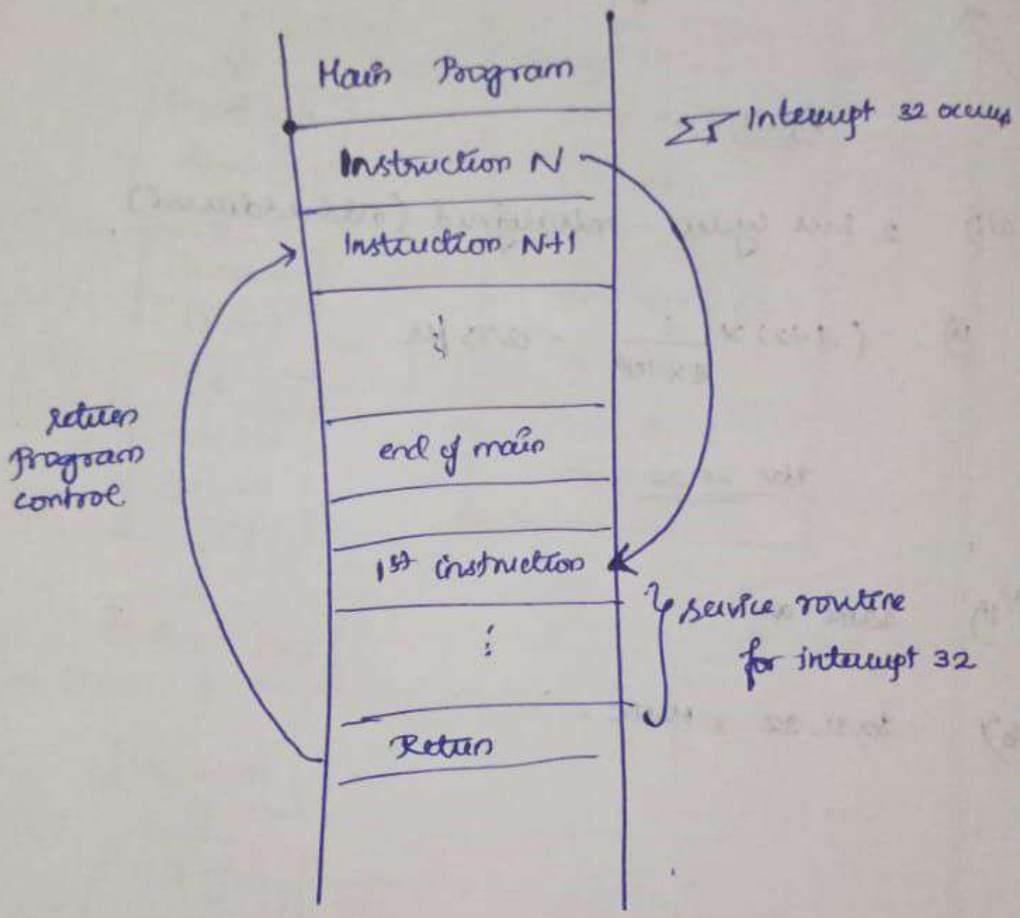
CO4

1) Interrupt Mechanism, types & Priority

→ The section of program to which control is passed : Interrupt service routine

- well defined context switching mechanism for changing program environments



256 Interrupts

↳ 5 groups.

↓

Increasing priority

- Reset
- Internal interrupts & exceptions } not user defined. dedicated system function
- Software interrupts
- Non Maskable interrupts } User defined
- External hardware interrupts

↓

Keyboard, printer, timer.

Type 0 - highest priority

↓

Type 255 - lowest priority interrupt

Internal interrupt

- Type 0 - Divide error (higher)
- Type 4 - Overflow ← lowest priority internal interrupt

