9/12/2017 :- Prof. Banka (soft computing)

Threshold logic.

Fuzzy logic :-> Data preprocessing

Optimization :-

Hybrid Techniques. (Combination of different techniques)

Supervised learning :-
→ not good in scalability.

Unsupervised learning :-
→ good at scalable
→ not good in performance vs. supervised.

Criticism

Soft computing (characteristics)

ANN
Simon Haykin
↳ chapter-1

B. Yegnarayana
↳ chapter-1

1 - 3
James A. Freeman
2
David SP

DATE
PAGE No.

9/12/2017 :- Prof. Banka (soft computing)

Threshold logic.

Fuzzy logic :-> Data preprocessing

Optimization :-

Hybrid Techniques. (Combination of different techniques)

Supervised learning :-
→ not good in scalability.

Unsupervised learning :-
→ good at scalable
→ not good in performance vs. supervised.

Criticism

Soft computing (characteristics)

ANN
Simon Haykin
↳ chapter-1

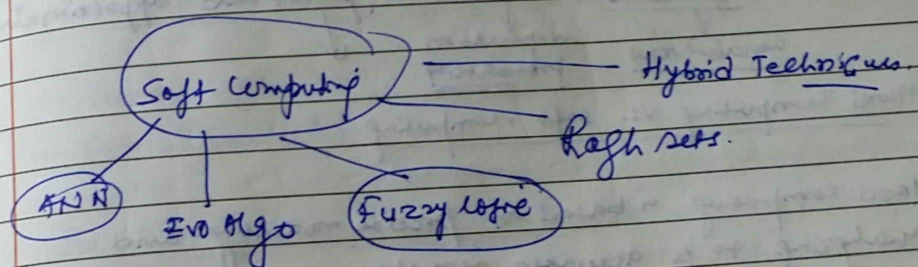
B. Yegnarayana
↳ chapter-1

1 - 3
James A. Freeman
2
David SP

Kreator

Evolution Algorithms

1. D.E. Goldberg. (1-3 chapters)
2. M.O.E.A (Multi objective evolutionary Algos).
   K. Deb (NSGA II)
   (chapter 1-6)



DATE
PAGE No.

Soft computing.

(1) Basic deterministic model for human mind.

(2) soft computing is foundation of ~~computational~~ computatal intelligence in machine.

(3) Tolerant of imprecision, uncertainties, partial and approximation.
unambiguous   definition probability.

Hard computing vs. soft computing

(1) Hard computing is based on precise modelling and analyzing to a accurate results.

(2) works wells for simple problem but bounded to NP-completeness.

P-class Problem:- Problem that is solved

NP-class problem   can solve in polynomial

Any of soft computing

(1) Subservient to hard computing.

(2)

(3) ~~refinements~~

(4)

DATE ______
PAGE No. ______

4. Difference b/w Hard computing and Soft computing

Unique features of Soft computing

Basic components of Soft computing

- Fuzzy logic
- Evaluation algorithm

Biological vs. conventional

1) speed
robustness
Flexibility / adapability.

ANN
:- definition by simon Haykin

Neuron (Biological)
ANN
Architecture
Training & weight / connections.
Testing / performance
XOR problem

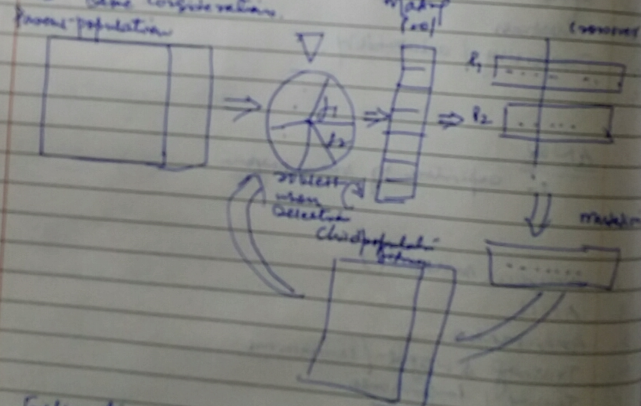
Boolean AND

x_1	x_2	
0	0	0
0	1	0
1	0	0
1	1	1

Two layer neural perceptron can't solve the XOR problem ?

Genetic Algorithm

1. encoded solⁿ (Chromozome)
2. Population
3. Genetic operators
4. Algo.
5. Basic consideration



Exploration & Exploitation Dilemma

Page No.
Date : 9/12/2017

Introduction :-

* Hard computing / conventional computing
* Soft computing : explorative search is used (not exhaustive).
  ↳ gives acceptable result / solution.

* Fuzzy logic : degree of membership (measured with value b/w 0 & 1).

* Squared Error : Difference between desired and actual result i.e.
  $(D-A)^2$
  for 1 result set : $(D_1 - A_1)^2$
  for 2 result set : $(D_1 - A_1)^2 + (D_2 - A_2)^2$

why squared error? → bcoz small error can be further reduced & higher error can be further amplified.
Also, squared error will take care of both +ve & -ve differences.

Learning →
1. Teacher
2. Learning Element / learning system.

Scalablitity :- If input increases & system can be balanced with visibility to user then, it is said to be scalable system.

Types of Learning →
1. Supervised learning 2. Unsupervised learning
3. critic(?)

Page No.
Date : / /

2. Hard comp works well for simple problems but bounded for NP completeness.

explanation →

* Ambiguous data situation or imprecise data situation i.e. data is not precise
* uncertainity → concerned with probability (may or may not happen)

$f(x_1, x_2, \ldots\ldots, x_n) = \ldots$

$x_i \in [0, 1]$

```
for x1 = 0, 1
    for x2 = 0, 1
        f = ........
```

] for 2 variables there are 2 nested loops

But a compiler can't allow many nested loops (e.g. C compiler will allow max 10 loops).

for 'n' variables

there will be 2^n → inputs

$$\begin{cases} 0 \quad 0 \quad 0 \ldots\ldots \quad 0 \\ \vdots \\ \vdots \\ 1 \quad 1 \quad 1 \ldots\ldots \quad 1 \end{cases}$$

$f_1 \quad f_2 \quad f_3 \ldots$

How many functions will be possible?

for 2^n inputs, there will be 2^{2^n} functions.

* Soft computing is well suited for real world problems.
* It can't give best results but it can give acceptable results.
* conventional paradigm shift
* If model is not present, then also soft computing can be u[sed] to provide solution

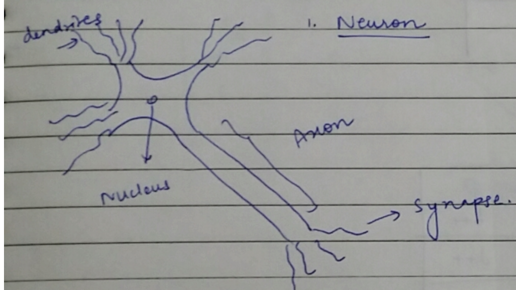
Page No.
Date : / /

Biological network is
* Adaptability → i.e. adaptaive & flexible
* contact sensitivity

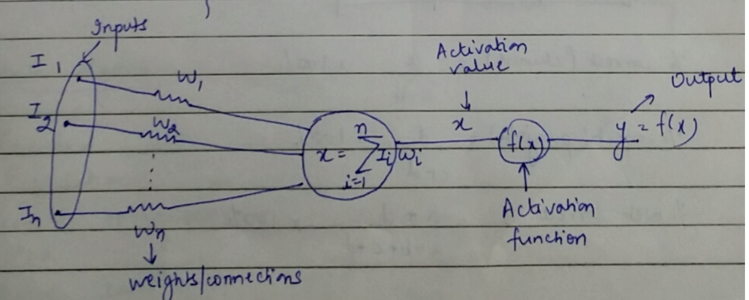
Artificial Neural Network →
highly parallel & distributive in nature having natural capability for storing potential knowledge.
It is similar to human brain in 2 aspects :-
1. knowledge is captured through learning process.

