Cryptography and Network security

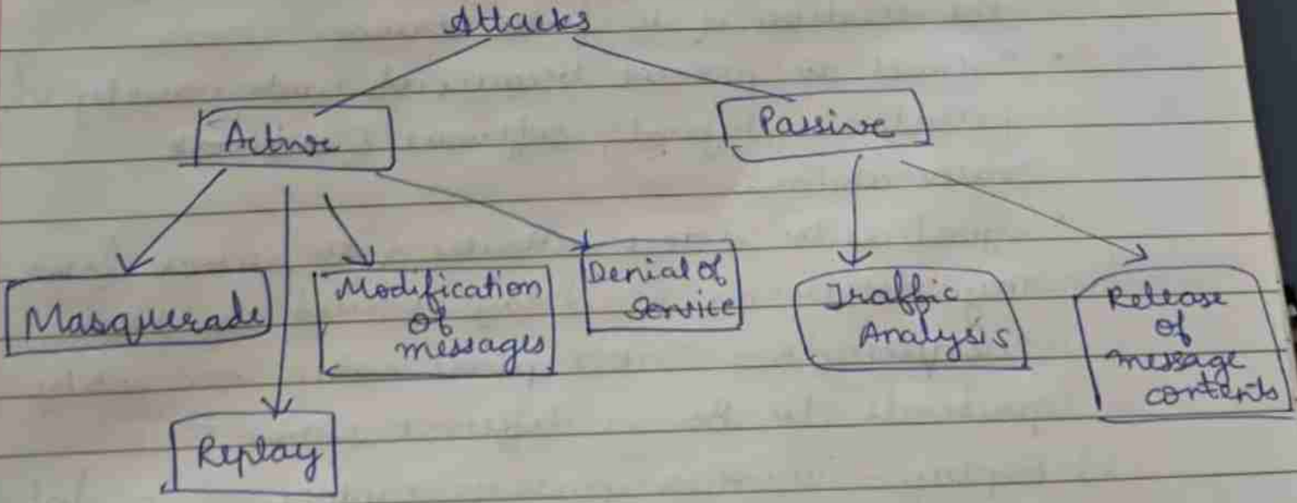
MOD 1

- Introduction

- Information security ensures that both physical and digital data is protected from ~~both phys~~ unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction
- cyber security is a process or measures taken by organisations or experts to protect devices, computer networks, or data from malicious activities.
- Network security is a subset of cybersecurity, aim to protect any data that is being sent through devices in your network to ensure that the information is not changed or intercepted.

Security attacks

- Actions that compromises the security of an organisation or an individual



- Passive attacks are in the nature of eaves-dropping on, or monitoring of transmissions. Goal is to obtain information that is being transmitted.
Passive attack attempts to learn or make use of information from the system but does not affect system resources.
  - release of message content - intruder is reading the messages.
  - traffic analysis - monitor traffic flow to determine location & identity of communicating hosts and could observe the frequency & length of messages being exchanged.

These attacks are difficult to detect because they do not involve any alteration of data

- Active Attack - An active attack attempts to alter system resources or affect their operation.
  - Involves some modification of data stream or the creation of a false stream.
  - Difficult to prevent because of wide variety of potential physical, software & network vulnerabilities.
  - Goal is to detect attacks & to recover from any disruption or delays caused by them.

1) Masquerade - Takes place when one entity pretends to be a different entity.
2) Replay - Involves passive capture of a data unit & its subsequent retransmission to produce an unauthorized effect.

3) Modification of messages - some portion of a legitimate message is altered, or messages are delayed or reordered to produce an unauthorized effect.

4) Denial of service - Prevents or inhibits the normal use or management of communications facilities.

Security Goals (CIA)

1] Confidentiality

Only authorized users access information.

2] Integrity

Ensure completeness, accuracy and an absence of unauthorized modifications.

3] Availability

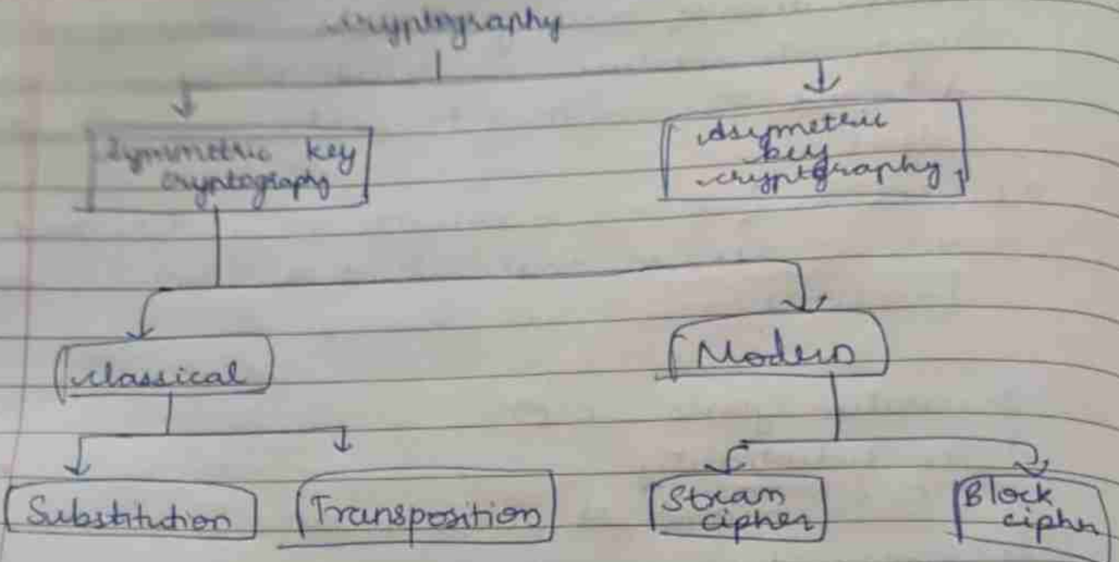
Available and operational when required by users.

Authentication - a mechanism by which a user is identified & uses some token to prove who they are.

Non repudiation - a way to guarantee that the sender of a message cannot later deny having sent the messarage and that the recipient cannot deny having received the messages.

☆ cryptography is technique of securing information and communications through use of codes so that only those person for whom the information is intended can understand it and process it.
Thus preventing unauthorized access to information.

$$\begin{tikzcd} \text{Cryto} \arrow[d] & \text{graphy} \arrow[d] \\ \text{hidden} & \text{writing} \end{tikzcd}$$



Symmetric Key Cryptography (Classical in diag)

Modern ciphers
1) Stream
2) Block.

Modern Block Ciphers
- widely used
- Provide secrecy / authentication services
- Focus on DES (Data Encryption standard)

Block ciphers work on block / word at a time
which is some number of bits. All these bits
have to be available before the block can
be processed. Block ciphers like a substitution
on ~~every~~ very big characters 64 bits or
move.

