CPE 633
Chapter 3 – Information Redundancy

Dr. Rhonda Kay Gaede

UAH

Electrical and Computer Engineering

Introduction

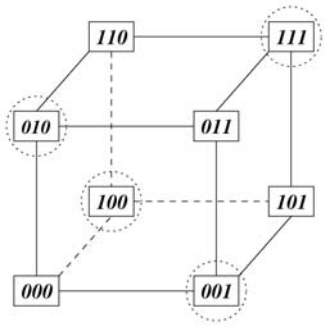
- The most common form of information redundancy is _____, which adds _____ _____ to the data, allowing us to verify the correctness of data and, in some cases, _____ it.
- Information redundancy can be practiced on larger _____ than an individual word, best known example is _____.
- At an even higher level, data can be _____ among processors.
- We will consider _____ fault tolerance for applications with large amounts of _____.

3.1 Coding – Basics

- A _____ data word is encoded into a _____ code word, _____.
- Not all 2^c binary combinations are valid _____.
- A code is the set of all _____ codewords.
- Performance parameters include
  - _____
  - _____
- Overhead
  - _____
  - _____

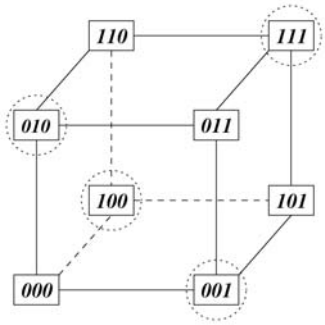
3.1 Coding – Hamming Distance

- The Hamming distance between two codewords is the number of _____ in which the two words differ.
- A Hamming distance of _____ between two codewords guarantees that a _____ error in any of the two words will not change it into the other.



3.1 Coding – Code Distance

- The code distance is the _____ Hamming distance between any two valid codewords.
- To detect up to ____ errors, the code distance must be at least _____.
- To correct up to ____ errors, the code distance must be at least _____.



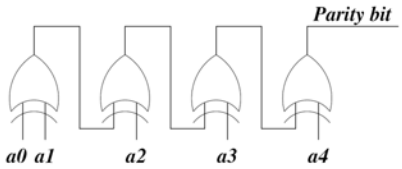
3.1 Coding – Separability

- A _____ code has separate fields for the _____ and _____ bits.
- Separable Codes
  - Decoding simply consists of _____ the data bits and _____ the check bits.
  - The _____ must still be processed separately to determine the correctness of the data.
- Nonseparable Codes
  - _____ the data requires some processing
  - The check bits must still be _____ to determine the correctness of the data.

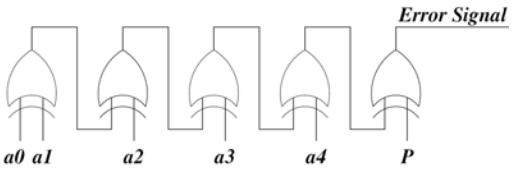
3.1.1 Parity Codes – Properties

- The simplest codes of all the codes are the _____ codes.
- Most basic form – ___ data bits plus __ check bit
- In an even(odd) parity code, this extra bit is set so that the total number of 1s in the whole (c=d+1)-bit word is even(odd).
- The _____ fraction is (c-d)/d = 1/d
- A parity code has a Hamming distance of __ and will detect all _____ errors and provides _____ _____.

3.1.1 Parity Codes – Even Parity Encoding and Decoding



(a) Encoder



(b) Decoder

3.1.1 Parity Codes – Variations of the Basic Parity Code

- _____
  - Have one bit per _____ rather than one bit per _____
  - Overhead increases from ___ to _____
  - Detect up to ___ errors
- _____ parity code
  - If d = a_{64}, a_{63}, a_{62}, …, a_0, use eight parity bits.
  - C_1 is parity for a_{63}, a_{55}, a_{47}, a_{39}, a_{31}, a_{23}, a_{15}, a_7
- _____ Parity
  - Can provide _____
  - Even parity rows
  - Even parity columns
  - Pair of parity bits identifies faulty bit _____

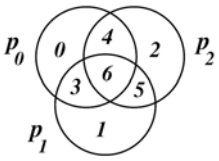
0	0	0	1	1	1	1
1	0	1	0	1	1	0
1	1	0	0	0	0	0
0	0	0	1	1	1	1
1	1	1	1	1	1	0
1	0	0	1	0	0	0

3.1.1 Parity Codes – More on Overlapping Parity Codes

- Each bit is _____ by _____ parity bit.
- Our goal is to identify every _____ bit.
- With d data bits, how many _____ are needed and _____ should they cover?
- Let r be the number of _____ bits, codeword size is _____. There are d+r _____ where in state i, the ith bit of the codeword is _____. There is also the no error state, total number of states is _____.
- For r parity checks, there are 2^r different check _____.
- The minimum number of parity bits is the _____ that satisfies $2^r \geq d+r+1$

3.1.1 Parity Codes – Selecting Parity Bit Coverage of Data Bits

- Example: d=4 data bits, $a_3a_2a_1a_0$
- r must be at least 3, $p_2p_1p_0$
- d+r+1 = 4+3+1 = 8 possible states
- codeword $a_3a_2a_1a_0\ p_2p_1p_0$



State	Erroneous parity check(s)	Syndrome
No errors	None	000
Bit 0 (p_0) error	p_0	001
Bit 1 (p_1) error	p_1	010
Bit 2 (p_2) error	p_2	100
Bit 3 (a_0) error	p_0, p_1	011
Bit 4 (a_1) error	p_0, p_2	101
Bit 5 (a_2) error	p_1, p_2	110
Bit 6 (a_3) error	p_0, p_1, p_2	111

3.1.1 Parity Codes – Syndrome Definition

- Suppose that the codeword 1100001 experiences a _____ and becomes 1000001. The _____ parity bits $p_2p_1p_0$ for 1000001 are 111.
- They should be ____. The difference between what they are and what they should be (_____) is the _____, in this case, 110.
- From previous table, a syndrome of 110 indicates that ____ is in error and should be 1, not 0.
- This code is called a (7,4) Hamming _____ _____ (SEC) code.

3.1.1 Parity Codes – Syndrome Calculation

- The syndrome can be calculated directly from the _____ in one step using a matrix operation with the _____. (All matrix additions are _____).

$$\begin{matrix} a_3 & a_2 & a_1 & a_0 & p_2 & p_1 & p_0 \end{matrix}$$
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} s_2 & s_1 & s_0 \end{bmatrix}$$

Parity Check Matrix

$p_2 = a_3 \oplus a_2 \oplus a_1$

$p_1 = a_3 \oplus a_2 \oplus a_0$

$p_0 = a_3 \oplus a_1 \oplus a_0$

3.1.1 Parity Codes – Syndrome Calculation

- We can modify the _____ of states to the _____ so that the calculated syndrome provides the _____ of the bit in error.
- Indices are now 7 downto 1.
- This assignment leads to a new _____.