1. Neuron (Biological Network)
2. Artificial Neuron Network
3. Architecture Network
4. Training & weight / connections.
5. Testing / Performance
6. XOR Problem.

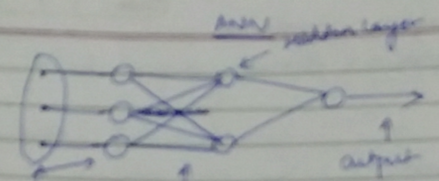
1. Neuron



2.



ANN



$W_{ij}(t+1) = W_{ij}(t) + \Delta_{ij} \cdot W(t)$ → change in weights (determined by learning algorithm)

$W_{ij}(t+1)$: next weight

t: step number (iteration)

$\Delta_{ij} W(t) = \eta \, \delta_j \, x_i$

$W_{ij}(t)$: current weight

eg - chinese or punjabi

	height	weight	complexion	class
1	h_1	w_1	c_1	
2	h_2	w_2	c_2	
⋮	⋮			
500	h_{500}	w_{500}	c_{500}	

→ 500 sample records "training set"

confusion matrix

	C	P
C	a	b
P	c	d

$\% \text{ correct (chinese)} = \frac{a}{a+b} \times 100\%$

$\% \text{ correct (punjabi)} = \frac{d}{c+d} \times 100\%$

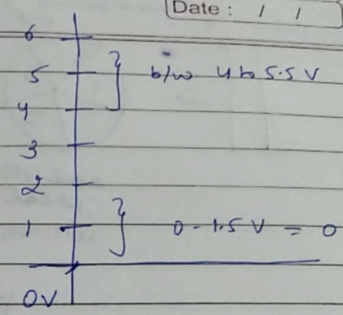
$\% \text{ net correct} = \frac{a+d}{a+b+c+d} \times 100\%$

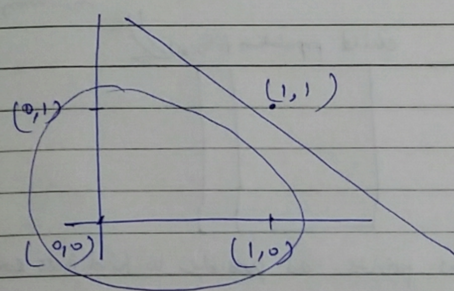
Page No.
Date : / /

Boolean AND

x_1	x_2	
0	0	0
0	1	0
1	0	0
1	1	1

↓

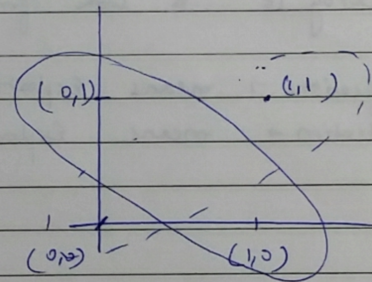




"decision boundary"
→ can be separated by a line.

XOR

x_1	x_2	
0	0	0
0	1	1
1	0	1
1	1	0

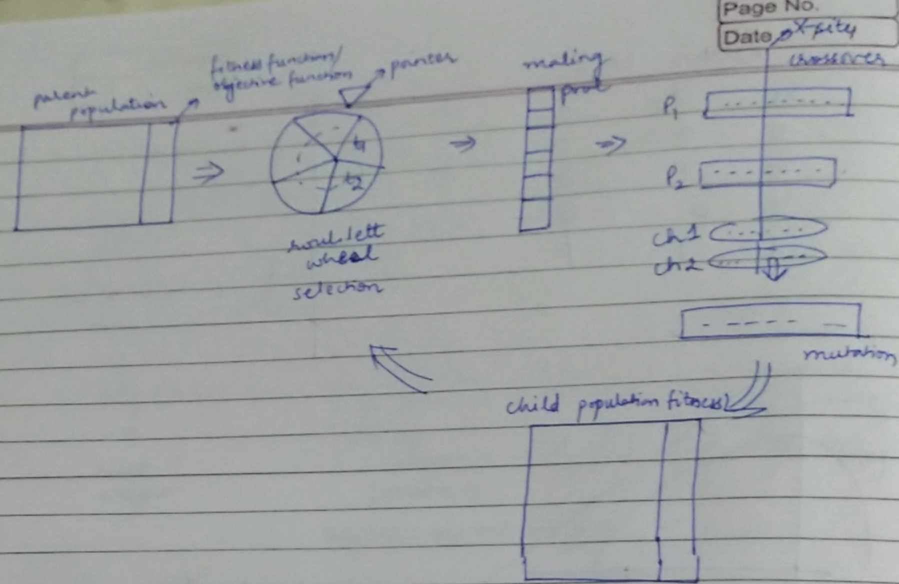


↓
can't be solved using linear boundary.

Genetic Algorithm

1. Encoded sol^n (chromosome)
2. Population chromosome.
3. Genetic operators
4. Algo.
5. Basic considerations

Page No.
Date



* Exploration → search as many new points as possible to find best solⁿ
* Exploitation →
  whatever best solⁿ we have so far, we search nearby it to look for any better solⁿ

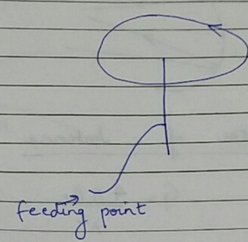
* Intensification → means Exploitation
* Diversification → means Exploration

(Follow slides provided by sir)

16/12/17

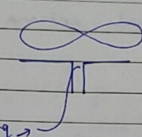
* **Antenna**

Omnidirectional antenna →



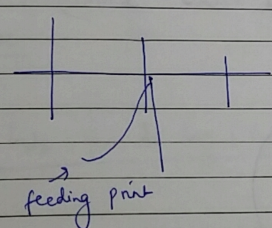
Directional :-

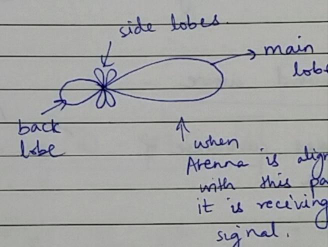
1. Dipole Antenna



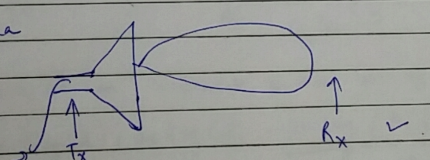
feeding point : from where we put/feed electrical wav
& from where we are collecting electrical signals from antenna.

2. Yagi Uda



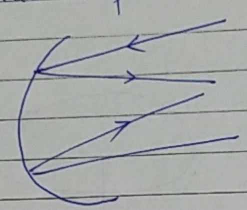


3. Horn Antenna



Page No.
Date : / /

4. Parabolic reflector (like Dish TV, Tata Sky)



Gain factor of Antenna

$$G = \frac{4 \pi A_e}{\lambda^2}$$

$\lambda = \frac{c}{f}$
↑
carrier wavelength

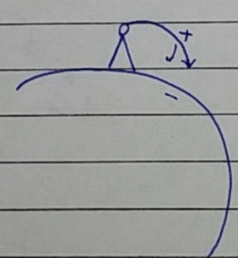
$G = 1$ for only omnidirectional antenna.

