

**Blockchains Architecture, Design and Use Cases**  
**Prof. Sandip Chakraborty**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture – 07**  
**Bitcoin Basic – I**

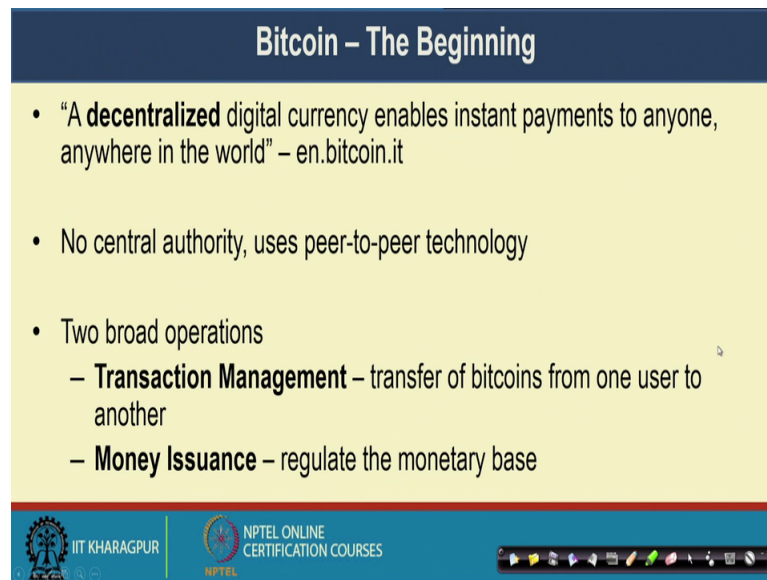
Welcome to the course on Blockchain. So, till now we have looked into the basic blockchain architecture, its use cases like Bitcoin and smart contracts and then we have looked into the basic cryptographic primitives in blockchain like we have looked into the hash and cryptographic hash, functions and digital signatures and we have also looked into that how digital signatures and cryptographic hash functions are used in the context of blockchain to make it secure and tamper proof.

So, now will look into the interesting topic of Bitcoin, which is emerged as a digital cryptocurrency; But the fundamentals or the base of Bitcoin is depending on the blockchain architecture. So, thus entire Bitcoin network it supports blockchain, and blockchain works as a fundamental building block behind the development of Bitcoin.

So, we look into the different aspects of Bitcoin how this digital currency is designed, how you take the technical aspects as well as the economical aspects together and have a nice currency system, which can replace the existing currency system with the properties like with the properties like permission list where anyone can joint in that cryptocurrency system or there will be no controller on that particular currency system, no government or bank will have control on that. So, how we can utilize this concept of blockchain to develop into such kind of cryptocurrency that we will look into the next three lectures.

So, let us start with the discussion of Bitcoin.

(Refer Slide Time: 02:11)



**Bitcoin – The Beginning**

- “A **decentralized** digital currency enables instant payments to anyone, anywhere in the world” – en.bitcoin.it
- No central authority, uses peer-to-peer technology
- Two broad operations
  - **Transaction Management** – transfer of bitcoins from one user to another
  - **Money Issuance** – regulate the monetary base

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, by the definition so, this definition I have taken from one Bitcoin websites; so, it details that Bitcoin is a decentralized digital currency of which enable instant payments to anyone anywhere in the world. So, it is a cross country payment system that was one of the primary objective behind the development of Bitcoin and another objective was that we want to have a cross country transaction support, such that no government organization will have a control over it.

And the nice properties of Bitcoins are this decentralized architecture, which actually helps it to get a system quarrel one has external control over this currency system it works; So, over a complete peer to peer network and it supports different levels of securities so, that the entire system becomes tamper proof, but it works in a nice way.

So, we do not have any kind of central authority. So, the entire Bitcoin network works in the peer to peer principal peer to peer technology principals. So, in Bitcoin we have two broad operations one is the transaction management; that means, transferring of Bitcoins from one user to another like you want to purchase something you can purchase it by utilizing the Bitcoin or you want to make some transfer of money from someone in India to someone say in USA.

So, you can do it using Bitcoins. And a second interesting aspect at the important aspect of Bitcoin is the money issuance; where you need to regulate the monetary base of Bitcoin like the economical aspects of a coin base of a digital currency that need to be

ensured where in our normal banking system, we have the banking authority the central bank which will regulate the money inside the country.

They will create the new money or with time, they can demolish or they can drop the old money and that control that in what rate the moneys will be generated that is in general controlled by the central banking system. But in case of a digital currency system, which works over a peer to peer network ensuring that is a kind of interesting, but difficult thing and Bitcoin actually solves that problem by utilizing set of technical concepts that we look into little details.

So, the basics of the Bitcoin it is creation of coins the economic aspects, that I have mentioned earlier like you require a kind of control supply of money. So, you must limit the currency to have their value. If you put up a lot of currency in the system, it will gradually reduce the value of that particular currency. So, that comes from the economic concept. So, any such maliciously generated currency, you need to reject those and you need to accept the actual currency which are flowing in your system.

Now, this Bitcoins the new Bitcoins that need to be generated like a normal currency so, this new Bitcoins they are generated during the Bitcoin mining timing.

