

Memory Segmentation

- Memory is used to store information. This information can be either programs or data.
- Programs are a set of instruction stored in sequential manner.
- After every instruction the address gets incremented.
- Data on the other hand can be stored sequentially or just randomly.
- Moreover there is an organized form of structure data called stack.
- Stack grows in reverse manner which mean after every push the address is decremented.
- Stack and data will eventually start overlapping on one another and will be difficult to manage as they grow.
- To prevent this the memory is organized into 4 segments.
→ Code → Stack → Data → Extra.

Code Segment

This segment is used to hold the program to be executed. Instructions are fetched from code segment.

Code segment holds 16 bit base address.

Instruction pointer holds 16 bits offset address.

Data Segment

This segment is used to hold general data.

This segment also holds source operands during string operation. DS register holds 16 bit base address for this segment.

BX register used to hold 16 bit offset

SI hold 16 bit offset address during string operation.

Stack Segment

This segment holds stack memory which operate in LIFO manner. SS hold base address.

SP hold 16 bit offset address of top of stack.

BP hold 16 bit offset during random access.

TOTAL

Extra segmentation

- used to hold general data
- CS segment is used as destination during string op
- ES hold base address.
- DI hold offset address during string op.

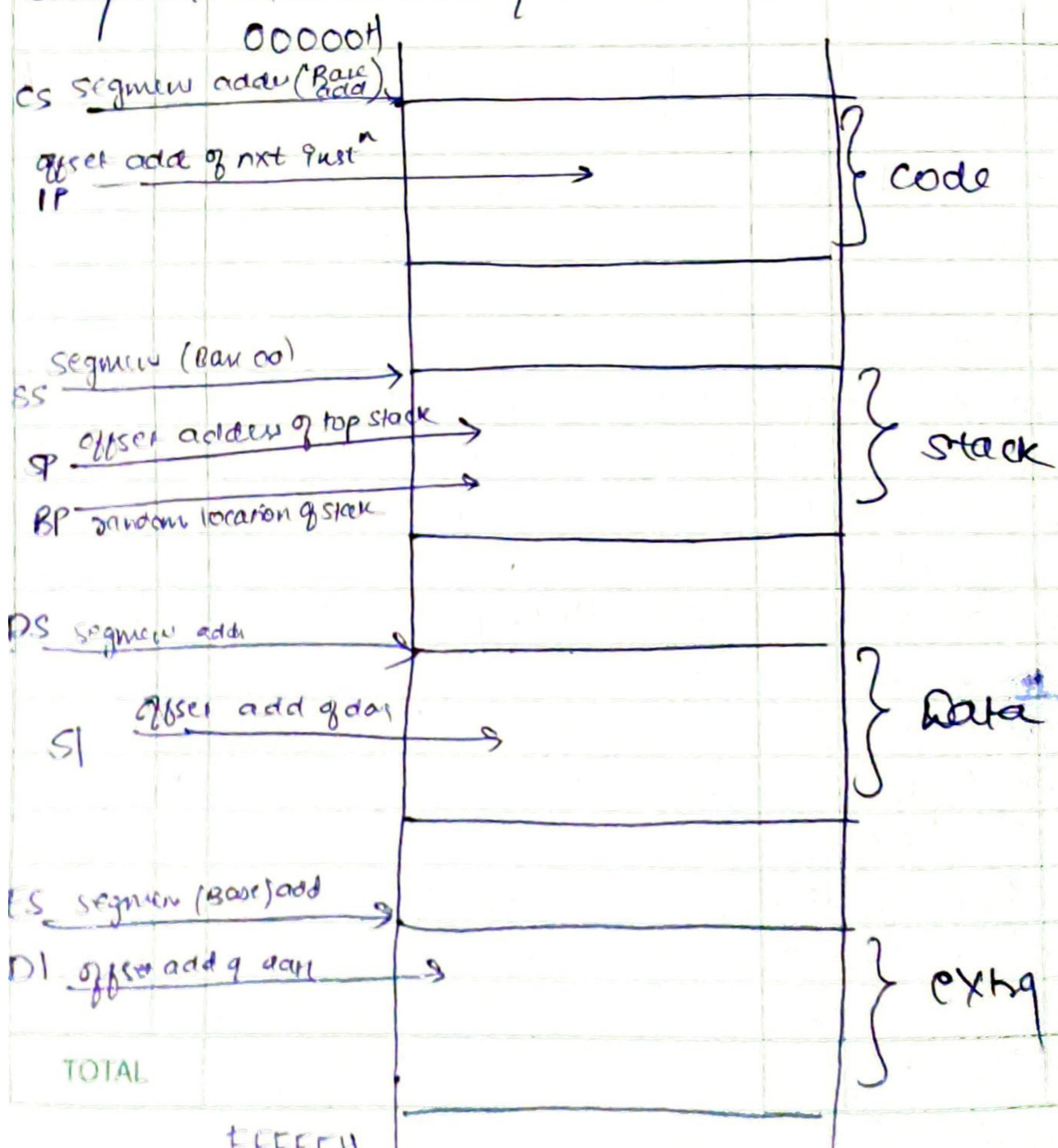
Advantages

It permits the programmer to access 1 MB using only 6 bit address.

It divides the memory logically to store instruction, data, stack separately.

Disadvantage

Although the total memory is $16 \times 64\text{ KB}$, at a time only $4 \times 64\text{ KB}$ memory can be accessed.



Microprocessor

Imp.

Q] Memory Segmentation and Advantages . 8086.

Bisection into code, stack, data, extra.

In 8086 architecture, memory segmentation is used to overcome the limitation of 16 bit address bw.

- 8086 has 20 bits of physical address.
- Here 20 bits of physical address is not byte compatible.
- To avoid the issue of byte compatibility microprocessor 8086 uses memory segmentation. In which we use concept of virtual address.
 - Here memory is bisected into 4 segments
 - Code Segment
 - Stack Segment
 - Data Segment
 - Extra segment
- To calculate physical address from virtual address we should know segment address and offset address.
 - segment address is held by segment registers (CS, SS, DS, ES)
 - offset address is held by offset pointers [IP(SP, BP), SI, DI]

TOTAL

Memory Banking In 8086.

- ⇒ 8086 has 16 bit data bus. This means it need to transfer 16 bit data in one cycle.
- ⇒ But we know that our location carries only one byte (8-bit)
- ⇒ 16 bit data will be stored in two consecutive location.
- ⇒ Hence, we need to devise a method to read two consecutive loc.