→ Broader range of applications

Camlin Page
Date / /

- stream ciphers processes messages a bit or byte at a time when en/decrypting

* ~~Feistal~~ Feistel cipher structure (Horst Feistel)

Partitions input block into two halves which are processed through multiple rounds which perform a substitution on left data half, based on round function of right half & subkey, and then have permutation swapping halves.

One layer of S-boxes & the following P-box are used to form the round function

Feistal cipher Design Elements
- block size → increasing size improves security, but slows cipher
- key size → increasing size improves security, makes exhaustive key searching harder, but slow cipher.
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis - for easier validation & testing of strength.

→ increasing no of rounds improves security but slows cipher.

→ greater complexity can make analysis harder, but slows cipher.

Data Encryption standard (DES)

· encrypts 64 bit data using 56 bit key.

· LUCIFER Algorithm is a Feistal block cipher that operates on block of 64 bits using a key size of 128 bits.

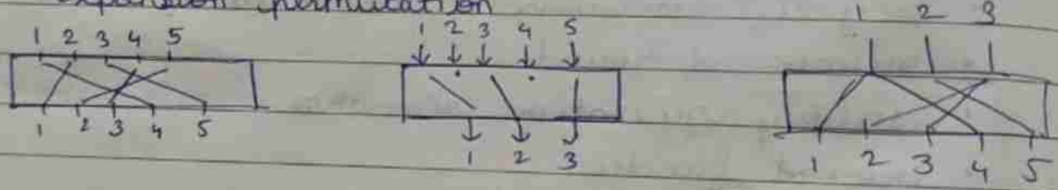
DES design controversy (cover size of key & classified design criteria)

· Applications of DES in Financial applications

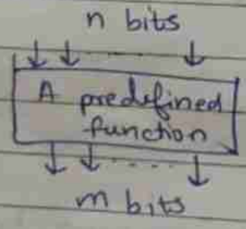
components of a modern block cipher

1) Transposition

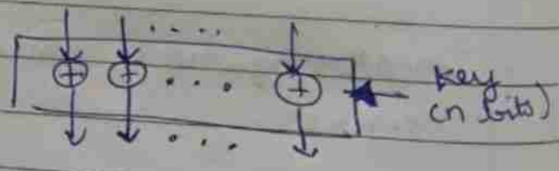
- straight permutation
- compression permutation
- expansion permutation



