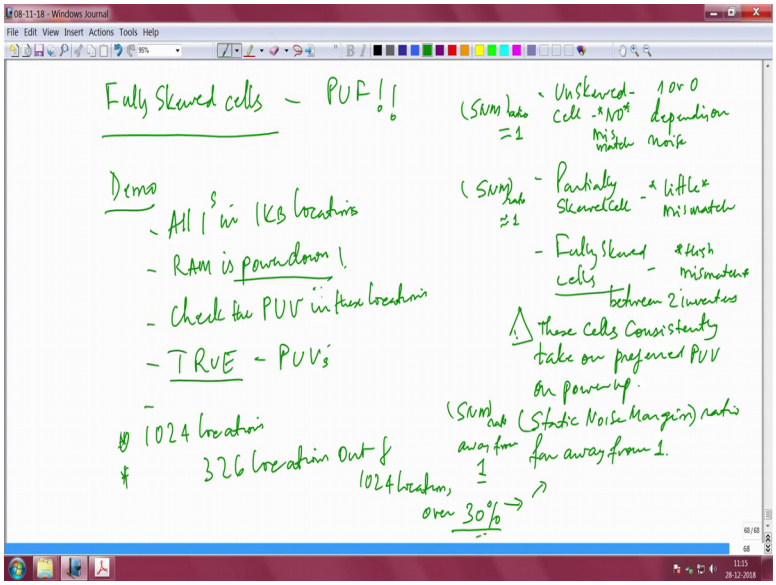
So, you could perhaps seriously consider using using SRAM as a possible hardware part for creating your un cloneable function. So, let us see what you can do ok. So, the reason why random variations happen in a SRAMs is because of doping concentration. I am just getting a little more detailed so, that you understand where a what do you mean by variations. It is basically because, of go doping concentration and channel length ok; you have these two issues. And therefore, because of essentially when you talk about RAM you are talking about inverters and pass transistors right.

So you so, a RAM comprises of inverters digital I mean inverters inverters and transistors I would not go into the detail of construction of a SRAM cell SRAM cell, but essentially this is what it will have. So, all this doping concentration and channel length variation random variations is because of these two reasons. So, it is hard to reproduce hard to reproduce hence, you have a nice probability of duplicating the hardware feature is close to 0 ok; probability of duplicating the hardware duplicating the hardware is and cannot say 0 is low I would say; duplicating the hardware probability is low is extremely low for references cannot be 0. So, it has to be close to 0 so, that would be a very guarded way of saying this whole thing.

So, essentially the semiconductor variation that we are talking about what does it mean; it simply means that the voltage voltage transfer transfer characteristics voltage transfer characteristics are VTC is is different from is different different from from one SRAM cell SRAM sorry SRAM R A M; one SRAM cell to another that is the reason. And therefore, you can actually go ahead and calculate the noise margins per VTC and all that; you can calculate the voltage transfer characteristics. Calculate the smallest noise

margin you can do that, you can calculate the static noise margin ratio; you can calculate the you can do several things.

(Refer Slide Time: 24:09)



Ultimately you are looking for fully skewed fully skewed cells. If you have unskewed cell, if you have an un skewed cell there is no miss no there is practically no mismatch between inverters ok. No mismatch in inverters the cell can settle either to 1 or 0 depending on the noise. So, we can put that down un skewed un skewed you can either settle down to 1 or 0 depending on on noise ok. Then you can have partially skewed, what do you mean partially skewed partially skewed cell. So, un skewed cell partially skewed cell here there is again little match.

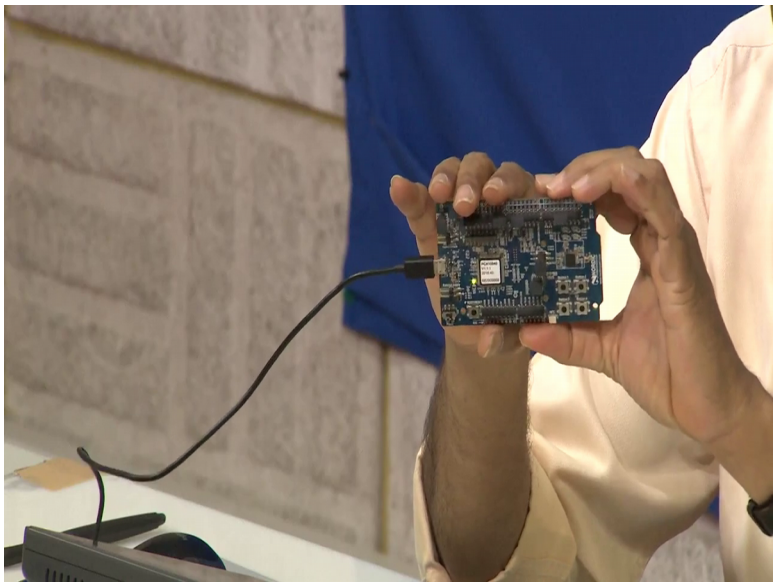
So, here there is answer is no no match at all no mismatch at all, here there is little the answer here is little here is no I will just put it no mismatch no mismatch little mismatch, here fully skewed. This is interesting here we have high mismatch high mismatch high mismatch between what 2 inverters which make up the SRAM cell thus, what will happen these cells. So, here is the key takeaway these cells these cells consistently consistently take on take on take on preferred power up value on power up.

