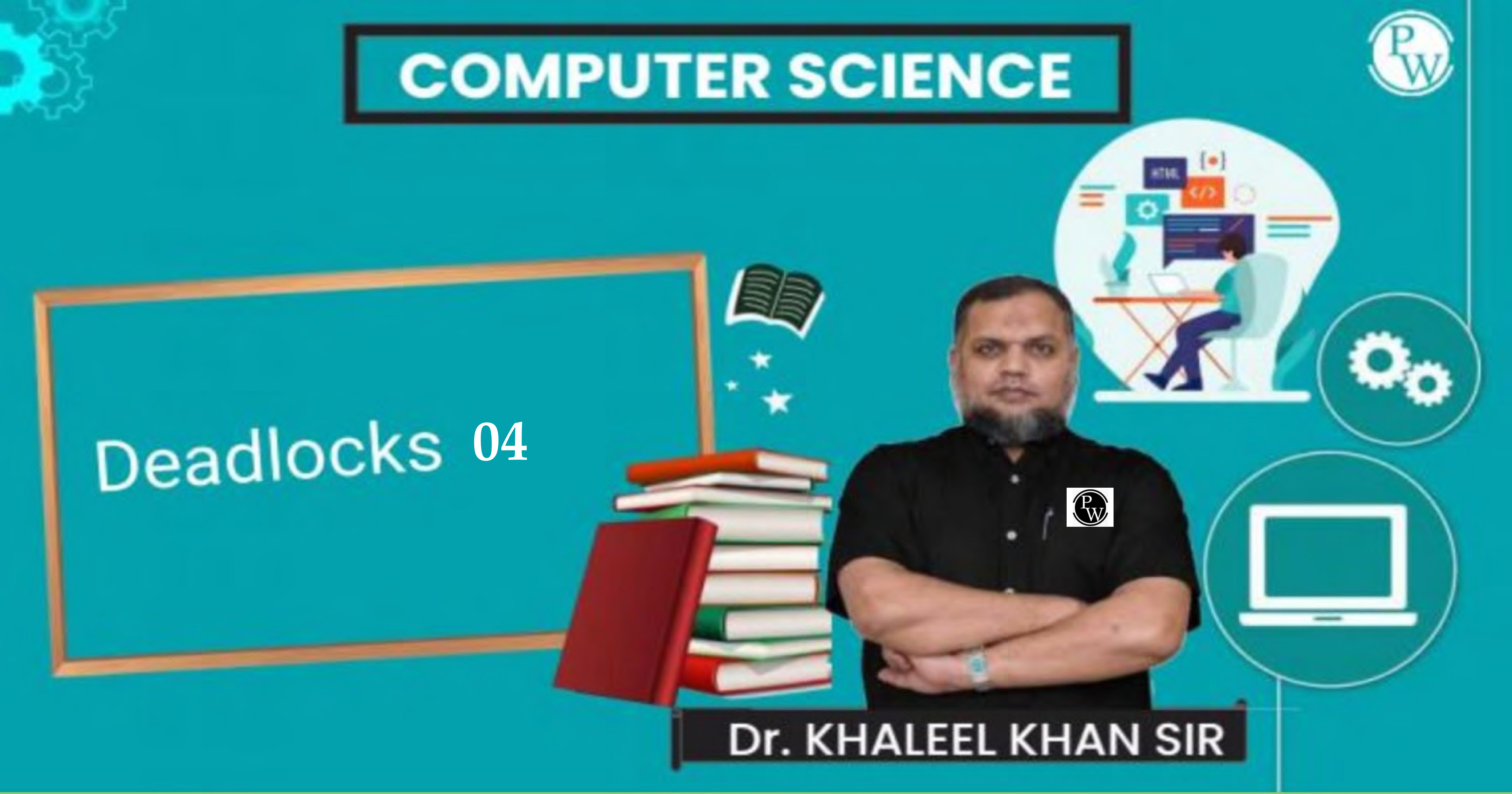
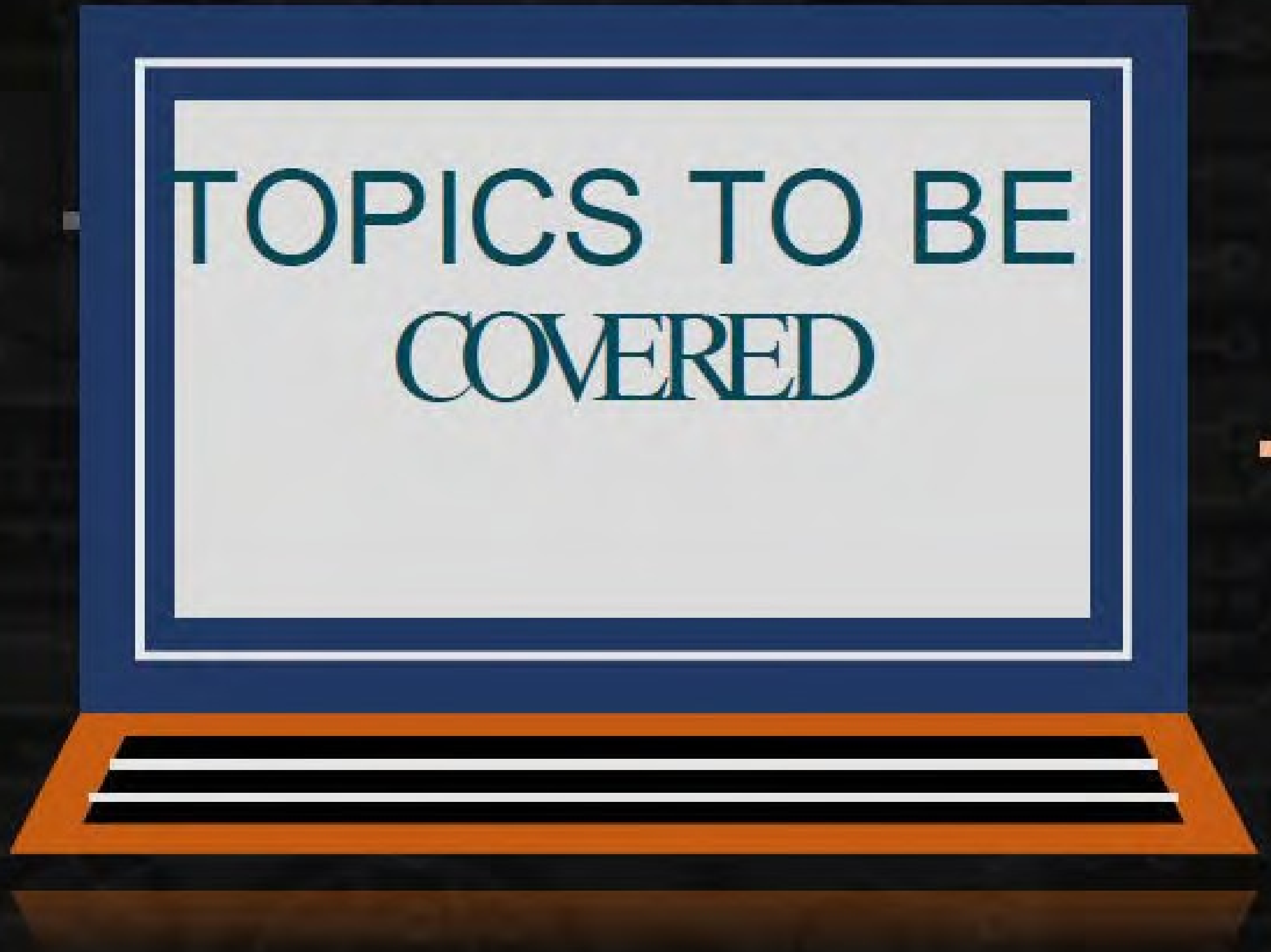