Propagation methods

- Ground wave Propagation (below 2 MHz)
- Sky wave " (b/w 2-30 MHz).
- Line of sight " (e.g. light wave bcoz its frequ is above 30 MHz)

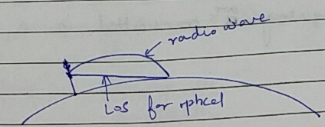
Ground wave :-

follows curvature of earth :-



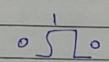
Page No.
Date : / /

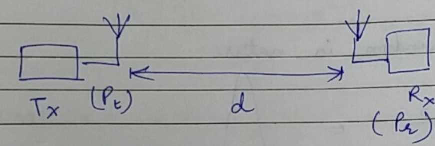
For optical wave :- LOS (Line of sight comm.)
For radio wave :- Reflection / refraction / diffraction anything may happen



Attenuation :-

* use amplifier to transmit signal appropriately.





free space loss $\rightarrow \frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2}$

$$\boxed{L_{dB} = 20 \log(f) + 20 \log(d) - 147.56 \text{ dB}}$$

$G_r = \frac{4\pi}{\lambda^2} \cdot A_e$ → effective Area

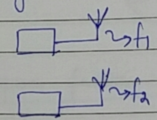
$\frac{P_t}{P_r} = \frac{(4\pi d)^2}{G_t G_r \lambda^2} = \frac{(\lambda d)^2}{A_t A_r}$

for directive antenna, we can compensate this loss by :-

↳ $L_{dB} = -20 \log(f) + 20 \log(d) \underbrace{-10 \log(A_t A_r)}_{\text{compensate}} + 169$

Noise

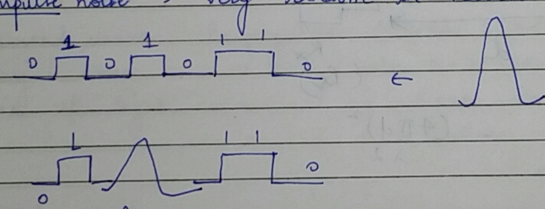
- Thermal noise
- Intermodulation noise :-
  signals with diff. frequency are transmitted in a shared medium



So, mixing of signals will produce energy at freq. $(f_1 \pm f_2)$ ~~or~~ their multiples i.e. $f_1 f_2$

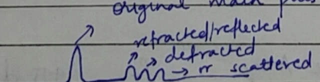
- Crosstalk noise
  ↳ unintentionally selecting f_2 freq. instead of f_1.

- Impulse noise → very random in nature



↑ Impulse noise has washed out these 3 bits.

Multipath



Fading channel

$K = 0 \rightarrow$ Rayleigh

$K = \infty \rightarrow$ AWGN

Page No.
Date : / /

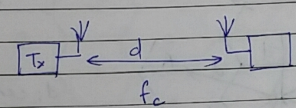
* Error correction
* Adaptive Equalization.

Delay = Transmission time + Propagation time + processing time + Queuing time.

Transmission time = $\frac{\text{Message size}}{\text{Bandwidth}}$

Propagation time = $\frac{\text{Distance}}{\text{Propagation speed}}$

Energy consumption = $P_t \times \frac{\text{No. of bits transmitted}}{\text{Bit Rate}}$



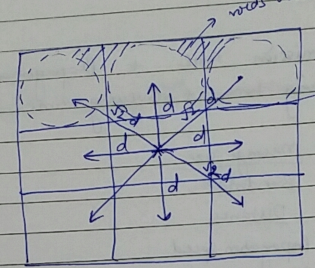
TDM → f is fixed
FDM → t is fixed.

(Numericals → photos).

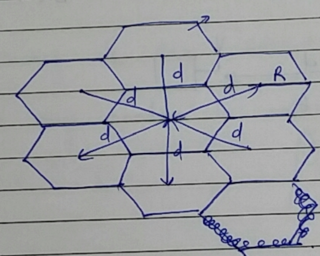
Page No.
Date : / /

17/12/2017

CELLULAR NETWORKS



BTS (Best Transreceiving station)



d: centre to centre distance

R: Radius.

$d = \sqrt{3}R$

D = min. distance b/w centre of cells that use same frequency band called co-channels.

R = Radius of a cell

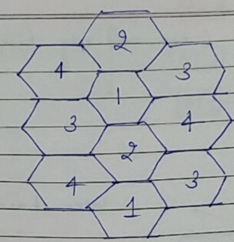
d = distance b/w centres of adjacent cell. $= \sqrt{3}R$

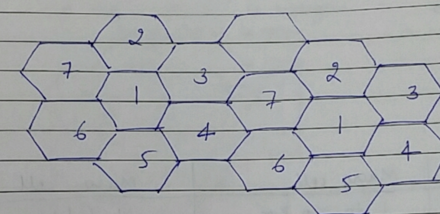
N = No. of cells in a repeater's pattern (each cell in pattern uses a unique set of frequency bands) termed the reuse factor).

$N = I^2 + J^2 + (I \times J)$ where $I, J = 0, 1, 2, \ldots$

$\therefore N = 1, 3, 4, 7, 9, \ldots$

(Graph coloring problems)





↑

7 cell structure.

$$\frac{D}{R} = \sqrt{3N} \qquad (\text{bcoz } d = \sqrt{3}R)$$

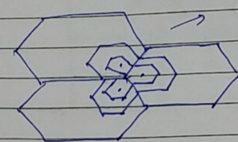
$$\therefore \boxed{\frac{D}{d} = \sqrt{N}}$$

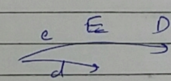
Q. $K = 395$, $N = 7$. what is max. freq. which can be achiev[ed] for individual cells?

$$f = \frac{K}{N} = \frac{395}{7} \approx 57$$

1. Adding new channels. (can add only finite no. of channels, not possible everytime)
2. Frequency borrowing
3. cell splitting.

inter-channel interference.

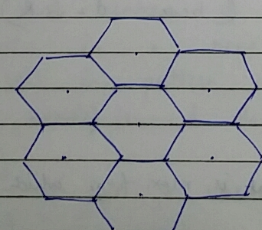




E>e.

* ~~macro cell~~

	Macro cell	Micro cell.
cell radius	1 to 20 km	0.1 to 1 km
Transmission Power	1 to 10 W	0.1 to 1 W.
Average delay	1 to 10 μs	10 to 100 ~~ns~~.
Bit Rate	0.3 Mps	1 Mbps

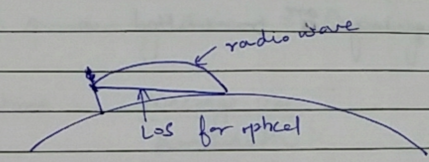


M
T
S
O

Mobile Telecommunication Switching office.

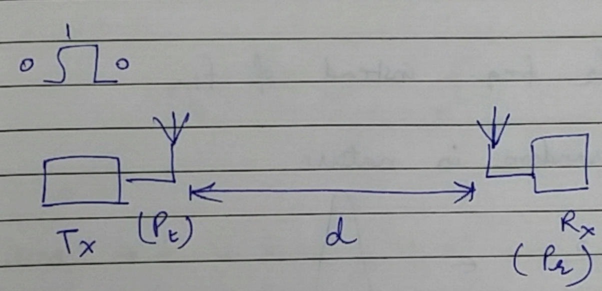
Page No.
Date : / /

For optical wave :- LOS (line of sight comm.)
For radio wave :- Reflection / refraction / diffraction anything may happen



Attenuation :-

* use amplifier to transmit signal appropriately.



Free space Loss, $\frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2}$

$$\boxed{L_{dB} = 20\log(f) + 20\log(d) - 147.56 \text{ dB}}$$

$G = \frac{4\pi}{\lambda^2} \cdot A_e \rightarrow$ effective Area

$\frac{P_t}{P_r} = \frac{(4\pi d)^2}{G_t G_r \lambda^2} = \frac{(\lambda d)}{A_t}$

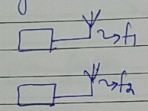
For directive antenna, we can compensate ⇓ this loss by :-

↳ $L_{dB} = -20\log(f) + 20\log(d) \underbrace{-10\log(A_t A_r)}_{\text{compensate}} + 16$

Noise

• Thermal noise

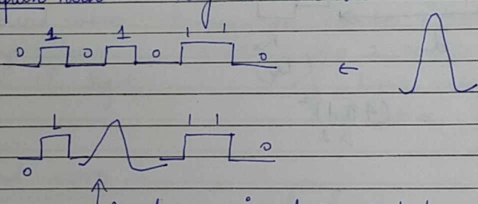
• Intermodulation noise :-
signals with diff. frequency are transmitted in a shared medium.