State	Erroneous Parity Checks	Syndrome
No errors	None	000
Bit 0 (p_0) error	p_0	001
Bit 1 (p_1) error	p_1	010
Bit 2 (a_0) error	p_1,p_0	011
Bit 3 (p_2) error	p_2	100
Bit 4 (a_1) error	p_2,p_0	101
Bit 5 (a_2) error	p_2,p_1	110
Bit 6 (a_3) error	p_2,p_1,p_0	111

3.1.1 Parity Codes – Parity Check Matrix Choices

- If $2^r > d+r+1$, we need to select _____ out of the 2^r combinations to serve as _____.
- For d=3, r=3 8 > 3+3+1, let's look at _____ parity check matrices, (a) uses the combination ____, (b) does not. (_____ ones are desirable)
- Matrix (a) requires two XOR gates to generate p_0 while matrix (b) requires only one. They both require one XOR gate each to generate p_2 and p_1.

$$\begin{matrix} a_2 & a_1 & a_0 & p_2 & p_1 & p_0 \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$
(a)

$$\begin{matrix} a_2 & a_1 & a_0 & p_2 & p_1 & p_0 \end{matrix}$$
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$
(b)

3.1.1 Parity Codes – Adding Double Error Detecting

- Going back to our (7,4) code. It is capable of correcting every _____ error but cannot _____ a _____ error.
- Consider 11000001 becoming 1010001 due to a double error (a_2 and a_1). The calculated syndrome would be _____ erroneously indicating an error in a_0.
  - •We can add another check bit which is the _____ _____ in the codeword.
  - •The resulting code is an ____ ____ and _____ _____ (DED) code.

$$\begin{matrix} a_3 & a_2 & a_1 & a_0 & p_3 & p_2 & p_1 & p_0 \end{matrix}$$
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} s_3 & s_2 & s_1 & s_0 \end{bmatrix}$$

3.1.1 Parity Codes – Double Error Detecting (Method 2)

- By restricting ourselves to the use of syndromes that include an _____ (for any single-bit error), a double error will result in a syndrome with an _____ number of 1s, indicating an error that cannot be corrected. One such matrix is shown below.
- Limiting ourselves to only odd syndromes implies that we use only ___ out of the ___ possible combinations.
- We need _____ for an SED Hamming code.

a_3	a_2	a_1	a_0	p_3	p_2	p_1	p_0
0	1	1	1	1	0	0	0
1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1

3.1.1 Parity Codes – Limitations of SEC Codes

- As d _____, the probability of having an error that is _____ by an SEC code _____.
- As d _____, the overhead r/d _____.
- f - probability of a bit error & assume bit errors occur independently of one another
- Probability of _____ in a field of d+r bits -

$$\Phi(d,r) = 1-(1-f)^{d+r} - (d+r)f(1-f)^{d+r-1}$$

$$\approx 0.5(d+r)(d+r-1)f^2 \quad (for\ f << 1)$$

3.1.1 Parity Codes – Limitations of SEC Codes

- To _____ this probability, we may partition the d data bits into _____ and encode each ____ separately using an appropriate (d+r,d) SEC Hamming code.
- The _____ is between the probability of having an uncorrectable error and the overhead.
- The probability that there is an _____ error in _____ of the D/d slices is

$$\Psi(D,d,r) = 1-[1-\Phi(d,r)]^{D/d}$$

$$\approx (D/d)\cdot\Phi(d,r) \quad (\textit{for } \Phi(d,r) << 1)$$

3.1.1 Parity Codes – Quantifying the Tradeoff(D=1024, f = 10^{-11})

d	r	Overhead r/d	Ψ(D, d, r)
2	3	1.5000	0.5120E-16
4	3	0.7500	0.5376E-16
8	4	0.5000	0.8448E-16
16	5	0.3125	0.1344E-15
32	6	0.1875	0.2250E-15
64	7	0.1094	0.3976E-15
128	8	0.0625	0.7344E-15
256	9	0.0352	0.1399E-14
512	10	0.0195	0.2720E-14
1024	11	0.0107	0.5351E-14

3.1.2 Checksum

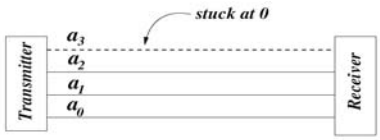
- A _____ is used to detect errors in transmission through _____.
- The basic idea is to _____ and transmit the _____ along with the _____.
- The receiver _____ a sum and compares with the _____ sum, if different, error.
- Single Precision - add modulo-2^d
- Double Precision – add modulo-2^{2d}
- Residue – add carry out of MSB to LSB
- Honeywell – concatenate two words and add modulo-2^{2d}

3.1.2 Checksum - Examples

(a) Single-precision	(b) Double-precision	(c) Residue	(d) Honeywell
0000	0000	0000	
0101	0101	0101	
1111	1111	1111	00000101
0010	0010	0010	11110010
0110	**00010110**	**0111**	**11110111**

All the checksum errors allow _____ but not _____, and the entire block of data must be _____ if an error is detected.

3.1.2 Checksum – Comparison when Line s-a-0



1000	0000
1011	0011
0000	0000
1100	0100
1111	**0111**
Transmitted	*Received*

(b) Single-Precision

10001011	00000011
00001100	00000100
10010111	**00010111**
Transmitted	*Received*

(c) Honeywell

3.1.3 M-of-N Codes – A Unidirectional Error-Detecting Code

- In an M-of-N code, every _____ codeword has exactly ___ bits that are 1, resulting in _____ codewords
- Any single-bit error will change the number of 1s to either _____ or _____
- Example 2 of 5 code
- Non-separable

Digit	Codeword
0	00011
1	00101
2	00110
3	01001
4	01010
5	01100
6	10001
7	10010
8	10100
9	11000

3.14 Berger Code

- A _____l error detecting code that is _____ and has a much lower _____ is the Berger code.
- Encoding - count the _____ in a word, then _____ the binary representation of the _____ and append to data bits
  - 11101 → 11101011
- Overhead – _____ - for d data bits, there can be at most d 1s
- If d = 2k-1 for an integer k, then the number of check bits, r = k and the resulting code is called a _____ Berger code.
- For unidirectional error detecting, the Berger code requires the _____ _____ of all known separable codes.

d	r	Overhead
8	4	0.5000
15	4	0.2667
16	5	0.3125
31	5	0.1613
32	6	0.1875
63	6	0.0952
64	7	0.1094
127	7	0.0551
128	8	0.0625
255	8	0.0314
256	9	0.0352