Power-down, keyboard - assign high priority

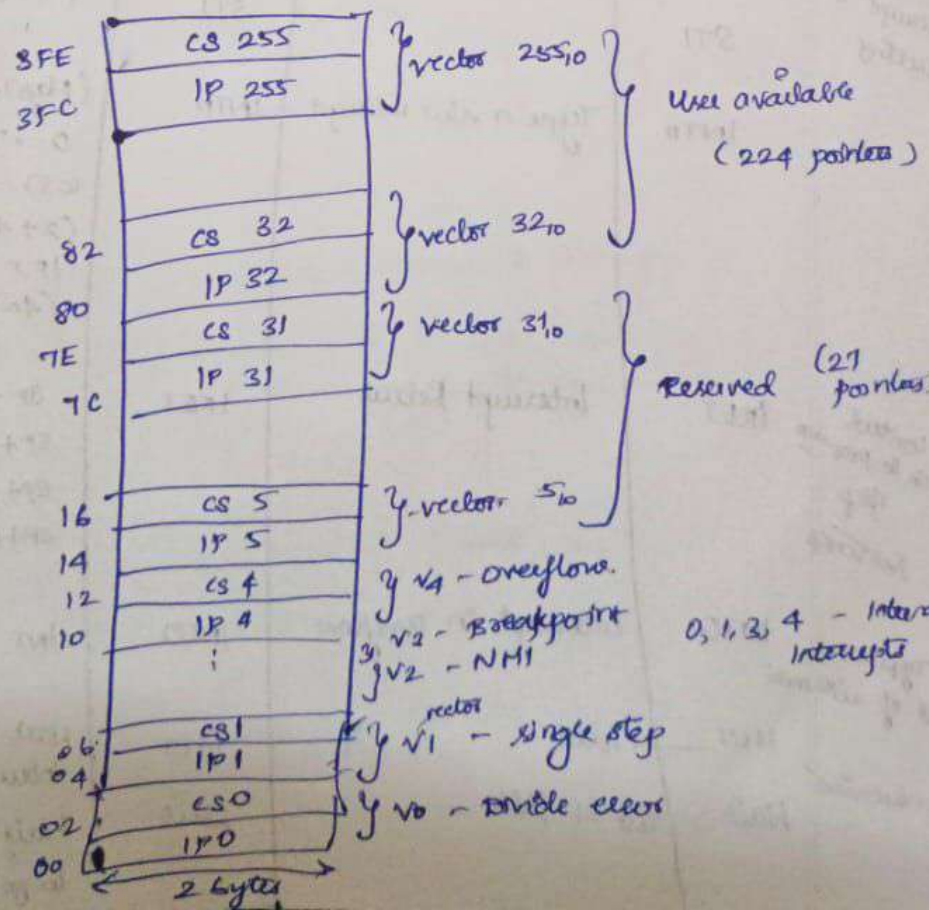
2) Interrupt vector table

- Address-pointer table - link interrupt type numbers to locations of their service routines.
- these contents can either held as firmware in EPROM or loaded into RAM

Address 00000_{16} to $003FE_{16}$ → 1st 1KB of memory

⇒ each 256 pointers require 2 words (4 bytes) & always stored at even-address boundary.

⇒ The higher addressed word of 2-word vector is called base address.



Eg:
type 255 IP_{255}, CS_{255} address $003FC_{16}$ $003FE_{16}$

CS_{255} & IP_{255}

Eg

At what address are CS_{50} & IP_{50} stored in memory?

Address = 4 × 50 = 200

(multiply type num (50) by 4 (4 bytes for each))

$200 \Rightarrow 11001000_2$

$C8_{16}$

IP_{50} is stored in $000C8_{16}$

$CS_{50} \Rightarrow 000CA_{16}$

Interrupt Instructions

Mnemonic	Meaning	Format	Operation	Flags affected
CLI	clear interrupt flag	CLI	0 → (IF)	IF
STI		STI	1 → (IF)	IF
INTn	Type n s/w interrupt	INTn	(Flags) → (SP-2) 0 → TF, IF (CS) → (SP-4) (2+4n) → (CS) (IP) → (SP-6) (4n) → (IP)	TF, IF
IRET	Interrupt Return	IRET	SP → IP SP+2 → CS SP+4 → Flags SP+6 → SP	All
INTO	Interrupt on overflow	INTO	INT 4 steps	TF, IF
HLT	Idle,	HLT	Wait for external interrupt or reset to occur	None
Wait	Test = 1, Idle	Wait	Wait for TEST to go active	None

Disables external interrupt i/p by resetting

Pass control back to program & restore flags

(Type 4) precede at end of arithmetic

Synchronize

Enabling / Disabling of Interrupts   initially, clear IF

IF - affects only external h/w interface

INTR Instr

STI, CLI - s/w

reenable → INTE & $\xrightarrow{set}$ IF

otherwise, reenabled by IRET.

5) External h/w interrupt interface signals

Min mode

o INT_{32}

⋮

o INT_{255}

→ refer book for complete dia.