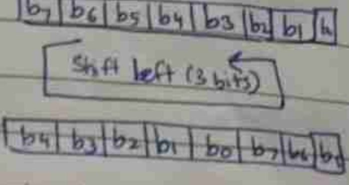
2] Substitution



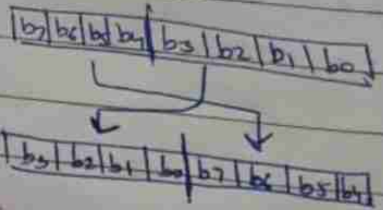
3] Exclusive OR



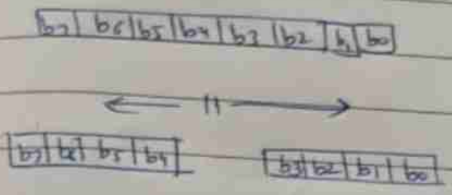
4] Shift



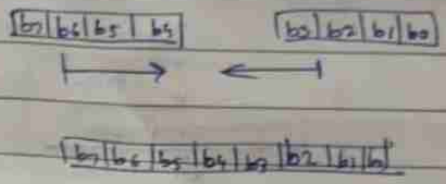
5] Swap



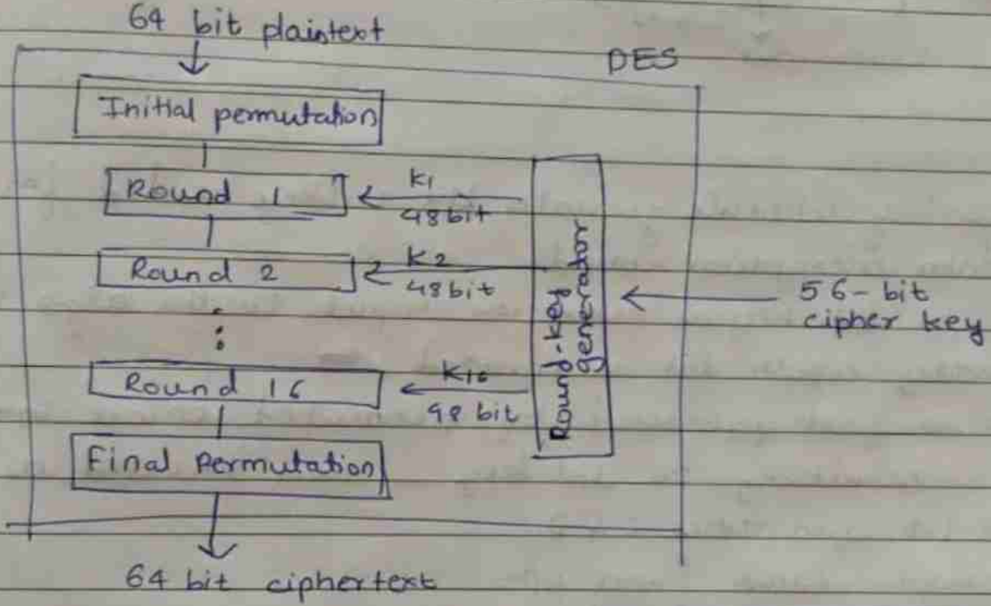
6] split

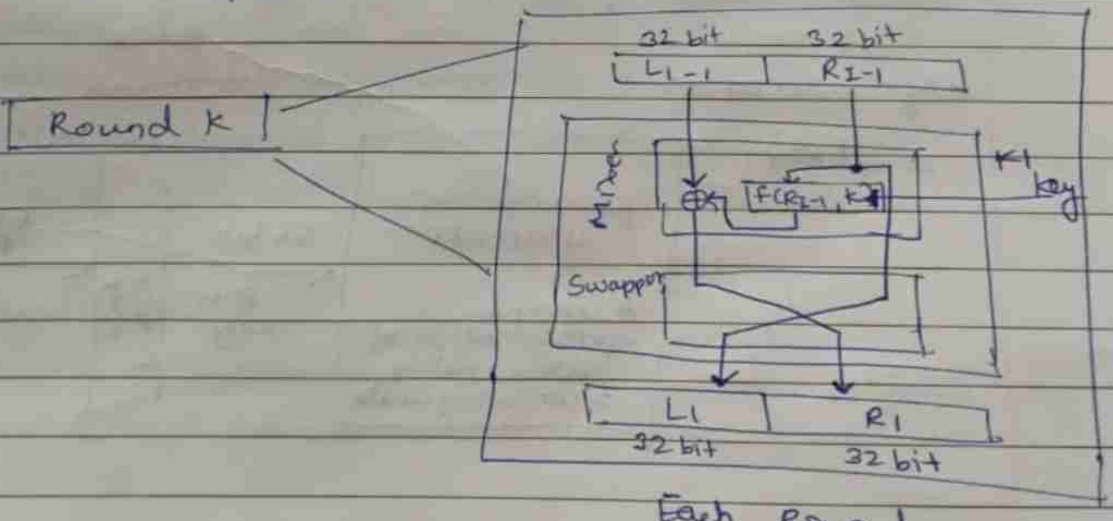


7] Combine



General structure of DES

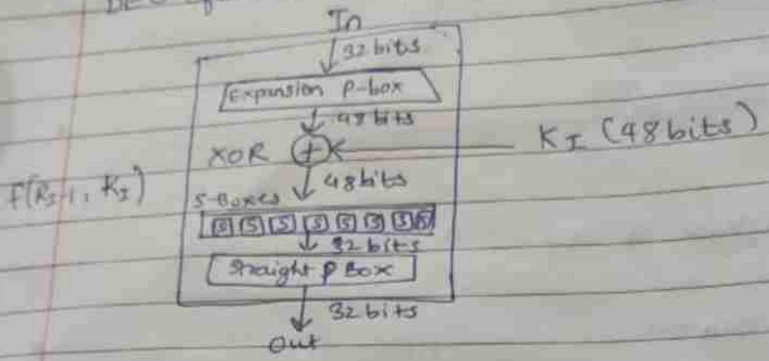




Each Round

Camlin Page Date / /

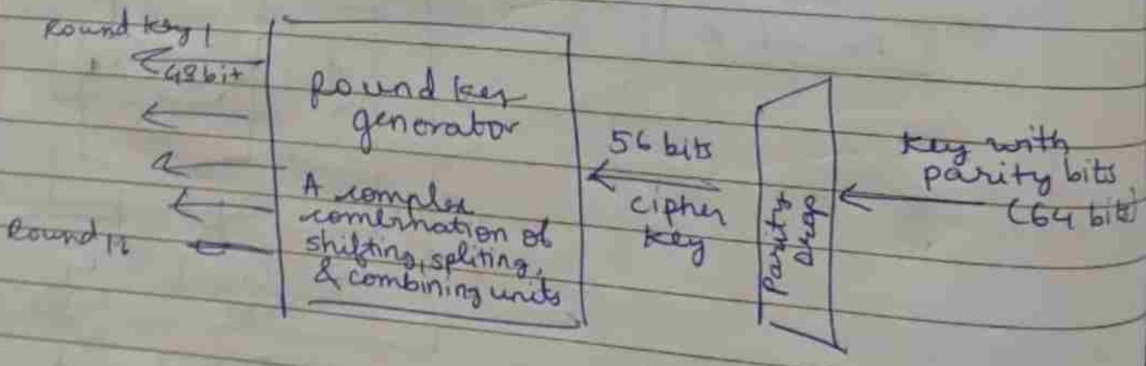
DES function



- DES key schedule generates the subkeys needed for each data encryption round.
- A 64 bit key is used as input to the algo, though every eighth bit is ignored,
- It is first processed by permuted choice one.
- The resulting 56 bit key is then treated as two 28-bit quantities C & D.

Example Read from ppt.

Key generation


Round key 1
48 bit
Round 16
Round key generator
A complex combination of shifting, spliting, & combining units
56 bits
cipher key
Parity drop
key with parity bits (64 bits)

Advance encryption standard

- Increase in block size from 64 bit upto 128 bits. & key from 128 to 256 bits.
- Triple DES - is secure & well understood but is slow

AES competition Requirements

- symmetric key block cipher
- 128 bit data, 128/192/256 bit keys
- stronger & faster than Triple DES
- Provide full specification & design details

Evaluation criteria

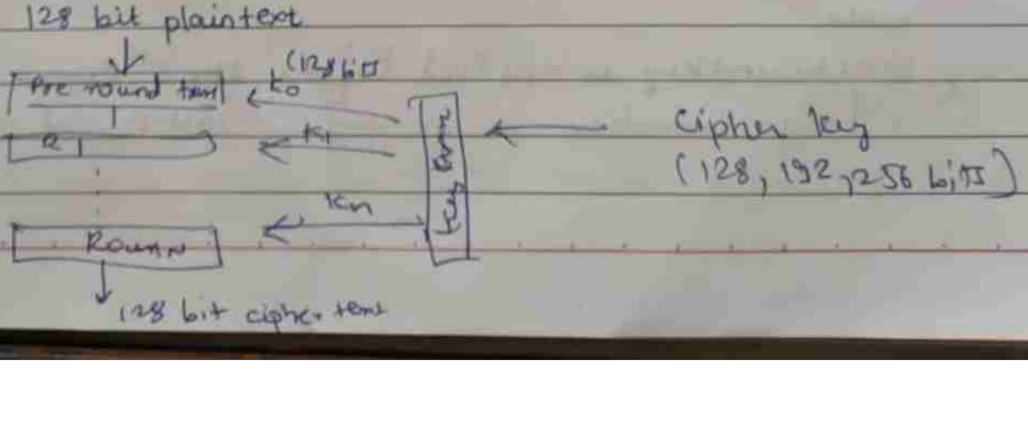
- general security
- ease of soft & hardware implementation
- implementation attacks.
- flexibility

AES cipher - Rijndael design

- simplicity
- 128/192/256 bit keys, 128 bits data
- resistant against known attacks
- speed and code compactness on many CPUs

Multiple Rounds

- Rounds are (almost) identical.
- First & last round are a little different.



No of Rounds	keysize
10	128
12	192
14	256

1) Key Expansion
- Round keys are derived from cipher key using Ryndael's key schedule.

2) Initial Round
AddRound key: Each byte of the state is combined with the round key using bitwise xor.

3) Rounds (Except ~~1st &~~ Last Round)
- SubBytes : non linear substitution step
- Shift Rows : transposition step
- Mix columns: mixing operation of each column.
- AddRoundKey

4) Final Round
- SubBytes
- Shift Rows
- AddRoundKey

$\boxed{\text{No Mix Columns}}$

128 bit values
- Data block viewed as 4-by-4 table of bytes.
- Represented as 4 by 4 matrix of 8 bit bytes.

