merged_topic_0: checksum, data, items, layer, protocols, numbers, sum, send, message

- The checksum is used in the Internet by several protocols although not at the data link layer
- The receiver can add all the numbers received (including the checksum)
- The sender initializes the checksum to and adds all data items and the checksum (the checksum is considered as one data item and is shown in color)
- The sender now sends six data items to the receiver including the checksum
- Since the value of the checksum is this means that the data is not corrupted
- The receiver drops the checksum and keeps the other data items
- If the checksum is not zero the entire packet is dropped
- The checksum is sent with the data
- If the value of checksum is the message is accepted otherwise it is rejected
- Note that if there is any corruption the checksum recalculated by the receiver is not all s
- ++++
- In the Internet the checksum technique is mostly used at the network and transport layer rather than the datalink layer
- The generator then creates anextra extram mbitbit unit called the checksum which is sent with the message
- At the destination the checker creates a new checksum from the combination of the message and sent checksum
- If the checksup is all s the message is accepted other wise the message is discarded
- Example Suppose our data is a list of five bit numbers that we want to send to a destination
- In addition to sending these numbers we send the sum of the numbers
- For example if the set of numbers is ( ) we send ( ) where is the sum of the original numbers
- The receiver adds the five numbers and compares the result with the sum
- If the two are the same the receiver assumes no error accepts the five numbers and discards the sum
- For Simplicity Example We can make the job of the receiver easier if we send the negative (complement) of the sum called the checksum

- In this case we send ( )
- The receiver can add all the numbers received (including the checksum)

merged_topic_1: hamming, distance, minimum, dmin, scheme, coding, block

- Table A code for error correction (Example) Note The Hamming distance between two words is the number of differences between corresponding bits
- Example Let us find the Hamming distance between two pairs of words
- The Hamming distance d ( ) is because
- The Hamming distance d ( ) is because Note The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words
- Example Find the minimum Hamming distance of the coding scheme in Table
- Solution We first find all Hamming distances
- Solution We first find all the Hamming distances
- Note To guarantee the detection of up to s errors in all cases the minimum Hamming distance in a block code must be dmin s
- Example The minimum Hamming distance for our first code scheme (Table) is
- Example A code scheme has a Hamming distance dmin
- Error correction codes need to have an odd minimum distance (
- So the minimum Hamming distance is dmin
- Note All Hamming codes discussed in this book have dmin
- ++++
- Coding schemes are divided into two broad categories Block coding Convolution coding Dataword Codeword Block Coding Message Messageis isdivided dividedinto intoblocks blockseach eachof ofk kbits bitscalled calleddatawords datawords We Weadd addr rredundant redundantbits bitsto toeach eachblock blockto tomake makethe thelength lengthn n k k r r The Theresulting resultingn nbitbitblocks blocksare arecalled calledcodewords codewords Datawords and codewords in block coding Block Coding With k bits we can create a combination of k datawords
- Error Detection in Block coding Two conditions satisfy the error detection
- Hamming Distance It is one of the central concept in coding for error control
- Why do you think Hamming distance is important for error detection
- The reason is that the Hamming distance between the received codeword and the