Interface circuitry - Identify pending active interrupts with highest priority & pass the type number to microprocessor.

INTR, $\overline{INTA}$

↳ level triggered
- maintain at level-1 until tested
- logic 1 must be removed before service

routine runs to completion, otherwise, same interrupt may get acknowledged 2nd time.

$\overline{INTA}$

2 pulse

pulse 1 → acknowledged, prepare to send type no

" 2 → puts type no on data bus. ($AD_0 - AD_7$)

Max

Diff: Bus controller produces $\overline{INTA}$, ALE.

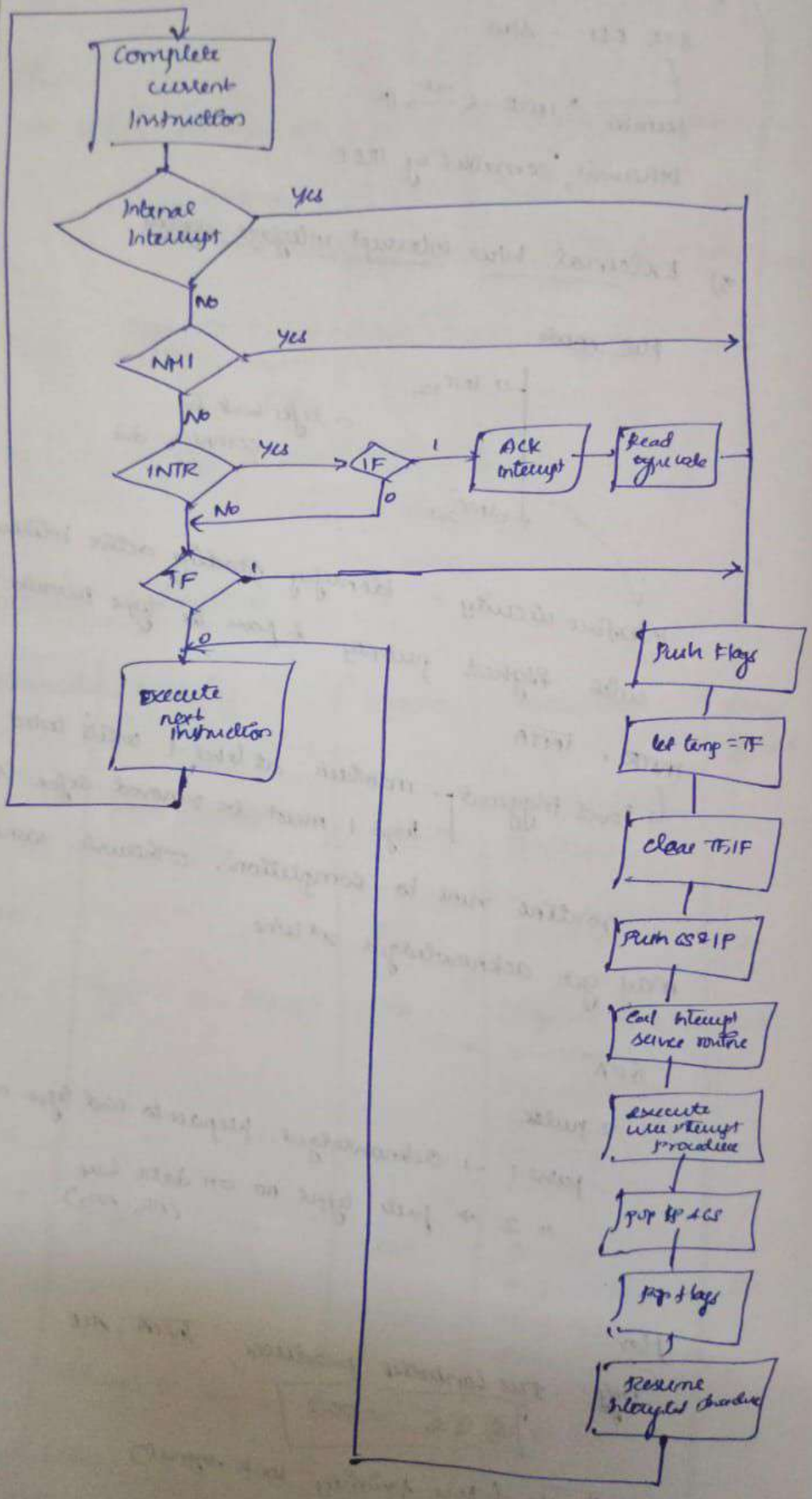
$\overline{S_2}\,\overline{S_1}\,\overline{S_0} = 000$