3.1.5 Cyclic Codes

- In cyclic codes, encoding of data consists of _____ (modulo-2) the data word by a constant number and the _____ is the resulting _____.
- Decoding is done by _____ by the same constant, a remainder of _____ indicates no error.
- These codes are called cyclic because, if you _____ a codeword, you also get another codeword.
- Cyclic codes are widely used in both _____ and _____.
- Only a small sampling is presented here.
- If ___ is the number of data bits, the ____ codeword is obtained by multiplying the _____ by a number that is _____ data bits long

3.1.5 Cyclic Codes – Generator Polynomials

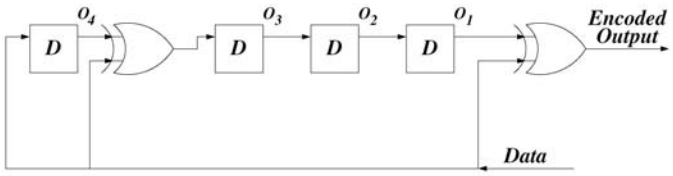
- In cyclic coding theory, the multiplier is represented as a _____ with the 1s and 0s treated as _____.
- For a multiplier of 11001, the generator polynomial $G(X) = 1 \bullet X^4 + 1 \bullet X^3 + 0 \bullet X^2 + 0 \bullet X^1 + 1 \bullet X^0 = X^4 + X^3 + 1$.
- A cyclic code using a _____ of degree n – k and a codeword of size n is called an _____ cyclic code.
- An (n, k) cyclic code can detect all _____ errors and also all runs of _____ bit errors, so long as these runs are shorter than ____ (burst errors)
- For a polynomial of degree n – k to serve as a _____ of an (n, k) cyclic code, it must be a _____ of X^n-1

3.1.5 Cyclic Codes – Generator Polynomials

- For N=15, $X^{15} – 1$ has five prime factors
  $X^{15} - 1 = (X + 1)(X^2 + X + 1)(X^4 + X + 1)$
  $(X^4 + X^3 + 1)(X^4 + X^3 + X^2 + X + 1)$
- Any ___ of these five factors and any _____ of two (or more) of these factors can serve as a _____ _____ for a cyclic code.
- For example, the product of $(X + 1)$ and $(X^2 + X + 1)$ is $X^3 + 1$ which generates a (15, 12) cyclic code.
- Cyclic codes are _____.
- Look at codeword generation for a ____ cyclic code – generator polynomial is _____.

3.1.5 Cyclic Codes – Hardware Implementation

- Multiplication can be implemented using _____ and _____.
- The generator polynomial is _____ by the connections used, the circuit here uses $X^4 + X^3 + 1$

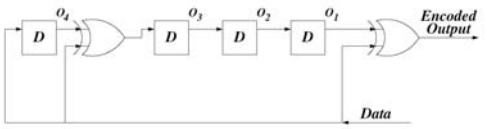


3.1.5 Cyclic Codes – Conceptual Encoding

- The _____ form of multiplication is shown here.
- In actuality, the data words are fed in _____, starting with the _____.
- The least significant bit of the _____ has only one _____.
- We accumulate _____ _____.
- This code is _____.

```
  10001100101
×     11001
-----------
  10001100101
  00000000000
  00000000000
  10001100101
  10001100101
-----------
1100010001101
```

3.1.5 Cyclic Codes – Encoding Example



shift clock	input data	O_4	i_3	O_3, O_2, O_1	encoded output	shift clock	input data	O_4	i_3	O_3, O_2, O_1	encoded output

3.1.5 Cyclic Codes – Conceptual Decoding

Error Free With Error

3.1.5 Cyclic Codes – Conceptual Decoding (Three Bit Errors)

Non-adjacent

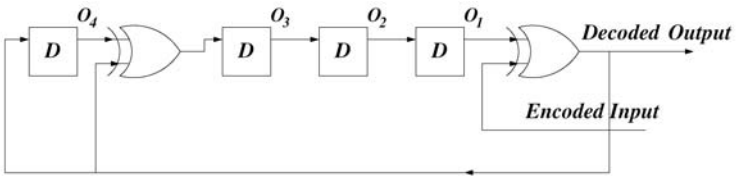
```
11000011101010 1 : 11001 = 10001101101
11001
 10111
 11001
  11100
  11001
   10110
   11001
    11111
    11001
     11001
     11001
     00000
```

Adjacent

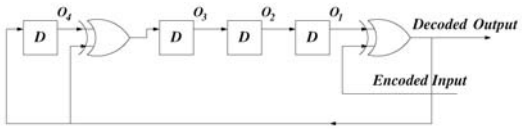
```
11000001101110 1 : 11001 = 10001110011
11001
 10011
 11001
  10100
  11001
   11011
   11001
    10110
    11001
     11111
     11001
     00110
```


3.1.5 Cyclic Codes – Hardware Implementation of Division

- Let the _____ be E(X), G(X) be the _____ _____, D(X) be the _____.
- For _____ D(X) = E(X)/G(X)
- $E(X) = D(X)G(X) = D(X)\{X^4 + X^3 + 1\}$
  $= D(X)\{X^4 + X^3\} + D(X)$
  $D(X) = E(X) - D(X)\{X^4 + X^3\}$
  $D(X) = E(X) + D(X)\{X^4 + X^3\}$



3.1.5 Cyclic Codes – Decoding Example



shift clock	encoded input	i_4	O_4	i_3	O_3, O_2, O_1	decoded output	shift clock	encoded input	i_4	O_4	i_3	O_3, O_2, O_1	decoded output

3.1.5 Cyclic Codes – Standard Generator Polynomials

- Many applications need to make sure that all _____ of length _____ or less will be detected.
- Cyclic codes of the type _____ are used
- The generating polynomial should be selected to allow a _____ _____ (use same circuit for different sizes of data blocks).
- Most commonly used :

  CRC-16 (16-bit Cyclic Redundancy Code)

  $$G(X) = X^{16} + X^{15} + X^2 + 1$$

  CRC-CCITT

  $$G(X) = X^{16} + X^{12} + X^5 + 1$$

3.1.5 Cyclic Codes – A Separable Version

- Advantage – data can be used before _____ complete.
- Data word $D(X) = d_{k-1}X^{k-1} + d_{k-2}X^{k-2} + \ldots + d_0$
- Append (n-k) zeroes to D(X) to obtain
  $D'(X) = d_{k-1}Xn^{-1} + d_{k-2}Xn^{-2} + \ldots + d_0X^{n-k}$
- Divide by G(X): D'(X) = Q(X)G(X) + R(X), degree of R(X) < n-k
- Codeword C(X) = D'(X) – R(X) has G(X) as a factor
- Divide C(X) by G(X) - if non-zero $\Rightarrow$ error
- In C(X) : first k bits data, last n-k check bits
- Example: (5,4) code with G(X)=X+1: data 0110, 1110