So, mixing of signals will produce energy at freq. $(f_1 + f_2)$ or their multiples i.e. $f_1 f_2$

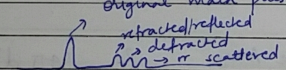
• Crosstalk noise
↳ unintentionally selecting f_2 freq. instead of f_1.

• Impulse noise → very random in nature



↑ Impulse noise has washed out these 3 bits.

Multipath



Fading channel

$K = 0 \rightarrow$ Rayleigh

$K = \infty \rightarrow$ AWGN

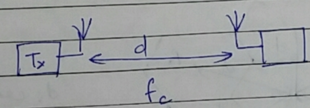
* Error correction
* Adaptive Equalization.

Delay = Transmission time + Propagation time + processing time + Queuing time.

Transmission time = $\frac{\text{Message size}}{\text{Bandwidth}}$

Propagation time = $\frac{\text{Distance}}{\text{Propagation speed}}$

Energy consumption = $P_t \times \frac{\text{No. of bits transmitted}}{\text{Bit Rate}}$

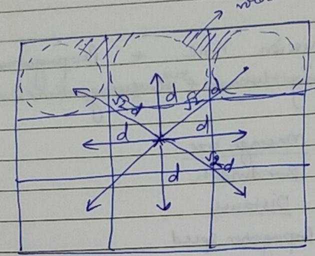


TDM → f is fixed
FDM → t is fixed.

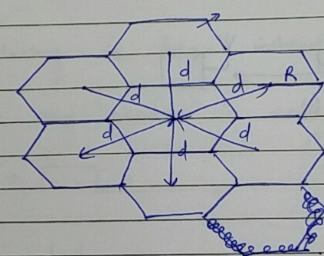
(Numericals → photos).

17/12/2017

CELLULAR NETWORKS



BTS (Best Transreceiving station)



d: centre to centre distance

R: Radius.

$d = \sqrt{3} R$

D = min. distance b/w centre of cells that use same frequency band, called co-channels.

R = Radius of a cell

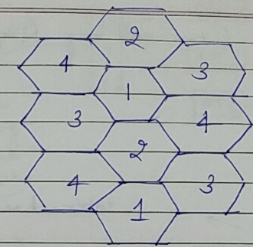
d = distance b/w centres of adjacent cell. $= \sqrt{3} R$

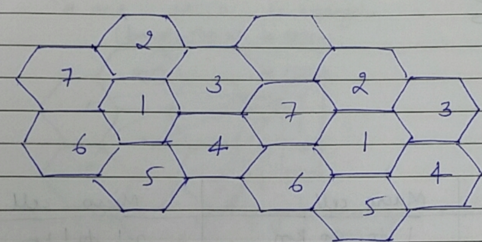
N = No. of cells in a repeater's pattern (each cell in pattern uses a unique set of frequency bands), termed the <u>reuse factor</u>).

$N = I^2 + J^2 + (I \times J)$ where $I, J = 0, 1, 2, \ldots$

$N = 1, 3, 4, 7, 9, \ldots$

(Graph coloring problems) ?





7 cell structure.

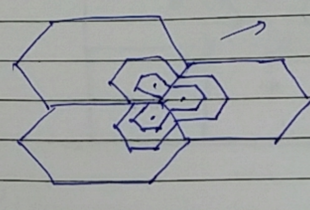
$\frac{D}{R} = \sqrt{3}\, N$ (bcoz $d = \sqrt{3}R$)

$\therefore \boxed{\frac{D}{d} = \sqrt{N}}$

Q. $K = 395$, $N = 7$. what is max. freq. which can be achieved for individual cells?

$f = \frac{K}{N} = \frac{395}{7} \approx 57$

1. Adding new channels. (can add only finite no. of channels, not possible everytime)
2. Frequency borrowing
3. cell splitting.



inter-channel interference.

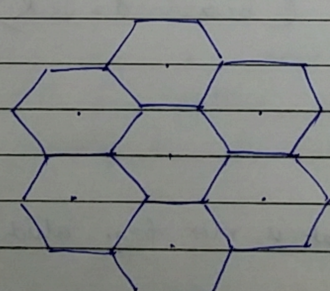
e E D

d

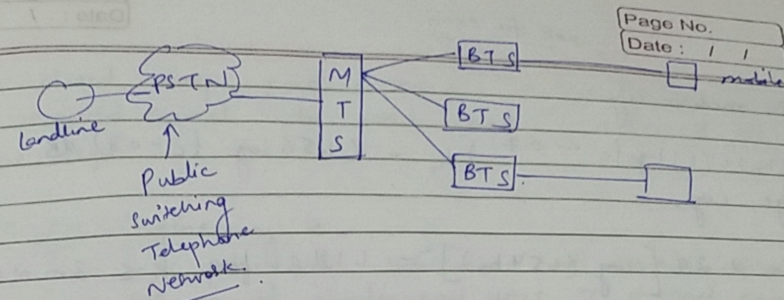
E>e.

* ~~macro cell~~

	Macro cell	Micro cell.
cell radius	1 to 20 km	0.1 to 1 km
Transmission Power	1 to 10 W	0.1 to 1W.
Average delay	1 to 10 μs	10 to 100 ns.
Bit Rate	0.3 Mps	1Mbps



M Mobile
T Telecommunication
S Switching
O office.

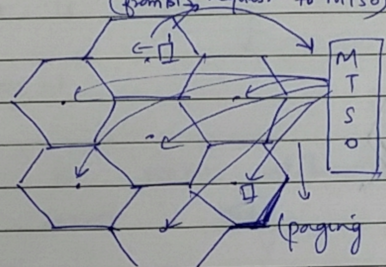


1. mobile → BTS → MTS → BTS → mobile. (call to)
2. Landline → BTS → MTS → PSTN → Landline.

for local calls :-

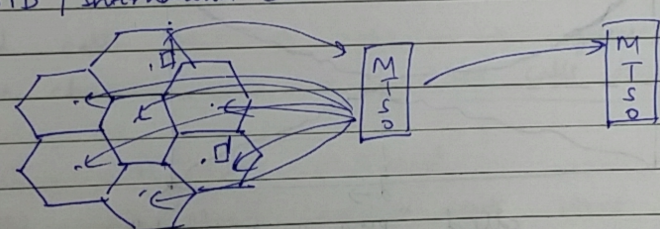
* one MTSO is enough.

(from BTS, request to MTSO)



(paging message is broadcasted to a BTS)
to search for subscriber

for STD / International calls :



{for Information only → not for Exam:-

Hata Model. 1968 → 1980.

$L_{dB} = 69.55 + 26.16 \log f_c - 13.82 \log h_t - A(h_r) + (44.9 - 6.55$

Page No.
Date : / /

* for closed loop power control:- signal power level, received signal to noise ratio, received bit rate are reqd.

Traffic Engineering ⇒

L = potential subscriber or mobile units.

N = no. of simultaneous users or channel capacity.

λ = min. rate of ~~~~ calls ~~~~ attended per unit time.

h = min. holding time per successful call.

$A = \lambda h$
↑
"Traffic intensity"

unit is "erlang"

$A = \lambda \cancel{h} = \rho N$

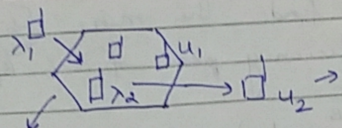
where ρ : is server utilization or fraction of time server is busy.

1. LCD : Lost Calls Delayed
Blocked calls can be put in a queue waiting for a free channel is called "lost calls delayed".

2. LCC : lost calls cleared
If the user hangs up & waits some random time interval before ~~~~ another call attempt this is known as "lost calls cleared."

3. LCH : Lost calls Held
If the user repeatedly attempts calling, it is known as "Lost calls Held"

λ = ...
h = The mean holding time per ...



$\rightarrow$ death rate of cell

$\lambda \rightarrow$ Birth rate of cell

Note : $\Sigma \lambda_i \Leftrightarrow \Sigma \mu_i$ (bcoz no. of channel is fixed) equal

Q.1. At room temperature, what is thermal noise power density?