COMPUTER SCIENCE

Deadlocks 04

Dr. KHALEEL KHAN SIR

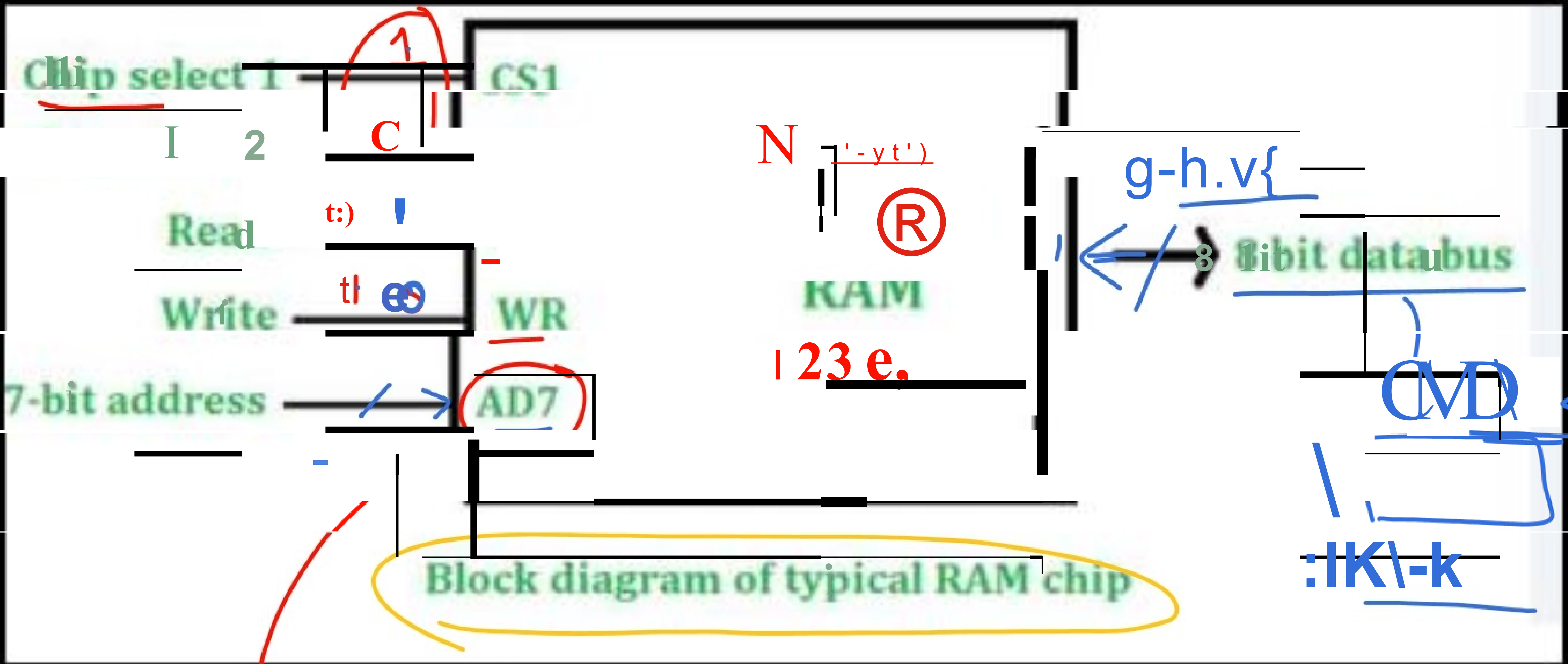


TOPICS TO BE COVERED

1.Problem Solvings

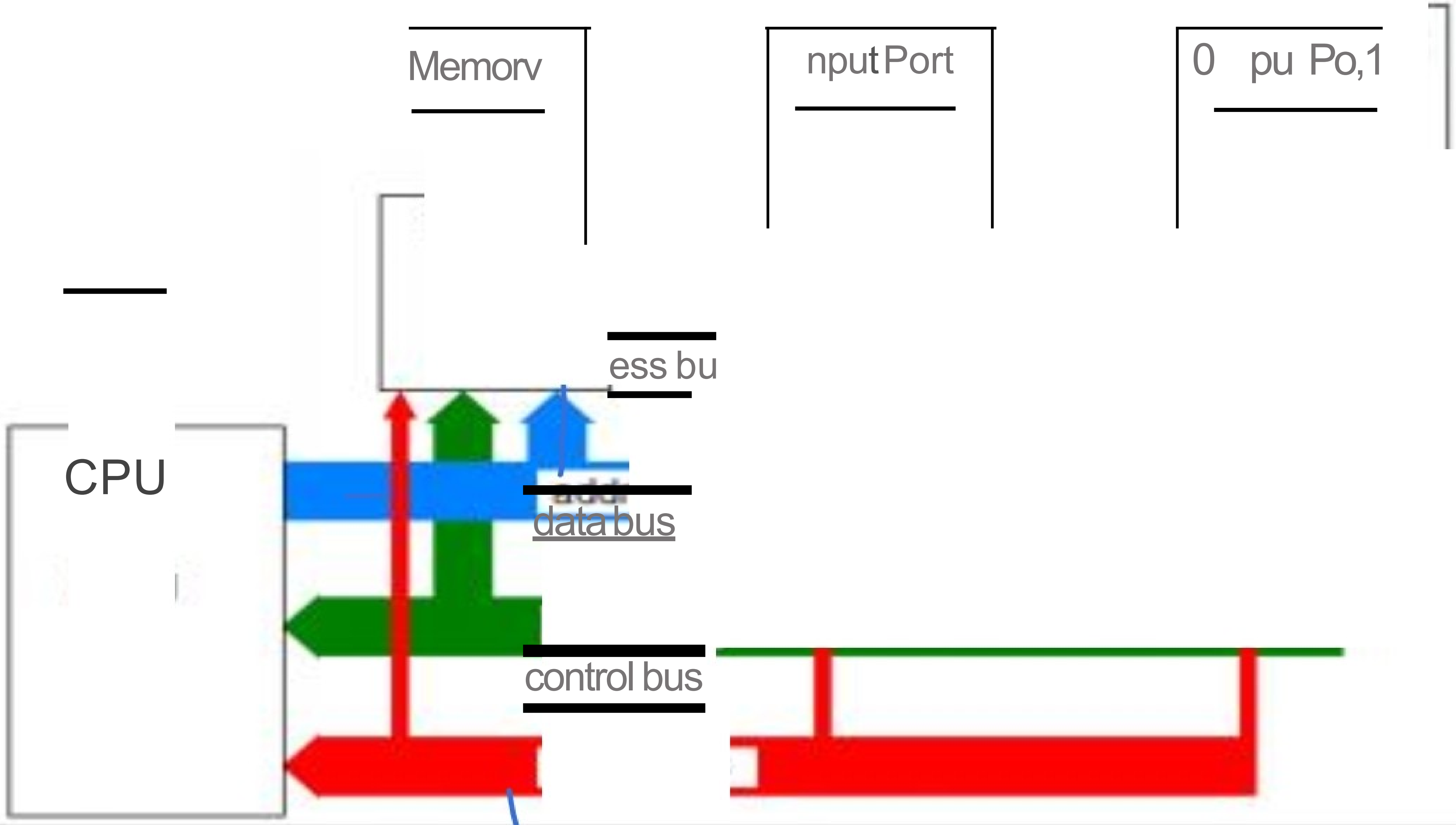
Session-II

26/11/2022

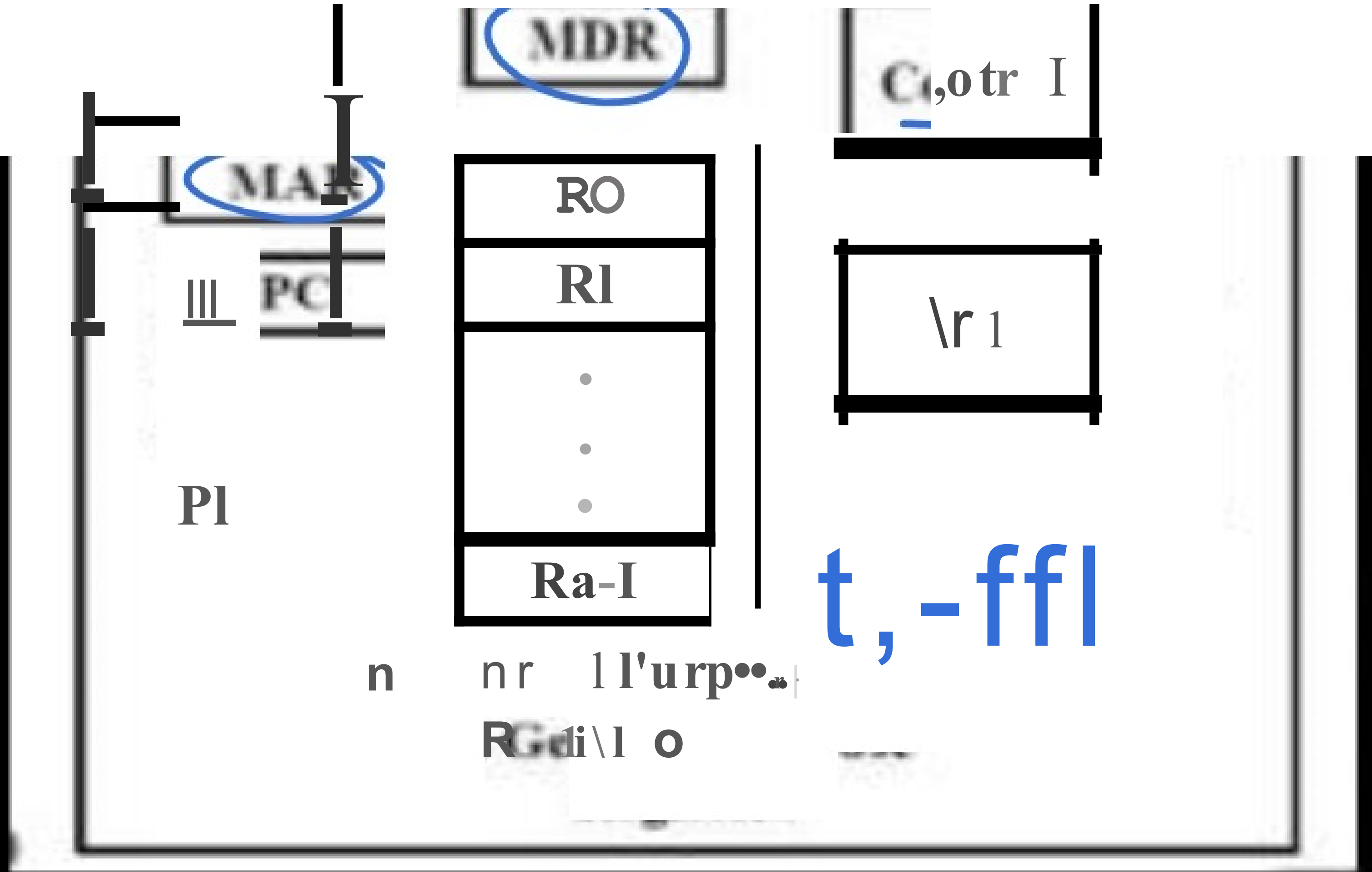


Block diagram of typical RAM chip

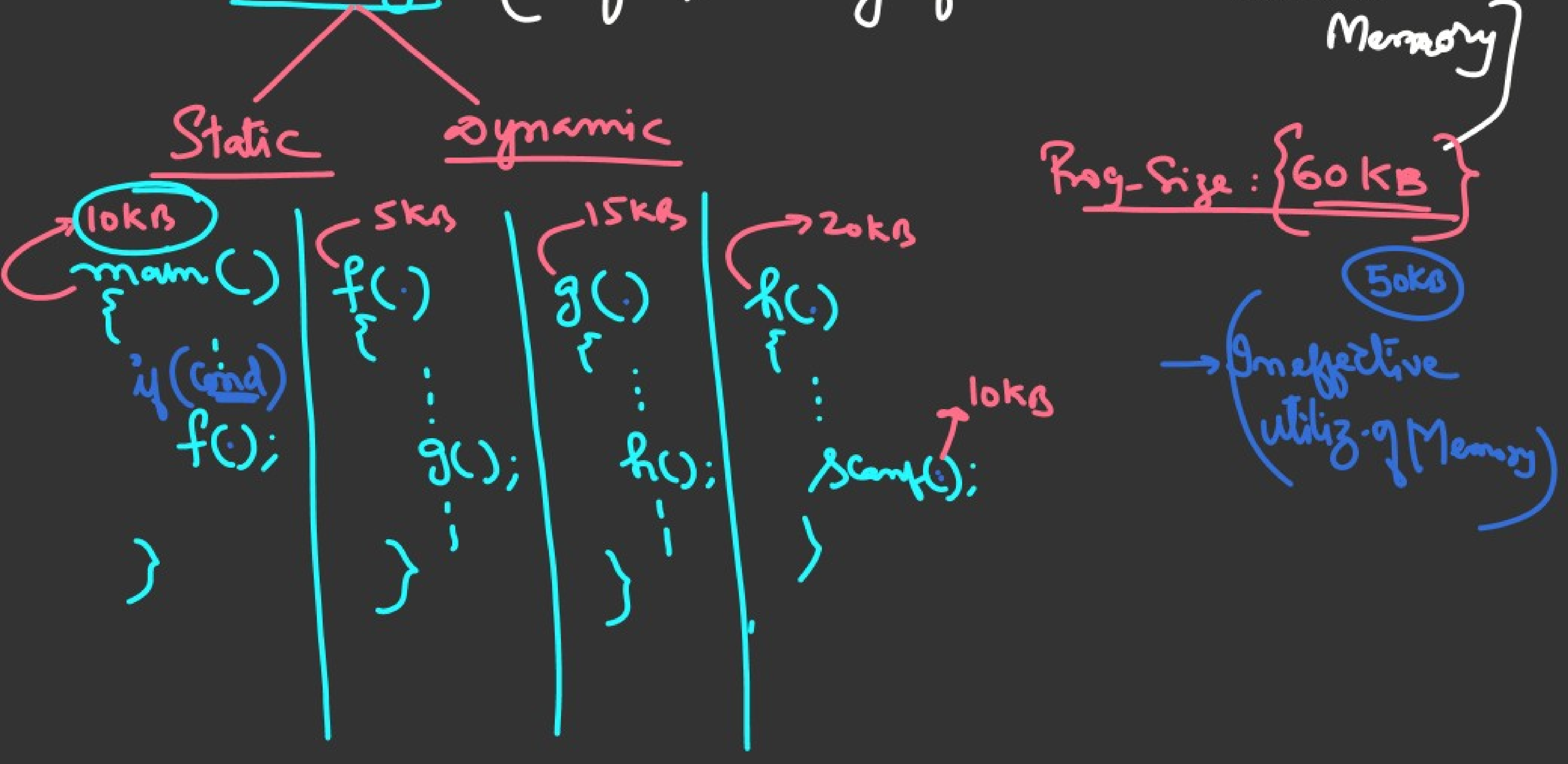
Address lines/bus


Memorv
nput Port
0 pu Po,1
CPU
ess bu
data bus
control bus

R|W


MDR
Cq,o tr l