3.1.6 Arithmetic Codes

- Arithmetic codes allow us to detect errors which may occur during the _____ of an _____ _____ in the defined set.
- This error detection can be achieved by _____ the arithmetic unit but lower cost detection can be achieved through _____.
- An arithmetic code is one that is _____ under an arithmetic operation.
- Definition: An error code is _____ under an arithmetic operation $*$ if for any two operands X and Y and the corresponding encoded entities X' and Y' there is an operation $\otimes$ satisfying $X' \otimes Y' = (X * Y)'$

3.1.6 Arithmetic Codes – Error Detection

- Arithmetic codes should be able to detect all _____ errors
- A _____ error in an operand or an intermediate result may cause a _____ in the final result
- Example - when adding two binary numbers, if _____ of the adder is faulty, all the remaining _____ _____ digits may be erroneous

3.1.6 Arithmetic Codes – Nonseparable AN Codes

- Formed by _____ the operands by a _____.
- X' = AX, $*$ and $\otimes$ are identical for _____ and _____.
- All error magnitudes that are _____ will not be detected
- A should not be _____
- An _____ A is best - it will detect every _____ fault
- A=3 - _____ AN-code that enables _____ of all single bit errors
- Example - the number 0110
- Representation in the AN-code with A=3 is
  - 10010
- A fault in bit position 3 may give the erroneous result $11010_2 = 26_{10}$
- The error is easily detectable - 26 is not a multiple of 3

3.1.6 Arithmetic Codes – Separable Residue Codes

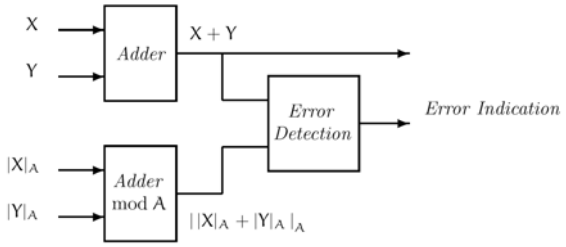
- Every _____ gets a separable check symbol, _____.
- For the residue code, _____ = X mod A = $|X|_A$, here A is called the _____.
- For the _____ residue code, C(X) = A – (X mod A)
- C(X) ⊗ C(Y) = C(X ∗ Y) for _____ and _____
- $|X + Y|_A = ||X|_A + |Y|_A|_A$, $|X \bullet Y|_A = ||X|_A \bullet |Y|_A|_A$
- Example, A = 3, X = 7, and Y = 5

3.1.6 Arithmetic Codes – Separable Residue Codes

- For division, the equation X – S = Q D is satisfied, where X is the _____, D the _____, Q the _____, and S the _____.
- The corresponding _____ is therefore $||X|_A - |S|_A|_A = ||Q|_A \bullet |D|_A|_A$
- Example, A = 3, X = 7, D = 5, the results are Q = 1 and S = 2

3.1.6 Arithmetic Codes – Comparison of AN and Residue Codes

- A residue code with _____ of ___ detects the same errors as the ____ code.
- The _____ for both involves calculating the result modulo-A and the _____ |log2A| is the same.
- Big difference, _____.



3.1.6 Arithmetic Codes – Low Cost Arithmetic Codes

- The AN and residue codes with _____ are the simplest examples of arithmetic codes that use a value of A of the form _____, for some _____.
- This choice _____ the calculation of the remainder when _____, thus these are called _____ arithmetic codes.
- The calculation of the remainder when dividing by 2a – 1 is simple, because the equation $|z_i r^i|_{r-1} = |z^i|_{r-1}$, $r = 2^a$ allows the use of modulo-$(2^a - 1)$ summation of the _____ that compose the number .

3.1.6 Arithmetic Codes – Low Cost Arithmetic Codes

- Example, X = 11110101011, divide by A = 7 = $2^3 - 1$. Partition X into (z_3, z_2, z_1, z_0) = (11, 110, 101, 011). Add modulo-7, a carry-out has a weight of 8, $|8|_7 = 1$, so add end around carry

3.1.6 Arithmetic Codes – Signed Operands

- If we wish to include _____ operands, we must require that the code be _____ with respect to R, where R is either 2^n (_____) or $2^n - 1$ (_____) and n is the number of bits in the _____.
- So, the _____ of each code word must also be _____.
- For AN, R – AX must be divisible by A, and A must be a factor of R. For A odd, R cannot be equal to 2^n, so R must be $2^n - 1$.

3.1.6 Arithmetic Codes – Ones Complement from Twos Complement

- $|2^n - X|_A = |2^n - 1 - X + 1|_A = |2^n - 1 - X|_A + |1|_A$
- 2s comp = 1s comp + 1, 1s comp = 2s comp – 1
- Carry out has weight of 2^n, for modulo $2^n - 1$, still need end around carry.
- Example, X = -10, Y = 13

3.1.6 Arithmetic Codes – Bi Residue Codes

- Using _____ creates interdependence between _____ and _____ units.
- A fault effect might be _____.
- It has been shown that a _____ is always detectable.
- Error _____ can be achieved by using ____ or more residue checks.
- Simplest case, ____ residue checks, _____.
- If n is the bits in the operand, select ___ and ___ such that n is the _____.
- If $A_1 = 2^a - 1$ and $A_2 = 2^b - 1$, any _____ can be corrected.

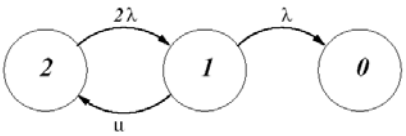
3.2.1 RAID Level 1

- Coding at a higher level.
- RAID – _____
- There is a level ___ which means _____.
- In RAID1, each original disk has been _____.
- If one disk fails, the other can continue to service requests.
- With both disks _____, reads can be divided among the disks, _____ execution.
- With both disks working, writes are _____ because both disks must _____ before the operation can complete.