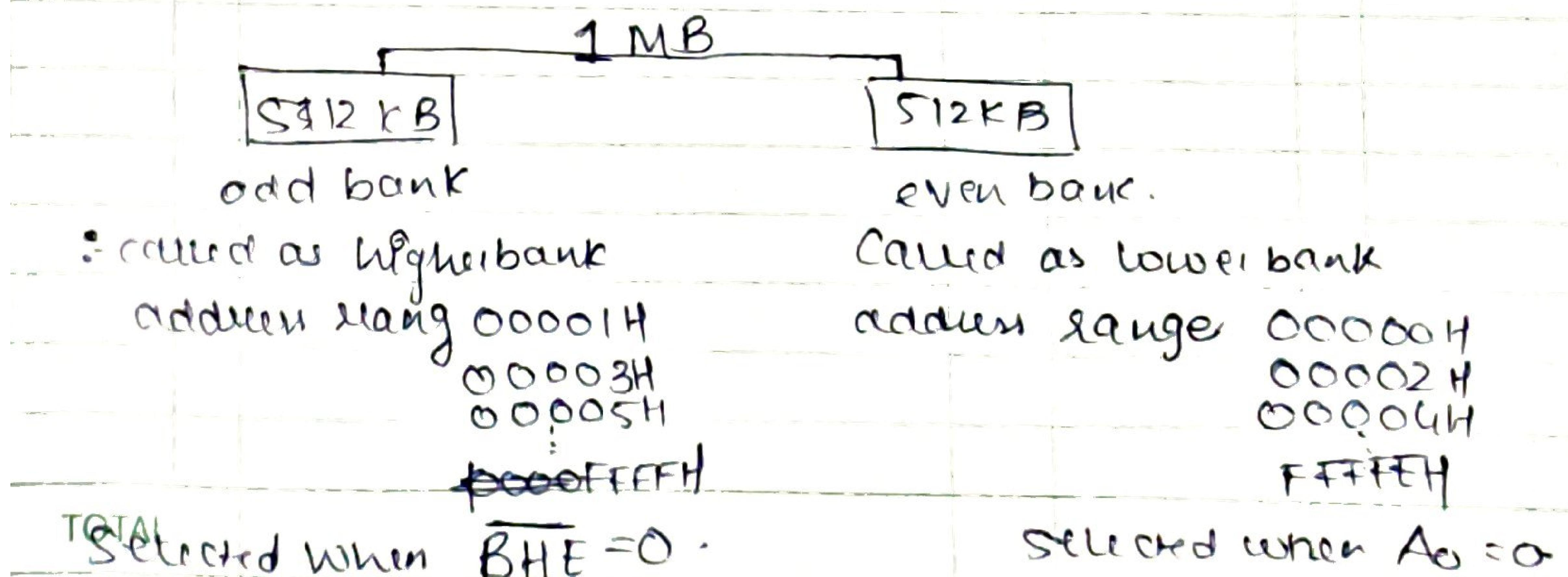
To do so memory PC divided into two banks. It has two separate chips:

- one chip has even locations 0, 2, 4, 6.. called Even bank
- other has odd location 1, 3, 5, 7.. called Odd bank.

Generally for 16 bit operation Even bank provide lower byte and odd bank provide higher byte

Hence even bank is called lower bank
odd bank is called higher bank.

- 16 bit data cannot stored in one location
- Reason is, if we make every location 16 bit width then it mean every operation will have to be 16bit operation.
But 8086 can perform 8bit as well as 16bit operation
consider programming wanting to store 8bit number like 25H
- If every location is 16 bit then we would have to occupy 16 bit location in storing 8bit.
- This simply means 8bit will get wasted.



BHE	Ao	Operation
0	0	R/W 16 bPIS from both bank
0	1	R/W 8 bPi from higher bank.
1	0	R/W 8 bPi from lower bank.
1	1	No operation (Idle mode)

Ao and BHE are used to select the banks.

8bit Operation on even bank.

e.g. MOV BL,[2000H]

This is an 8 bit operation.

BL register gets 8bit value stored at offset add 2000H.

Since it's 8bit operation MP will select Only one bank.

Since address is even MP will Select even bank.

Now Ao will be 0 and BHE will be 1.

8bit op on odd bank.

One bank

Odd bank

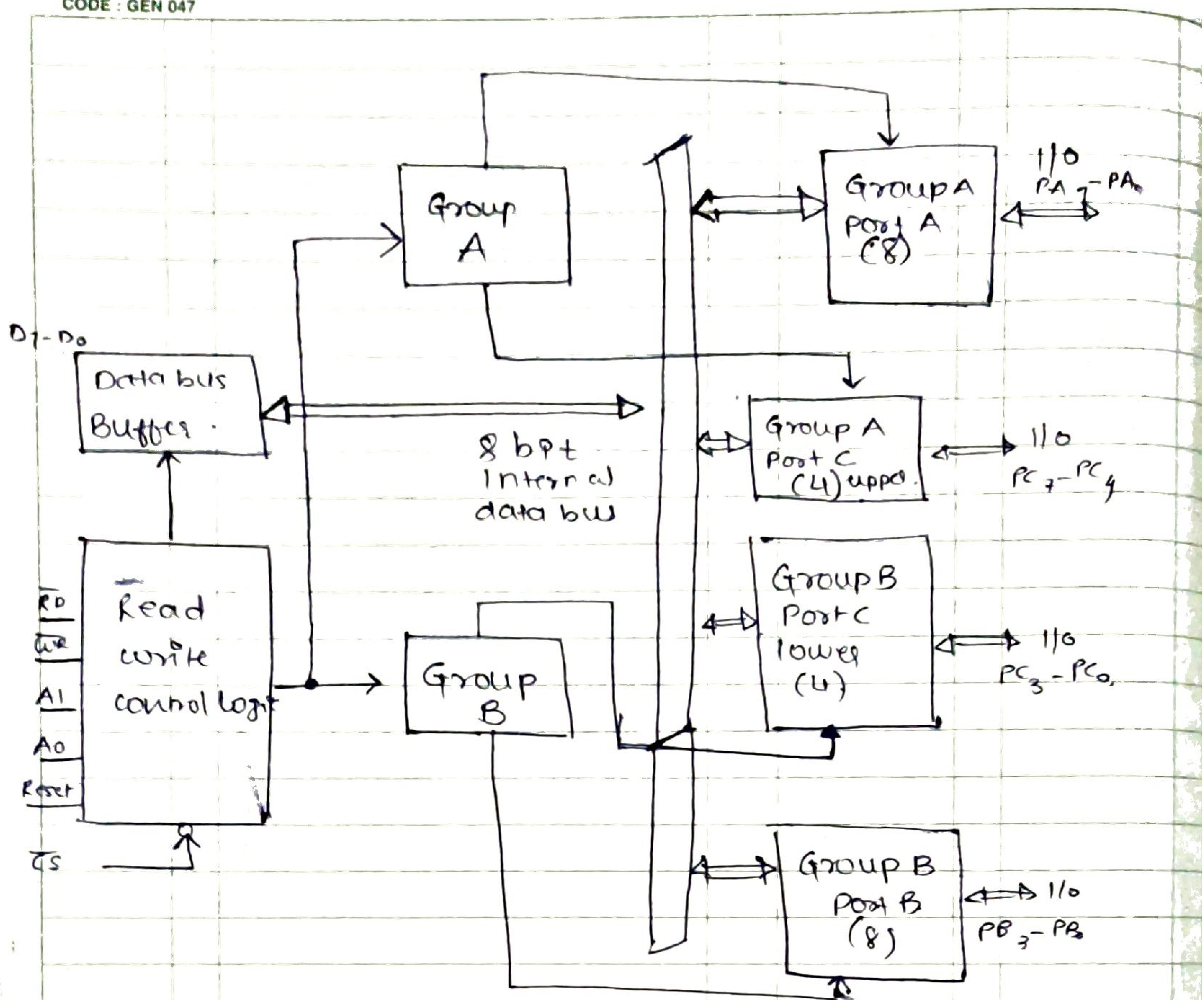
Ao will be 1 and BHE = 0.