Key is expanded to array of 32 bit words

Note:
1) OneRoundkey is applied before the 1st Round.
2) 3rd transformation is missing in last Round.

$$\begin{tikzcd}
\text{State} \arrow[d] & \\
\boxed{\text{SubBytes}} \arrow[d] & \\
\boxed{\text{State}} \arrow[d] & \\
\boxed{\text{ShiftRows}} \arrow[d] & \\
\boxed{\text{State}} \arrow[d] & \\
\boxed{\text{MixColumns}} \arrow[d] & \\
\boxed{\text{State}} \arrow[d] & \\
\boxed{\text{AddRoundKey}} \arrow[d] & \text{Round key} \arrow[l] \\
\text{State} &
\end{tikzcd}$$

1) SubBytes : Byte substitution
- simple substitution of each byte
  provide a confusion
- Uses one S-box of 16×16 bytes containing a permutation of all 256 8 bit values
- Each byte of state is replaced by byte indexed by row (left 4 bits) & column (right 4 bits)
  - S-box constructed using defined transformation of values in Galois Field - GF (2^8)
- Involves 16 independent byte to byte transformations.

2) Shift Rows.

Shifting which permuts the bytes
A circular byte shift in each Row
- 1st row unchanged
- 2nd row does 1 byte circular shift to left
- 3rd " does 2 byte " " " "
- 4th " does 3 byte " " " "

The transformation is called ShiftRows.
In decryption it is called InvShiftRows & shifting is to the Right.

3) Mix Columns
- Shift Rows & Mix Column provide diffusion to the cipher.
- Each column is processed separately.
- Each byte is replaced by value dependent on all 4 bytes in column
- Effectively a matrix multiplication in GF (2^8)

4) AddRoundkey
- XOR state with 128 bits of round key
- It proceeds one column at a time.
  - adds a round key with each state column matrix.
  - operation is matrix addition.
- Inverse for decryption identical
  - since XOR own inverse, with reversed keys

Camlin Page
Date / /

Mess start
20th July

Introduction to Parallel computing

- A parallel computer is a collection of processing elements that communicate and co-operate to solve large problems fast.
- Processing of multiple tasks simultaneous on multiple processor is called parallel processing.
- Parallel computing is type of computation

Types of Parallelism
- Instruction Parallelism
- Thread or Task level Parallelism
- Data Level Parallelism
- Bit Level Parallelism

* Motivation of Parallel Computing

* Scope of Parallel ~~computation~~ computation
Applications
→ Commercial computing
  - Weather forcasting
  - Remote sensors, Image Processing
  - Process optimization, operations research
→ Scientific & Engineering Computing.

Concurrency Vs Parallelism
- concurrency is when two tasks can start, run and complete in overlapping time periods.

Parallelism - 1) Pipelining
2) Data Parallelising
3) Partitioning
4) Overlapping
5) Concurrency
6) Replication
7) Time sharing
8) space sharing
9) Multitasking
10) Multiprogramming
11) Multi threading
12) Distributed computing

control structure of Parallel Platforms
• Parallel tasks can be seperated based on granuality.

computation / communication Ratio:
In parallel computing, granuality is a qualitative

1) Fine grain Parallelism
2) Coarse grain Parallelism

SIMD & MIMD Architecture (SIMD / MIMD comparison)

MIMD more scalable
SIMD is simple architecture compared to MIMD

```
IF (B==0)
    C=A;
else
    C = A/B;
```

• Platform that provide shared data space are called shared address space machiches or multiprocessors.
• Platform that provide messasing are also called message passing platform or multicomputers.

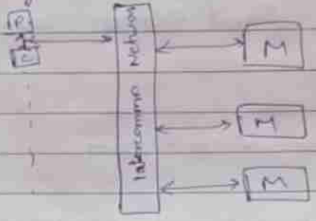
Camlin Page
Date / /

Two primary forms
1) shared data space
2) exchanging messages

- Communication Model of Parallel Platform (Shared Address Space Platforms)

PRAM
Parallel Random Access Machine

Typical shared address space architecture
(a) Uniform Memory Access (UMA)
- Here all processors share the physical memory uniformly.



Non Uniform Memory Access (NUMA)

PRAM (
n processors working on n number of operations which are independent. It may result to accessing same memory space ~~[illegible]~~ by two or more processors.

Architecture of an Ideal Parallel Computer (pram)
1) Exclusive Read, exclusive write. (EREW) (Eg Mutual Exclusion)
2) Concurrent Read, Exclusive write (CREW) (Eg websites, blogs)
3) Exclusive Read, Concurrent write (ERCW)
4) Concurrent Read, Concurrent write (CRCW)
↳ (Eg Cloud services)

(Eg Developer with DBA)

→ weak as more like sequential model

(Cache Coherancy)

Resolving concurrent writes

- Common : write only if all values are identical
- Arbitary : write the data from a randomly selected processor.
- Priority : follow a predetermined priority order
- Sum : write the sum of all data items.

- Message Passing vs Shared Address space Platform

PRAM
↓
Also stand for no. of processors

Methods to implement the PRAM model,

i) Shared Memory model - (OpenMP) lib
ii) Message passing model (MPI) lib
iii) Distributed Memory Model

i) Shared Memory - emphasizes on control parallelism than data parallelism.

- Multiple processes execute on different processors independently ~~by~~ but they share common memory space.

ii) Message passing ~~mod~~ (Distributed memory systems)

- logical machine view of a message passing platform consists of p-processing nodes.
- All processors have their own local memory.

Distributed Memory systems require comm network to connect inter processor memory.

- No cache Coherancy in Distributed memory.

* Physical complexity of an Ideal Parallel computer

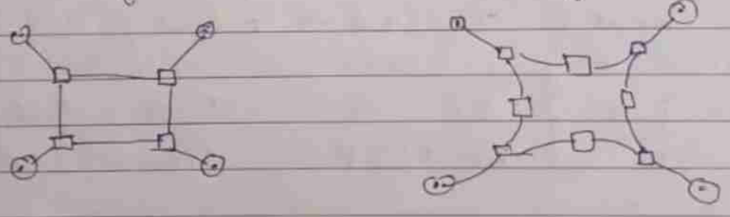
i) Processors and memories are connected via switches.

i) Since switch operates in $O(1)$ time at the level of word for a a system of p processors & m words, the switch complexity is $O(mp)$.

·) For meaningful value of p and m a true PRAM is not realizable.

* Interconnection Networks for Parallel computers

- Interconnection network carry data between processors and to memory.
- Made of switch & links
- classified as static or dynamic



Static Dynamic

- Switch maps a fixed number of inputs to outputs.
- Total no of parts on a switch is degree of switch
- cost of switch grows as the square of degree of the switch.

Peripheral hardware $\propto$ degree

Packaging costs $\propto$ no of pins.

- Processors talk to network via a network interface

Network topology - 1) Buses (without cache & with cache)
2) Crossbar
3) Multistage Networks

A crossbar network uses an p×m grid of switches to connect p inputs in a non blocking manner.

- crossbar have excellent performance scalability but poor cost scalability.
- Buses have excellent cost scalability but poor performance scalability.
- Multistage networks interconnects strike a compriss between these 2.