Ans1. $N_0 = K \times T \rightarrow$ Temperature
$\downarrow$
Boltzman constant

Room temp = $T = 17^\circ C$ = 290 K

$N_0 = 1.38 \times 10^{-23} \times 290$ W/Hz $10\log_{10} N_0 \Rightarrow$ units (dB)

Bandwidth
$B = 10$ MHz

$N_0 = KTB = 1.38 \times 10^{-23} \times 290 \times 10$

Q2. Suppose a signal encoding technique requires that $E_b/N_0 = 8.4$ dB for a bit error rate of 10^{-4}. If the effective noise temp. is 290 K & data rate is 2400 Bps. what received signal level is reqd. to overcome thermal noise?

Take, $10\log_{10}$ on RHS ~~both sides~~ to convert to dB

$$\frac{E_b}{N_0} = \frac{S}{KTR}$$

$\Rightarrow 8.4 = 10\log_{10} S + 10\log_{10} K - 10\log_{10} T - 10\log_{10}$

$8.4 = 10\log_{10} S - 10\log_{10}(1.38 \times 10^{-23}) - 10\log_{10}(290)$ ~~$- 10\log_{10}($~~

Date 21/1/18

Wireless LAN -

1. limited by signal range.
2. ISM band
   (Industry Scientific Medicine)
3. 802.11 (WLAN)
4. 802.16 (WMAN)
5. 802.15 (Personal area N/W, Bluetooth)

SDM - (Space division Multiplex.)

2. Modes of operations.

→ ① presence of controlled Module (CM) BS → AP 802.1
   (Control mobile station

→

→ ② Adhoc commⁿ - Peer to Peer communicati
   where there is no control module (CM)

Applications -
Ex - LAN extension
   cross building interconnection

⇒ 1) WLAN Requirement (high through
   Good use of bandwidth

2. Number of nodes – May be large – 100's

3. good connection to LAN background N/W

4. good service (Range)

5. Minimum battery power consumption.

6. Transmision security & robustness

7. Coallocated N/W operation

8. lisence free operation (2.4 GHz)

9. Cell handoff / N/W roaming
   - dynamic management
   - adaptive MAC
   - address management
   - automatic addition.
     deletion
     relocation

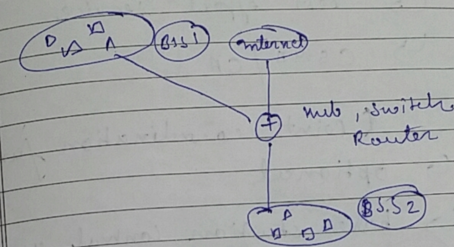
10. choice of physical solution:

(1997 . 802.11) 1 or 2 Mbps
. FHSS
. DSSS

2000 – 802.11 b
HR – DSSS – 11 Mbps

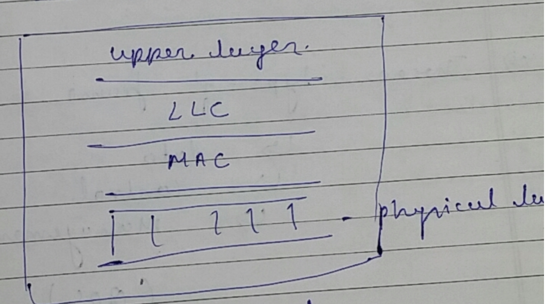
2001
802.11 g,
OFDM, 54 Mbps.

⇒



basic service set
(wireless host & access point

Simple LAN architecture.



802.11 access control
→ avoid collision.
CSMA
→ No collision detection.

⇒ 802.11 protocol Architecture –
Two modes

1) DCF (~~Manditatory~~) (mandatory)
(distributed co-ordinate function)

↳ best effort service

↳ CSMA / CA

2) PCF (point co-ordination function)

↳ optional

↳ base station control users to the medium using polling mechanism with high priority to the users.

Three type of frame

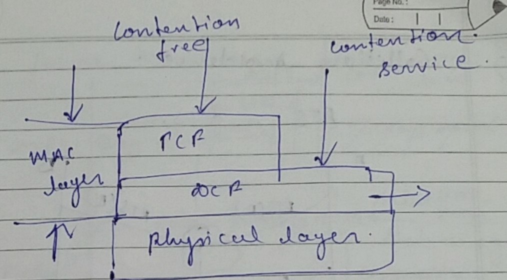
↳ data

↳ control

↳ Management (provide services, priority)

(QOS)
↓
Quality of service.

Page No.:
Date:

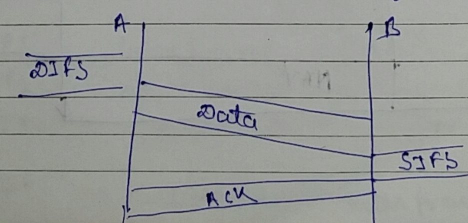


⇒ 802.11 sender

↳ if sends the channel if it is idle atleast for (DIFS) then
transmit the entire frame (no co)
(with contention service)

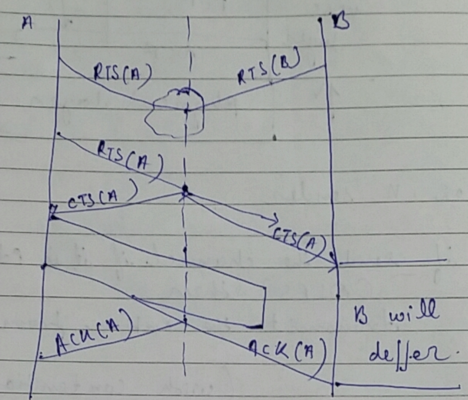
↳ if sends the channel busy then start random back off time while channe
transmit when time expires, if no ACK, ~~incres~~ increas random back off interval repeat 2.

⇒ 802.11 receive
- if frame received OK
return ACK after SIFS.



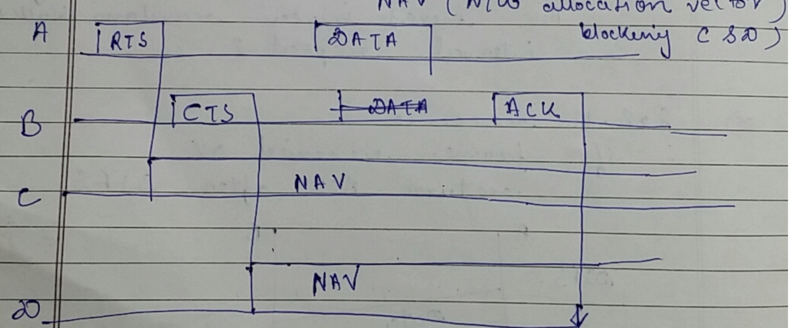
⇒ Collision Avoidence.

RTS - CTS.