(Refer Slide Time: 05:41)

**Bitcoin Basics – Creation of Coins**

- **Controlled Supply:** Must be limited for the currency to have value – any maliciously generated currency needs to be rejected by the network
- Bitcoins are generated **during the mining** – each time a user discovers a new block
- The rate of block creation is adjusted every 2016 blocks to aim for a **constant two week adjustment period**

Information Source: <https://en.bitcoin.it/wiki/>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

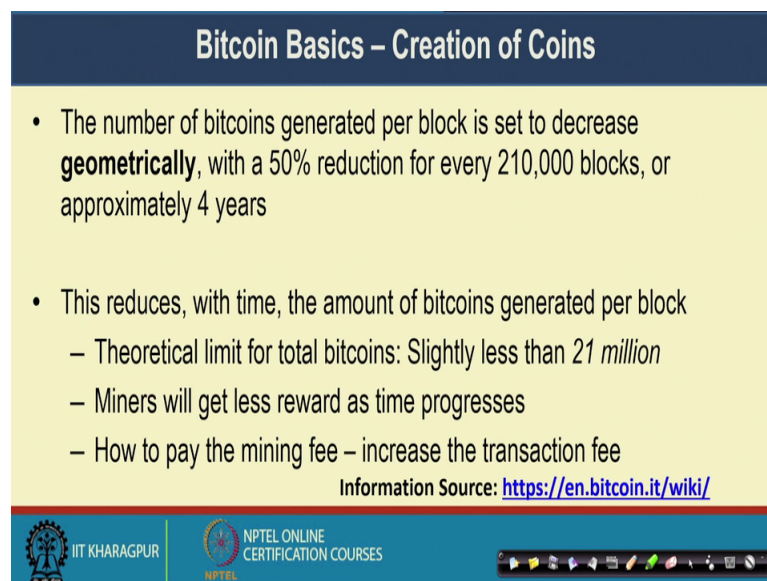
So, each time a user discovers a new block that particular procedure we call mining. So, the methodology for Bitcoin mining we look into more details, but whenever a particular

person generates a new block during that time, the system generates some new Bitcoin and that person who is investing his or her power and system power and the time to generate the new blocks by participating in the mining procedure.

For those person they are awarded with that newly generated Bitcoin. But as I have mentioned that this flow of money or the generation of new Bitcoins that need to be regulated; so, you should have some rate at which you need to adjust a block creation.

So, in case of standard Bitcoin architecture, the rate of block creation it is adjusted over every 2016 blocks, and the aim is to have a constant of 2 week adjustment period; like at every two week you will readjust the amount of money that has been generated by mining the blocks. So, to again to regulate the money the number of Bitcoins which are generated per block it is said to decrease geometrically.

(Refer Slide Time: 07:06)



**Bitcoin Basics – Creation of Coins**

- The number of bitcoins generated per block is set to decrease **geometrically**, with a 50% reduction for every 210,000 blocks, or approximately 4 years
- This reduces, with time, the amount of bitcoins generated per block
  - Theoretical limit for total bitcoins: Slightly less than 21 million
  - Miners will get less reward as time progresses
  - How to pay the mining fee – increase the transaction fee

Information Source: <https://en.bitcoin.it/wiki/>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Like with time the amount of money that is awarded by for mining a new block that got drop that get decreased with a rate of 50 percent reduction for every 2 lakh 10000 blocks, which takes approximately 4 years; that means, at approximately 4 years the price of mining like if you participate in the mining procedure and if you can successfully mine a new block, the amount of award that are you will get from the Bitcoin network that will gradually drop at a that that gradually drop at every 4 years.

So, this reduction with time, this reduces the amount of Bitcoins which is generated per block; so, theoretical limit for the total Bitcoins which you can generate with this particular limit, which it is slightly less than 21 million.

So, once you have generated some 21 millions of Bitcoin after that you will find that by participating in the mining procedure, you will not get any reward. So, that way as the time progresses the miner will get less reward, but the question comes that how can you fee pay them are how what should be the incentive for them, for participating in the mining procedure.

Because the currently the miners participate in the mining procedure because they can get some Bitcoin by participating in the mining in the mining procedure they have to solve some mathematical puzzle, for which they have to invest the system powers and the time; by investing that system powers and time in the mining procedure if the mining is successful there you added with some Bitcoin. But as that at that as that will get decreased with time and once this approximately 21 millions Bitcoins are generated, the system will not generate any new Bitcoin. So, the miners will not get paid from the network for participating in the mining procedure.

So, in that case what can be there incentive like, you can use the transaction fee you can you can increase the transaction fee or get more transaction fee from the participating user to pay them, to pay the miners for participating in the mining procedure. So, that way this way that is that is the normal rule of economy, where you establish a system and one once the system is well established.

Then you imply some charges on the users will make a transaction and once the user will make a transaction, with the transaction they have to provide certain transaction fee, and the transaction fee will be utilized to make a payment to the miners, who will participate in the new block mining procedure. So, that is that is a kind of broad objective.

(Refer Slide Time: 10:04)

Projected Bitcoins									
Date reached	Block	Reward Era	BTC/block	Year (estimate)	Start BTC	BTC Added	End BTC	BTC Increase	End BTC % of Limit
2009-01-03	0	1	50.00	2009	0	2625000	2625000	Infinite	12.500%
2010-04-22	52500	1	50.00	2010	2625000	2625000	5250000	100.00%	25.000%
2011-01-28	105000	1	50.00	2011*	5250000	2625000	7875000	50.00%	37.500%
2011-12-14	157500	1	50.00	2012	7875000	2625000	10500000	33.33%	50.000%
2012-11-28	210000	2	25.00	2013	10500000	1312500	11812500	12.50%	56.250%
2013-10-09	262500	2	25.00	2014	11812500	1312500	13125000	11.11%	62.500%
2014-08-11	315000	2	25.00	2015	13125000	1312500	14437500	10.00%	68.750%
2015-07-29	367500	2	25.00	2016	14437500	1312500	15750000	9.09%	75.000%
2016-07-09	420000	3	12.50	2016	15750000	656250	16406250	4.17%	78.125%
2017-06-23	472500	3	12.50	2018	16406250	656250	17062500	4.00%	81.250%
	525000	3	12.50	2019	17062500	656250	17718750	3.85%	84.375%
	577500	3	12.50	2020	17718750	656250	18375000	3.70%	87.500%
	630000	4	6.25	2021	18375000	328125	18703125	1.79%	89.063%
	682500	4	6.25	2022	18703125	328125	19031250	1.75%	90.625%
	735000	4	6.25	2023	19031250	328125	19359375	1.72%	92.188%
	787500	4	6.25	2024	19359375	328125	19687500	1.69%	93.750%