3.2.1 RAID Level 1 – Reliability

- Assumptions
  - Disks fail independently, each at a constant rate λ
  - The time to repair is exponentially distributed with a mean of 1/μ
- Reliability at time t

$$R(t) = P_1(t) + P_2(t) = 1 - P_0(t)$$



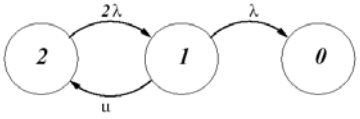
$$\frac{dP_2(t)}{dt} = -2\lambda P_2(t) + \mu P_1(t)$$

$$\frac{dP_1(t)}{dt} = -(\lambda + \mu)P_1(t) + 2\lambda P_2(t)$$

$$P_0(t) = 1 - P_1(t) - P_2(t)$$

$$P_2(0) = 1; \qquad P_0(0) = P_1(0) = 0$$

3.2.1 RAID Level 1 – Mean Time to Data Loss (MTTDL)



- Mean time before state 1 is entered – 1/2λ
- Mean time to stay in state 1 –1/μ
- Probability of going from state 1 to state 2 – μ/(λ + μ)
- Probability of going from state 1 to state 0 – λ/(λ + μ)
- Probability of n visits to state 1 before transition to state 0 is $q^{n-1}p$
- Mean time to enter state 0 :

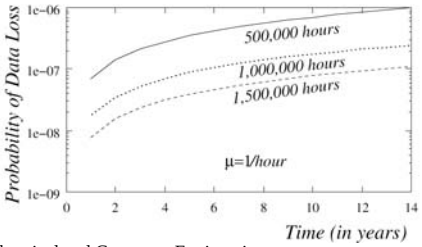
$$T_{2\to 0}(n) = n\left(\frac{1}{2\lambda} + \frac{1}{\lambda+\mu}\right) = n\frac{3\lambda+\mu}{2\lambda(\lambda+\mu)}$$

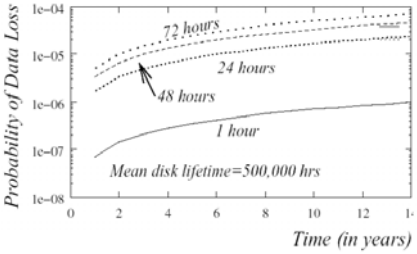
$$MTTDL = \sum_{n=1}^{\infty} q^{n-1} p T_{2\to 0}(n) = \sum_{n=1}^{\infty} n q^{n-1} p T_{2\to 0}(1) = \frac{T_{2\to 0}(1)}{p} = \frac{3\lambda+\mu}{2\lambda^2}$$

3.2.1 RAID Level 1 – Approximate Reliability

- For μ >> λ, MTTDL ≈ μ
- R(t) ≈ e $^{-t/MTTDL}$
- Availability is the same as that for a _____

Impact of mean disk lifetime





Impact of mean disk repair time

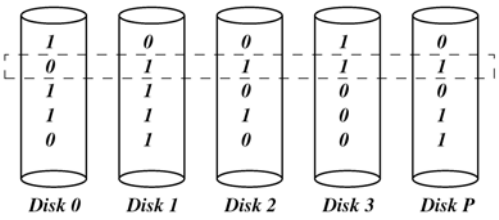
3.2.2 RAID Level 2

- A bank of _____ plus _____ disks
- d data disks and c check disks
- i-th bit of each disk - bit of a c+d-bit codeword
- From Hamming code theory - to permit the _____ per word –

$$2^c \geq c + d + 1$$

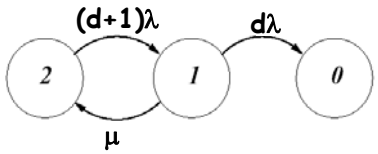
- We will not spend more time on RAID2 because other RAID designs impose much _____

3.2.3 RAID Level 3



- RAID3 consists of a bank of _____ together with ____ _____ disk.
- The data are _____ across the data disks, and the ith position of the parity disk contains the _____ associated with the bits in the ith position of each of the data disks.
- Each disk has _____ coding per _____.
- The _____ indicates the disk in error, the _____ can be recovered from the other d disks.
- As with parity, only _____ can be handled.
- If _____, we have data loss.

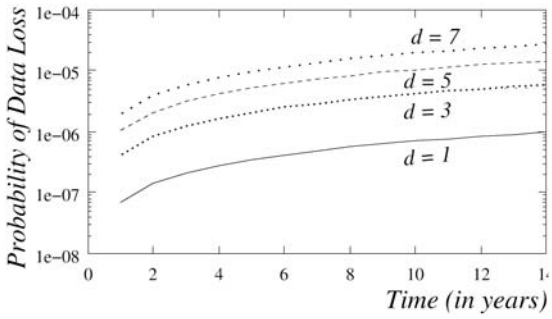
3.2.3 RAID Level 3 – Reliability Analysis



- The Markov chains are very similar to _____.
- In RAID1, __ disks per group, here _____ disks per group.
- In both cases, data loss occurs if _____ disks fail.

$$MTTDL = \frac{(2d+1)\lambda + \mu}{d(d+1)\lambda^2} \qquad R(t) = e^{-t/MTTDL}$$

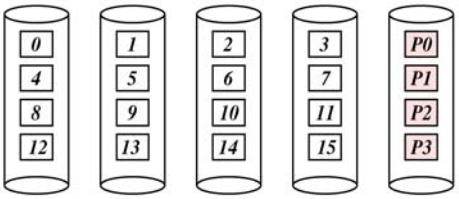
3.2.3 RAID Level 3 – Numerical Results



- d = 1 is the _____ case.
- As d _____, the reliability _____.

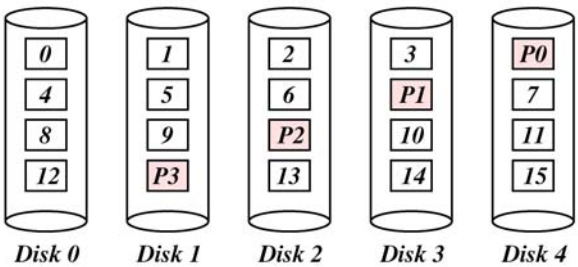
3.2.4 RAID Level 4

- The unit of interleaving changes from a _____ to a _____ of arbitrary size, called a _____.
- When individual bits were interleaved, _____ had to be accessed for a _____.
- A read may involve only _____.
- A write may involve only _____ and _____
- Same _____ as RAID3.



3.2.5 RAID Level 5

- For RAID4, the parity _____ can be the _____.
- _____ parity bits among the disks.
- The reliability model is the same as _____.