sent codeword is the number of bits that are corrupted during transmission
- dmin used to define the minimum Hamming distance in a coding scheme
- The dmin in this case is
- Table A code for error detection (Example) BLOCK CODING BLOCK CODING Minimum Hamming Distance Minimum Hamming Distance Find the minimum Hamming distance of the coding scheme in Table
- Table A code for error correction (Example) BLOCK CODING Three Parameters For any coding scheme we need three parameters
- Code word size n Dataword size k Minimum Hamming distance dmin A coding scheme C is written is C (n k) with a separate expression for dmin For example C ( ) with dmin and C ( ) with dmin
- Relationship in between Hamming distance and errors occurring Hamming distance between the sent and received codewords is the number of bits affected by the error
- For example send codeword received codeword bits are in error and the Hamming distance is d ( ) Minimum Hamming Distance for Error Detection If s errors occur during transmission the Hamming distance between the sent codeword and received codeword is s If it is necessary to detect upto s errors the minimum hamming distance between the valid codes must be s so that the received codeword does not match a valid codeword
- To guarantee the detection of up to s errors in all cases the minimum Hamming distance in a block code must be dmins The minimum Hamming distance for our first code scheme from the table is dmin s or s S So this code guarantees detection of only a single error
- Dataword Codeword BLOCK CODING Minimum Distance for Error Correction To guarantee correction of up to tt errors in all cases the minimum Hamming distance in a block code must be dmin t
- BLOCK CODING Minimum Distance for Error Minimum Distance for Error Correction Correction Example BLOCK CODING A code scheme has a Hamming distance dmin
- Error correction codes need to have anodd oddminimum minimumdistance distance (
- Linear Block Codes This is the widespread used coding scheme
- A linear block code is a code in which the exclusive OR of two valid codeword creates another valid codeword
- Hamming codes Hamming codes were originally designed with dmin which means that can detect up to two errors or correct one single error
- Syndrome Error None q q b q b b b Cyclic Codes Cyclic codes are special linear block

codes with one extra property

merged_topic_2: hamming, distance, minimum, dmin, scheme, words, pairs, distances

- Table A code for error correction (Example) Note The Hamming distance between two words is the number of differences between corresponding bits
- Example Let us find the Hamming distance between two pairs of words
- The Hamming distance d ( ) is because
- The Hamming distance d ( ) is because Note The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words
- Example Find the minimum Hamming distance of the coding scheme in Table
- Solution We first find all Hamming distances
- Solution We first find all the Hamming distances
- Note To guarantee the detection of up to s errors in all cases the minimum Hamming distance in a block code must be dmin s
- Example The minimum Hamming distance for our first code scheme (Table) is
- Example A code scheme has a Hamming distance dmin
- Error correction codes need to have an odd minimum distance (
- So the minimum Hamming distance is dmin
- Note All Hamming codes discussed in this book have dmin
- ++++
- The Hamming distance between two words (of the same size) as the number of difference between the corresponding bits
- Let us find the Hamming distance between two pairs of words
- The Hamming distance d ( ) is because
- The Hamming distance d ( ) is because Hamming Distance Let us find the Hamming distance between two pairs of words
- The Hamming distance d ( ) is because BLOCK CODING Minimum Hamming Distance Minimum Hamming Distance The minimum Hamming distance is the smallest Hamming distance all possible pairs in a set of words
- Solution We first find all Hamming distances
- Solution We first find all the Hamming distances

merged_topic_3: received, codeword, transmission, corrupted, accepts,

receiver, dataword, syndrome

- The codeword is corrupted during transmission and is received
- The receiver accepts the received codeword and the errors are undetected
- No error occurs the received codeword is
- The received codeword is
- The codeword is received
- ++++
- Hence out of codewords are used for message transfer and rest are unused
- If the receiver receives an invalid codeword this indicates that the data was corrupted during transmission
- The receiver has a list of valid codewords
- Let k and n Dataword Codeword Let the sender encodes dataword as and send it receiver
- Receiver receives it is valid codeword
- Dataword is extracted by receiver
- Receiver incorrectly extracts dataword
- Dataword Codeword An errordetecting code can detect only the types of errors for which it is designed other types of errors may remain undetected
- If the syndrome is there is no error in the received codeword codeword is accepted as the dataword if the syndrome is the data portion of the received codeword is discarded
- The codeword created from this dataword is which is sent to the receiver
- The dataword is created at the receiver
- Note that here the dataword is wrongly created due to the syndrome valueThe Thesimply simplyparity paritycheckcheckdecoder decodercannot cannotdetect detectan aneven evennumbers numbersof oferrors errors
- Dataword becomes codeword is received syndrome no error
- The dataword is sent as codeword
- Relationship among the sent codeword error received codeword and the generator
- The receiver divides the received codeword by g (x) to get the syndrome

merged_topic_4: generator, note, simple, errors, isolated, paritycheck, odd, number, parity

