NAME - AADITYA RAVASO KHOT

PRN - 21610051

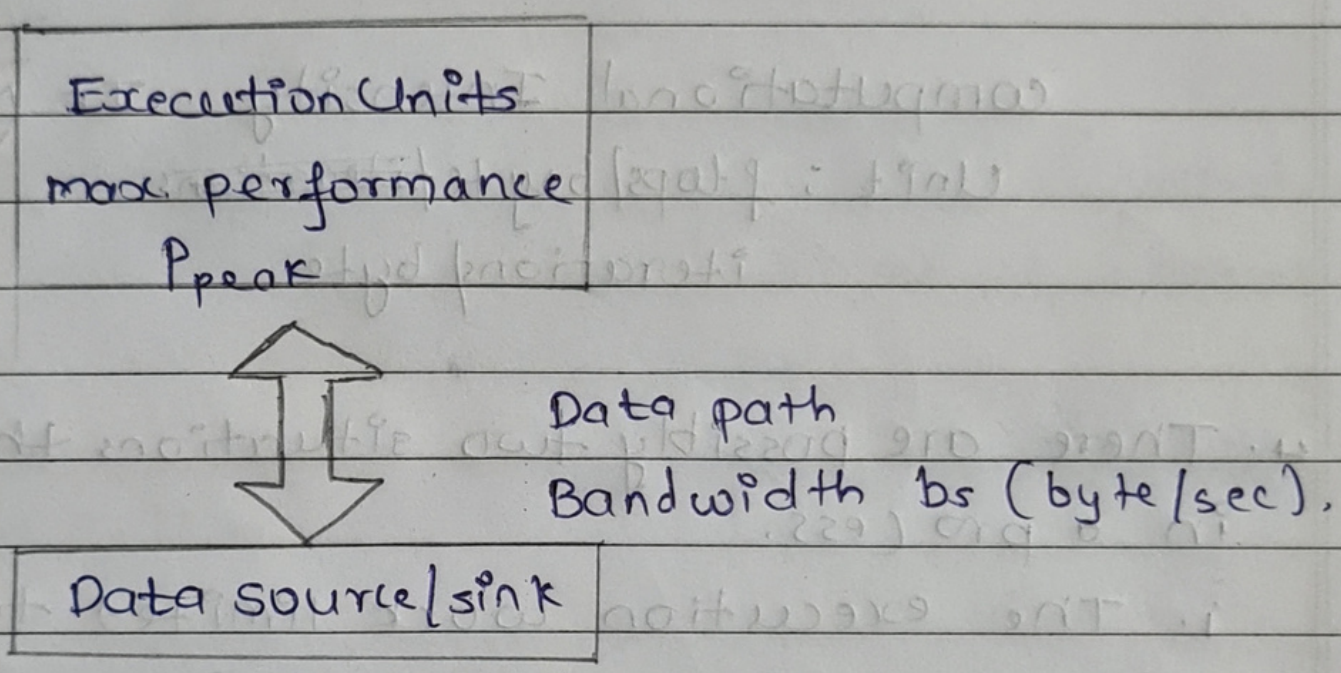
SUBJECT - COMPUTER ALGO. LABS

ASSIGNMENT NO. 2

Native Roofline model (Simple Roofline model).

1. The native roofline model is probably the simplest but still useful performance model for steady state loops in high performance computing.

2. Hardware View



Hardware is viewed as two units

i. Execution Unit of processor (at max. performance)
   units: mega flops/s or mega loops/s
   or iterations/s

ii. Data source/sink:
   Main memory interface which can store data or deliver data at maximum speed (bandwidth/s)
   unit: byte/s
   Not necessary it should be memory interface

iii.