MAR
PC
R0
R1
Ra-I
\r 1
Pl
t,-ffl
n nr l l'urp
RGdi\l o

Loading : [Refers to Loading of .exe from Disk to Memory]



```
main()        f()          g()          h()
{             {            {            {
  if (cond)     :            :            :
    f();        g();         h();         scanf();
                :            :
}             }            }            }
```

Dynamic Loading :

Loading the Modules/Segments of the Program on Demand @ R.T

	Static vs	Dynamic
Space	→ Ineff	Efficient
Time	→ efficient	Inefficient

Linking : Resolving (Finding Addresses) the external Refs used in the Program.

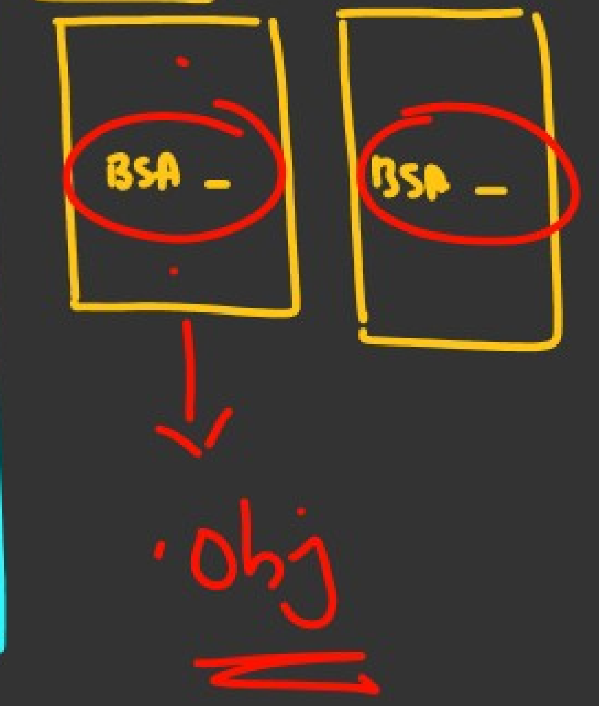
external Refs → Function, Global entities

BSA: Branch & Save Address

```
→ #include <stdio.h>
extern int x; — unresolved
main()
{
   :
BSA —;  f();
}  :
f()
{
   :
BSA —; → g();
}  :
```