- Note A simple paritycheck code is a singlebit errordetecting code in which n k with dmin
- This shows that the simple parity check guaranteed to detect one single error can also find any odd number of errors
- Note A simple paritycheck code can detect an odd number of errors
- Note In a cyclic code those e (x) errors that are divisible by g (x) are not caught
- Note If the generator has more than one term and the coefficient of x is all single errors can be caught
- Example Find the status of the following generators related to two isolated singlebit errors
- This is a very poor choice for a generator
- A codeword with two isolated errors up to bits apart can be detected by this generator
- Note A generator that contains a factor of x can detect all oddnumbered errors
- Note A good polynomial generator needs to have the following characteristics
- ++++
- Simple parityCheck Code In Simple paritycheck code a kbit dataword is changed to an nbit codeword where nk The extra bit called the parity bit The extra bit called the parity bit is selected to make the total number of s in the codeword even
- Although some implementations specify an odd number of s we discuss the even case
- A simple paritycheck code is a singlebit errordetecting code in which n k with dmin
- At Generator ro aaaa (modulo) Where if number of s is even the result is zero if number of s is odd the result is one Minimum Distance for Linear Block Codes Simple paritycheck code C ( ) Minimum Distance for Linear Block Codes At Generator ro aaaa (modulo) Where if number of s is even the result is zero if number of s is odd the result is one The checker at the receiver does the same thing as the generator in the sender with one exception The addition is done over all bits
- Simple paritycheck (Cont) Assume the sender sends the dataword
- Simple paritycheck (Cont)
- This shows that the simple parity check guaranteed to detect one single error can also find any odd number of errors
- The total number s in each bit combination (dataword bits parity bit) must be even

merged_topic_5: codeword, received, receiver, table, dataword, syndrome

- First the receiver finds that the received codeword is not in the table

- The receiver assuming that there is only bit corrupted uses the following strategy to guess the correct dataword
- Comparing the received codeword with the first codeword in the table (versus) the receiver decides that the first codeword is not the one that was sent because there are two different bits
- The original codeword must be the second one in the table because this is the only one that differs from the received codeword by bit
- ++++
- Hence out of codewords are used for message transfer and rest are unused
- If the receiver receives an invalid codeword this indicates that the data was corrupted during transmission
- The receiver has a list of valid codewords
- Let k and n Dataword Codeword Let the sender encodes dataword as and send it receiver
- Receiver receives it is valid codeword
- Dataword is extracted by receiver
- Receiver incorrectly extracts dataword
- Dataword Codeword An errordetecting code can detect only the types of errors for which it is designed other types of errors may remain undetected
- If the syndrome is there is no error in the received codeword codeword is accepted as the dataword if the syndrome is the data portion of the received codeword is discarded
- The codeword created from this dataword is which is sent to the receiver
- The dataword is created at the receiver
- Note that here the dataword is wrongly created due to the syndrome valueThe Thesimply simplyparity paritycheckcheckdecoder decodercannot cannotdetect detectan aneven evennumbers numbersof oferrors errors
- Dataword becomes codeword is received syndrome no error
- The dataword is sent as codeword
- Relationship among the sent codeword error received codeword and the generator
- The receiver divides the received codeword by g (x) to get the syndrome

merged_topic_6: complement, arithmetic, ones, number, using, represent

- Note In moduloN arithmetic we use only the integers in the range to N inclusive
- Example How can we represent the number in ones complement arithmetic using only