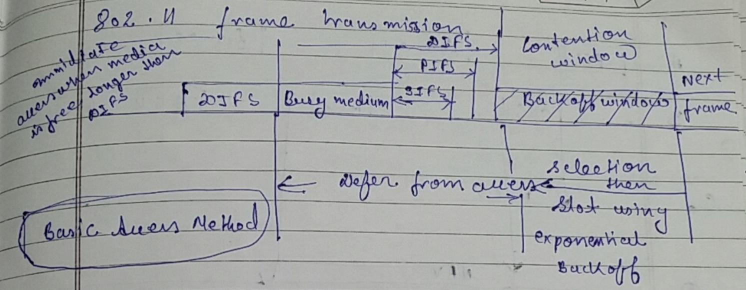


⇒ the use of virtual channel sensing using CSMA/CA

NAV (N/W allocation vector)
blocking CSD)

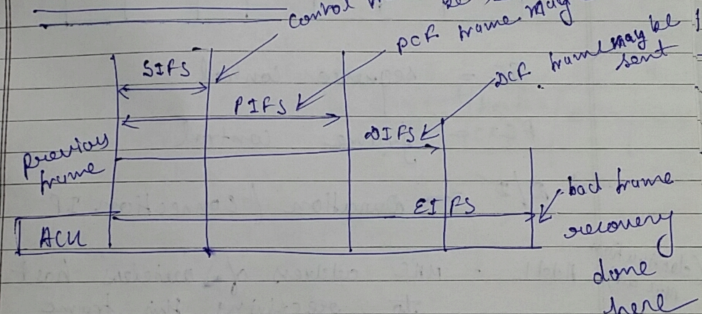


802.11 frame transmission



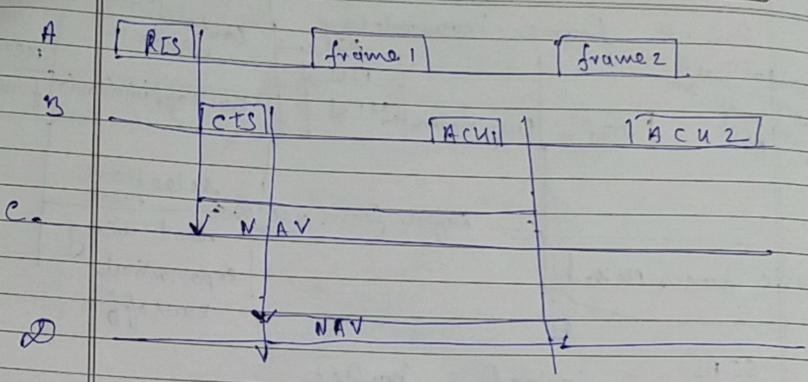
SIFS — control traffic
PIFS — optional traffic
DIFS — general traffic

⇒ IFS 802.11



IFS = interframe space.
DIFS = distributed interframe space
PIFS = point co-ordination function interframe space
EIFS = Extended interframe space.

when large frame divided into multiple frame



802.11 MAC frame fregment

2	2	6	6	6	2	6	0.2312	4
FC	D/I	Add1	Add2	Add3	SC	Add4	FB	CRC

SC = sequence control

FC = frame control

D/I = Duration / Connection IP

(destination Mac add) Add1 = MAC address of wireless host or AP to receiving this frame

(Source Mac add) Add 2 = Mac address of wireless host or AP transmiting this frame

Add 3 = Mac address of router interface to which access point is attached.

Add 4 = used only in Adhoc mode.

⇒ reliable = automatic ~~sequence~~ resequence of re-transm^n ?
of ARQ

* 802.16

frame-format

a) data frame — QAM - 16, QAM - 64

b) control — QPSK

(bandwidth request frame)

a) data frame

1	1	6	1	1	2	11	16	8	11	4
0	E O	type		C I	ER	length	Content ID	Header CRC	data	CRC

control frame

1	1	6	11	16	8
1	0	type	Byte	Connection ID	Header CRC

Connection ID = binds the end points through the system

Bits

EO - Encrypted payload

CI - check sum input

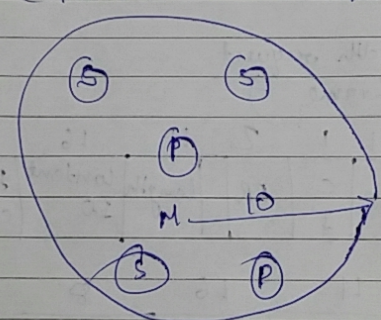
EK - Encryption technique.

Page No.:
Date:

⇒ Service classes

- Content bit rate service (voice)
- real time variable bit rate service (video)
- Non real time variable bit rate service (high quality data)
- Best effort service (data, http etc)

⇒ 802.15
PAN (personal area N/W)



Page No.
Date : / /

Solⁿ → use extra register.

value of R_1 will be available after 5th cycle.

* Data Hazard

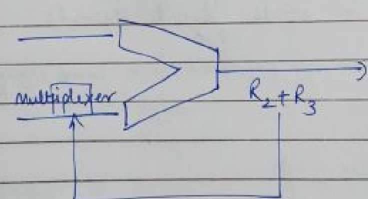
Ex :-

					1	2	3	4	5	6
DADD	R_1	R_2	R_3	-	IF	ID	EX	MEM	WB	
DSUB	R_4	R_1	R_5			IF	stall	stall	stall	ID
AND	R_6	R_1	R_7							
OR	R_8	R_1	R_9							
XOR	R_{10}	R_1	R_{11}							

Data Hazard is that value of R_1 is available after 5th cycle but in subtract operation it uses "old" value of R_1 in the 4th cycle to execute subtract instruction. This is incorrect.
∴ It is a data Hazard.

Solution :- Data Forwarding

DADD	R_1	R_2	R_3	IF	ID	EX	MEM	WB	
DSUB	R_4	R_1	R_5		IF	ID	EX		
AND						IF	ID	EX	
OR									
XOR									



Computation value of R_2+R_3 is forwarded for execute stage of DSUB.

Page No.
Date: / /

Speedup pipelining $= \dfrac{\text{A.Ex. Time unpipelined}}{\text{A.Ex. Time pipelining}}$

$= \dfrac{IC_{\text{unpipelined}}}{IC_{\text{pipelined}}} \times \dfrac{CPI_{\text{unpipelined}}}{CPI_{\text{pipelined}}} \times \dfrac{CCT_{\text{unpipelined}}}{CCT_{\text{pipelined}}}$

$= \dfrac{CPI_{\text{unpipelined}}}{CPI_{\text{pipelined}} + ?} = \dfrac{CPI_{\text{unpipelined}}}{1 + ?} \simeq \dfrac{\text{pipeline depth}}{1 + (?)^*}$

with overhead — with overhead

↳ called "hazards"

$CPI_{\text{pipeline with hazards}} > 1$.

Types of Hazards →

- Structural Hazards
- Data Hazard
- Control Hazard

speed up pipelining with Hazards $= \dfrac{\text{pipeline depth}}{1 + \text{stalls due to Hazards}}$ $(?)^*$

Structural Hazards

IF & MEM are performed with same register

	1cc	2cc	3cc	4cc	5cc	6cc	7cc	8cc
I_1	IF	ID	EX	MEM	WB			
I_2		IF	ID	EX	MEM	WB		
I_3			IF	ID	EX	MEM	WB	
I_4				~~stall~~	~~IF~~ stall	~~stall~~	IF	

not allowed to use same register for 2 operations in same clock cycle

so, I_4 is stalled

8 cc without structural Hazard → 11cc with structural Hazard

Ans. 4.

$$\text{CPU time} = \sum_{i=1}^{n} IC_i \times CPI_i \times CCT$$

$n = 3$

$1 \rightarrow 50\%$ $\quad CPI_1 = 3$
$2 \rightarrow 30\%$ $\quad CPI_2 = 4$
$3 \rightarrow 20\%$ $\quad CPI_3 = 5$

$$\therefore \text{CPU time} = IC \times CPI_{overall} \times CCT$$

$$= 10^9 \times CPI_{overall} \times \frac{1}{2\,GHz}$$

$$= 10^9 \times CPI_{overall} \times 0.5 \times 10^{-9}$$

$$CPI_{overall} = 0.5 \times 3 + 0.3 \times 4 + 0.2 \times 5$$
$$= 3.7$$

$$\text{CPU time} = 10^9 \times 3.7 \times 0.5 \times 10^{-9}$$

$$\frac{\text{CPU Time}_{old}}{\text{CPU Time}_{new}} = \frac{\cancel{IC_{old}}}{\cancel{IC_{New}}} \times \frac{CPI_{old}}{CPI_{new}} \times \frac{\cancel{CCT_{old}}}{\cancel{CCT_{new}}}$$

(If same program)

e.g.