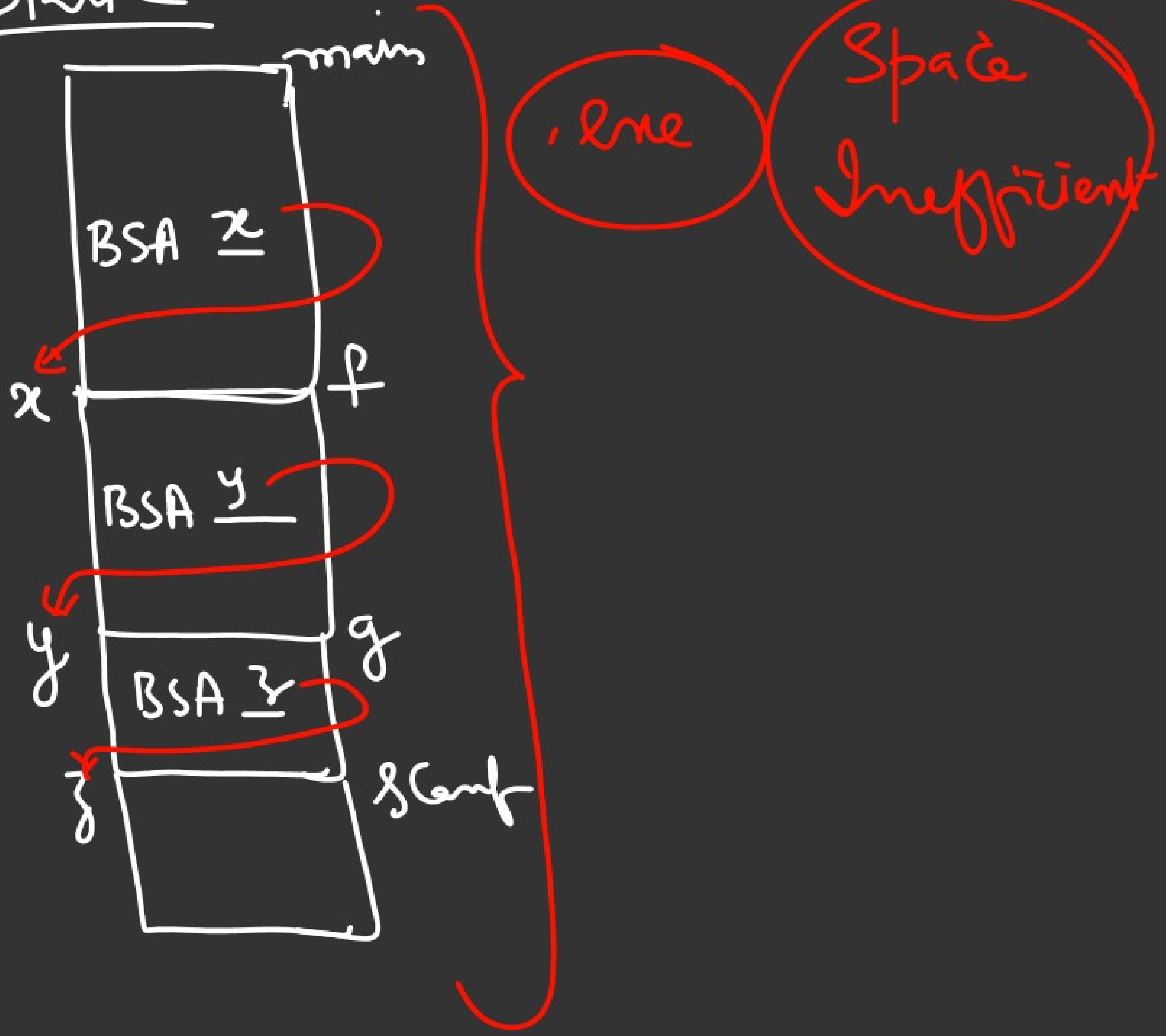
```
g()
{
   :
 scanf(..); BSA —;
}
```

Linker:



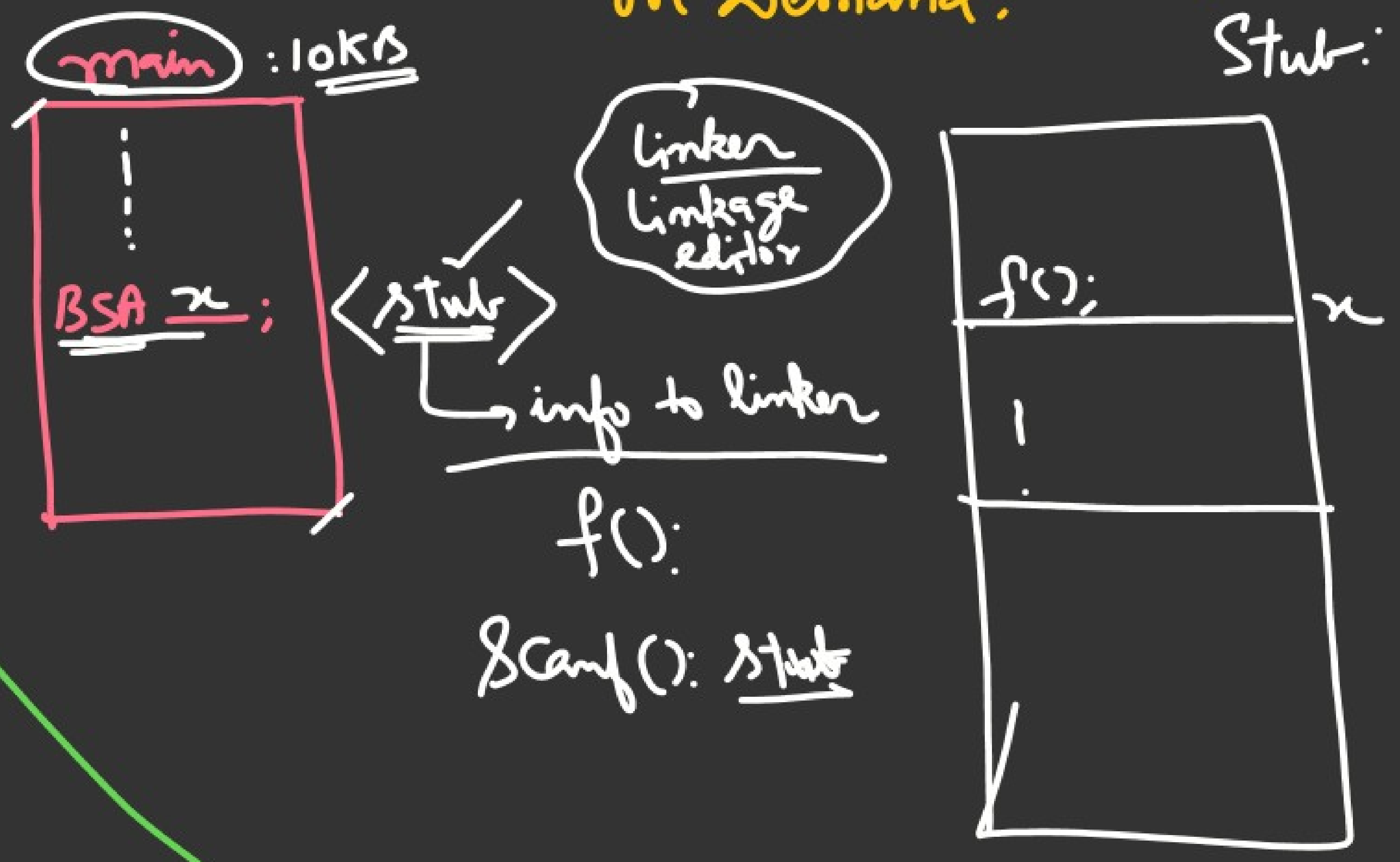
Linking < Static, Dynamic

Static



* **Dynamic Linking:** linking the modules/fn's/Enties @ R.T on demand;

Stub: A small piece of code



Galvin

Such Libraries that are linked with the applications @ R.T on demand are known as ______;

Dynamic link Libraries

(DLL)

Advantages:

1) Space efficiency

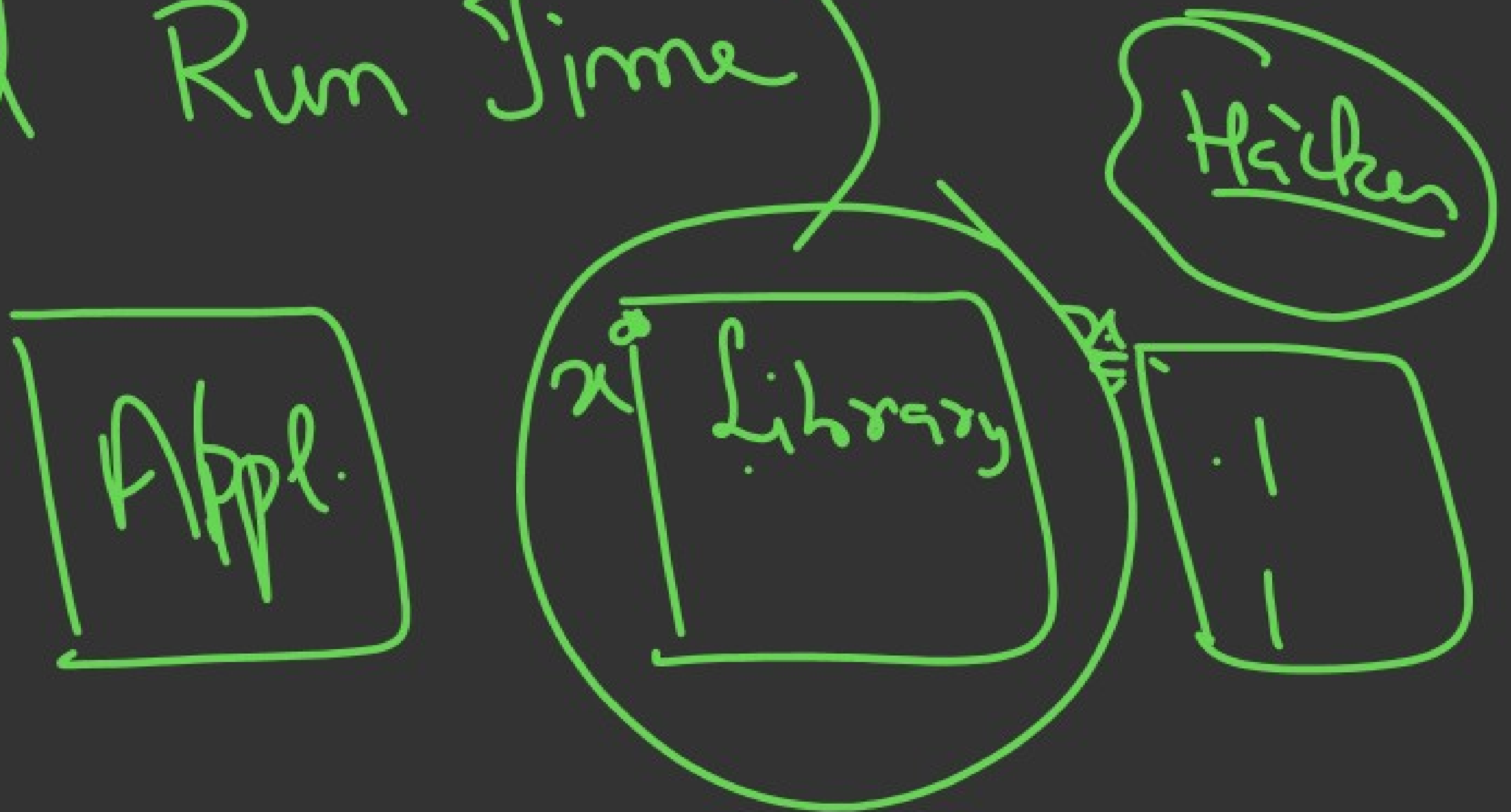
2) Reusability
(Shared Library)

3) Flexibility
(Change the Impl.)

Drawback:

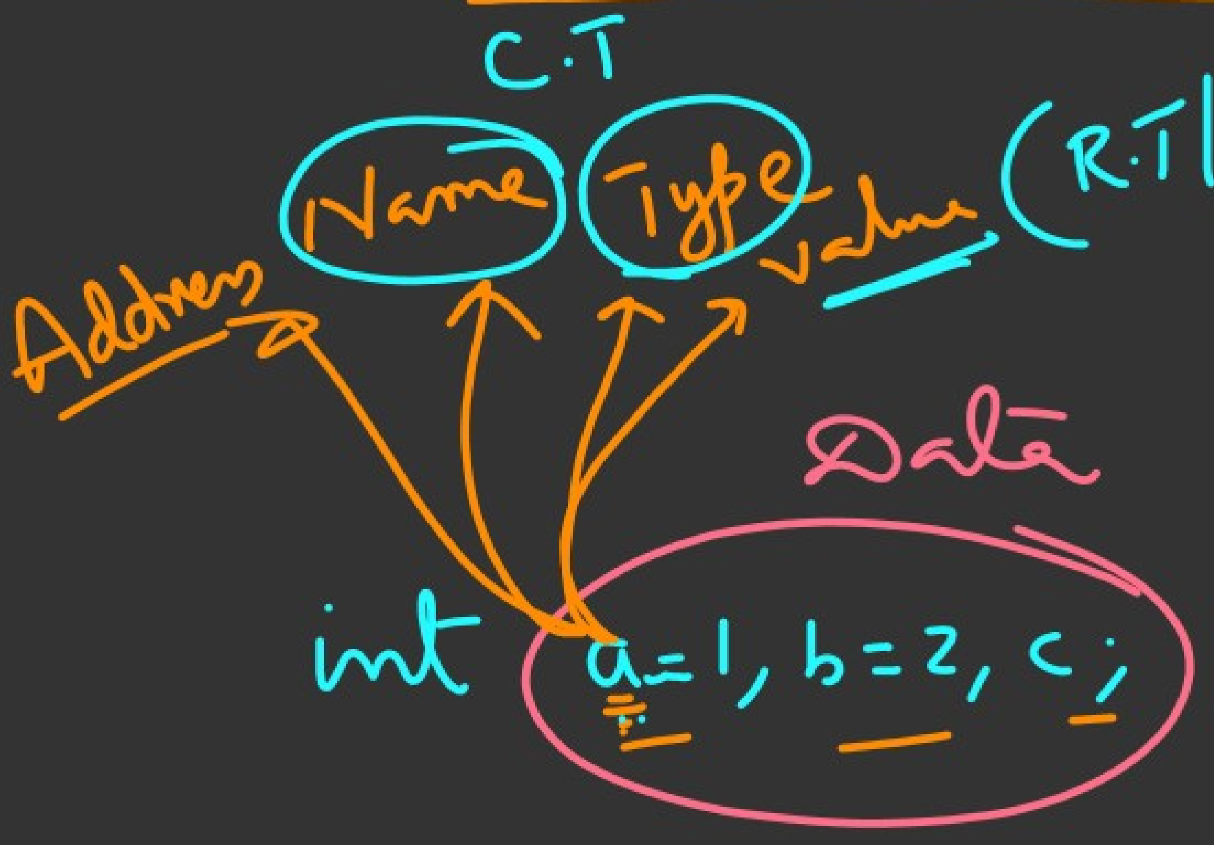
1) Time inefficient

2) Less Secure:
(Because the Path to the module is not known until Run Time)



Address Binding:

Association of Program Instrn's & Data units to Memory Locations/Address is known as Address Binding (AB)



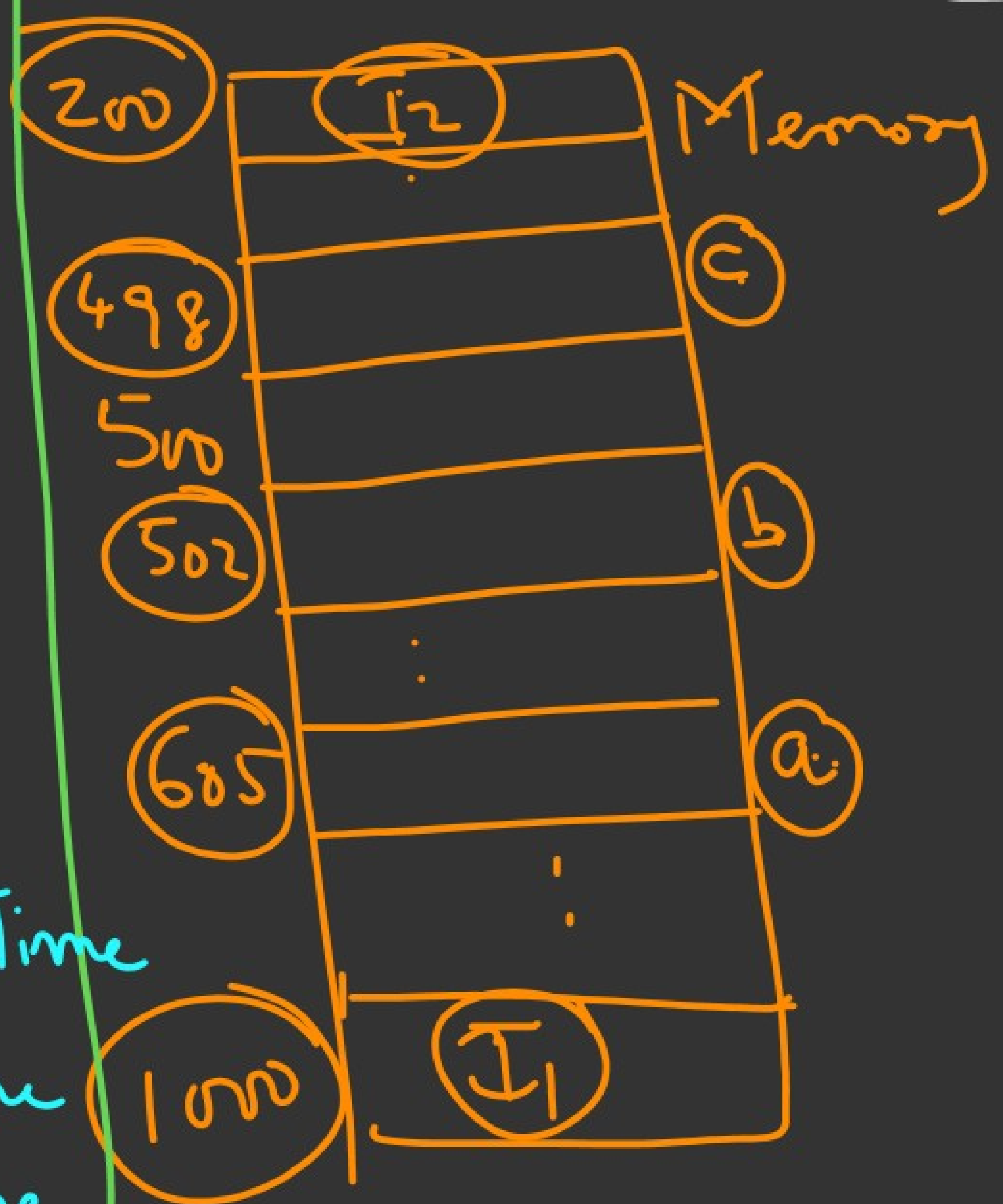
int a=1, b=2, c;

Code

```
c = a+b;  ⟹  I1: Load R1, a;
              I2: Load R2, b;
              I3: Add R1, R2;
              I4: Store c, R1
```

Time @ while (AB) takes place is called Binding Time

(i) Compile Time
(ii) Load Time
(iii) Run Time



Static {

I. Compile - Time :

(MSDOS)

Compiler → I: , D: } to Memory Addresses;