$\overline{LOCK}$ (bus priority lock signal)

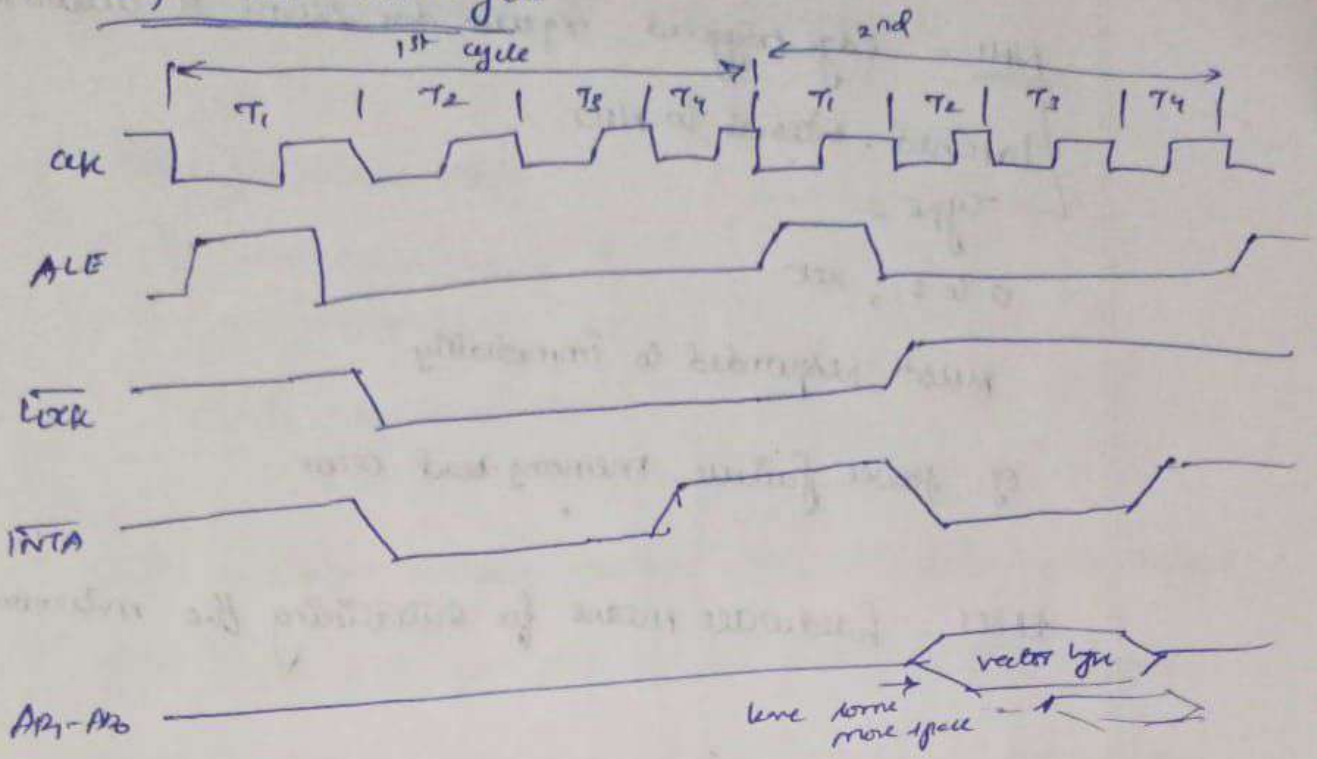
input to → bus arbiter

INTR → 1 → if IF=0, external interrupts masked out, else ignore

Interrupt Processing Sequence



Interrupt ack bus cycle



Interrupt service routine

To save registers & parameters on the stack { PUSH XX, " YY, " ZZ

main body of service routine { ⋮

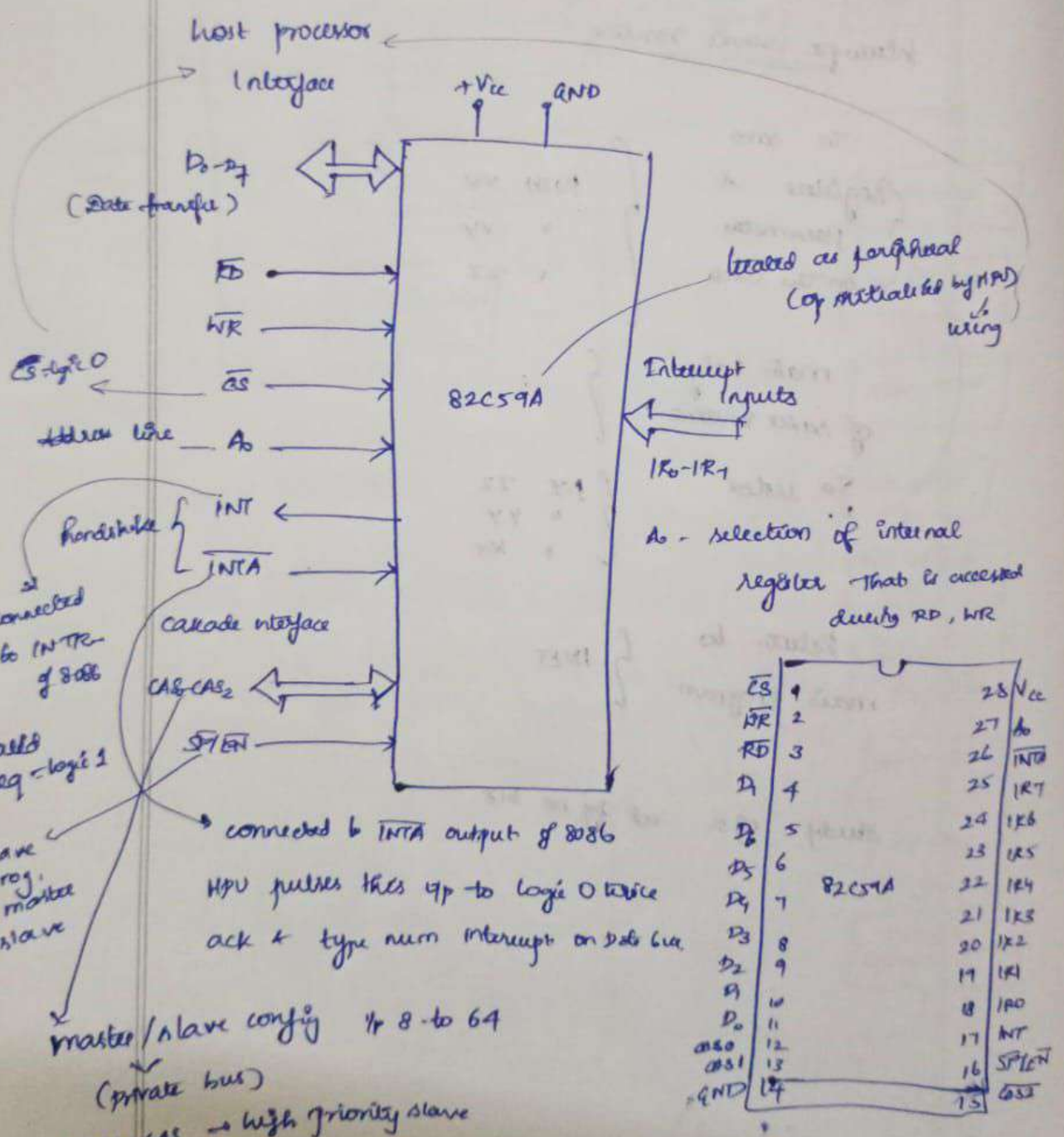
To restor { POP ZZ, " YY, " XX

Return to main program { IRET

NMI - edge triggered - request for service is automatically latched. internal to MPU
- Type 2
0 to 1., set
must responded to immediately
eg: power failure, memory read error

RESET - hardware means for initializing the microcomputer

82C59A PIC



$IR_0 - IR_7$ - level sensitive or edge triggered

1.

(req for service must be removed bfore service routine completion)

Logic 0 - service missed

Prob

some external devices produce short-duration pulse instead of fixed logic level for use as an interrupt req signal.

If MPU is busy servicing a higher priority interrupt, this req will be completely missed if 82C59A is in level sensitive mode

To overcome this, edge-triggered is used.