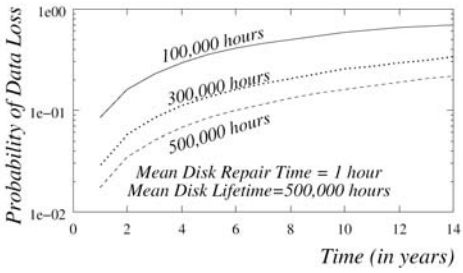


3.2.6 Modeling Correlated Failures

- Previous reliability and availability analysis assumed _____ of disks.
- The reality is that _____ and _____ are typically _____ among multiple disks.
- Disk _____ consist of disks housed in one enclosure that share _____, _____, _____, and _____, each of which can cause the entire string to fail.
- Let λ_{str} be the failure rate of the _____ elements of a string.

$$\lambda_{total} = \lambda_{str} + \lambda_{indep} \qquad R_{total}(t) = e^{-\lambda_{total} t}$$

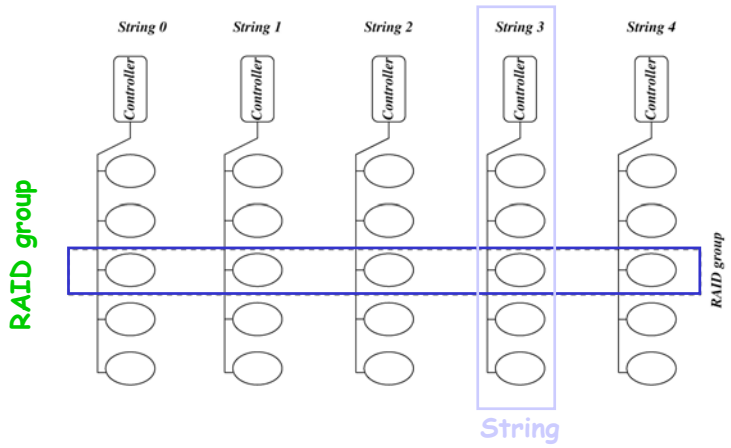
3.2.6 Modeling Correlated Failures



Mean String Lifetime

- To _____ this situation, use an _____ arrangement of strings and RAID groups.
- Thus, the failure of _____ affects only _____ in each RAID group.

3.2.6 Modeling Correlated Failures – Orthogonal Arrangement of Strings and Groups



3.2.6 Modeling Correlated Failures – Approximate MTTDL and Reliability

- Each RAID _____ has d + 1 disks, with __ groups, there are (d + 1)g disks _____.
- No longer assume repair times are _____ _____, let $f_{disk}(t)$ denote the _____ of the disk repair time.
- The approximate rate at which individual failures _____ in a given disk is given by $\lambda_{disk}\pi_{indiv}$, where λ_{disk} is the _____ of a single disk and π_{indiv} is the probability that a given _____ triggers data loss.
  - π_{indiv} is the probability that _____ in the affected RAID group while the previous failure has not _____.
- Disk failures can happen either due to an _____ _____ failure or a _____ failure, failures happen at the rate $d(\lambda_{disk} + \lambda_{str})$.

3.2.6 Modeling Correlated Failures – Failure Rate due to an Individual Disk Failure

- Let τ denote the random _____.

$$Prob\{Data\ \ Loss \mid repair\ \ takes\ \ \tau\} = 1 - e^{-d(\lambda_{disk} + \lambda_{str})\tau}$$

- _____ probability of data loss

$$\begin{aligned}\pi_{\text{indiv}} &= \int_0^\infty \text{Prob\{Data loss|the repair takes } \tau\} \cdot f_{\text{disk}}(\tau)\,d\tau \\ &= \int_0^\infty (1 - e^{-d(\lambda_{\text{disk}}+\lambda_{\text{str}})\tau}) f_{\text{disk}}(\tau)\,d\tau \\ &= \int_0^\infty f_{\text{disk}}(\tau)\,d\tau - \int_0^\infty e^{-d(\lambda_{\text{disk}}+\lambda_{\text{str}})\tau} f_{\text{disk}}(\tau)\,d\tau \\ &= 1 - F^*_{\text{disk}}(d[\lambda_{\text{disk}} + \lambda_{\text{str}}])\end{aligned}$$

- $F^*_{disk}()$ is the Laplace transform of $f_{disk}()$
- Approximate rate at which _____ is triggered by _____

$$\Lambda_{\text{indiv}} \approx (d+1)g\lambda_{\text{disk}}\{1 - F^*_{\text{disk}}(d[\lambda_{\text{disk}} + \lambda_{\text{str}}])\}$$

3.2.6 Modeling Correlated Failures – Failure Rate due to a String Failure

- The total rate at which _____ is $(d + 1)\lambda_{str}$
- On _____, repair string, then repair affected disks.
- Two Cases
  - _____ – failure can happen if _____ _____ occurs anywhere before all of the groups are restored.
  - _____ – affected disks are _____ to further failure until the string and its affected disks are _____.

3.2.6 Modeling Correlated Failures – Pessimistic Calculation

- τ - (random) time taken to repair the failed string and all disks affected by it
- $f_{str}(\tau)$ - probability density function of τ
- $F^{*}_{str}(\tau)$ - Laplace transform of $f_{str}(\tau)$
- Pessimistic assumption - rate of additional failures

$$\lambda_{pess} = (d+1)\lambda_{str} + (d+1)g\lambda_{disk}$$

- Conditioning upon τ - the probability of data loss

$$p_{pess} = 1 - e^{-\lambda_{pess}\tau}$$

- Integrating on τ - unconditional pessimistic probability of data loss

$$\pi_{pess} = 1 - F^{*}_{str}(\lambda_{pess})$$

3.2.6 Modeling Correlated Failures – Optimistic Calculation

- Optimistic assumption - rate of additional failures

$$\lambda_{opt} = d\lambda_{str} + dg\lambda_{disk}$$

- Conditioning upon τ - the probability of data loss is

$$p_{opt} = 1 - e^{-\lambda_{opt}\tau}$$

- Integrating on τ - unconditional optimistic probability of data loss

$$\pi_{opt} = 1 - F^{*}_{str}(\lambda_{opt})$$

3.2.6 Modeling Correlated Failures – Reliability of Orthogonal Configuration

- Rate of string failures triggering data loss –

$$\Lambda_{str} = (d+1)\lambda_{str}\pi; \quad (\pi_{pess} \quad or \quad \pi_{opt})$$

- Approximate rate of data loss in the system -

$$\Lambda_{data\_loss} \approx \Lambda_{indiv} + \Lambda_{str}$$

- Mean Time To Data Loss - $MTTDL \approx \frac{1}{\Lambda_{data\_loss}}$
- System reliability - $R(t) \approx e^{-\Lambda_{data\_loss} t}$

3.3 Data Replication – Introduction

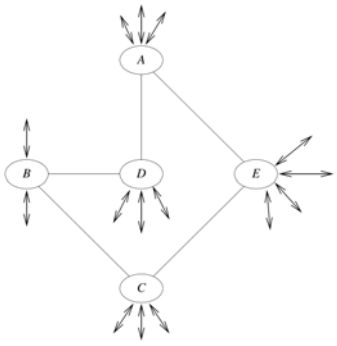
- Data replication consists of holding _____ copies of data on _____ nodes in a _____ system
- Data replicates must be kept _____ despite _____ in the system.
- Managing replication: _____ and _____ voting schemes.
- Voting is used to specify _____ of nodes that need to be updated for _____ or that need to be accessed for _____.