four bits
- Solution The number in binary is (it needs five bits)
- We can wrap the leftmost bit and add it to the four rightmost bits
- Solution In ones complement arithmetic the negative or complement of a number is found by inverting all bits
- Positive is negative is
- If we consider only unsigned numbers this is
- In other words the complement of is
- Another way to find the complement of a number in ones complement arithmetic is to subtract the number from n (in this case)
- Example Let us redo Exercise using ones complement arithmetic
- However can not be expressed in bits
- The extra two bits are wrapped and added with the sum to create the wrapped sum value
- The sum is wrapped and becomes
- All words are added using ones complement addition
- ++++
- Ones Complement The previous example has one major drawback
- We have solution to use ones complement arithmetic
- In this ones complement arithmetic we can represent unsigned numbers between and mm using only m bits
- In ones complement arithmetic we have two s one positive and one negative which are complements of each other
- How can we represent the number in ones complement arithmetic using only four bits
- We have ( ) ( ) ( ) ( ) How can we represent the number in ones complement arithmetic using only four bits
- Solution In ones complement arithmetic the negative or complement of a number is found by inverting all bits
- Positive is Ones complement of is If we consider only unsigned number this is
- In other words the complement of is

merged_topic_7: valid, codeword, match, sent, errors, receiver, dataword, received, syndrome

- For example if the third codeword ( ) is sent and one error occurs the received codeword does not match any valid codeword
- If two errors occur however the received codeword may match a valid codeword and the errors are not detected
- Again we see that when any of the valid codewords is sent two errors create a codeword which is not in the table of valid codewords
- However some combinations of three errors change a valid codeword to another valid codeword
- Otherwise there is an error somewhere and the data are not accepted
- ++++
- Hence out of codewords are used for message transfer and rest are unused
- If the receiver receives an invalid codeword this indicates that the data was corrupted during transmission
- The receiver has a list of valid codewords
- Let k and n Dataword Codeword Let the sender encodes dataword as and send it receiver
- Receiver receives it is valid codeword
- Dataword is extracted by receiver
- Receiver incorrectly extracts dataword
- Dataword Codeword An errordetecting code can detect only the types of errors for which it is designed other types of errors may remain undetected
- If the syndrome is there is no error in the received codeword codeword is accepted as the dataword if the syndrome is the data portion of the received codeword is discarded
- The codeword created from this dataword is which is sent to the receiver
- The dataword is created at the receiver
- Note that here the dataword is wrongly created due to the syndrome valueThe Thesimply simplyparity paritycheckcheckdecoder decodercannot cannotdetect detectan aneven evennumbers numbersof oferrors errors
- Dataword becomes codeword is received syndrome no error
- The dataword is sent as codeword
- Relationship among the sent codeword error received codeword and the generator
- The receiver divides the received codeword by g (x) to get the syndrome

merged_topic_8: divisible, divide, useless, greater, polynomial, remainder, xoois, xig

- No xi can be divisible by x
- If i is equal to or greater than xi is divisible by g (x)
- c All values of i make xi divisible by g (x)
- This g (x) is useless
- d This polynomial can not divide xt if t is less than
- It should not divide xt for t between and n
- ++++
- If the term does not have remainder (syndrome) either e (x) or e (x) is divisible by g (x)
- then xig (x) will have a remainder
- If g (x) have at least two terms and the coefficient ofx xoois is then e (x) can not be divided by g (x) ie there will be some remainder
- No xi can be divisible by x
- If i is equal to or greater than xi is divisible by g (x)
- c All values of i make xi divisible by g (x)

merged_topic_9: error, discuss, detection, correction, section, data, applications, require, chances, corrupt

- Some applications require that errors be detected and corrected
- INTRODUCTION Let us first discuss some issues related directly or indirectly to error detection and correction
- Let us first discuss some issues related directly or indirectly to error detection and correction
- Topics discussed in this section Types of Errors Redundancy Detection Versus Correction Forward Error Correction Versus Retransmission Coding Modular Arithmetic Topics discussed in this section Note In a singlebit error only bit in the data unit has changed
- Table A code for error detection (Example) Note An errordetecting code can detect only the types of errors for which it is designed other types of errors may remain undetected
- What is the error detection and correction capability of this scheme
- In other words if this code is used for error correction part of its capability is wasted
- For each case what is the error that can not be caught