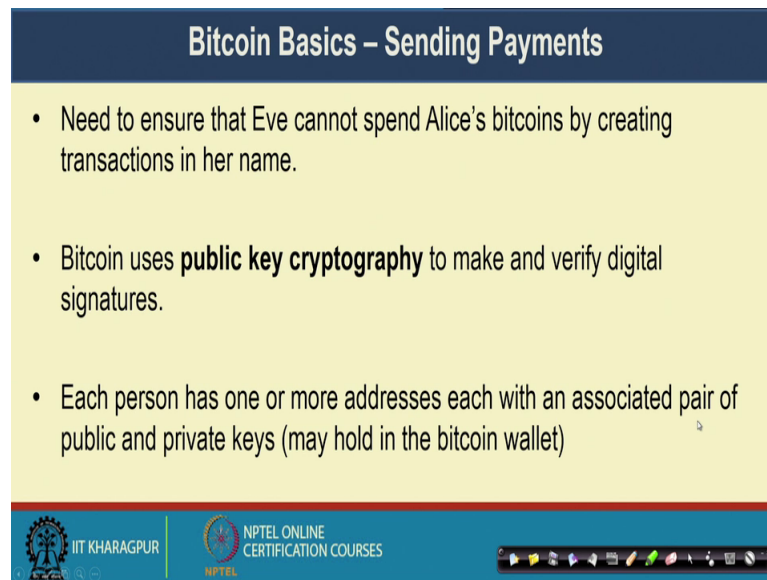
Information Source: <https://en.bitcoin.it/wiki/>

Now, this is a kind of projected Bitcoins with respect to time. So, around this third January 2009 during that time, when the initial time the block was generated during that time you can get around 50 Bitcoin per block. So, if you if you generate a new block for every block you can get 50 Bitcoin.

So, as the time progress, for the first 4 years it was 50 Bitcoin then from 28 11 two 2012, it reduces to 25 Bitcoin per block, then from 9 7 2016 it will get 12.50 Bitcoin per block, then it is again gradually reduced to 6.25 Bitcoin per block, that way gradually the price of Bitcoin at the reward amount that you can get by participating in the mining procedure are a generating a new block, that will get reduce over time.

So, this chart it gives a nice indication about how much time it will take, to completely generated the targeted Bitcoin means by the time this Bitcoin reward per block that will get nearly 0 or close to 0, and by doing this mathematical calculation you can find out that once it reaches to 21 million Bitcoins in the network, then you will not get any further reward from the system ok.

(Refer Slide Time: 11:43)



**Bitcoin Basics – Sending Payments**

- Need to ensure that Eve cannot spend Alice's bitcoins by creating transactions in her name.
- Bitcoin uses **public key cryptography** to make and verify digital signatures.
- Each person has one or more addresses each with an associated pair of public and private keys (may hold in the bitcoin wallet)

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, whenever you are sending the payments in the Bitcoin network, you need to ensure that any other person cannot spend the Bitcoin owned by one person. So, if cannot spend the Bitcoin which is owned by Alice, by creating the transaction in her name. So, the intruders in the network they will not be able to create some kind of false transaction.

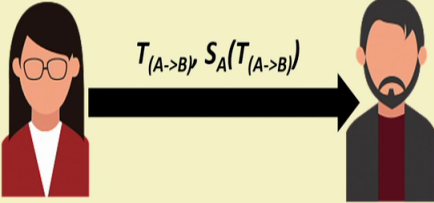
To ensure that Bitcoin uses the public key cryptography as I have mentioned earlier, that it uses this digital signature concept along with e c d s a digital signature algorithm to make and verify the transactions using digital signatures. So, each person who are participating in the Bitcoin network, they has one or more search addresses with an associated pair of public and private key.

So, every user can have one or more address based on your wallet or how many addresses you can create, but every Bitcoin address they will get associated to it appeared of public and a private key.

(Refer Slide Time: 12:58)

### Bitcoin Basics – Sending Payments

- Alice wish to transfer some bitcoin to Bob.
  - Alice can sign a transaction with her private key
  - Anyone can validate the transaction with Alice's public key



The diagram illustrates the process of sending a payment. On the left, a woman representing Alice is shown. An arrow points from her to a man representing Bob on the right. Above the arrow, the text  $T_{(A \rightarrow B)}$  and  $S_A(T_{(A \rightarrow B)})$  is written, indicating that Alice is sending a transaction and its signature to Bob.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, say Alice wants to transfer some Bitcoin to bob. So, what Alice can do, Alice can sign the transaction. So, Alice can create a transaction and along with the transaction Alice can put her signature and send that to bob. And during putting the signature as we have discussed earlier during the crypto primitives that, Alice can use her private key to sign that particular transaction and send the signature along with the transaction.

Now anyone in the network, they can validate this transaction with Alice's public key. So, they can decrypt the transaction with the public key and validate that whether the transaction is originated from Alice or not.



(Refer Slide Time: 13:45)



The slide is titled "Bitcoin Basics – Sending Payments" in a dark blue header. The main content is on a light yellow background and consists of a bulleted list of five steps. At the bottom of the slide, there is a footer with logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a small video inset showing a man speaking. The text "Information Source: <https://en.bitco>" is also present in the footer area.