3.3.1 Voting: Non-Hierarchical Organization

- Simplest voting scheme:
  - Assign _____ to _____ of a datum
  - S is the set of _____ with _____
  - $v = \Sigma_{i \in S}$, $r + w > v$, $w > v/2$, r and w integers
  - V(X) denotes the _____ assigned to copies in _____ of nodes.
  - To complete a ____, it is necessary to _____ from _____ of a set $R \subset S$ such that $V(R) \geq r$. Similarly, to complete a _____, we must find a set $W \subset S$ such that $V(W) \geq w$, and execute that write on _____.
  - For any sets R and W, we must have $R \cap W \neq \phi$ (because $r + w > v$)
  - For any two sets W_1 and W_2, $W_1 \cap W_2 \neq \phi$ (because $w_1 + w_2 > v$)

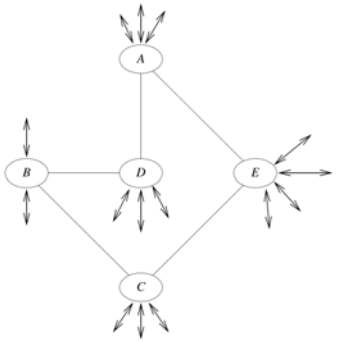
3.3.1 Voting: Non-Hierarchical Organization

- A _____ is any set R such that $V(R) \geq r$ and a _____ is any set W such that $V(W) \geq r$.
- Example:



  - Assume one vote/node, v = 5.
  - For $w > 5/2$, $w \in \{3, 4, 5\}$, $r + w > v \rightarrow r > v - w$
  - $(r, w) \in \{(3, 3), (4, 3), (5, 3), (2, 4), (3, 4), (4, 4), (5, 4), (1, 5), (2, 5), (3, 5), (4, 5), (5, 5)\}$
  - Consider (r, w) = (1, 5). A ____ _____ can be successfully completed by reading _____ of the ____ copies.

3.3.1 Voting: Non-Hierarchical Organization



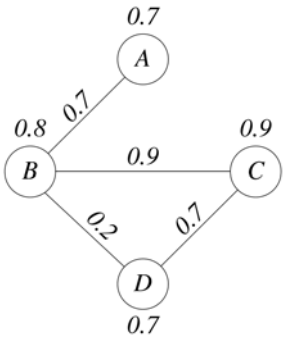
- As another example, consider (r, w) = (3, 3). Only _____ copies have to be _____ for a successful _____.
- However, each _____ takes longer because _____ _____ have to be accessed. _____ suffers but _____ increases because it is still possible to satisfy r = w = 3 with ____ _____.
- If there are many _____ than _____, (1, 5) allows better _____ but worse _____ since the system cannot satisfy _____ if A is disconnected.

3.3.1 Voting: Non-Hierarchical Organization

- System _____ is the probability that both _____ are available.
- The problem of _____ such that availability is maximized is very hard, a _____ gets us close.
- Definitions: node and link availability, $a_n(i)$ and $a_l(i)$, set of links incident on node I, L(i) (all at some t)
- Heuristic 1
  - Assign to node i a vote $v(i) = a_n(i)\sum_{j\in L(i)}a_l(j)$ _____ to the _____. If the _____ assigned to nodes is even, give _____ to one of the nodes with the _____ of votes.
- Heuristic 2
  - Let k(i, j) be the node _____ to node _____. Assign to node i a vote $v(i) = a_n(i) + \sum_{j\in L(i)} a_l(j)a_n(k(i, j))$ rounded to the nearest integer. Give one extra vote as with heuristic 1.

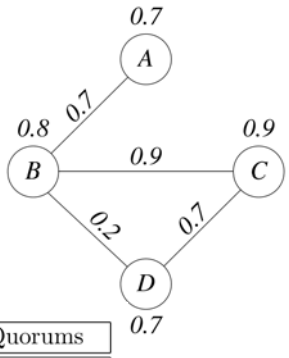
3.3.1 Voting: Non-Hierarchical Organization – Heuristic 1 Example

- Vote Assignments
  - v(A) = round(_____) = __
  - v(B) = round(_____) = __
  - v(C) = round(_____) = __
  - v(D) = round(_____) = __
  - r + w > __, w > _____, w ∈ {_____}
    - For w=__, r=__ is the smallest read quorum; possible read/write quorums are {_____}
    - For w=__, r=__ is the smallest read quorum; possible read quorums are {_____}, only write quorum is {_____}



3.3.1 Voting: Non-Hierarchical Organization – Heuristic 2 Example

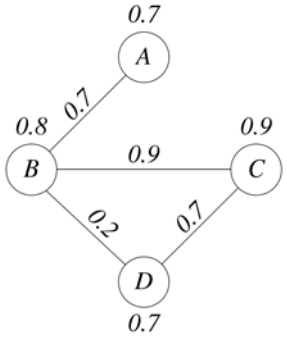
- Vote Assignments
  - v(A) = round(_____) = __
  - v(B) = round(_____ _____) = __
  - v(C) = round(_____ _____) = __
  - v(D) = round(_____ _____) = __
  - r + w > __, w > ___, w ∈ {_____



r	w	Read Quorums	Write Quorums
4	4	AB, BC, BD, ACD	AB, BC, BD, ACD
3	5	B, AC, CD	BC, ABD
2	6	B, C, AD	ABC, BCD
1	7	A, B, C, D	ABCD

3.3.1 Voting: Non-Hierarchical Organization – Availability Example

- Consider (r, w) = (4, 4)
- Availability in this case is the probability that _____ one of the quorums _____ can be used.
- System availability is not a ___ of quorum availability because they are not _____ events.
- Instead, list _____ _____ of system components' states and add up the probabilities for those combinations _____ _____.
- Each _____ can be _____ _____, consider 256 possibilities here.