- Table Standard polynomials CHECKSUM The last error detection method we discuss here is called the checksum
- However we briefly discuss it here to complete our discussion on error checking The last error detection method we discuss here is called the checksum
- However we briefly discuss it here to complete our discussion on error checking Topics discussed in this section Idea Ones Complement Internet Checksum Topics discussed in this section Example Suppose our data is a list of five bit numbers that we want to send to a destination
- If the result is it assumes no error otherwise there is an error
- ++++
- Data Link Layer Data can be corrupted during transmission
- during transmissionSome applications require that Some applications require that errors be detected and corrected
- Data can be corrupted Error Detection and Correction Network must be able to transfer data from one devices to another with acceptable accuracy
- The chances of data being corrupt can not be ignored
- The number of errors and the size of the message are important factors
- What is the error detection and correction capability of this scheme

merged_topic_10: generator, note, simple, errors, isolated, divisible, caught, useless, criteria

- Note A simple paritycheck code is a singlebit errordetecting code in which n k with dmin
- This shows that the simple parity check guaranteed to detect one single error can also find any odd number of errors
- Note A simple paritycheck code can detect an odd number of errors
- Note In a cyclic code those e (x) errors that are divisible by g (x) are not caught
- Note If the generator has more than one term and the coefficient of x is all single errors can be caught
- Example Find the status of the following generators related to two isolated singlebit errors
- This is a very poor choice for a generator
- A codeword with two isolated errors up to bits apart can be detected by this generator

- Note A generator that contains a factor of x can detect all oddnumbered errors
- Note A good polynomial generator needs to have the following characteristics
- ++++
- To find the criteria that must be imposed on the generator g (x) to detect the type of error that need to be detected
- Those errors that are not divisible by g (x) are not In a cyclic code those e (x) errors that are divisible by g (x) are not caught
- Note If the generator has more than one term and the coefficient of x is all single errors can be caught
- This g (x) is useless

file1_topic_11: blocks, redundant, add, make, nbit

- We add r redundant bits to each block to make the length n k r The resulting nbit blocks are called codewords
- In block coding we divide our message into blocks each of k bits called datawords
- We add redundant bits to the bit dataword to make bit codewords

file1_topic_12: set, acceptable, satisfies, condition, relationship

- In this coding scheme k and n
- The relationship between m and n in these codes is n m
- Calculate values of k and n that satisfy this requirement
- Solution We need to make k n m greater than or equal to or m m
- If we set m the result is n and k or which is not acceptable
- If we set m then n and k which satisfies the condition

file1_topic_13: dataword, flipping, changing, derive, wrong

- As we saw we have k datawords and n codewords
- Later we will see how to derive a codeword from a dataword
- Assume the dataword is
- The dataword becomes the codeword
- After flipping b (changing the to) the final dataword is

- After flipping b we get the wrong dataword

file1_topic_14: column, digit, shows, datawords, process

- Table shows the list of datawords and codewords
- Table shows the datawords and codewords
- We keep the rightmost digit ( ) and insert the leftmost digit ( ) as the carry in the second column
- The process is repeated for each column

file1_topic_15: continued, example, follows, examine, procedure

- Example (continued)
- We examine five cases
- It should have at least two terms
- Example (continued) The receiver follows the same procedure as the sender
- We leave this an exercise

file1_topic_16: singlebit, caught, error, corrupted, changes

- Two corrupted bits have made the error undetectable
- One singlebit error changes a
- One singlebit error changes r
- Three bitsa a and aare changed by errors
- No bit is corrupted
- Some bits are corrupted but the decoder failed to detect them
- Example Which of the following g (x) values guarantees that a singlebit error is caught
- Any singlebit error can be caught
- All singlebit errors in positions to are caught
- No singlebit error can be caught

file1_topic_17: dmin, case, code, second

- The dmin in this case is
- Example Our second block code scheme (Table) has dmin
- So in this code we have dmin

file1_topic_18: guarantees, correct, code, errors, detection

- This code guarantees detection of only a single error
- This code can detect up to two errors
- Solution This code guarantees the detection of up to three errors (s) but it can correct up to one error
- This shows that our code can not correct two errors
- Any two errors next to each other can not be detected