(Dynamic)
Multistage - Omega Network (commonly used)
Network consists of $\log_2 p$ stages, where p is no of inputs / outputs.
For input i connected to output j if

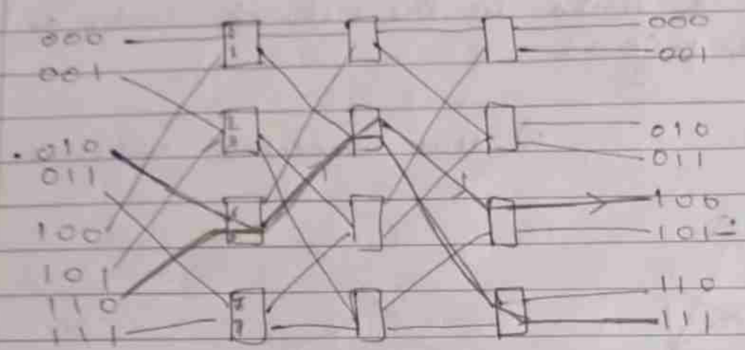
$$j = \begin{cases} 2i & 0 \le i \le p/2 - 1 \\ 2i + 1 - p & p/2 \le i \le p - 1 \end{cases}$$

- Perfect shuffle patterns are connected using 2×2 switchs
- Switches operate in 2 modes
  1) crossover
  2) Pass through

An omega network has $p/2 \times \log p$ switching node & cost of network grows as $\Theta(p \log p)$

s - source d - destination (binary representation)
Data traverses the link to the 1st switch node. If the most significant bits of s & t are same then data is ~~distribu~~ routed in pass through or else crossover.

• This process is repeated for



From 010 to 111

Different so switching crossover

It points to 101 & 101 is connected to 3 so we go to 3 in stage 2

2nd bit is same so passthrough

It points to 011 & 011 is connected to 6

Now 3rd bit is diff so cross over

From 110 to 100

same bit pass through

It points ~~010~~ 101 & 101 is connected to 3

2nd bit diff so crossover so it points to 010 & 010 is pointing to 4 or (100)

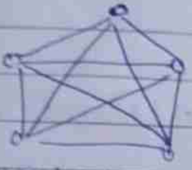
last bit same so pass through.

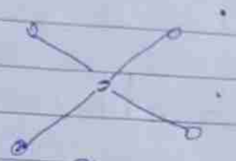
we get 100.

Network topologies : 1) completely connected network
- Each processor is connected to every other processor
- No of links in the network scales as $O(p^2)$.

No of links = $\frac{p(p-1)}{2}$

2) Star connected Networks.


completely connected


Star connected (static counter part of buses)

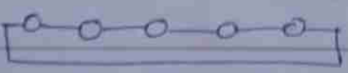
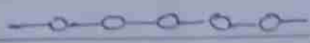
(Not fault tolerant)

Network topologies : Linear Arrays, Meshes & k-d Meshes

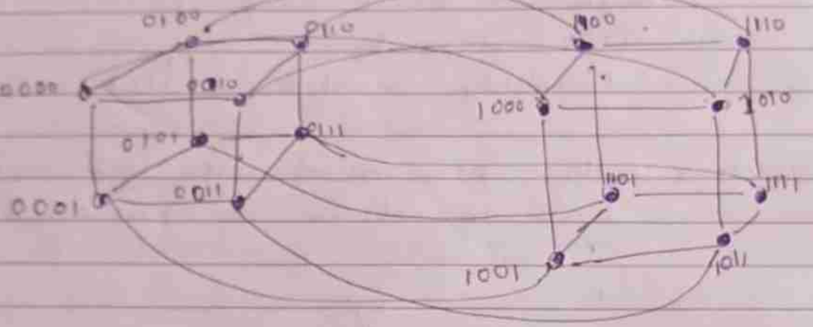
In linear array, each node has 2 neighbors one to its left & one to its right. If the nodes at either end are connected, we refer to it as a 1-D torus or a ring.

- A generalization to 2 dimensions has nodes with 4 neighbors, to north, south, east & west.
- A further generalization to d dimensions has nodes with 2d neighbors.
- A special case of d-dimensional mesh is a hypercube. d = log p where p is the total no. of nodes.

Linear array ① with wraparound links. ② without wraparound

①



Network topologies : Hypercubes and their Construction



4 D Hypercube made of 2 (3D Hypercube)

Sending data from 0000 to 0010
1 link req & 1 bit diff
0000 to 0110
2 link req & 2 bit diff
0000 to 1111
4 link req & 4 bit diff

∴ ~~[illegible]~~

• Dist betwⁿ 2 ~~two~~ nodes is at most log p

Network topologies : Tree - Based Networks.
complete binary tree network : ① Static ② dynamic

• Dist btwⁿ two nodes is no more than 2logp.
Fat tree is for communications.

Evaluating static Interconnection Networks

- Diameter - dis btw^n 2 farthest nodes in network.

  Linear array - dia = $p-1$

  Mesh - dia = $2(\sqrt{p}-1)$

  Tree & Hypercube = $\log p$.

  Completely connected = $O(1)$

- Bisection width - Min no. of wires you must cut to divide the network into two equal parts.

  linear array = 1

  mesh = $\sqrt{p}$

  hypercube = $p/2$

  completely connected = $p^2/4$

- Cost : No of links ~~of~~ or switches is meaningful measure of cost. linear = p k-d = $p \times d$

  Mesh = $2(p-\sqrt{p})$ Hypercube: $p \frac{\log p}{2}$

  Arc

- Connectivity: Mesh $\sqrt{p}$ Wrap Around $2\sqrt{p}$

Channel Width - No of bits that can be communicated simultaneously ~~b/w~~ in a link connecting two nodes.

Channel Rate - ~~Rat~~

Channel Bandwidth = Channel Rate × Channel Width

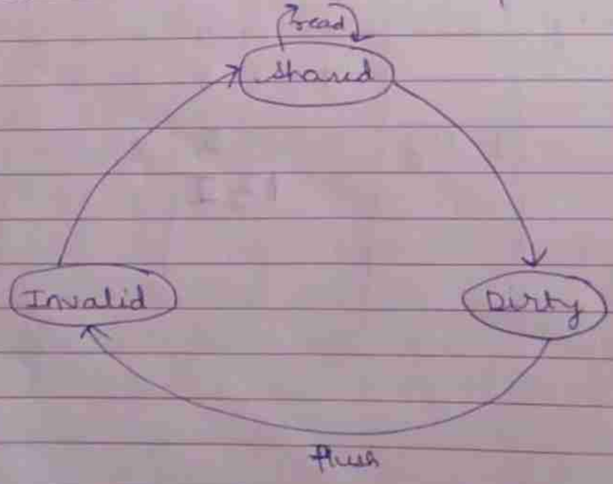
Cache Coherence in multiprocessor systems
· In case of shared address space machines, additional hardware is req to coordinate access to data that might have multiple copies in network.