3.3.1 Voting: Non-Hierarchical Organization – Dynamic Vote Assignment

- The requirement of _____ may be very hard to maintain as _____, even though a _____ of the system _____.
- _____ can counter this problem., involves keeping _____ for each datum.
- Notation:
  - VN_i - _____ of data at node i
  - SC_i - _____ at node i - number of nodes _____ of this data
  - When system starts operation, SC_i is initialized to the _____ in the system
  - S_i - set of nodes _____ i can communicate
  - M - maximum _____ in S_i
  - I - _____ set of S_i having _____
  - N - _____ update sites cardinality (S_i) of nodes in I

3.3.1 Voting: Non-Hierarchical Organization – Assignment Algorithm

1. If an update request arrives at node i, node i computes the following quantities:
   - $M = \max\{VN_j, j \in S_i\}$ (where S_i is the set of nodes with which node i can communicate, including i itself), i.e., the maximum version number of the concerned datum, among all the nodes with which node i can communicate.
   - $I = \{j | VN_j = M, j \in S_i\}$, i.e., the set of all nodes whose version number is equal to the maximum.
   - $N = \max\{SC_j, j \in I\}$, i.e., the maximum update sites cardinality associated with all the nodes in I.
2. If $\|I\| > N/2$, then node i can raise a write quorum and is allowed to carry out the update on all nodes in I; otherwise the update is not allowed. The update is carried out and the version number of each copy of that datum in I is incremented, i.e., VN_i is incremented for each $i \in I$. Also, for each $i \in I$, we set $SC_i = \|I\|$. This entire step must be done atomically: all these operations must be done at each node in I, or none of them can be done.

||I|| is the _____ of I

3.3.1 Voting: Non-Hierarchical Organization – Dynamic Example

- Seven nodes – state at t_0

	A	B	C	D	E	F	G
VN	5	5	5	5	5	5	5
SC	7	7	7	7	7	7	7

- _____ → {A, B, C, D}{E, F, G}
- E receives _____ at $t_1 > t_0$, E needs __ only has __, rejects update
- __ receives update request at $t_2 > t_0$, __ needs __, has __, request is honored.
- New state at t_2

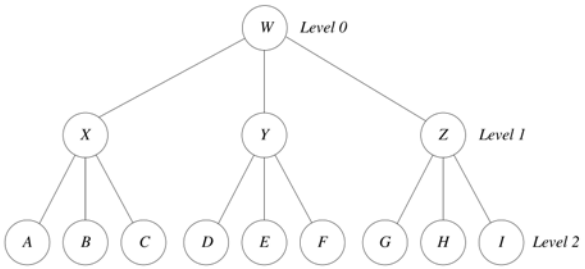
	A	B	C	D	E	F	G
VN	6	6	6	6	5	5	5
SC	4	4	4	4	7	7	7

- Disconnection at $t_3 > t_2$ → {A, B, C}{D}{E, F, G}
- __ receives update request at $t_4 > t_3$, __ needs __, has __, request is honored
- New state at t_4

	A	B	C	D	E	F	G
VN	7	7	7	6	5	5	5
SC	3	3	3	4	7	7	7

3.3.2 Voting: Hierarchical Organization

- Construct m-level ___.
  - Let all nodes holding copies of the data be the _____ at level m-1.
  - Add virtual nodes at _____ to level 0.
  - Each node at level l will have the same _____ _____, denoted by l_{i+1}. Here, $l_1 = l_2 = 3$

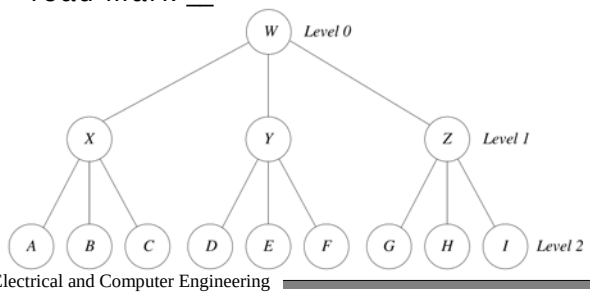


3.3.2 Voting: Hierarchical Organization - Algorithm

- Assign _____ to each _____.
- Define r_i and w_i at level l to satisfy $r_i + w_i > l_i$, $w_i < l_i/2$
- Algorithm
  - Read-mark the root at level 0
  - At level 1 - read-mark r_1 nodes
  - Proceeding from level i to level i+1 - read-mark r_{i+1} children of each of the nodes read-marked at level i
  - You cannot read-mark a node which does not have at least r_{i+1} non-faulty children
  - Proceed until i = m-1

3.3.2 Voting: Hierarchical Organization - Algorithm Example

- Select _____ for I = ___ and set r_i = _____
- Starting at _____, read-mark _____
- Moving to _____, read-mark _____ and _____
- The read quorum is _____
- If ___ had been faulty, read-mark __ instead.
- If _____ faulty, can't read-mark __, go back and read-mark __



Quorum size is 4 compared to at least 5 with Non-Hierarchical Approach

3.3.3 Primary Backup Approach

- One node is designated as the _____, route _____ through that node.
- Designate other nodes as _____.
- Under normal operation, copy _____ to the primary to all _____ backups.
- When the primary _____, choose _____ _____ to take its place.

3.4 Algorithm-Based Fault Tolerance

- Data replication at the _____ level.
- Well-suited for _____ of data.
- Use _____.
- Given an n x m matrix A, the _____ matrix A_C is

$$A_C = \begin{bmatrix} A \\ eA \end{bmatrix} \quad \text{where} \quad e = [111\cdots1]$$

- The _____ matrix, A_R, is similar

$$A_R = [A\ Af] \quad \text{where} \quad f = [111\cdots1]^T$$

- The _____ matrix, A_F of size _____ is

$$A_F = \begin{bmatrix} A & Af \\ eA & eAf \end{bmatrix}$$

- Column and row checksums detect _____, both allow _____.

3.4 Algorithm-Based Fault Tolerance

- To allow locating and correcting by adding only rows or columns but not both, add an additional row or column.

$$A_C = \begin{bmatrix} A \\ eA \\ e_wA \end{bmatrix} \quad \text{where} \quad e_w = [1,2\cdots2^{n-1}]$$

$$A_R = [A\ Af\ Af_w]$$

$$\text{where} \quad f_w = \begin{bmatrix} 1,2\cdots2^{m-1} \end{bmatrix}^T$$

$$A_F = \begin{bmatrix} A & Af & Af_w \\ eA & eAf & eAf_w \\ e_wA & e_wAf & e_wAf_w \end{bmatrix}$$

3.4 Algorithm-Based Fault Tolerance
- Weighted Checksum Code

- Example for _____ correction:
  - Suppose an error detected in _____
  - WCS1/WCS2 _____
    checksum eA/e_wA for column j
  - Calculate _____:

$$A_C = \begin{bmatrix} A \\ eA \\ e_wA \end{bmatrix}$$

$$S_1 = \sum_{i=1}^{n} a_{i,j} - WCS1 \qquad S_2 = \sum_{i=1}^{n} 2^{i-1} a_{i,j} - WCS2$$

  - If _____ syndrome is nonzero – the checksum is wrong. If both are nonzero _____ implying that _____ is in error and can be corrected through

$$a'_{k,j} = a_{k,j} - S_1$$

3.4 Algorithm-Based Fault Tolerance
- Weighted Checksum Code Example