{I1 : 4002, I2 : 4004, I3 : 4008}

II. Load Time :

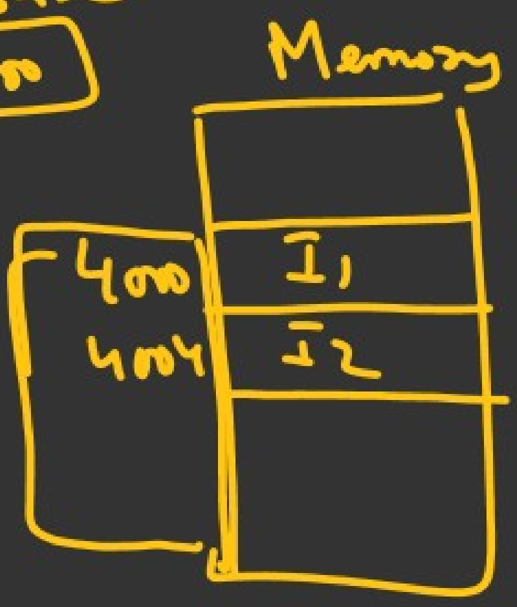
Base 4000

Loader

I1 : 0

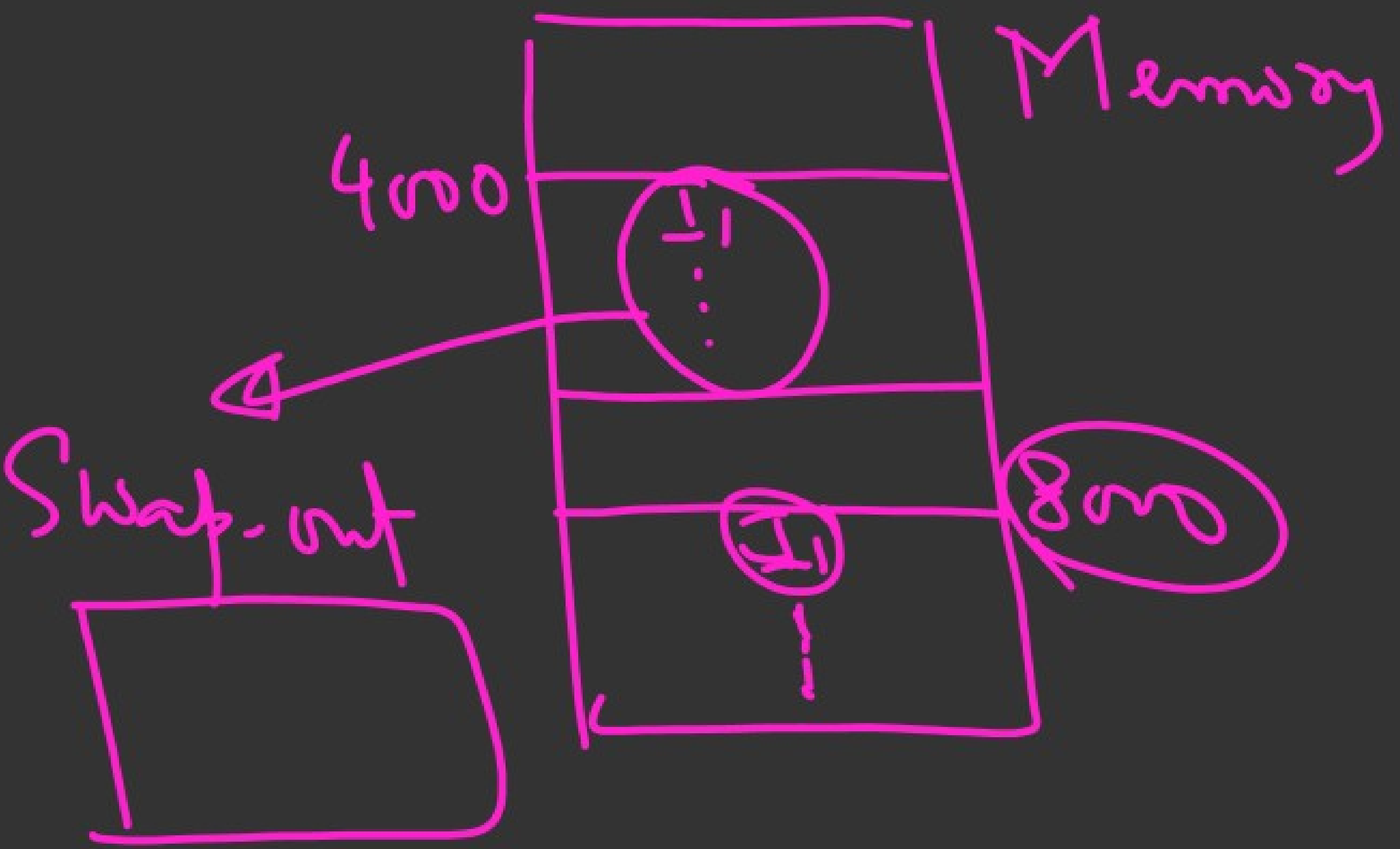
I2 : 4

I3 : 8



Static

(@ R.T, Program Cannot get relocated)



(Flexible)

3) Dynamic | R.T Address Binding :

Prog. Instn's/Data units Can get their addresses changed during R.T;

< Dynamic Relocation >

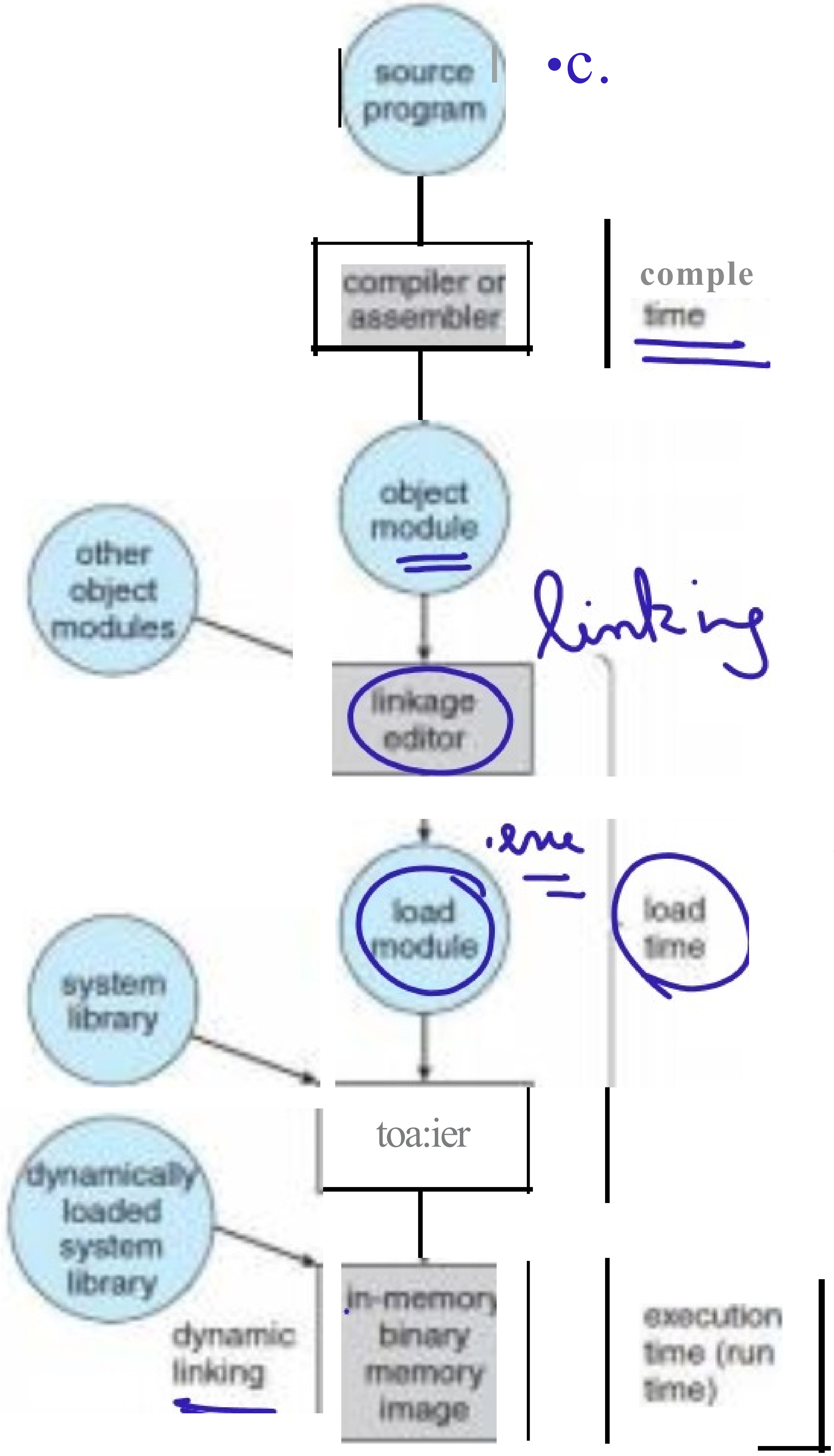


Figure 8.3 Multistep processing of a user program

Galvin

→ dynamic Loading

→ 〃 Linking

→ R.T Addr. Binding

(*) (Mem. Mgmt. Techniques)?

THANK YOU!