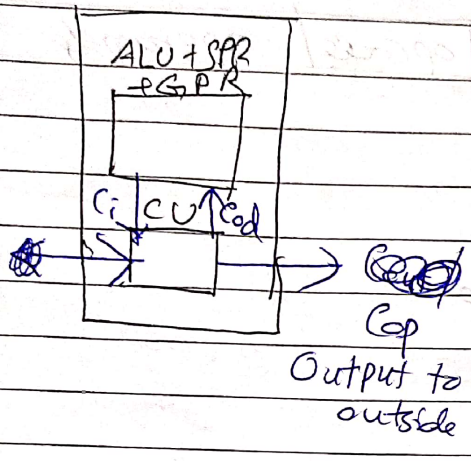
15/2/23 Wed Feb

Date __/__/____

Saathi

Control Unit Design



CU does instruction sequencing
↓
Fetch instruction from memory
Generate sequence of control signals & then execute it
May used instruction decoder to decode instruction

Content of AC will go to CU
Depending on this CU generate control signal
Based on this send signal to ALU
Cod → Control Output to Data → sent to ALU.

Can have more than 1 CU.

Synchronise other CUs → Main / Supervisor
Slave CU → Take input from supervised CU

Before

opcode	operands	Next instruction address

→ Disadv. is instruction length ↑
∴ ↑ Vol. of memory for simple prog.

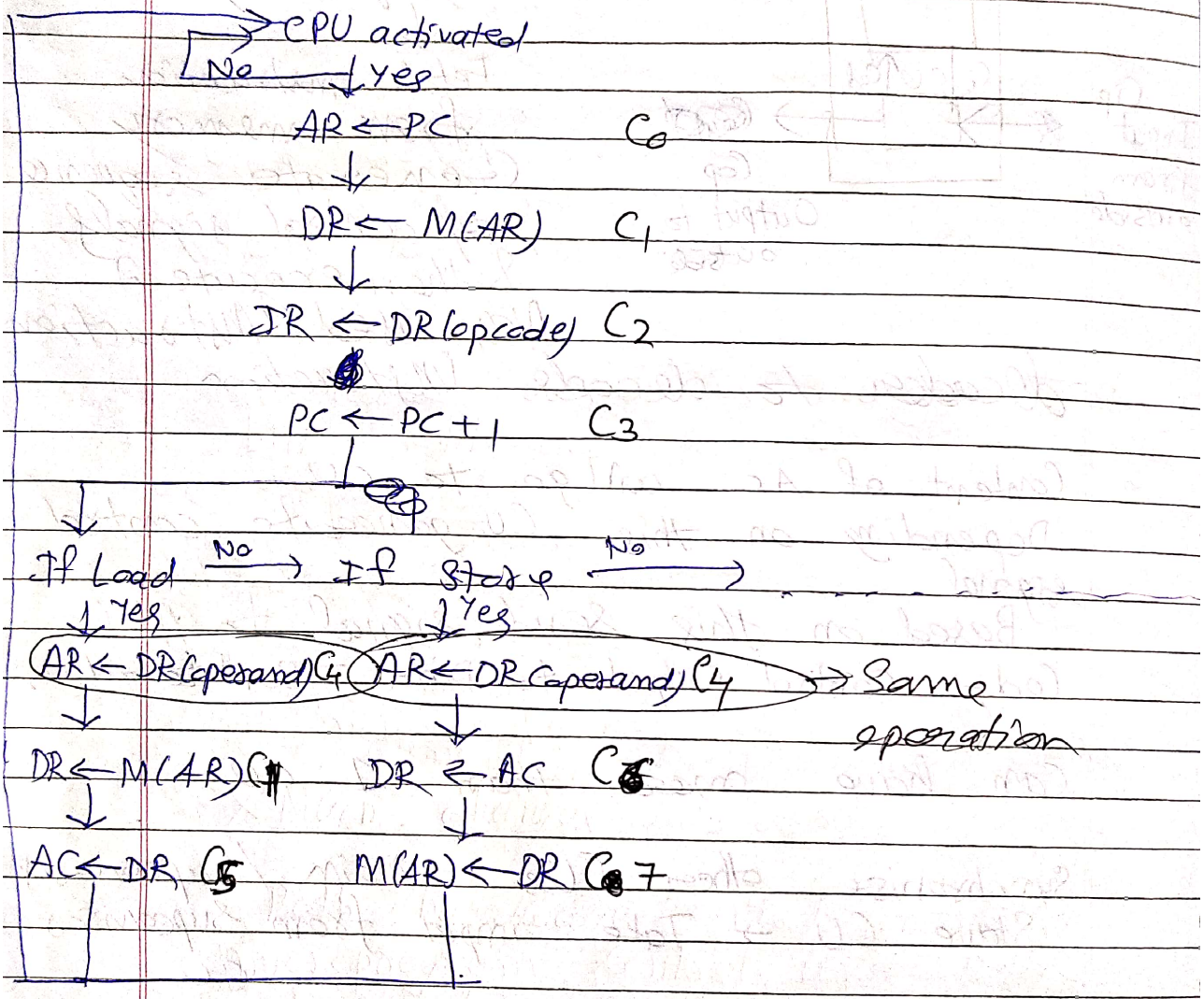
PC play imp role in instruction sequencing