file1_topic_19: linear, block, codes, valid, code

- LINEAR BLOCK CODES Almost all block codes used today belong to a subset called linear block codes
- A linear block code is a code in which the exclusive OR (addition modulo) of two valid codewords creates another valid codeword
- Almost all block codes used today belong to a subset called linear block codes
- Topics discussed in this section Minimum Distance for Linear Block Codes Some Linear Block Codes Topics discussed in this section Note In a linear block code the exclusive OR (XOR) of any two valid codewords creates another valid codeword
- Example Let us see if the two codes we defined in Table and Table belong to the class of linear block codes
- The scheme in Table is a linear block code because the result of XORing any codeword with any other codeword is a valid codeword
- For example the XORing of the second and third codewords creates the fourth one
- The scheme in Table is also a linear block code
- In a cyclic code if a codeword is cyclically shifted (rotated) the result is another codeword
- Cyclic codes are special linear block codes with one extra property

file1_topic_20: created, dataword, wrongly, value, syndrome

- The dataword is created
- No dataword is created
- Note that here the dataword is wrongly created due to the syndrome value
- The dataword is not created

file1_topic_21: burst, length, slip, errors

- Note All burst errors with L r will be detected
- All burst errors with L r will be detected with probability ( ) r
- All burst errors with L r will be detected with probability ( ) r Example Find the suitability of the following generators in relation to burst errors of different lengths
- This generator can detect all burst errors with a length less than or equal to bits out of burst errors with length will slip by out of burst errors of length or more will slip by
- This generator can detect all burst errors with a length less than or equal to bits out of million burst errors with length will slip by out of million burst errors of length or more will slip by
- c This generator can detect all burst errors with a length less than or equal to bits out of billion burst errors with length will slip by out of billion burst errors of length or more will slip by

file1_topic_22: numbers, sum, send, receiver, easier

- In addition to sending these numbers we send the sum of the numbers
- For example if the set of numbers is ( ) we send ( ) where is the sum of the original numbers
- The receiver adds the five numbers and compares the result with the sum
- If the two are the same the receiver assumes no error accepts the five numbers and discards the sum
- Example We can make the job of the receiver easier if we send the negative (complement) of the sum called the checksum

file1_topic_23: complemented, checksum, sum, value, new

- The sum is then complemented resulting in the checksum value ( )

- The value of the checksum word is set to
- All words including the checksum are added using ones complement addition
- The sum is complemented and becomes the checksum
- The sum is complemented and becomes the new checksum
- Example Let us calculate the checksum for a text of characters (Forouzan)

file1_topic_24: divided, byte, words, bit, message

- The message is divided into bit words
- The message (including checksum) is divided into bit words
- The text needs to be divided into byte (bit) words
- We use ASCII (see Appendix A) to change each byte to a digit hexadecimal number

file2_topic_25: redundant, bits, data, second, correct

- If a data is sent at mbps then each one bit last only for second ((u) micro second)
- Instead of repeating the entire data stream a shorter group of bits may be appended to the end of each unit
- This technique is called Redundancy These extra bits are discarded as soon as the accuracy of the transmission has been determined
- To detect or correct errors we need to send extra (redundant) bits with data
- Forward Error Correction Process in which the receiver tries to guess the message by Redundant bit
- (usually not all errors are detected) Coding The sender adds redundant bits The receiver checks the relationships between the two sets of bits to detect or correct the errors
- At the source the message is first divided into mbits units

file2_topic_26: bits, zero, codewords, augment, means

- Since n bits we can create a combination of n codewords
- Since n k The possibility codeword is greater than possible data words
- This means we have n k codewords extra
- Eg if k and n we have dataword and codewords