- Alice wants to send bitcoin to Bob
  - Bob sends his address to Alice
  - Alice adds Bob's address and the amount of bitcoins to transfer in a "transaction" message
  - Alice signs the transaction with her private key, and announces her public key for signature verification
  - Alice broadcasts the transaction on the Bitcoin network for all to see

Information Source: <https://en.bitco>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, these are the set of states that Alice has to follow if she wants to make some transfer of certain bitcoins to Bob. So, first Bob sends his address to Alice. So, this is a cryptographically generated address which is transferred to Alice, now Alice adds Bob address and the amount of Bitcoin that need to be transferred in a transaction message. So, she constructs a transaction message with that particular transaction information, that the transaction is from Alice's address to Bob address and along with that, she puts the amount of Bitcoin that need to be transferred.

Now, Alice signs the transaction with her private key and announces the public key with which anyone can validate that transaction, then Alice broadcasts this transaction in the Bitcoin network for all to see that transaction. So, the broad idea is that Alice constructs the transaction, puts her signature along with the transaction, and also puts the public key to validate the transaction, then broadcasts the transaction in the Bitcoin network.

(Refer Slide Time: 14:55)

**Double Spending**

- Same bitcoin is used for more than one transactions
- In a centralized system, the bank prevents double spending
- How can we prevent double spending in a decentralized network?

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, whenever you are creating the transaction or the Alice is creating the transaction, in a digital currency you may have the problem of double spending. So, what is double spending it is like that the same Bitcoin, it is used for more than one transaction. Say for example, Alice has total 50 Bitcoin and the same amount of Bitcoin she has transferred to Bob and say she has transferred to Charlie.

Now, these transactions are kind of double spending that both the transactions cannot be valid simultaneously. Now, in a centralized system like a banking agency this is very easy to validate such that kind of double spending. So, whenever you are submitting the transaction to the bank, the bank can validate that where are whether you are doing a double spending like you have say 1000 rupees in your hand, but you are making two transactions worth 1000 rupees.

So, bank can validate that accordingly can take the necessary actions, but the question comes that in a decentralized network how can you prevent double spending so, that someone will not be able to make two transactions with the same Bitcoins.




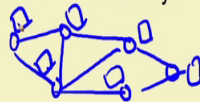
So, to prevent double spending we use this blockchain in Bitcoin. So, the details about the transactions which are sent, these transactions are forwarded to all the users in the Bitcoin network or as many as other computers as possible and we use this blockchain, which is a constantly growing chain of blocks to contain a record of all such transactions and this blockchain is maintained by all the peers in the Bitcoin network.

(Refer Slide Time: 16:33)

### Handle Double Spending using Blockchain

- Details about the transaction are sent and forwarded to all or as many other computers as possible
- Use **Blockchain** – a constantly growing chain of blocks that contain a record of all transactions
- The blockchain is maintained by all peers in the Bitcoin network – everyone has a copy of the blockchain

Information Source: <https://en.bitcoin.it/wiki/>



So, everyone has a copy of the blockchain. So, in case of a Bitcoin network so, all the nodes who are there in the network, they maintain a copy of blockchain with them.




So, everyone has the same copy of blockchain which contains all the transactions. So, the transactions they are put inside this blockchains and everyone can validate those transactions as well.

(Refer Slide Time: 17:12)

### Handle Double Spending using Blockchain

- To be accepted in the chain, transaction blocks must be valid and must include **proof of work** – a computationally difficult hash generated by the mining procedure
- Blockchain ensures that, if any of the block is modified, all following blocks will have to be recomputed

Information Source: <https://en.bitcoin.it/wiki/>



Now, to get a transaction accepted for a chain the transaction blocks, they need to be validated and they also should include something called the proof of work. So, this proof

of work is the Bitcoin consensus algorithm that we will discuss later on, where the miners need to solve a computationally difficult hash problem based on challenge and they need to solve a hash puzzle that we have discussed earlier; that means, they need to find out something like this, like they need to find out  $y$  with hash value of  $x$  and some nonce.

So, the miners need to find out this nonce where  $x$  and  $y$  are known. So, ideally in case of Bitcoin  $x$  is known  $x$  is a record of the previous hash, and the set of transactions organized in a merkle tree and a merkle root; and  $y$  hash certain constants like  $y$  should have certain number of zeros at the beginning. So, that was the constant which is imposed on  $y$ .

Now, the target of the miner is to find out this nonce so, that they can find out a hash value  $y$  with this constant like there would be certain number of zeros at the prefix of  $y$ . So,  $y$  will look like something like there is a set of 0's and followed by say some numbers. So,  $y$  will look something like this. So, there would be certain number of zeros at the beginning, which is put as the puzzle for this particular mining problem in bitcoin. So, the attacker not the attacker sorry the miners need to find out the nonce so, that  $y$  looks like this.

So, finding out this kind of the solution for this kind the puzzle as we have discussed earlier it is a very difficult you do not have any kind of computationally efficient algorithm to solve this kind of puzzle, and the best way of doing it is to try with different random nonce or different values of nonce.

So, the miners actually find out this nonce values which work like a proof of work. So, the response to this particular challenge or the solution of this particular puzzle is this nonce value which, work as the proof of work. So, this particular algorithm is known as the proof of work, which is used to validate a particular block in the blockchain.

Now, blockchain it ensures that as you have discussed earlier like blockchain ensures because of this hash pointer property and every block also contain the hash of the previous block. So, the entire thing is tamper proof. So, if any block is modified, you need to modify all the subsequent blocks and the hash for all the blocks need to be recomputed.

(Refer Slide Time: 20:21)

**Handle Double Spending using Blockchain**

- When multiple valid continuation to this chain appear, only the longest such branch is accepted and it is then extended further (**longest chain**)
- Once a transaction is committed in the blockchain, everyone in the network can validate all the transactions by using Alice's public address
- The validation prevents double spending in bitcoin

Information Source: <https://en.bitcoin.it/wiki/>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, this things we have mentioned earlier briefly that when multiple valid continuation to this chain appear, only the longest such branch is accepted and it is then extended further. So, we always consider the longest chain in the Bitcoin network we will go to that little details in the next lecture, that how we actually achieve this.