Saathi

Date __/__/__

Design CU with 8 macro instructions, PE

∴ Instruction is | opcode | operands |

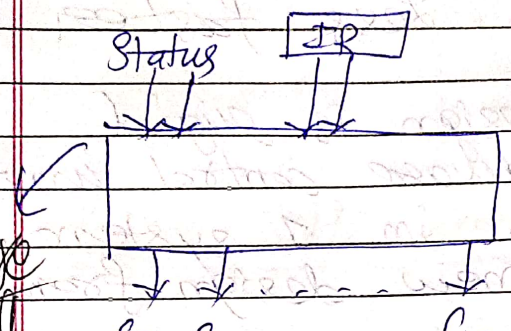


A micro instructions can have ≥1 micro operations

For each micro operation to be realized, we have to activate control signal (Ci)

We will have ≈ 13 control signals for all 8 macro instructions taken together.

We need control unit / logic ckt that generates these 13 control signals.



For load
C_0, C_1, C_2, C_3
C_4
C_5
in this sequence generate

↓

Can make ckt that generate $C_0 - C_3$ & then based on ckt, generate additional C_i's & go back to initial state.

→ Load CPU

C_0	Activate C_0
C_1	Activate C_1
C_2	Activate C_2
C_3	Activate C_3
If load	
C_4	Activate C_4
C_5	Activate C_5

This is micro prog. control unit
Instruction of micro instruction fetched one by one & execute

Before hardware control unit dominated (till 1970s) over micro prog. control unit

Which unit dominates depends on cost of software / hardware / purpose

SUCCESSFUL
Date
Page No.

Hardware design faster than software

As in micro prog control unit we need memory. : Less faster

But hardware design is rigid
Once design hardware control unit
X change design in 1 system
Have to use new design from scratch

Hardware design is harder to debug but software is easier to debug.

~~Application Specific Machi~~
In 1980s micro programme control unit dominate

But now software became ↑ costly.
Now want ↑ fast machine
X Much changes in design

∴ Both hardware control unit & microprogram control unit both dominate.

Disad

SUCCESSFUL
Date
Page No.

Hardware Control

- State Table Method
- Delay Element Method
- Sequence Control Method

State Table Method

There are finite states
Output : $C_0 - C_{12}$
Input : Z flag, IR, Status

If in a path, know sequence in which control signals generate
i.e, Depends on present state

Draw state diag., draw sequential ckt.

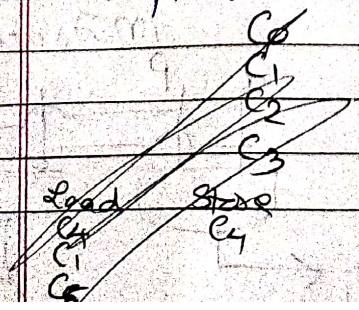
If know states, i/p → can draw state table & hence design