3. Software View:
   comprises of several back to back loops which is sufficiently large to startup and windown effects like pipelining, prefetching. i.e. steady state behaviour.

```
! may be multiple levels
do i=1  <sufficient>
  <complicated stuff doing
    N flops causing
     V bytes of data
       transfer>
end do
```

computational Intensity $I = \frac{N}{V}$

Unit : ~~flops/byte / iterations~~
iterations/byte

4. There are possibly two situations that causes delay in a process.
   i. The execution work limited to max performance of execution units.
   ii. The bandwidth of data path

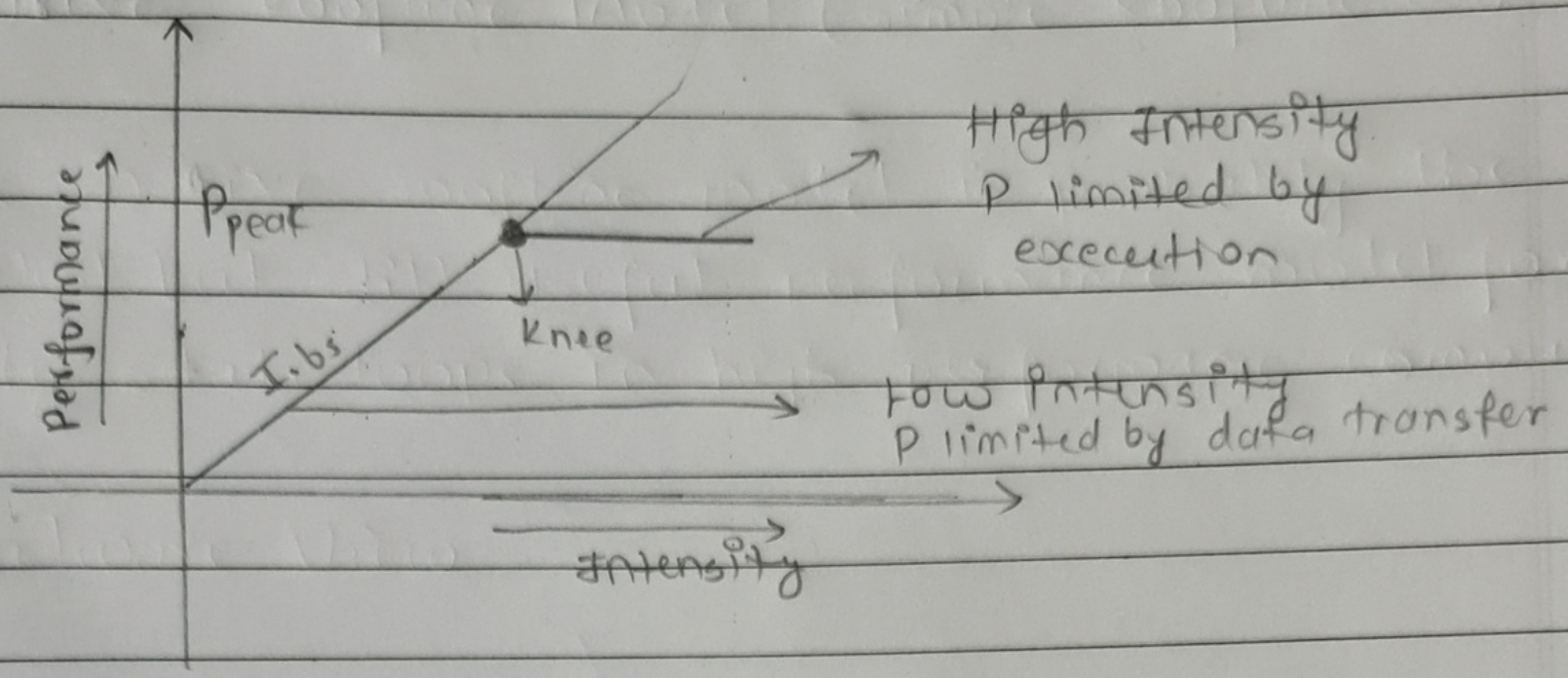
P_{peak} [flops/s]

$I \cdot b_s$ [flops/byte x bytes/s]

∴ At any time the upper limit of the final performance is minimum of P_{peak} and $I.b_s$

$$\therefore P = \min(P_{peak}, I.b_s)$$

5. Graphical Representation:



The intersection of P_{peak} and $I.b_s$ is called as "knee which is the point where best use of resources is observed i.e. max performance.

$\therefore P_{max} = I.b_s$

The model relies on several assumptions including perfect overlap of data transfers and computation, ignoring latency effects and assuming steady state code execution. Overall the native roofline model provides a simplified way to analyze the potential performance of a code on specific hardware platform, helping developers understand whether their code is limited by computation or data transfer and guiding optimization efforts to achieve better performance.