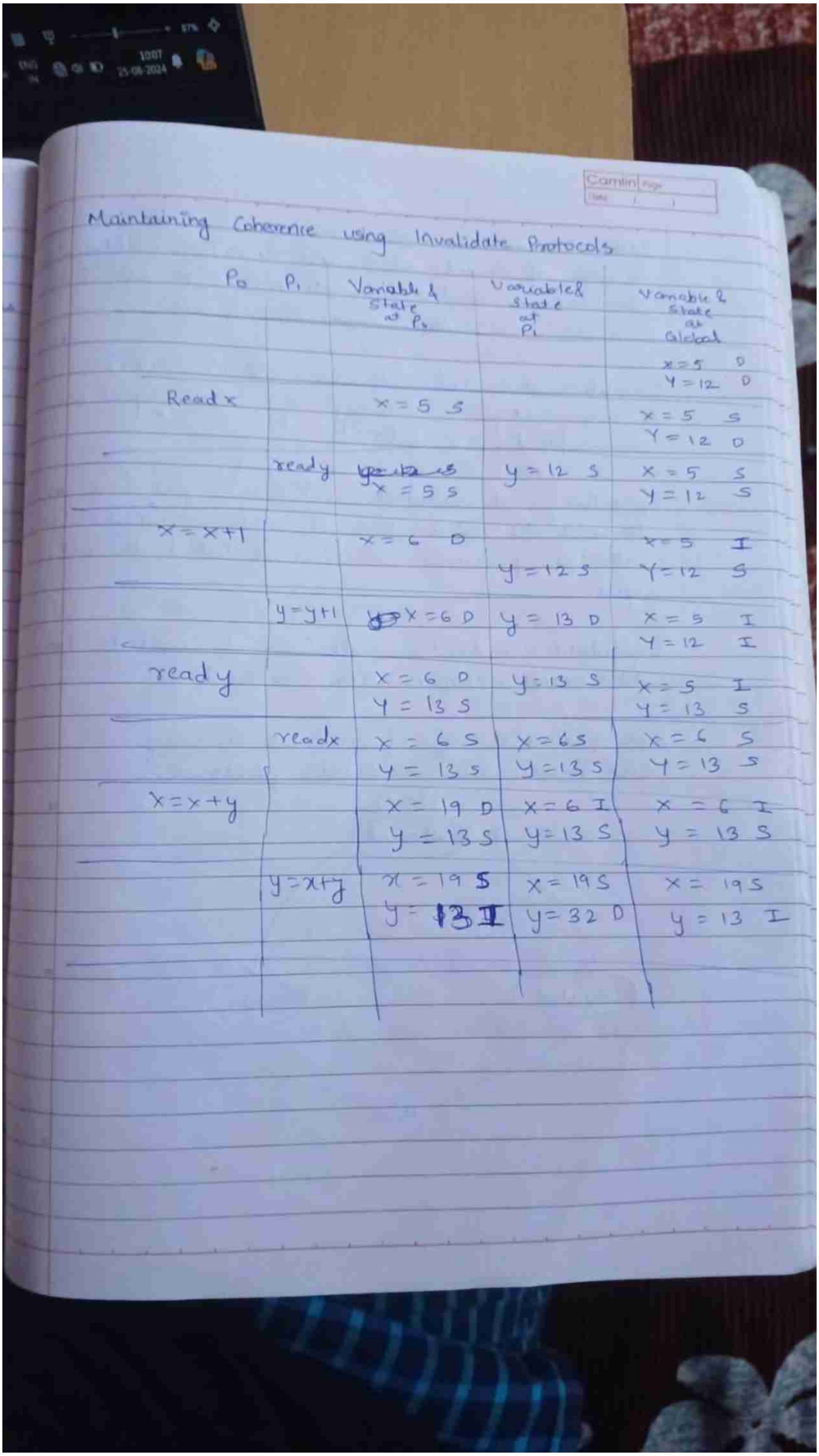
Invalidate protocol & Update Protocol (costlier)
If any processor is writing then other copies get invalidated or updated

If value is invalidated and other processor wants to read the value then it will communicate with the processor which invalidated the other copies.

Invalidate leads to idealing & update lead to excess communication (overhead)

Both protocols suffer from false sharing
Most devices use invalidate protocol.





Maintaining Coherence using Invalidate Protocols

P_0	P_1	Variable & State at P_0	Variable & State at P_1	Variable & State at Global