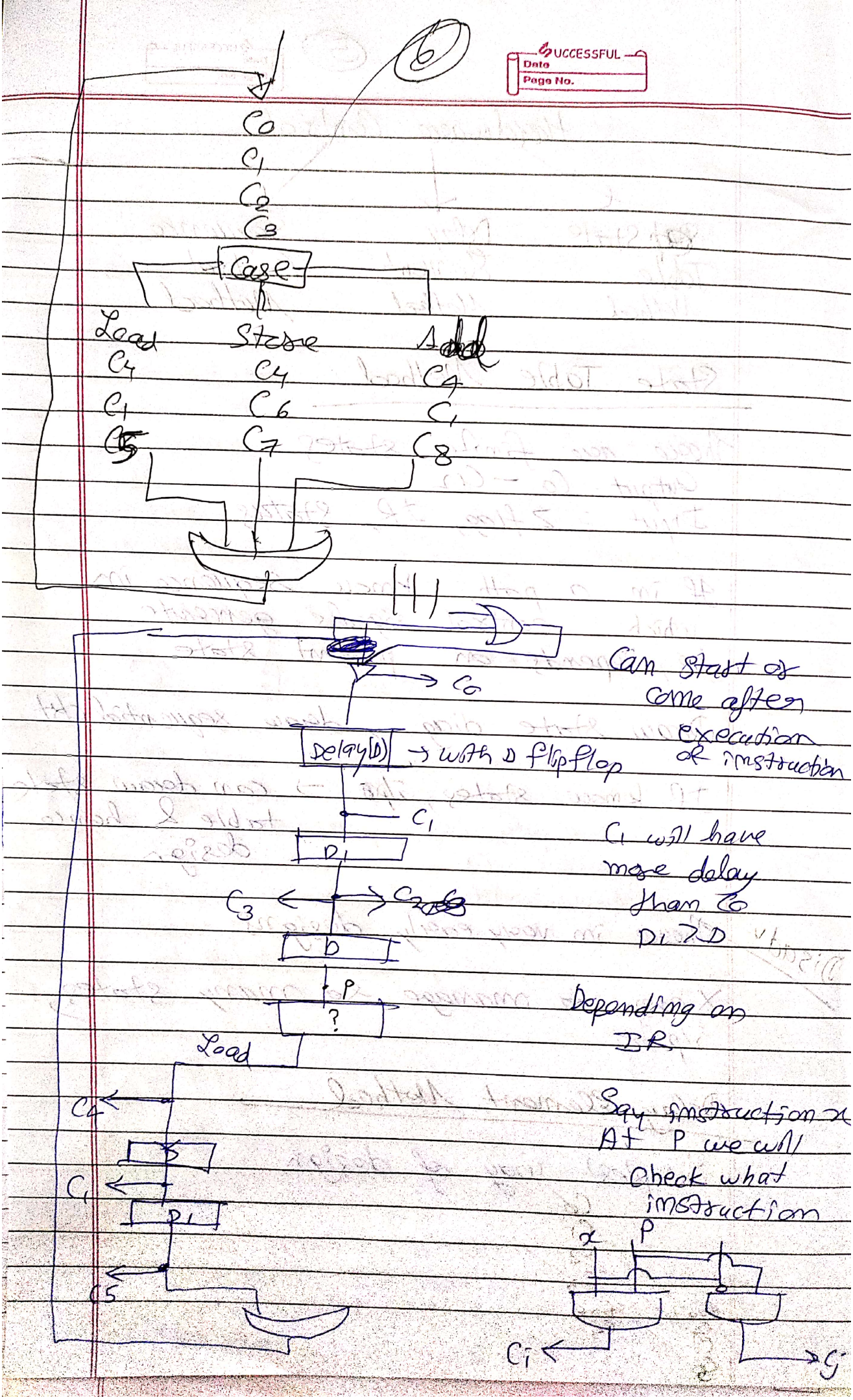
Disadv. There in very early designs.

X easy to manage so many states, i/p

Delay Element Method

Simplest way of design




C0
C1
C2
C3
Case
Load
Store
Add
C4
C5
C6
C7
C8
C0
Delay(D) → with D flipflop
C1
D1
C3
C2
D
P
?
Load
C4
D
C1
D1
C5
Can start or come after execution of instruction
C1 will have more delay than C0
D1 > D
Depending on IR
Say instruction x
At P we will check what instruction
x
P
C1
C5

Saathi

Date ___ / ___ / _____

Delay element method provides the simplest design.

Based on value of x appropriate C1/C5 generated.

Disadv. Now there is memory associated, Also multiple delays. Various types of delays have to be designed.

;; Costly

If say max. delay = Delay of all operations

Lot of time wasted, Very costly