And once a transaction is committed in the blockchain then everyone in the network can validate the transaction by using Alice's public address. So, this somehow spend or prevents the double chain double spending in Bitcoin. Like if there are two transactions with the same Bitcoin everyone will be able to see that and the miners will not be include both the transactions in the same block.

And if a transaction is already included in a block, then the same transaction same Bitcoin that will not be used in a in the next block or in any of the subsequent blocks; by maintaining this kind of blockchain architecture which are already get got validated, you can this way prevent the double spending in the network.

(Refer Slide Time: 21:36)



### Bitcoin Anonymity

- Bitcoin is permission-less, you do not need to setup any “account”, or required any e-mail address, user name or password to login to the wallet
- The public and the private keys do not need to be registered, the wallet can generate them for the users
- The **bitcoin address** is used for transaction, not the user name or identity

Information Source: <https://en.bitcoin.it/wiki/>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, another concept in the context of Bitcoin is anonymity like this Bitcoin is permission less, that you do not need any set up of any account or you do not require any email address user name or password to login to your wallet anytime you can join in the network. Now, whenever you are joining in the network, you do not again need to register your public and the private keys, the wallet can generate the public and the private keys for you.

Now, the interesting part is this Bitcoin address which is used for transaction; so, this Bitcoin address or not the user name or the identity. So, they do not carry the identity of a particular user, they are some kind of anonymous address through which you will not be able to guess who is the actual user. And a single user can have more than one addresses as well.

So, let us look into a brief that how this addresses are generated.

(Refer Slide Time: 22:30)

**Bitcoin Anonymity**

- A **bitcoin address** mathematically corresponds to a public key based on ECDSA – the digital signature algorithm used in bitcoin
- A sample bitcoin address: 1PHYrmdJ22MKbJevpb3MBNpVckjZht89hz
- Each person can have many such addresses, each with its own balance
  - Difficult to know which person owns what amount

Information Source: <https://en.bitcoin.it/wiki/>

*Handwritten notes: H160 ( PA PUB )*

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this Bitcoin addresses are basically corresponds to it is a mathematical correspondence, of the public key which is used by the user. So, the broad algorithm is something like this. So, you have generated your public key.

So, once you have generated your public key, say Alice's generated Alice's generated her public key say this is the public key for Alice. So, what you do that once you have this public key for Alice which is generated by this ECDSA algorithm, you apply a hash over that. Normally, we apply a 160 bit hash function on top of that. So, after applying the hash function on top of that, you extract the first few bits out of that hash and use that as your address.

So, a sample Bitcoin address look something like this so, this is a sample Bitcoin address, which is generated by applying the hash function over the public key and then performing certain other operations.

So, there are multiple versions of Bitcoin addresses different addresses applies different set of operations, but the broad idea is that you have the public key on the public key you apply a hash algorithm 160 bit hash algorithm. And then by taking the hash value and processing over that doing some for the processing over the hash value finally, you generate a address like this. So, by looking into this address it is difficult to guess who is the actual user.

Now, as I you have mention that each person can have more than one such a addresses, even some Bitcoin users they have many of such addresses and each of the address will have their own value. So, that way it is difficult to know that which persons owns the account. So, these particularly prevents the anonymity of Bitcoin transactions by you can always see the transactions inside the blockchain, but you have no way to identify who the corresponding user is.

(Refer Slide Time: 24:35)

**Bitcoin Script**

- Alice makes a transaction of BTC 20 to Bob. How Bob will claim those transactions?
- A transaction is characterized by two parameters
  - Alice sends some bitcoins: **the output (out) of the transaction**
  - Bob receives some bitcoins: **the input (in) of the transaction**
- We need to determine that **a transaction input correctly claims a transaction output**

Handwritten diagram:  $(T, SA, KA, PUB)$  with a note  $T(A \rightarrow B)$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, Bitcoin has a nice concept called Bitcoin script. So, what is Bitcoin script? Say Alice makes a transaction of some 20 Bitcoin to bob; Now, how Bob will claim that transactions, how Bob will know that those transactions are intendant for him and no one else other than Alice’s actually initiated that transaction.

Now, the broad concept we have discussed earlier, the broad concept says that you include the signature of Alice along with the transactions. So, along with the transactions you include two other things. So, whenever you are sending some transactions. So, along with the transactions, you send a signature and you send the public key say you send a you send a public key of public key of Alice.

So, this three information’s are transfer and if this three information are transferred then Bob can look into this information and Bob can find out that well in this transaction is from say A to B where A is the address of Alice, and B is the address of Bob and by verifying this signature with this public key, Alice can find out sorry Bob can find out



that this transaction is a valid transaction, which is actually came from Alice and no one else in the network and this is not a kind of force transaction in the network. So, after doing this validity check on this top of this transaction, then Bob can accept the transaction.

Now, every transaction it is characterized by two parameters one is called like the input parameter, the input parameter is something like that the input of a transaction is Bob is receiving some Bitcoins and the output of a transaction is Alice is sending some Bitcoin to bob.