				x = 5 D
				Y = 12 D

Read x		x = 5 S		x = 5 S
				Y = 12 D

	read y	~~y = 12 S~~
		x = 5 S
			y = 12 S	x = 5 S
				y = 12 S

x = x+1		x = 6 D	y = 12 S	x = 5 I
				Y = 12 S

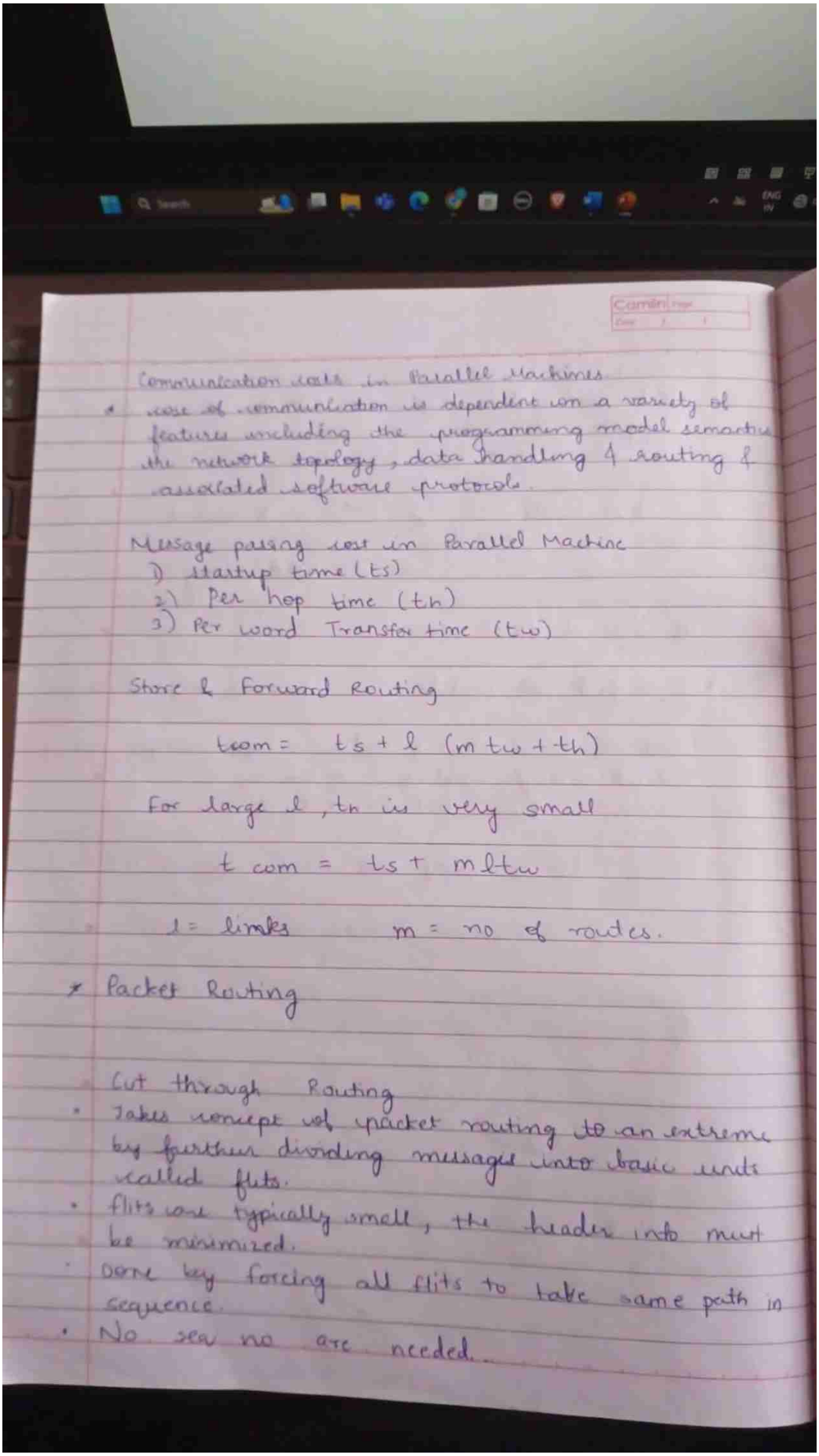
	y = y+1	x = 6 D	y = 13 D	x = 5 I
				Y = 12 I

read y		x = 6 D
		Y = 13 S
			y = 13 S	x = 5 I
				Y = 13 S

	read x	x = 6 S
		y = 13 S
			x = 6 S
			y = 13 S
				x = 6 S
				Y = 13 S

x = x+y		x = 19 D
		y = 13 S
			x = 6 I
			y = 13 S
				x = 6 I
				y = 13 S

	y = x+y	x = 19 S
		y = 13 I
			x = 19 S
			y = 32 D
				x = 19 S
				y = 13 I



Communication costs in Parallel Machines

- cost of communication is dependent on a variety of features including the programming model semantics the network topology, data handling & routing & associated software protocols.

Message passing cost in Parallel Machine

1) startup time (ts)
2) Per hop time (th)
3) Per word Transfer time (tw)

Store & Forward Routing

$$t_{com} = t_s + l\,(m\,t_w + t_h)$$

For large l, th is very small

$$t_{com} = t_s + m\,l\,t_w$$

l = links m = no of routes.

* Packet Routing

Cut through Routing

- Takes concept of packet routing to an extreme by further dividing messages into basic units called flits.
- flits are typically small, the header info must be minimized.
- Done by forcing all flits to take same path in sequence.
- No seq no are needed.

• Total communication time for cut through routing is approx

$$t_{com} = t_s + l t_h + t_w m$$

time with cut through $<$ time for Store & Forward

$<$ ~~Packet~~ Packet Routing

as size of flits is less

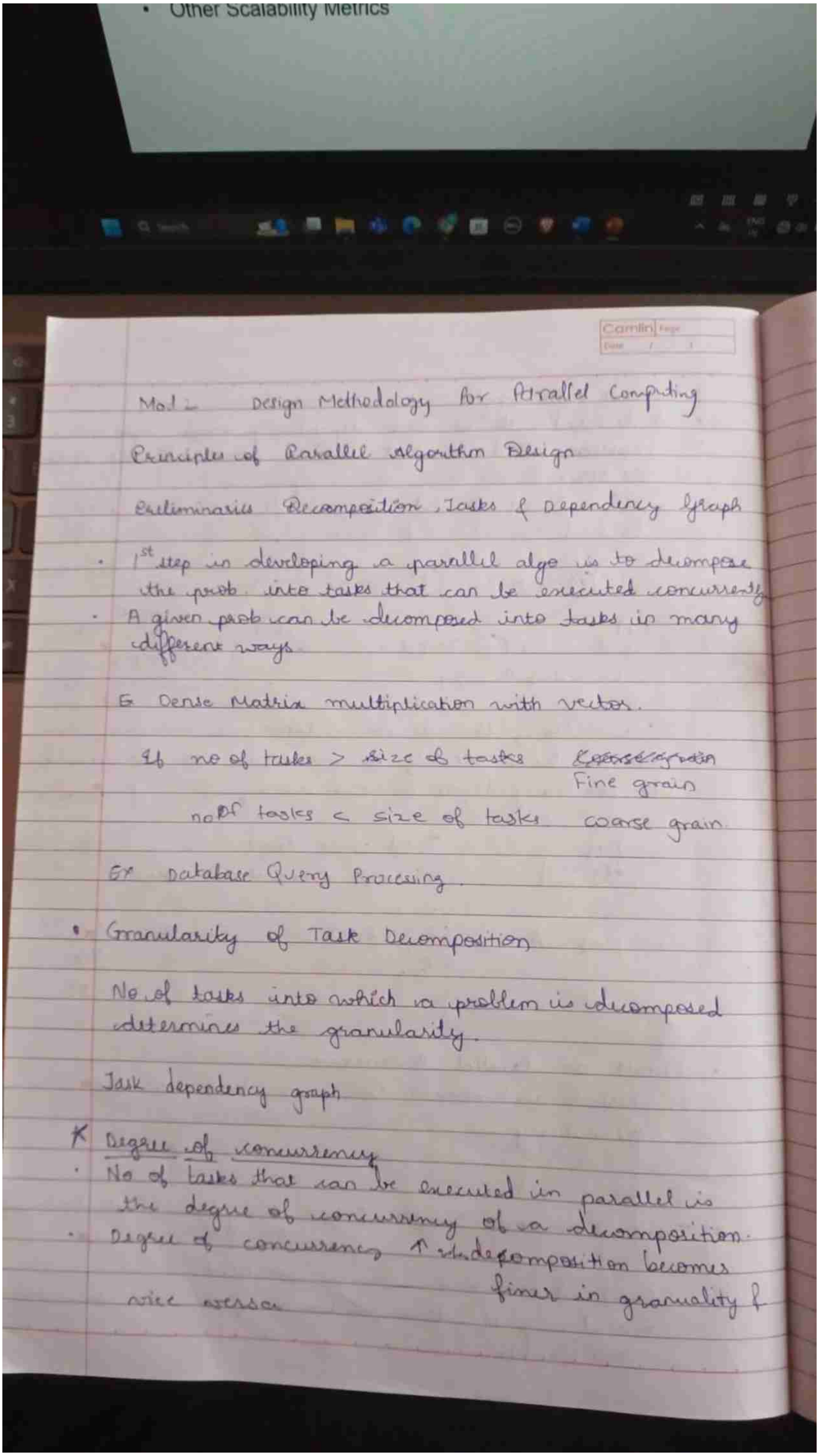
* Simplified Cost Models for Communicating Messages