Aligned / Misaligned operation :-

TOTAL

8255 Block diagram PPI

CODE : GEN 047



The architecture of 8255 has the following main components :

Data Bus Buffer.

This is a 8 bit bidirectional buffer used to interface the internal data of 8255 with external data bus.

The CPU transfers data to and from the 8255 through the data bus via this buffer.

TOTAL

Read / write logic

- It accept address and control signal from MP.
- The control signals determine whether it is read or write op.
- During read data will be transformed from 8255 to MP.
- During write data will be transferred from MP to 8255.
- There is a CHPP selection signal that select 8255 on basis of add.
- The reset signal is to reset 8255 and stop current transfer.
- There are two address line A1 and A0.
- These line are use to make partial selection within 8255.

A1	A0	A2	A1	Selection
0	0	0	0	Port A
0	1	0	1	Port B
1	0	1	0	Port C
1	1	1	1	control word

Group A control

- This control block control port A and Port C upper i.e PC7-PC4.
- It accept control signals from control word and forward them to respective ports.

Group B control

- This control block control Port B and Port C lower : PC3-PC0
- " "

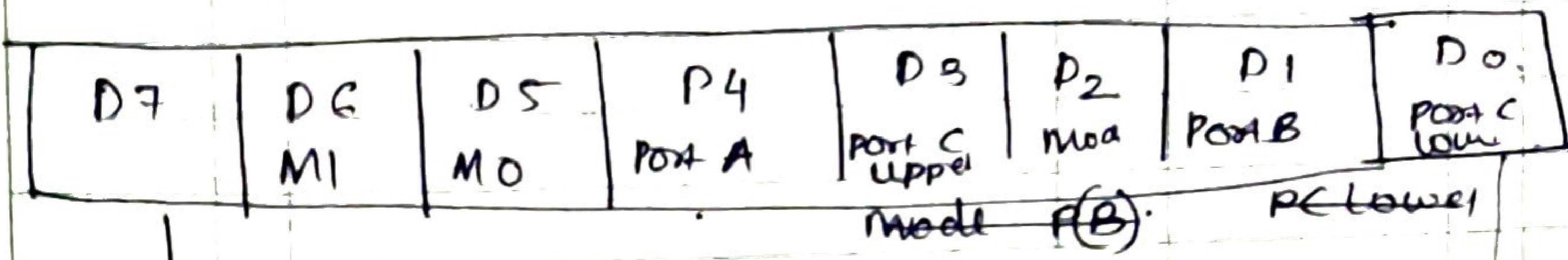
Port A, Port B, Port C.

These are 8 bits bidirectional ports.

Port	Mode 0	Mode 1	Mode 2
Port A	Yes	Yes	Yes
Port B	Yes	Yes	No
Port C	Yes	No (Hand sign)	No (Hang sign)

TOTAL

Control word Format 9855



D7 = 1 I/O mode

D7 = 0 BSR mode

D5 and D6 mode of group.

D6	P5		
0	0	0	mode 0.
0	0	1	mode 1
1	0	0	mode 2.
1	0	1	

D4 (Port A)

- D4 1 Input
- 0 Output

D3 (Port C Upper).

- D3 1 Input
- 0 Output

D2 (Mode of g1)

- 0 \Rightarrow mode 0
- 1 \Rightarrow mode 1

D1 (Port B)

- D1 = 1 Input
- 0 Output

D0 (Port C lower)

- D0 = 1 Input
- 0 Output

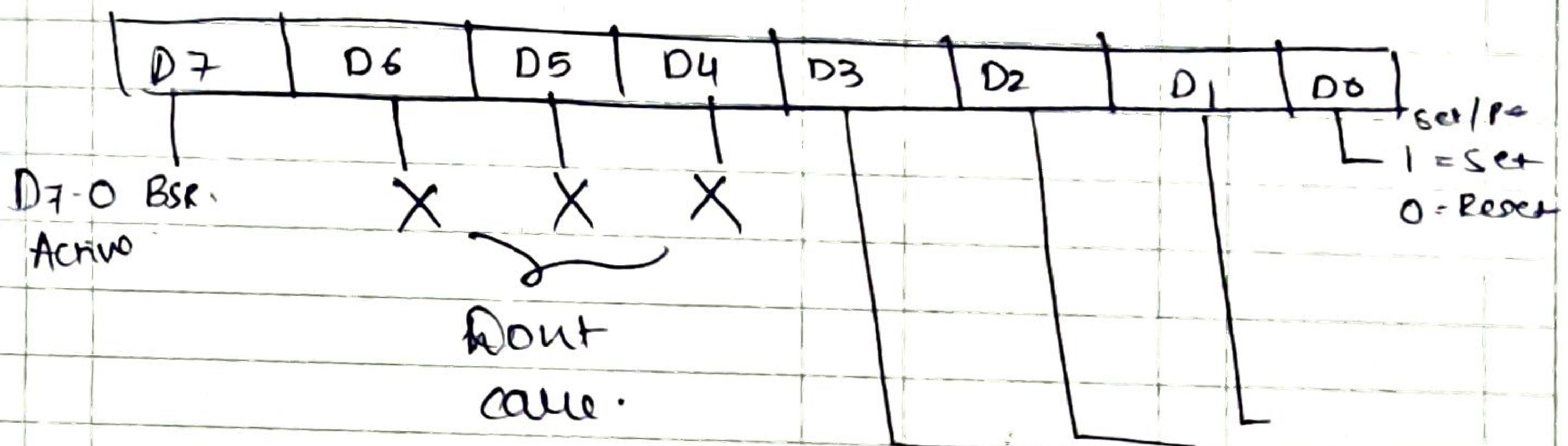
TOTAL

Aripan

control word format of BSR.