Raw material

$5 \times 100 = 500$ $\qquad CPI_{old} = \frac{500}{100} = 5$

cars

Pipeline fashion $= 104$ $\qquad CPI_{new} = \frac{104}{100} = 1.04 \approx 1$

Page No.
Date : / /

$$\frac{P_{new}}{P_{old}} = \frac{\text{CPU Time}_{old}}{\text{CPU time}_{new}} = \frac{IC_{old} \times CPI_{old} \times CCT_{old}}{IC_{new}\, CPI_{new}\; CCT_{new}}$$

$= 5$

Power v/s Energy

Power → watt

Energy → Joules.

$P = DP + LP$

DP → Dynamic Power

LP → Leakage Power

$DP \propto \text{activity} \times \text{capacitance} \times (\text{Voltage})^2 \times \text{frequency}$

$LP = f(V, I)$ — function of; V → voltage, I → current

→ $L.P. = 25\% \times$ Total Power

$LP \propto V$.

Ex :- A program runs for 3 seconds on 100 watt processor.

$E = P \times \text{Time}$

E → Energy consumption, P → Power

$= 100 \times 3 = \underline{300}$ Joules.

How to reduce DP, LP & Energy ?

There are 2 techniques :-

1. DFS (Dynamic Frequency Scaling).

Load store Architecture

$x = a + b$

Load ↳ LD R_1 $0(R_x)$ ← from this address.
Load ↳ LD R_2 $0(R_y)$
Add → ADD R_0 R_1 R_2
store → SD R_0 $0(R_z)$

Car Anology

stage1 stage2

stage1	stage2			
C_1				
C_2	C_1			
C_3	C_2	C_1		

Latency for each car = 5
but throughput is increasing
i.e. more no. of cars (instructions)
per second.

MIPS

↳ Multiprocessor without interlocked pipeline stages

- Five stages :-
  - IF (Instruction fetch)
  - ID (Instruction Decode)
  - EX (Execute)
  - MEM (Memory)
  - WB (Write Back)

* Processor Performance Equation

$$\text{CPU time} = \left(\text{CPU clock cycles of a program}\right) \times \left(\text{clock cycle time}\right)$$

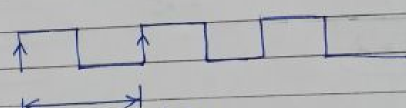
Q.3.

$$\underset{\substack{\uparrow \\ \text{clocks per Instruction}}}{CPI} = \frac{\text{CPU clock cycle of a program}}{\text{Instruction count (I.C.)}}$$

Ans

$$\text{CPU clock cycle of a program} = I.C. \times CPI$$

$$\therefore \boxed{\text{CPU Time} = I.C. \times CPI \times CCT} \text{ ns.}$$



$1\text{ns} = \underbrace{1\text{GHz}}_{\text{frequency}}$

$$T \propto \frac{1}{F}$$

Q.4. Suppose a ~~processor~~ program takes 1 billion instructions to execute on a processor running at 2 GHz and 50% of instructions whose CPI is 3 clock cycles per instruction and 30% of instruction is 4 clock cycles per instruction & 20% of instruction is 5 clock cycles per instruction. What is the CPU Time?

① DF Scaling

$DP \propto V^2 f$

Example

program X

	P	T
P_{old}	100 W	100 secs.

$= E_{old} = P \times t = 100 \times 100$
$= 10{,}000$

$100 \text{ secs.} \xrightarrow{f = 0.5\,GHz} 200 \text{ secs.}$

$f = 1\,GHz$
$f = 0.5\,GHz$

100 W → 75 W DP, 25 LP

$75\,W\ DP \xrightarrow{f = 0.5\,GHz} 37.5 \text{ watts}$

$25\,LP \xrightarrow{\text{watts}} 25\,LP$

$E = P \times t$

new $62.5 \times 200 = 12500$

P_{new}

E_{new} 12500
E_{old} 10000

$E_{new} = 1.25 \times E_{old}$

DVFS

	P	T
P_{old}	100 W	× 100 sec

$= 10000$

100 sec → 110 sec

100 W → DP 75 W, LP 25 W

$E_{old} = 10{,}000$
$E_{new} = 8470$
$E_{new} = 0.84 \times E_{old}$

$75\,W \xrightarrow{V = 0.9,\ = 1.1\,ns} 75 \times (0.9)^2 \times (0.9)$
$= 55\,W$

$25\,W \rightarrow 25 \times 0.9$
$= 22.5$

$P_{new} = 77\,W$

$E = P \times t$
$= 77 \times 110$
$= 8470$

Page No.
Date : / /

CPU Time = IC × CPI × CCT.

IC → no. of instructions in a program.

Lower IC, Lower CPU Time.

IC → job of compiler or user to minimize the IC.

CPI → no. of clocks per instruction.
this is ~~done~~ minimized by Computer Architecture

CCT → by H/W Techology.

$$\text{CPU Time} = \frac{\overset{\alpha}{\text{Instructions}}}{\text{program}} \times \frac{\overset{\beta}{\text{clock cycles}}}{\text{Instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

$$= \frac{\text{seconds}}{\text{program}}$$

Different types of Instructions :-

type - i
$i+1$
$i+2$
⋮
n

we have 'n' types of instructions.

$$\text{CPU clock cycles} = \sum_{i=1}^{n} I.C_i \times CPI_i$$

$$\text{CPU Time} = \sum_{i=1}^{n} IC_i \times CPI_i \times CCT$$

$$\text{overall CPI} = \frac{\sum IC_i \times CPI_i}{IC} = \sum_{i=1}^{n} \frac{IC_i}{IC} \times CPI_i$$

Page No.
Date: / /

Example:-

Program   X   P   T $\Rightarrow$

P = 100 W, T = 100 secs.

$E_{old} = P \times t = 100 \times 100 = 10000$

100 W $\to$ 75 W (DP), 25 W (LP)

$DP\downarrow \propto V^2 \times f\downarrow$

If we change frequency,

$f = 1\,GHz \to 0.5\,GHz$

$DP = 75\,W \to 37.5$ watts.

$LP = 25\,W \to 25\,W$ (no change in LP)

$T = 100\,secs \to 200\,secs.$

$P = 37.5 + 25 = 62.5$

$E_{new} = P \times t = 62.5 \times 200 = 12500$

$E_{new} = 12500$

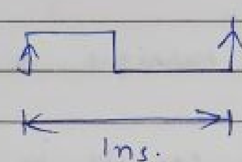
$E_{old} = 10000$

$\Rightarrow \boxed{E_{new} = 1.25 \times E_{old}}$

Page No.
Date : / /

2. D V F S
(Dynamic Voltage Frequency Scaling)

P	T
100 W	100 sec

1V = 

0.9 V = 

If voltage decreases, clock cycle-time increases.

P: 100 W → DP: 75W, LP: 25 W

T: 100 sec → 110 sec

$E_{old} = P \times T$
$= 100 \times 100 = 10000$

25 W → $25 \times 0.9 = 22.5\,W$

75W ↓
$V = 0.9$
$f = 1.1\,ns$
$= \frac{1}{1.1 \times 10^{-9}} = 0.9 \times 10^{9}\,Hz = 0.9\,GHz$

$75 \times (0.9)^2 \times 0.9$
$= 55W$

$P_{new} = 77.5W \approx 77$

$E_{new} = P \times t = 77 \times 110 = 8470.$

Page No.
Date : / /

4. Memory operation :- loading & storing takes place.

LD R_0 $0(R_x)$

SD R_0 $0(R_y)$

5. WB (write Back) :-

Addition is performed in 3^{rd} stage but its value is reflected only after 5^{th} stage.

ADD R_0 $R_1 + R_2$

fetch using PC

ADD R_0 $R_1 + R_2$

ADD → arithmetic; R_0 → Destination; R_1, R_2 → Source

In ID → we come to know source & dest. registers & that it is arithmetic operation

1cc	2cc	3cc	4cc	5cc
IF	ID	EX	MEM	WB.

It takes 5 clock cycles.

eg :-

LD R_0 $0(R_x)$

Load

↓

It takes 4 clock cycles.