(Refer Slide Time: 26:50)

**Bitcoin Script**

- Alice makes a transaction of BTC 20 to Bob. How Bob will claim those transactions?
- A transaction is characterized by two parameters
  - Alice sends some bitcoins: **the output (out) of the transaction**
  - Bob receives some bitcoins: **the input (in) of the transaction**
- We need to determine that **a transaction input correctly claims a transaction output**

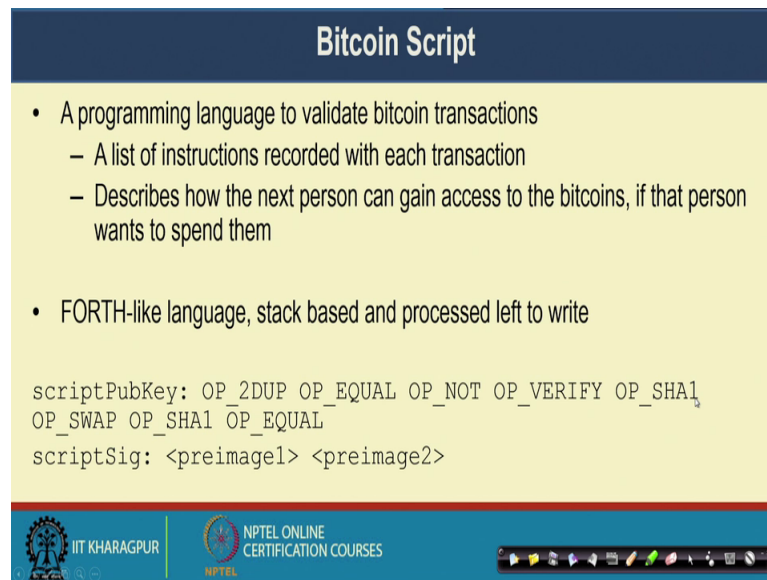
Handwritten diagram:  $A \text{ } \underline{B} \text{ } 20 \rightarrow B \text{ } \underline{\$} \text{ } 20$   
The word "output" is written below the underlined "B" and "20".  
The word "input" is written below the underlined "\$" and "20".

Footer: IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, it is something like this that Alice is making a transaction of say Bitcoin 20 to Bob. So, Bob is receiving this Bitcoin 20 from Alice. So, this is the input of the transaction that Bob is receiving Bitcoin 20, and this is the output of the transaction or out of the transaction which says that Alice is transferring some Bitcoin 20.

Now, what we need to determine here that the input of a transaction correctly claims the a transaction output. So, the transaction which has been made from Alice, it is a correct transactions and it is intended for Bob only.

(Refer Slide Time: 27:38)