- ⇒ BSR mode is used only for port C.
- ⇒ In this mode Individual bits of port C can be set or reset.
- ⇒ This is very useful as it provides 8 individually controllable lines which can be used while interfacing with device I/O A to D converter.
- ⇒ The Individual bits is selected, and set/preset through control word.
- ⇒ Since D7 bit of control word is 0, the BSR operation will not affect the I/O operation of 8288.

Control word



TOTAL

Interrupt Structure of 8086

An interrupt is a special condition that occurs during the working of CPU.

The CPU services it by executing a subroutine called ISR.

There are 3 source of interrupt

① External signal (Hardware Interrupt)

These interrupt occurs as signals on external pins of CPU.

- 8086 has two pins to accept hardware interrupt NMI & INTR

② Special Instruction (Software Interrupt)

- These interrupt are caused by writing the software interrupt instruction INTn where 'n' can be any value from 0 to 255
- Hence all 256 interrupts can be invoked by software

③ Condition produced by program (internally generated interrupt)

- 8086 is interrupted when some special condition arises.

e.g. - An error in division automatically causes the INTO interrupt

Dedicated interrupt (0-4)

INTO :- (divide error)

This interrupt occurs whenever there is division error.

INT1 :- (single step)

The CPU executes this interrupt after every 2^{24} if TF is set.
It puts CPU in single stepping mode i.e. CPU pauses after executing every instruction useful while debugging.

INT2 :- (Non maskable interrupt)

The CPU executes this ISR response to an interrupt or NMI here.

INT3 :- (Breakpoint interrupt)

This interrupt is used to cause breakpoint in program.

TOTAL

INT4 (Overflow interrupt)

- This interrupt occurs if the overflow flag is set AND the CPU executes the INTO instruction (Interrupt on overflow).
- It is used to detect overflow errors in signed arithmetic operations.

Reserved Interrupts - INT 31

These levels are reserved by INTEL to be used by higher processor like 80386, Pentium, They are not available to the user.

User defined Interrupts INT 32 - ... INT 255

These are user defined interrupt
(IVT entry point (Backside))

Hardware Interrupt

8086 has two hardware interrupts NMI, INTR
NMI is non maskable interrupt (higher priority)
INTR is maskable interrupt (lower priority)

NMI

When NMI interrupt is activated, then action take place

- complete the current instruction that is in progress.
- push the flag register value on the stack
- push the code segment value and instruction pointer value of the previous address on stack
- IP is loaded from word location 00008H
- CS is loaded from ~~next~~ word location 0000AH.

INTR

On receiving an interrupt on INTR line, the CPU emits 2 INTA pulses.

1st INTA pulse - the CPU will decide vector number

2nd INTA pulse - send vector number, N to CPU

Now MP multiplies $N \times 4$ and goes to LVT to obtain ISR address.

If PS maskable by making IF=0 through CLI instruction.

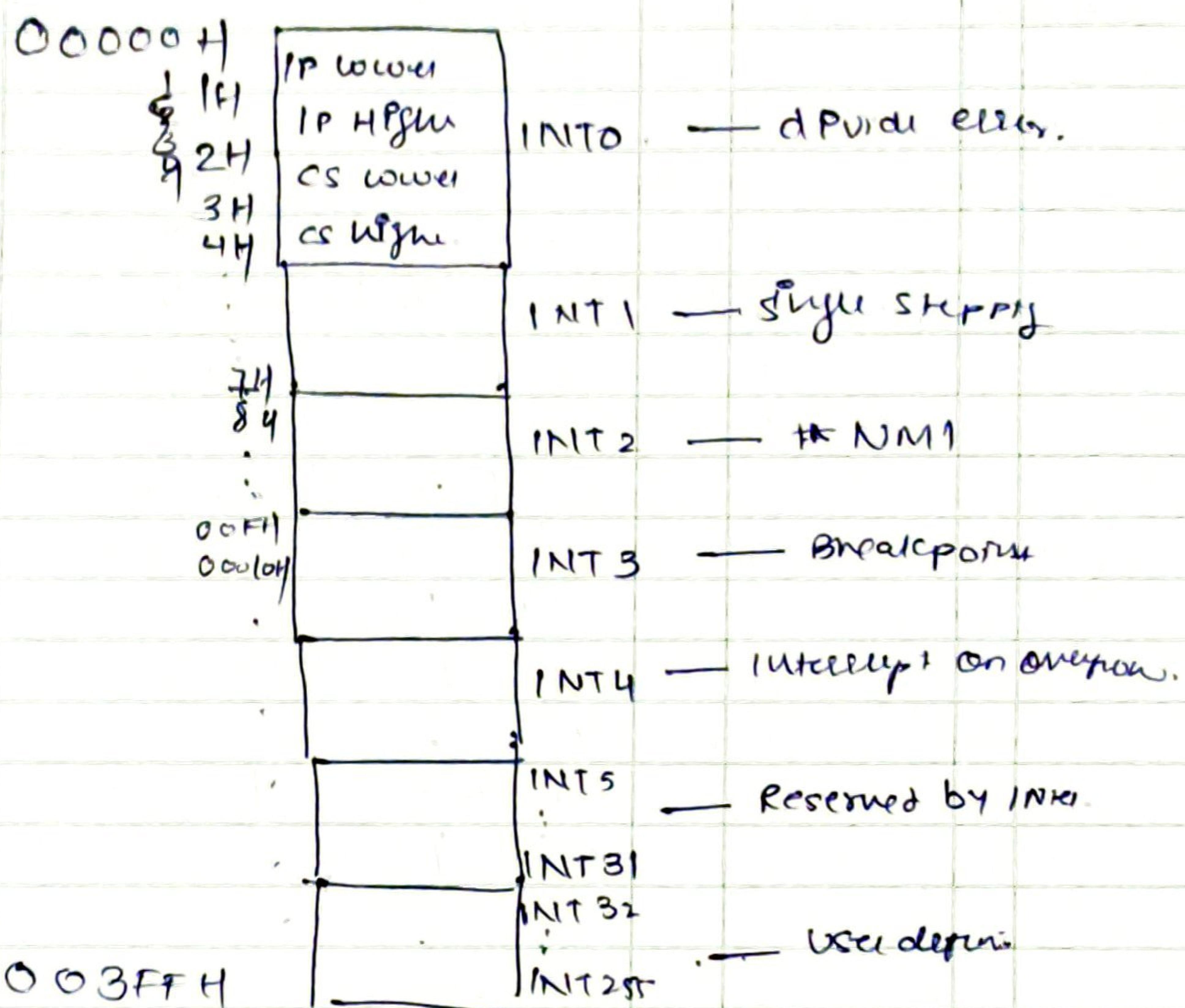
If PS nonmaskable by making IF=1 through STI instruction.