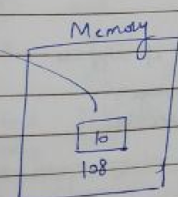
1cc	2cc	3cc	4cc
IF	ID	EX	MEM

EX ↓ $(0+R_x)$ address is found

LD R_0 $0(R_x)$

$0 + 108$

$= 108.$



SD R_0 $0(R_y)$ → also takes 4 cc's.
Store

Ex :- BNE Z

1 cc	2 cc
IF	ID

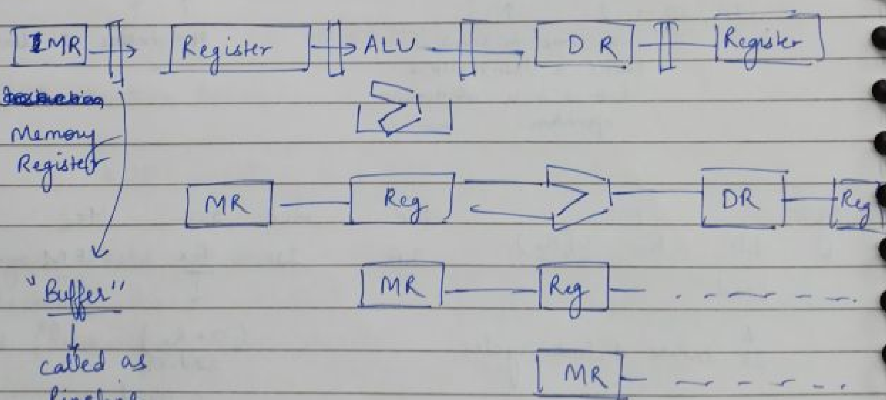
↓
Branch instruction with Jump or No Jump

↑
we require 2 clock cycles to detect whether it is a branch instruction with jump or no jump.

if(x = =0) go to L
- - - -
- - - -
L: - - - -

1cc	2cc	3cc	4cc	5cc
IF	ID	EX	MEM	WB

1. fetch PC →



~~Instruction~~ Memory Register

2.

3. "Buffer"
↓
called as Pipeline Registers.

4.

Buffer is required bcoz if suppose instruction come out of IF stage & ID is busy, then it can be stored in buffer.

↑
at this clockcycle 4 diff. instructions are at 4 diff. stages. This is called ILP.

Page No.
Date : / /

$$\text{Speedup pipelining} = \frac{\text{A. Ex. Time unpipelined}}{\text{A. Ex. Time pipelining}}$$

$$= \frac{\cancel{IC}_{unpipelined}}{\cancel{IC}\ pipelined} \times \frac{CPI_{unpipelined}}{CPI\ pipelined} \times \frac{\cancel{CCT}_{unpipelined}}{\cancel{CCT}\ pipelined}$$

$$= \frac{CPI_{unpipelined}}{CPI_{pipelined} + ?} = \frac{CPI_{unpipelined}}{1 + ?} \cong \frac{\text{pipeline depth}}{1 + (?)^*}$$

with overhead ↳ called "hazards"

with overhead

$CPI_{pipeline}$ with hazards > 1.

Types of Hazards →

- Structural Hazards
- Data Hazard
- Control Hazard.

$$\text{speed up pipelining with Hazards} = \frac{\text{pipeline depth}}{1 + \underbrace{\text{stalls due to Hazards}}_{(?)^*}}$$

Structural Hazards

IF & MEM opr. performed with same register R.

	1cc	2cc	3cc	4cc	5cc	6cc	7cc
i_1	IF	ID	EX	MEM	WB		
i_2		IF	ID	EX	MEM	WB	
i_3			IF	ID	EX	MEM	WB.
i_4				~~IF~~ stall	~~IF~~ stall	~~IF~~ stall	IF

not allowed to use same register for 2 operations in same clock cycle. so, it is stalled.

8 cc without structural Hazard → 11 cc with structural Hazard.

Date : / /

$E_{old} = 10000$

$E_{new} = 8470$

$E_{new} = 0.84 \times E_{old}.$

07/01/2018

Instruction Level Parallelism (ILP)

- Pipelining (best technique to achieve ILP)

Allow instructions to execute in overlap manner.

Instructions

Ex.

ADD R_0 R_1 + R_2

R_0, R_1, R_2 are registers. on which ADD operation is performed. sum of R_1 & R_2 is stored in R_0.

RISC

Full form:-

Instructions

Two ways

		called	example:-	
32-bit Instruction	⇒	word	⇒ ADD	R_0 R_1 + R_2
64-bit Instruction	⇒	double	DADD (double add)	R_0 R_1 R_2

direct address, indirect address, register address ⇒ refer to these diff. addressing modes from book.

Page No.
Date : / /

	clock cycle 1cc	2cc	3cc	4cc	5cc	6cc	7cc
PC = i	IF	ID	Ex	MEM	WB		
PC=PC+4 i+1		IF	ID	Ex	MEM	WB	
PC=PC+4 i+2			IF	ID	Ex	MEM	WB
PC=PC+4 i+3							

In every clock cycle, one instruction enters a pipeline.
In every clock cycle, one instruction will get completed.

* The objective of pipeline is $\boxed{CPI = 1}$

~~Instruction fetch:~~
1. IF: Job is to fetch instruction by using program counter (PC).

$PC = PC + 4$
↑
4 bytes i.e. 32 bits (1 word).

2. ID (Instruction Decode)

ex:- ADD R_0 R_1 R_2

ID stage checks whether values are present or not.
It decodes the instruction that what type of instruction it is (eg:- arithmetic instruction etc.)

3. Ex (Execution): performing integer operations or floating point operations.

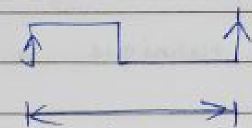
ADD R_0 $R_1 \oplus R_2$
↓
ALU (Arithmetic Logical Unit)
FU

Page No.
Date : / /

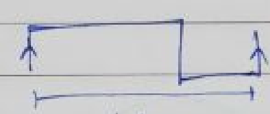
2. D V F S

(Dynamic Voltage Frequency Scaling)

P	T
100 W	100 sec

1V = 

1ns.

0.9V = 

1.1ns

If voltage decreases, clock cycle-time increases.

P

100 W

$$\begin{array}{ccc} & 100\,W & \\ \swarrow_{DP} & & \searrow^{LP} \\ 75\,W & & 25\,W \end{array}$$

T 100 sec $\downarrow$ 110 sec.

$E_{old} = P \times T$

$= 100 \times 100 = 10000$

$25\,W \rightarrow 25 \times 0.9 = 22.5\,W$

$75\,W \downarrow$

$V = 0.9$

$f = 1.1\,ns$

$= \frac{1}{1.1 \times 10^{-9}} = 0.9 \times 10^{9}\,Hz = 0.9\,GHz$

$75 \times (0.9)^2 \times 0.9$

$= 55\,W$

P_{new}

$= 77.5\,W \approx 77$

$E_{new} = P \times t = 77 \times 110 = 8470.$

4. Memory operation :- loading & storing takes place.

LD R_0 $0(R_x)$

SD R_0 $0(R_y)$

5. WB (write Back) :-

Addition is performed in 3^{rd} stage but its value is reflected only after 5^{th} stage.

ADD R_0 R_1 + R_2

fetch using PC

$\underbrace{\text{ADD}}_{\text{arithmetic}}$ $\underbrace{R_0}_{\text{Destination}}$ $\underbrace{R_1 \quad R_2}_{\text{Source}}$

In ID → we come to know source & dest. registers & that it is arithmetic operation

1cc	2cc	3cc	4cc	5cc
IF	ID	EX	MEM	WB.

It takes 5 clock cycles.

eg :- LD R_0 $0(R_x)$

Load

↓

It takes 4 clock cycles.

1cc	2cc	3cc	4cc
IF	ID	EX	MEM

EX ↓ $(0+R_x)$ address is found

LD R_0 $0(R_x)$

$0 + 108$

$= 108.$