**Bitcoin Script**

- A programming language to validate bitcoin transactions
  - A list of instructions recorded with each transaction
  - Describes how the next person can gain access to the bitcoins, if that person wants to spend them
- FORTH-like language, stack based and processed left to right

```
scriptPubKey: OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1
OP_SWAP OP_SHA1 OP_EQUAL
scriptSig: <preimage1> <preimage2>
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

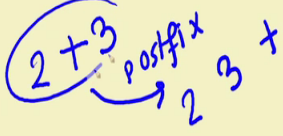
So, for that what we do rather than transferring the public key or transferring the signature what Bitcoin does Bitcoin transfer some code, which is some scripts. So, Bitcoins script is a programming language to validate Bitcoin transaction. So, it is a list of transactions or list of instruction records with each transaction, and it describes that how the next person who can gain access to the Bitcoin if that person wants to spend them.

Now, Bitcoin script is a forth like language, it is a stack based and processed left to right we will see an example, of this forth like language. So, this is an example of a Bitcoin script, we will go to the details of this particular script that what does it mean and what are the individual operators of this script looks like. So, before going to that as this Bitcoin script is a inspired from this forth language let us look briefly about how forth works.

(Refer Slide Time: 28:34)

### How FORTH Works

- A stacked based computer programming language originally designed by Charles Moore
  - A procedural programming language without type checking
  - Use a **stack** for recursive subroutine execution
  - Uses **reverse Polish notation (RPN)** or **postfix notation**

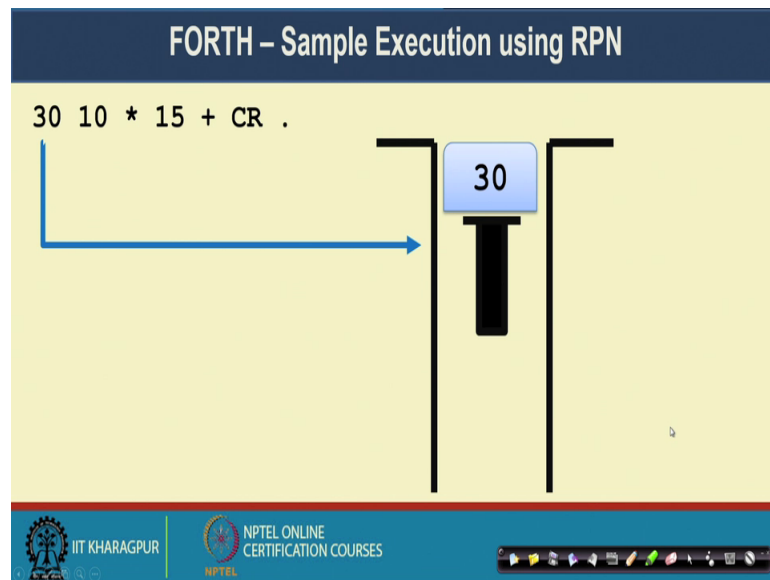


The slide features a blue header with the title 'How FORTH Works'. Below the header, a yellow background contains a bulleted list. The third bullet point is underlined. A handwritten blue diagram illustrates the conversion of the infix expression '2 + 3' to the postfix expression '2 3 +'. The numbers '2' and '3' in the infix expression are circled, and an arrow points from the '+' operator to the space between '2' and '3' in the postfix expression. The footer includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a Windows taskbar.

So, this forth is a stack based computer programming language, which was originally designed by Charles Moore. It is a procedural programming language without any type checking, it uses a stack for recursive subroutine computation and it uses this reverse polish notation or the postfix notation to make a computation. So, what is a postfix notation? Say in postfix notation say you want to make a addition of 2 plus 3 in a postfix notation, this operator is written after the operand.

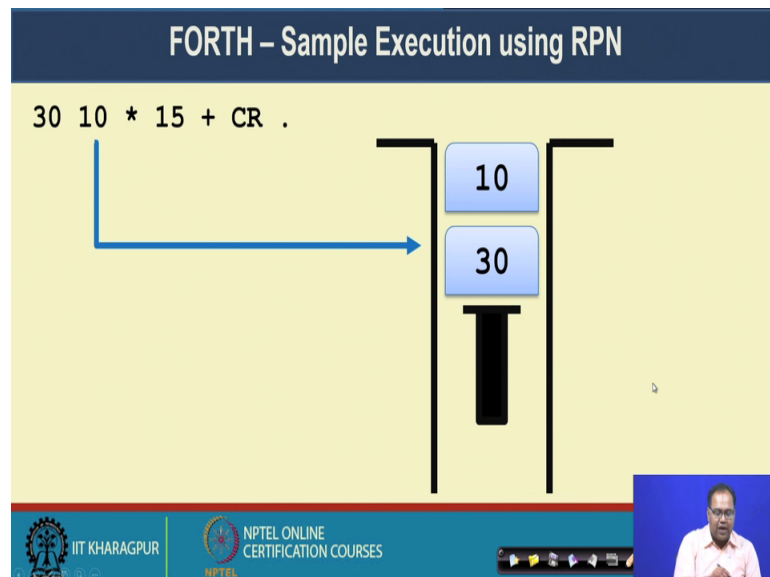
So, in the postfix notation it is written as 2 3 plus. So, if you convert infix operation. So, this operation we called as a infix operation. So, if you convert a infix operation to a postfix operation, the advantages is that you can use a stack to correctly do the computation of that particular postfix operation.

(Refer Slide Time: 29:37)

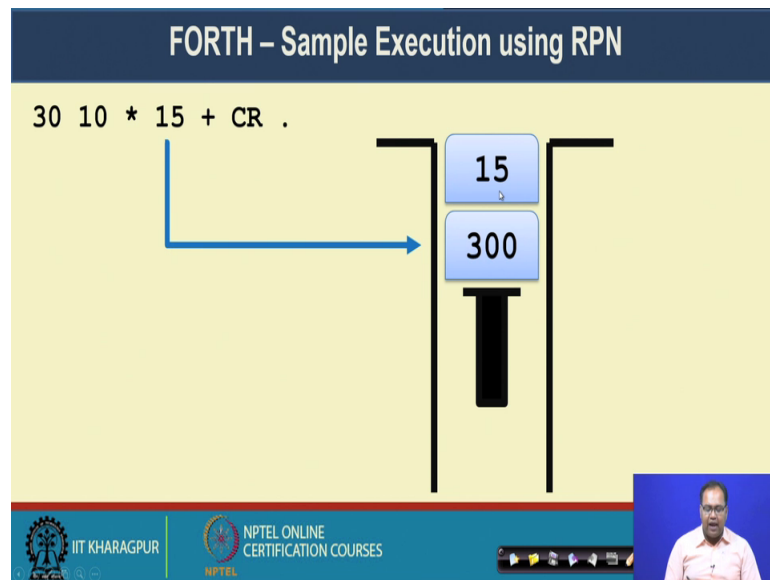


So, here is an sample execution of forth using this reverse polish notation, say you want to make this computation. If you want to make this computation you first put 30 to the stack, then once you have put that 30 to the stack, then you encounter 10, 10 is also pushed in the stack.

(Refer Slide Time: 29:52)



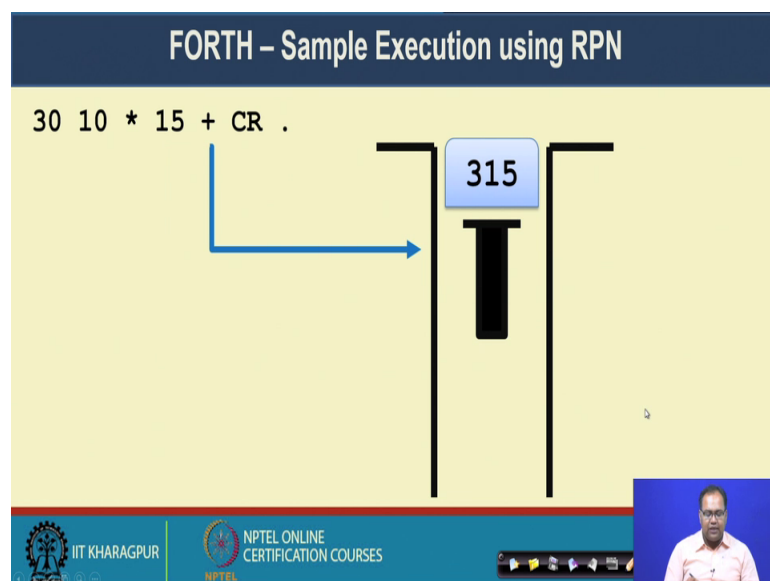
(Refer Slide Time: 29:56)



Then you come to the next thing, the next thing is a multiplication operator. Whenever you are getting a multiplication operator you pop out the top two element from the stack and do the multiplication.

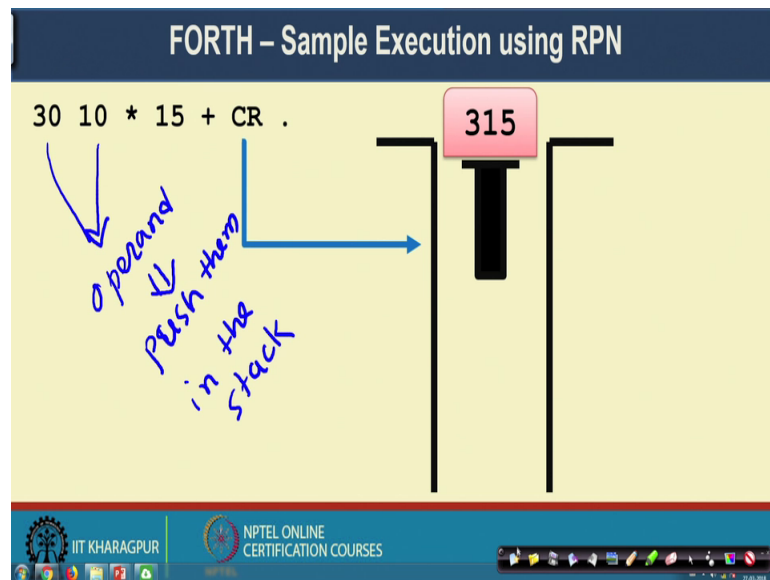
So, the multiplication result is 300 it is again pushed to the stack, then the next operand is 15, you if there is an operand you push it in the stack.

(Refer Slide Time: 30:18)



And after that you are getting a plus whenever you are getting a plus, you pop out the top two operand from the stack and perform the operation, push it again in the stack and you got three 115.

(Refer Slide Time: 30:32)



And finally you get this CR instruction whenever you are getting the CR instruction you finally, get the output which is there at the top of the stack as result.

So, the idea in this execution of RPN representation using a stack is that, whenever you are having some kind of operand like 30 or 10 and whenever you have the operand, then you push them in the stack, push them in the stack and whenever you are getting some operator like this start.

(Refer Slide Time: 31:14)

### FORTH – Sample Execution using RPN

30 10 \* 15 + CR .

The diagram shows a stack with the number 315 at the top. Handwritten notes in blue ink explain the execution of the RPN expression "30 10 \* 15 + CR .":

- An arrow points from the asterisk (\*) to the stack, labeled "Operator".
- An arrow points from the asterisk to the stack, labeled "pop out top two elements from the stack".
- An arrow points from the asterisk to the stack, labeled "perform the operation".
- An arrow points from the asterisk to the stack, labeled "push the result in the stack".

The stack is represented by two vertical lines with a horizontal bar at the top containing the number 315. The background is a light yellow color.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, whenever you have some operator, then you if this is a binary operator. So, you pop out top two element from the top two elements from the stack, then you perform the operation, and then push the result in the stack ok.

So, that way you can you can perform the execution using this reverse polish notation.

(Refer Slide Time: 32:03)

### FORTH – Sample Code

**FORTH Code:** `:FLOOR5 (n--n') DUP 6 < IF DROP 5 ELSE 1 - THEN;`

**Equivalent C Code:**