=

TOTAL

(IVT) - Interrupt Vector Table (Imp).

- The IVT contains ISR address for 286 Processor.
- Each ISR address is stored as .CS and IP.
- As each ISR address is of 4 bytes (2-CS, 2-IP)
- Each ISR address requires 4 location to be stored.
- The first 1KB of memory address 00000H - 0003FFH are reserved for IVT.
- Whenever an interrupt INTN occurs, CPU does NYU to get value of IP and CS from IVT. hence perform ISR.



TOTAL

MINIMUM MODE

- In 8086 microprocessor operate in minimum mode when $MN/MX = 1$
- In minimum mode, 8086 is the only processor in the system which provides all the control signals needed for memory operation.
- Hence circuit is simple but it does not support multiplexing.
- Other components like bus transceiver, latches, & clock generation, address decoder are also present in system.
- The address bus of 8086 is 20 bit long by this we can access 2^{20} bytes memory i.e. 1MB out of 20, 16 bits A₀ - A₁₅ are multiplexed with data bus.

Control Signals

Control signals are used to identify whether the bus is carrying valid address or not.

In which direction data is needed to transferred over bus.

When bus is a valid write done when to put read data on bus.

(Block diagram)

Latches 8282 (8bit)

The latches are buffered. They are used to separate the valid address from multiplexed address by using control signal ALE which is connected to strobe (STA).

Transceiver 8286 (8bit)

They are bidirectional buffers and also known as data amplifiers. They are used to separate valid data from multiplexed.

- Two such transceivers are needed because data bus is 16 bit long. 8286 is connected to DT/R' and DEN' signals.

They are enabled through DEN signals.

DEN	DTs	Action
1	X	Transceiver disabled
0	0	Receive data
0	1	Transmit data.

TOTAL

8086

8284 clock generator is used to provide clock
 $M/10 = 1$ then I/O transfer is performed over the bus
 $M/10 = 0$ min I/O operation is performed.

The signal RD' and WR' are used to identify whether a read bus cycle or write bus cycle is performing.

When WR' = 0, then PI provides valid output.

RD' indicates that 8086 is performing read data or I/O operation. During Read operation other control signals are also used namely UDEN (Data enabled) PI provides external device.

Control signals for all operation are generated by decoding M/10', RD', WR'.

M/10'	RD'	WR'	Action
1	0	1	memory read
1	1	0	memory write
0	0	1	I/O read
0	1	0	I/O write.

INTA and INTB

When INTB = 1, then there is an interrupt to 8086 by other devices for their service.

When INTA = 0, then PI indicates processor is ready to service them.

Timing diagram of minimum.

- All processor bus cycle is of at least 4 T states T_1, T_2, T_3, T_4 . The address is given by processor in T_1 state.
- In T_2 the bus is idle for changing direction, bus.
- The bus transfer take place between T_3 and T_4 .
- If the addressed device is slower, then wait state is inserted between T_3 and T_4 .

TOTAL

Maximum mode

- When $M_N/M_X=0$, 8086 work PS max mode.
- clock PS provided by 8288 clock generator.
- 8288 bus controller Address from address bus PS latched into 8282 8bit latch. Three such latches are required because address bus PS 20bit.
- The ALE PS connected to STB (strobe) of latch.
- The data bus is operated through transceiver. Two such transceivers PS required because data bus PS 16 bit.
- Write direction of data is controlled by DT/R signal DRA DIN PS connected to OE' and DT/R PS connected to T.

DEN	DT/R	Action
0	X	Transceiver PS disabled
1	0	receive data
1	1	transmit data

Control signals of all operation are generated by decoder S₁, S_{1'}, S₀ using 8288 bus controller.

S2	SI	SO	MP	control signal
0	0	0	Interrupt ACK	INTA'
0	0	1	Read I/O port	IOPC'
0	1	0	write I/O port	IOWC' and ALAR'
0	1	1	Halt	none
1	0	0	First" fetch	MRDC'
1	0	1	Memory Read	MRDC'
1	1	0	Memory write	MWTC' and AMWTC'
1	1	1	Inactive	none

Bus request PS done using EG/GT' pins integrated with 8086 RQ₀/GTO has more priority than EG, GT.
In max mode the advanced write signals get erased. This gives slower device more time to get ready to accept data therefore it reduces the number of cycles.

TOTAL

Advantages

- It helps to pull up more devices like 8087.
- It supports multiprocessing, therefore it helps to increase efficiency.

Disadvantages

- It has more complex circuit than MM mode.

MIN MODE

- It is a unprocessor mode.

8086 is the only processor in the circuit.

- Here MN/MX is connected to VCC.
- ALE for latch is given by 8086 itself.
- DEN and DT/R for transceiving are given by 8086 itself.

Control signals like M/IO, RD and WR are produced by 8086.

INTA for interrupt ack is produced by 8086.

Bus request are grant is handled using HOLD and HLDA signals.

Protocol is simpler but does not support multiprocesso-

TOTAL

does not need clock

MAX MODE

- It is a multiprocessor mode.

Along with 8086 there can be other processor like 8087, 8089.

- Here MN/MX is connected to Ground.

ALE for latch is given by 8288 bus.

As there are multiple processors DEN and DT/R for transceiving is given by 8288 bus controller.

Instead of control signal, all processor produce S₂, S₁, and S₀.

produced by 8288 bus controller

Bus request are grant is handled using RQ / GT sign.

The circuit is more complex to support multiplexing.

need clock

Addressing modes:-

Register addressing

Immediate addressing

Memory addressing - Direct, Register indexed, Register
Based addressing, indexed address.