So, essentially the I would say the start up (Refer Time: 26:56), the star static static noise margin margin ratio ratio start static noise margin ratio is far away from 1; here it is SNM. So, I will just put so, completeness you should say something about this right, this is closed. So, I will say SNM: Static Noise Margin ratio is close to 1 is close to 1 this is

SNM ratio is equal to 1 ok. And this is away from 1, here is the away from 1. Therefore, these cells consistently take on preferred power up values that is PUV as they call the Power Up Value which is on power up they take a value; they take a preferred power up value.

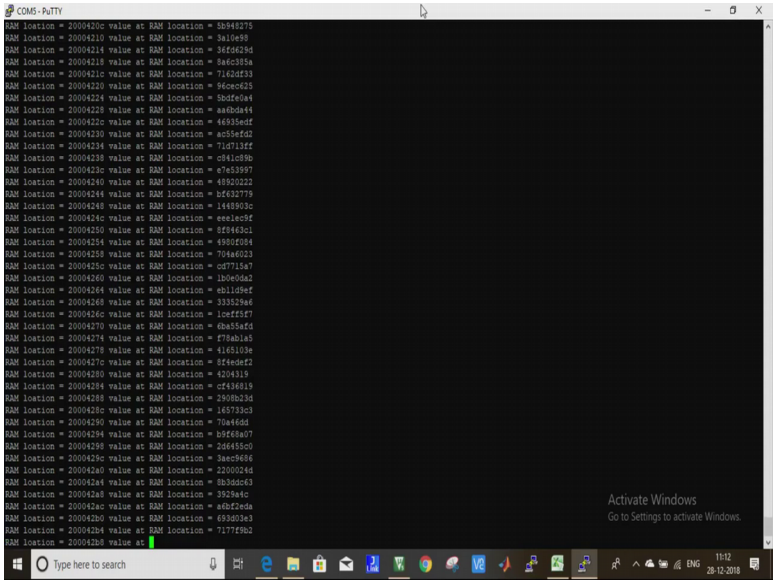
Because, the static noise margin ratio is far away from 1 ok; they that I think answers the the reason why s the SRAMs essentially have the same start of values repeatedly when you reboot the system or when you power up the system. So, now what should the demo have? In the demo you want to essentially see whether whatever we have discussed whether there is the this kind of a phenomena actually occurring. So, let us start like this I have Vasanth here with me, we will set up this demo and I will start showing you some hardware and then we will move on to the physical demonstration alright ok.

(Refer Slide Time: 29:27)



So, this is the board of interest. What is this board? This is a commercially available board you can buy it in any market. This is essentially the Nordic 52832 board, it has I says told you it has 512 kilobytes of RAM. It is a arm cortex M 4 based system and we are trying to see whether we can replicate this SRAM PUF in the in this demonstration ok. So, Vasanth is here and he will operate the screen from the beginning.

(Refer Slide Time: 30:07)



So, I would like you to look at the screen now; yes here in the screen we will first initially do a reset and the screen will show up alright. So, here is a connection; now look at what is happening. If you look carefully he has selected 1 kilobyte of RAM out of the 512 kilobytes of RAM and he is writing all f f into these locations ok. You can see all our f; that means, he is writing all 1's into the location demo all 1's in 1 kilo byte location. Now, let us see after this what will happen he will shut down the RAM.