```
int floor5(int v){  
    return (v<6)?5:(v-1);  
}
```

• Defines a new word (a subroutine) called **FLOOR5**

Code Source: <https://en.bitcoin.it/wiki/>

Handwritten annotations on the slide include:

- Blue circles around the expression `(n--n')` in the FORTH code.
- A blue equals sign between the FORTH and C code.
- A blue circle around the number 6 in the C code, with the word "False" written next to it.
- A blue circle around the number 7 in the C code.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this is again of forth sample code and the corresponding C code representation, this is a forth subroutine a forth subroutine is named as an word so, in this subroutine. So, this particular instruction in the brackets, it means that we are taking one input and we are

possibly producing some other output. So, this is in the forth representation, then this particular instruction is equivalent to c instruction that v less than 6, if v less than 6 is true then you return 5 otherwise you return v minus 1.

Now, whenever you are representing in forth. So, this dup operation basically make a duplicate of whatever is there in the stack top. So, it is like that whatever input you are getting here. So, that input will be inserted in the stack.

So, you are if you are getting a value say 7, 7 will be inserted in the stack then you will get a 6, 6 will be inserted in the stack, then you will get a operand whenever you are getting the operand you compare this top two element in the stack by comparing that that you find out whether that is true or false. So, here 6 is less than you making and this operand. So, 7 is less than 6 it is it is false. So, the output of this operation is false. So, you take a false and push it inside the stack.

(Refer Slide Time: 33:38)

**FORTH - Sample Code**

**FORTH Code:**  
:FLOOR5 (n--n') DUP 6 < IF DROP 5 ELSE 1 - THEN;

**Equivalent C Code:**  
int floor5(int v){  
    return (v<6)?5:(v-1);  
}

- Defines a new word (a subroutine) called **FLOOR5**

Code Source: <https://en.bitcoin.it/wiki/>

So, in the stack now, you will have false, then you execute this if statement. So, if it is false so; that means, you will you will directly jump to the else statement and in else statement it is saying that whatever you make a minus. So, this duplicate element has duplicated the entry in the stop the stack stop. So, you make a minus from there and after doing the minus whatever be the result you pop it out.



(Refer Slide Time: 34:10)

**FORTH - Sample Code**

**FORTH Code:**  
`:FLOOR5 (n--n') DUP 6 < IF DROP 5 ELSE 1 - THEN;`

**Equivalent C Code:**  

```
int floor5(int v){  
    return (v<6)?5:(v-1);  
}
```

- Defines a new **word** (a subroutine) called **FLOOR5**

Code Source: <https://en.bitcoin.it/wiki/>

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, the entire thing you can execute in the stack in this way. So, you have the stack there whatever be the input, say the 7 was the input you have put that input in the stack then there is a dup. So, this dup duplicate whatever there is the stack. So, this duplicate will duplicate the 7 in the stack then there is operand you push it in the stack, make this comparison whenever the comparison is there, the by doing the comparison the result is false.

So, you pop this out and put false there, then based on the if statement it comes to this else part because it is false. So, once it comes to false then whatever be there in the stack top you have a one. So, you push one there and then you have a operand minus then this minus you make a 7 minus 1, its it comes to be 6 and then 6 is returned from this execution.

So, it becomes equivalent to the corresponding C code. So, this is all about the forth programming language, in the next class will see that how Bitcoin script is actually extended on top of this forth programming language and it performs the input and the output operation, for the for the Bitcoin transactions. So, we will see those details in the next class. So, that is all about today's class.

Thank you.