String addressing

I/O addressing

Implicit addressing

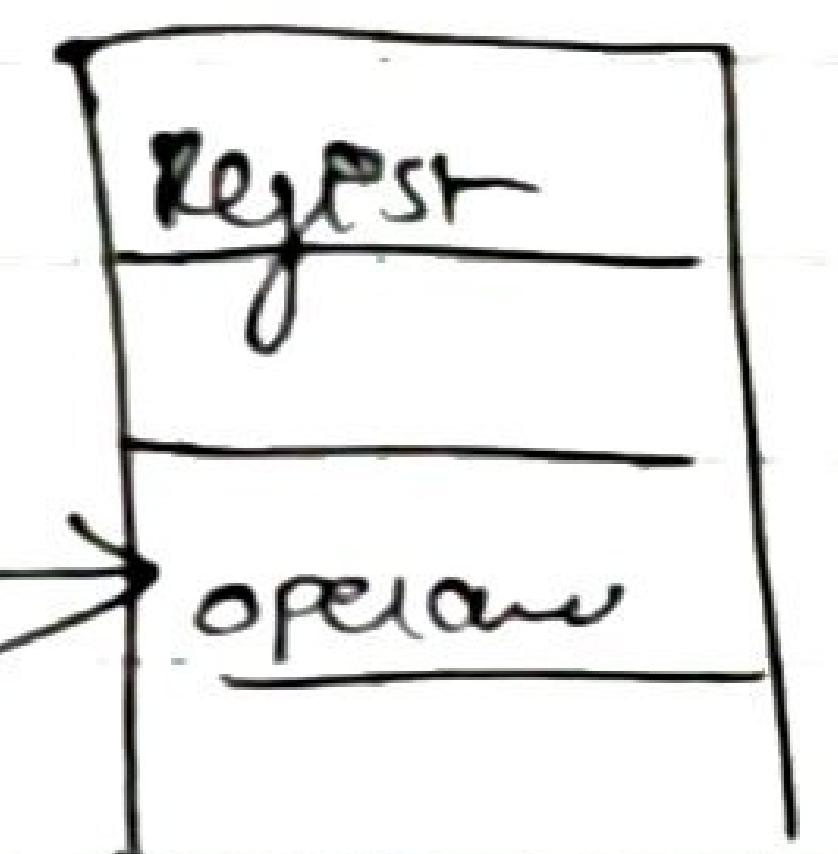
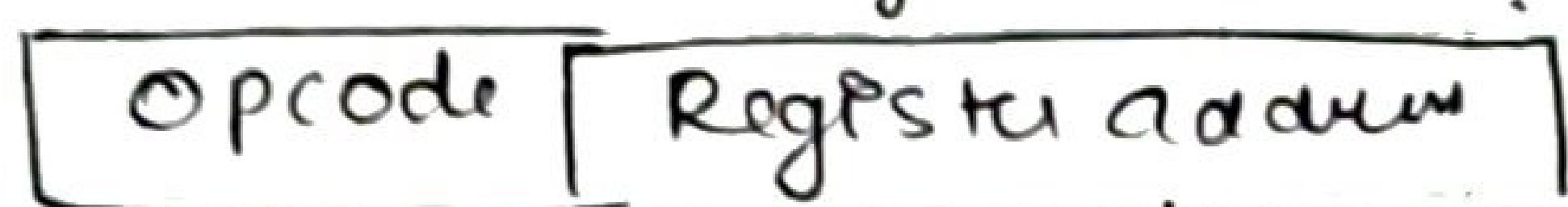
- Register addressing mode

Data is in register and instruction specifies the particular register

The Instⁿ will specify name of register which holds the data.

`MOV AX, BX`

The content of 8 bit register ~~BX~~ is moved to another register ~~AX~~



- Immediate addressing

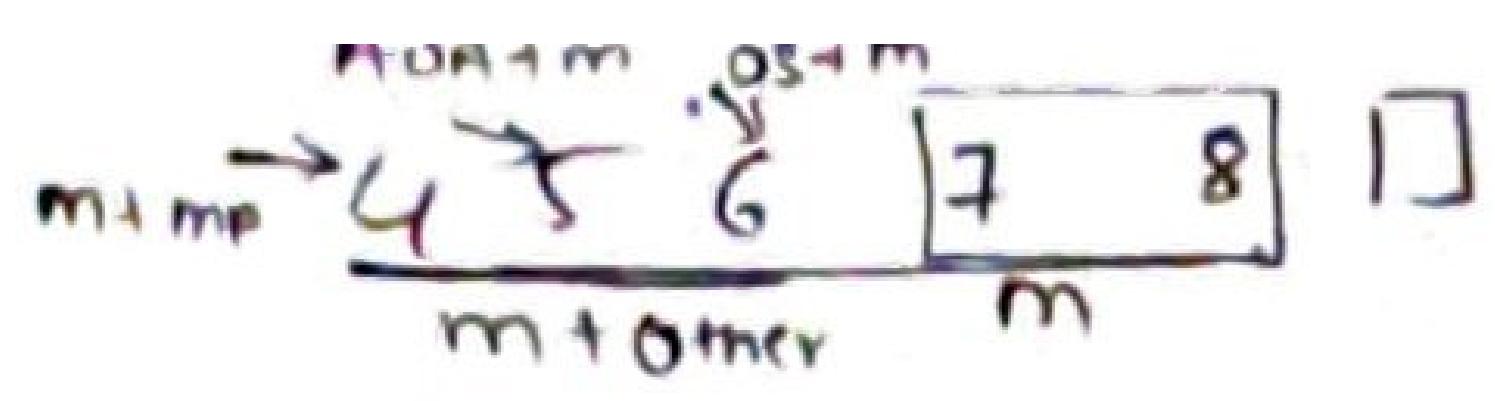
In immediate addressing mode, an 8 bit or 16 bit data is specified as part of instruction

Ex `MOV DL, 08H`

The 8 bit data (`08H`) given in Instⁿ is moved to DL
 $(DL) \leftarrow 08H$

`MOV AX, 0A9FH`

- The 16 bit data (`0A9FH`) given in Instⁿ is moved to AX
 $(AX) \leftarrow 0A9FH$



CODE : GEN 047

MOV A(CL), 4321

ASSUME DS = 5000H

PhysAdd.

DS * 10H + 4321

$$PA = \underline{5000} \times 10 + 4321$$

$$= 54321H$$

$$CL \leftarrow [54321H]$$

111000

5000 Bottom qps

DIRECT ADDRESSING

Here the effective address of memory location at which data operand is stored is given in the Inst^n.

The effective address is just a 16 bit number written directly in the Inst^n.

MOV BX,[13544]
MOV CL,[0400H]

13544H denotes the content of the memory location when executed the Inst^n will copy the content of memory location into BX Regist^r.

16 bit memory address (offset) directly specified in Inst^n.