- This means that we have n k codewords that are not used
- A codeword consists of n bits of which k are data bits and r are check bits
- Dataword (k) if codeword (n) bits
- Augment dataword by (nk) zero bits ie zero bits
- Augment DW by (nk) zero bits ie zero bits
- So general DP n bit if n is number of bits
- All our data can be written as a bitbitword word (they (theyare areless lessthan than)) except the checksum

file2_topic_27: invalid, valid, codewords, illegal, corrupted

- We call these codewords invalid or illegal
- The original codewords has changed to an invalid one
- Code word is corrupted
- This is not a valid codeword and it is discarded
- Codeword is corrupted but this is valid codeword

file2_topic_28: sbbbq, mod, raaa, modulo, syndrome

- The syndrome is when the number of s in the received codeword is even otherwise it is The syndrome is passed to the decision logic analyzer
- raaa (modulo) raaa (modulo) raaa (modulo) raaa (modulo) raaa (modulo) raaa (modulo) At Checker Sbbbq (mod) Sbbbq (mod) Sbbbq (mod) The bit syndrome creates different bit patterns (to) that can represent different conditions
- Syndrome Error None q q b q b b b Fig logical decision made by the correction logic Analyzer In Syndrome is not concerned with generator since there is error or an error in the parity bit
- In Syndrome one of the bits must be flipped (from to or from to) to find correct dataword
- Sbbbq (mod) Sbbbq (mod) Sbbbq (mod Dataword Codeword Cases
- After flipping b (changing to) the final dataword is obtained
- Syndrome b b b Dataword Codeword
- Sbbbq (mod) Sbbbq (mod) Sbbbq (mod) After flipping b we get (wrong dataword)

file2_topic_29: cyclic, generator, divisor, codeword, encoder

- In a cyclic code if a codeword is cyclically shifted (rotated) the result is another codeword
- For example if is a codeword and we cyclically leftshift then is also a codeword
- In this case if we call the bits in the first word a to a and the bits in the second word b to b we can shift the bits by using the following ba ba ba ba ba ba ba A category of cyclic code is Cyclic Redundancy Check (CRC) Used in LANs WANs Cyclic Redundancy Check The generator uses nk bit divisor which is predefined and agreed The remainder (bit) is appended to the dataword to make the final codeword At Encoder In encoder the dataword has k bits (here) The codeword has n bits (here) The size of the dataword is augmented by adding n k (here) The generator uses adivisor divisorof ofsize size nk (here) The quotient of the division is discarded
- The remainder (rrr) is appended to the dataword to create the codeword
- ie Let the divisor is (Predefined and agreed) (nk) digits The remainder (rrr) is appended to the dataword to create the codeword At decoder Decoder does the same division as encoder
- Polynomials A better way to understand cyclic codes and how they can be analyzed is to represent them as polynomials
- The divisor in a cyclic code is normally called the generator polynomial or simply the generator
- Cyclic Code Analysis The divisor is normally called the generator polynomial or generator Analysis Let f (x) is a polynomial with binary coefficient d (x) Dataword c (x) Codeword g (x) Generator s (x) Syndrome e (x) Error Note In a cyclic code If s (x) one or more bits is corrupted
- We can write this The first term at the right hand side of the equality has a remainder of zero (according to the definition of codeword)

file2_topic_30: right, shifting, bits, leftmost, rightmost

- The bit pattern in this case has bits
- (xxxx) (xx) (let the student do rest of the problem) xxxxx Shifting A binary pattern is often shifted a number of bits to the right or left
- Shifting to the left means adding s extra bit at right side
- Shifting to the Right means deleting some right most bits
- If the number has more thanm m bits the extra leftmost bits need to be added to them m rightmost bits (wrapping)

- Solution The number is binary is (it needs six bits)
- We can wrap the leftmost bit and add it to the four rightmost bits

file2_topic_31: terms, subtraction, multiplying, deleted, powers

- Addition Subtraction Addition or subtraction is done by combining terms and deleting pairs of identical terms
- (xxx) (xxx) x x X and x are deleted Multiplying or dividing terms Just add the powers of multiplication
- X x x Subtract the powers for division
- Xx x Multiplying two polynomials It is conceptually same as we did it for the encoderdecoder pairs equal terms are deleted