0 to 1 - active (+ve)

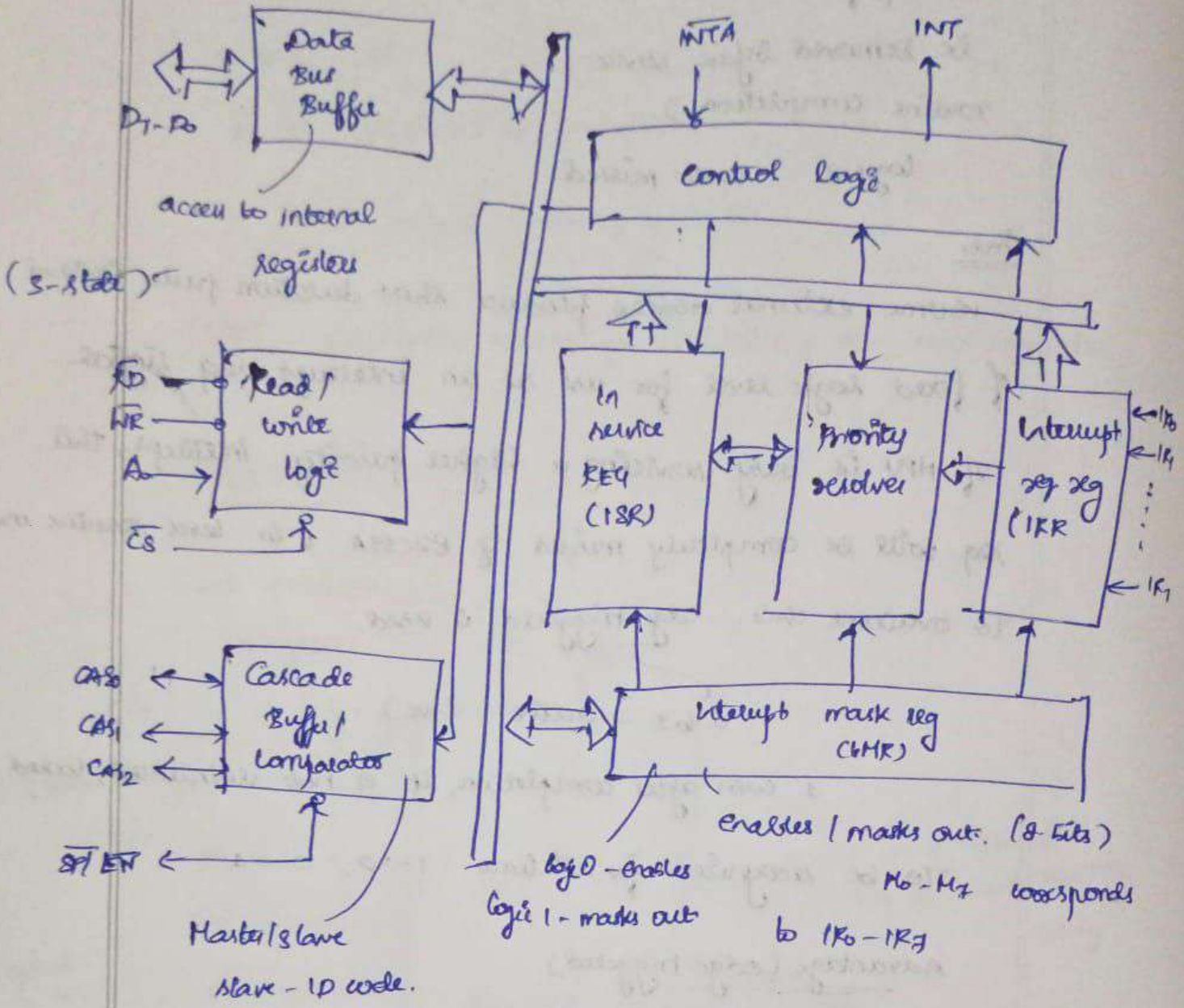
1 even after completion, it is not reinitialized, locked out

To be recognized for 2nd time, $1 \rightarrow 0$, $0 \rightarrow 1$

Advantage (edge-triggered)

If req at IR input is removed bfore MPU acknowledges service of the interrupt, its request is kept latched internal to 82C59A until serviced.

Internal architecture of 82C59A



Nov 2022

1) b)

INTR → INTA cycle (T1 to T3) → Interrupt vector Fetch cycle

→ stacking Flags, CS, IP (T4 to T5), Jump to ISR