~~Regstu Indirect addresses~~

MOV CL, [4321H] \rightarrow offset

Physical address cal :-

Assume DS = 5000H

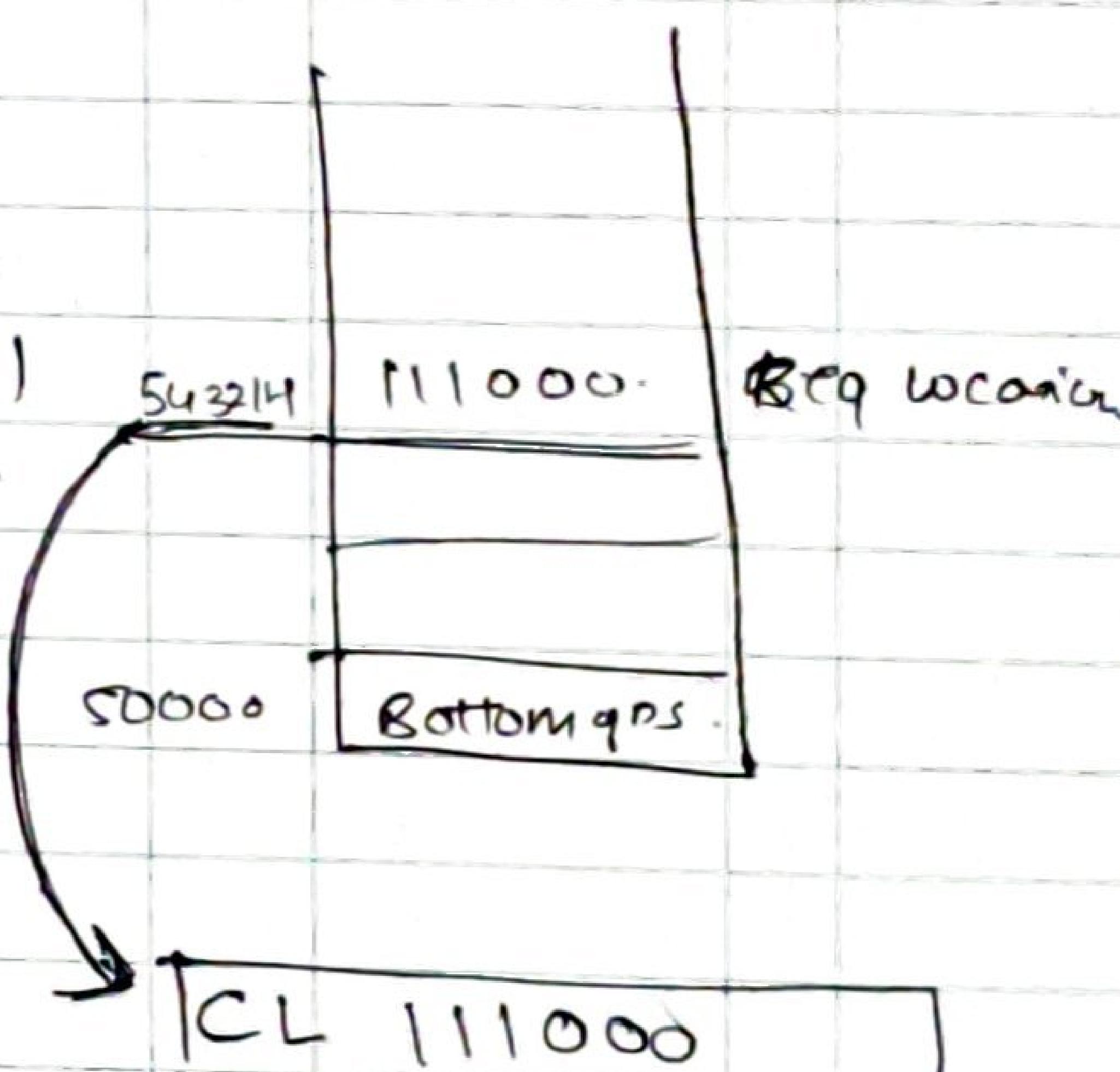
$$DS * 10H + 4321$$

$$5000 \times 10 + 4321$$

$$50000 + 4321$$

$$= 54321$$

$$CL \leftarrow 54321$$



TOTAL

Aug 12 - 1880 - Found in the soil at the will of Specie No. 12, just

Indirekt Addressierung mit

In this offset drawing Ps in either BX, SI, or
diff' signs & epth. DS or ES

(9) `MOV AL, [SI].`
Assume OS :- \$00C4
Assume SI :- 1102H (.)