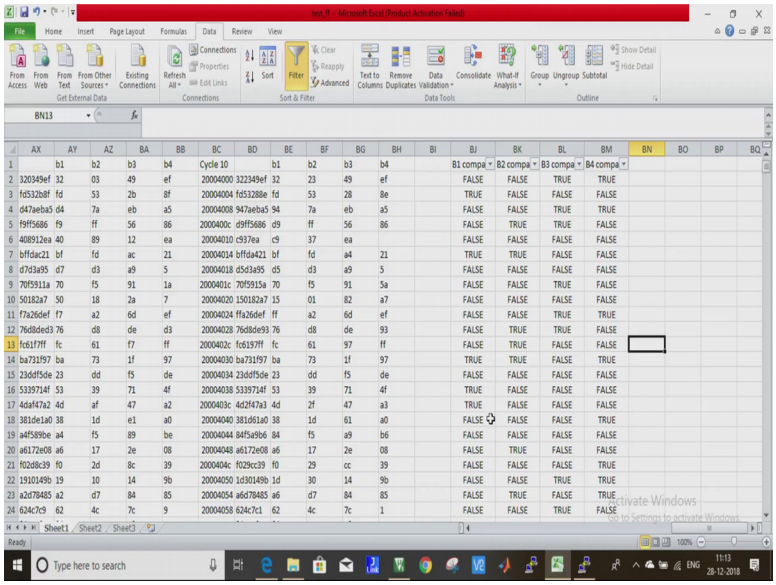
RAM is powered down you power down the RAM and then check the Power Up Value: PUV alright. So, it is still writing maybe let us see what what happens here it is good to examine this screen here. So, here is a small application program written by in embedded code which will allow you to do this experiment right turning off RAM for 3 seconds. And, then it starts again and then he starts reading byte by byte just to see whether the values are actually matching. So, you can see it says at RAM location some RAM location you can see there is a pause here. RAM location is so and so value at RAM location is this value right.

So, he has been reading that consistent he is been reading them. Now, any number of times you do it should read back the same values because, the power up value should be the same. And, let us see what he actually does; he has tabulated this result in a user friendly manner which will allow you to understand what is actually happening. If the start up values come back he calls it that location as true ok; true if the PUVs come back,

PUVs come back. In other words if it is fully he will only identify those which are fully skewed cells. And, then if they are fully skewed we will get back the same and then he will put a true there otherwise he will say it is false.

So, let us say if you have 1024 locations how many of them come back with a true and how many of them come back with a false; will be the final result that you want to see correct. So, let us do that and let us shift to the excel sheet. So, let us finish this let us finish this yeah it is done now. So, let us move on to the excel sheet right.

(Refer Slide Time: 33:30)



So, you can see the left side is essentially the number of cycles he has done this experiment. And he takes byte by byte, he does a comparison there is cycle 1 byte by byte cycle 2 cycle 3 cycle 4 something like; I think he goes some 10 cycles before he does a comparison and shows you this beautiful table false false false true and so on ah. So, you can see that you have to count every time it is true only then it will actually relate to a count right. So, this perhaps perhaps cannot be used for a over for total count because, this is saying false ah. So, first one false false true true does not make any sense.

So, you should see true true true true only then you make it as a count. So, let us see how many of them ultimately are in all the rounds in all the comparisons that they he does; how many of them actually indicate all true. So, come down so so now, let us see yeah so, let us see that so, here you go. You see a number here which says 300 and you can

read it down below there 326; just can you move the mouse there yes yes little bit up ah. So, you essentially have 326 locations out of 1024 locations come back with the same PUVs. Roughly I would say a little of about 30 percent n of the RAM locations can a little over 30 percent, can give you consistent you know a set of PUV values.

But the ta task is not over here what does it mean if the voltage rail voltage is you know the applied voltage is different; how will that change the game. Will this 30 percent become 20 percent? I was a little greater than a little greater it should not be just 30, it is a little over 30 I would say just write it as over 30. Over 30 percent will it become less than 30 percent, will it be 15 percent. These are tests and you know investigations that one have to complete before one can use them in any way useful in any particular manner to generate a nice signature

ah So, in other words this area of device identity using unclonable functions, signatures that are extracted from analog blocks and from digital blocks and perhaps all other blocks is a useful area. One may want to pursue by reading up sufficient amount of literature in this area; you can come out with some nice algorithms. And see how you can extract a signature first and then think about how can I use this signature; can I use it for creation of a key or can I use it for some other purpose of of you know uniquely identifying in its ones own imagination of how this data this can be used. People even talk about you know creating you know sort of a hash functions and so on and so forth, but that is all open how to use it.

So, at this juncture I wanted to introduce you to this exciting area of physically unclonable functions as a security module as part of this course.

Thank you very much.