Cost of comm btw^n 2 nodes l hops away using cut-through routing is given by

$$t_{com} = t_s + l t_h + t_w m$$

t_h is very less

$$t_{com} = t_s + t_w m$$



• Other Scalability Metrics

Mod 2 Design Methodology for Parallel Computing

Principles of Parallel Algorithm Design

Preliminaries Decomposition, Tasks & Dependency Graph

- 1st step in developing a parallel algo is to decompose the prob. into tasks that can be executed concurrently
- A given prob can be decomposed into tasks in many different ways

Ex Dense Matrix multiplication with vector.

If no of tasks > size of tasks	~~Coarse grain~~ Fine grain
no of tasks < size of tasks	coarse grain

Ex Database Query Processing.

- Granularity of Task Decomposition

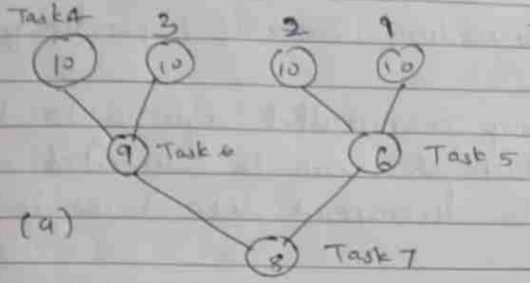
No. of tasks into which a problem is decomposed determines the granularity.

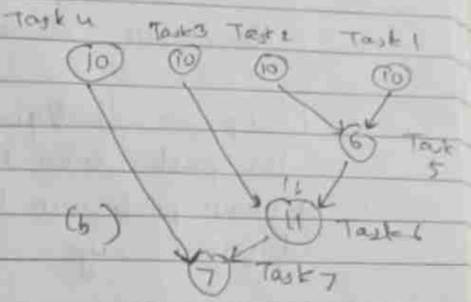
Task dependency graph

* Degree of concurrency
- No of tasks that can be executed in parallel is the degree of concurrency of a decomposition.
- Degree of concurrency ↑ as decomposition becomes finer in granularity & vice versa

Critical Path Length

- A directed path in task dependency graph represents a sequence of tasks that must ~~need~~ be processed one after the other





Critical path = 27 = 10+9+8

Critical path = 10+6+11+7 = 34

If we deploy towards max degree of concurrency then we can have idealing in processors. so we should use the avg degree of concurrency ~~for ease~~

$$\text{Avg degree of conc} = \frac{\text{Total work done}}{\text{Length of Critical Path Lengt}} = \left(\frac{\text{No of tasks}}{\text{No of levels}}\right)$$

For A $= \frac{\cancel{63}^{\cancel{21}\,7}}{\cancel{27}_{\cancel{9}\,3}} = 2.33$ For B $= \frac{\cancel{64}^{32}}{\cancel{34}_{17}} = 1.88$

- Limits on Parallel Performance
  1) There is an inherent bound on how fine the granularity

* Task Interaction Graphs. (How this two graph exchange the data)

More interaction more overhead.
Granularity of decomposition is finer, the associated overhead increases.

Sparse matrix → many elements are 0.

Processes and Mapping

- No of tasks in a decomposition exceeds the no of processing elements available.
- For this reason, a parallel algo. must also provide a mapping of tasks to processes.
- Mappings are determined by both task dependency & task interaction graphs.
- Task dependency graphs can be used to ensure that work is equally spread across at any point (minimum idling & optimal load balance).
- Task interaction graphs can be used to make sure that process need minimum interation with other processes (minimum communication)

If mapping is appropriate parallel time minimizes
1) Mapping independent task to diff processes
2) Assigning tasks on critical path to processes as soon as they become available.
3) Minimizing interaction between processes by mapping tasks with dense interactions to the same process.

* Decomposition Techniques:
1) Recursive decomposition
2) Data decomposition
3) exploratory decomposition
4) speculative decomposition

LLM
Large Language Model

1) Recursive decomposition

- Generally used in problems that can be solved using divide and conquer strategy.
- Task is divided into sub problems.
- sub tasks are divided until a desired granuarity is reached. (Quick sort example)

2) Output Data Decomposition

- Identify the data on which computations are performed.
- Partition this data across various task
- This partitioning induces a decomposition of the problems
- Can be done in various ways which can impact the parallel comp

1) Output data partitioning
2) Input data "
3) Intermediate " "

Input Data Partioning

Intermediate Data Partioning :

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ 4 \end{bmatrix} \cdot \boxed{1 \quad 2} = \begin{pmatrix} 2 & 4 \\ 4 & 8 \end{pmatrix}$$

$$+$$

$$\begin{bmatrix} 3 \\ 5 \end{bmatrix} \quad \begin{bmatrix} 3 & 4 \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 15 & 20 \end{bmatrix}$$

$$\begin{bmatrix} 11 & 16 \\ 19 & 28 \end{bmatrix}$$

16 Tiles Problem
4 x 4 box

3] Exploratory decomposition

$S = \frac{T_s}{T_p}$ Time taken sequentially
Time taken parallely

Super linear speedup $S > 2$
Normally $S \leq 2$

4) Speculative Decomposition

* Characteristics of Tasks
- Task generation

Static task generation
- Concurrent tasks can be identified a-priori. Typical matrix operations, graph algos, image processing applications & other regularly structured probs fall in this class. (Typically decomposed using recursive decomposition)

Dynamic Task Generation:
- Tasks are generated as we perform computation

Size of Data Associated with Tasks

- The size of data associated with a task may be small or large when viewed in context of size of task.

Characteristics of Task interaction

Tasks
i) Static - task & interactions are known a-priori. simpler to code into programs.

2) Dynamic - Timing ~~of~~ or interacting tasks

i) Regular - definite pattern.
2) Irregular - interactions lack well defined topologies
Ex Multiplication of sparse matrix with vector.

- Interactions may be readonly or read-write.
- In Read Only task just read data items associated wit a task

- Interactions may be one way or two way.
- A one-way interaction can be initiated & accomplished by one of the two interacting tasks.
- One way interactions are harder to code in message passing APIs.
- Two way interactions require interaction from both tasks.