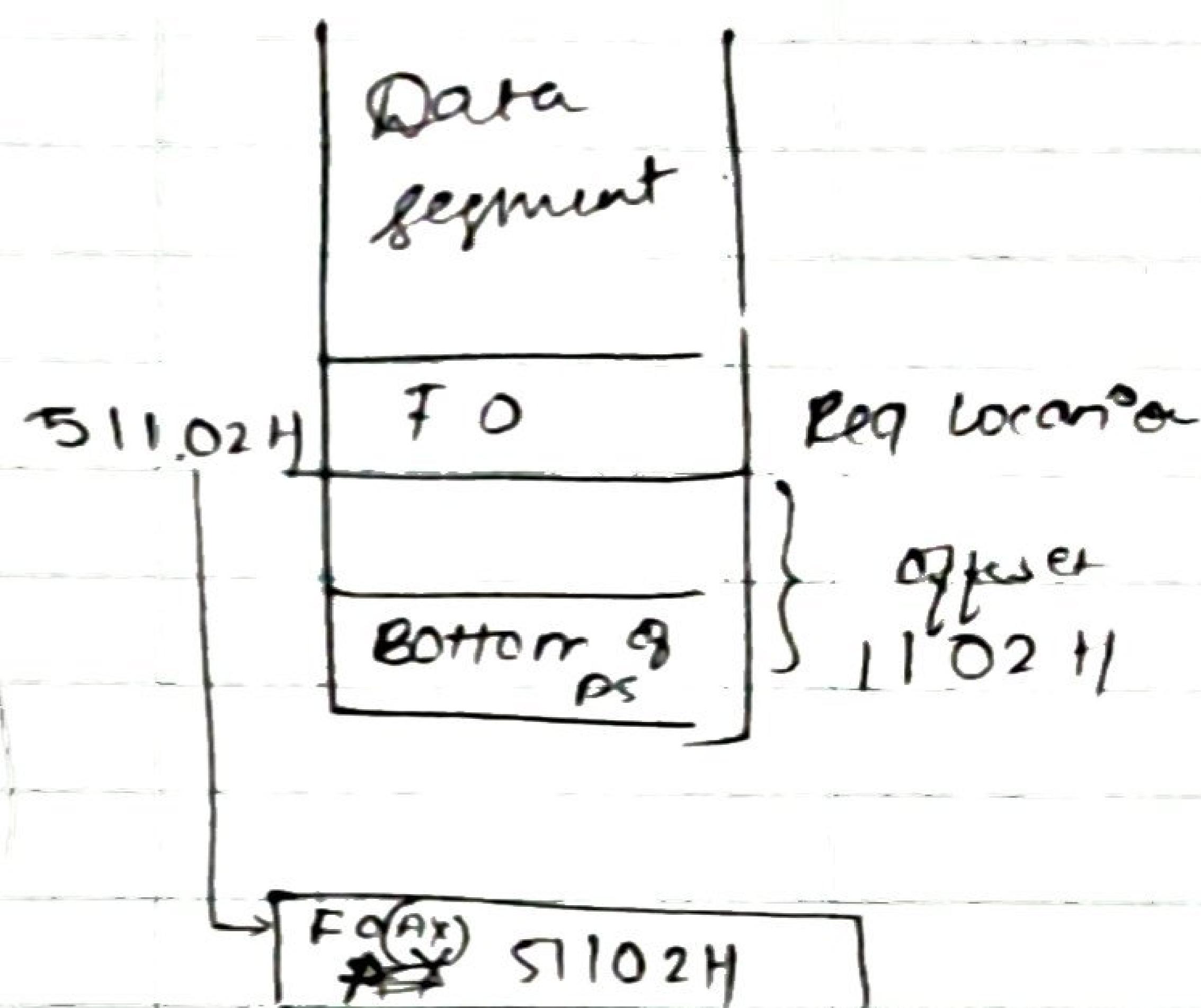
Physical cloud

$\text{DS}^+ \text{OH}^- + [\text{S}]\overline{\text{I}}$

$$p_1 = 50000 + 1102$$

= 51102 H

AL < [51102H]



Based addressing mode $(BX)(BP)$ Base

In Based address mode, **BX OR BP** is used to hold base value
of effective address.

signed 8 bit or unsigned 16 bit displacement will be specie

In case of 8 bit displacement In this mode EA
memory may be taken direct from one base register BX, BP.

Highly PS BX thin OS is by default segment register

Ex mov dl , [bx]

Assume PS = \$1000 H

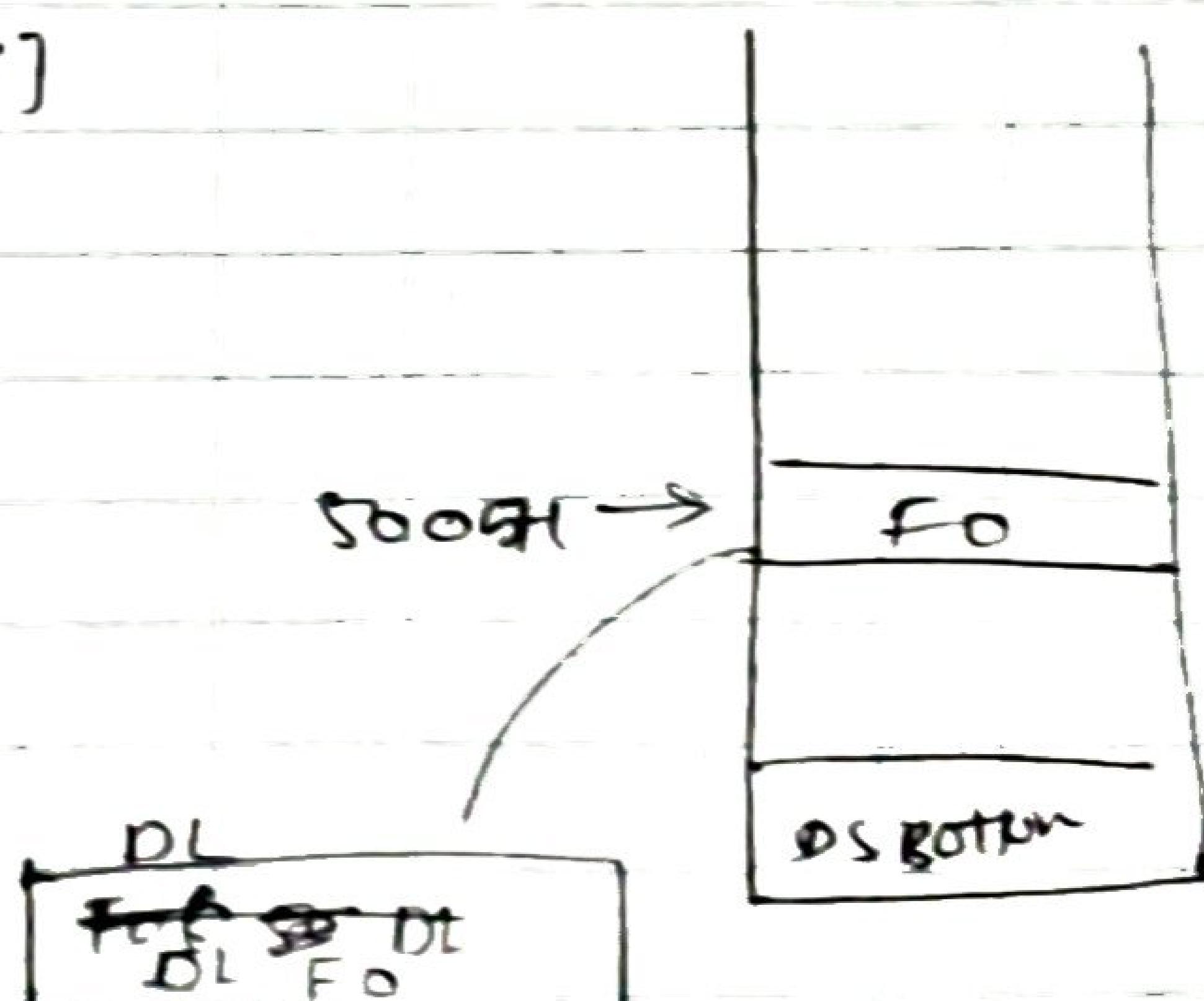
Assume $Bx = 0.005H$

DSY10H + 005H

50006 + 00054

STDOO 5+1

DL ≈ 50005^{±1}



TOTAL

memory Indexed addressing mode.

In this mode, EA of memory may be taken direct from index register specified by instruction.

If offset is SI then sum is by default segment register
DI the ES is segment register

MOV DL, [SI]

$$DS = 5000H$$

$$SI = 1002H$$

$$DS \times 10H + SI$$

$$50000 + 1002 = 51002H$$

$$DL \leftarrow [51002H]$$

Based indexed mode combination of above.

In this effective address is sum of base register and index register

Base reg: BX, BP
 Index: SI, DI

MOV LH, [BX + SI]

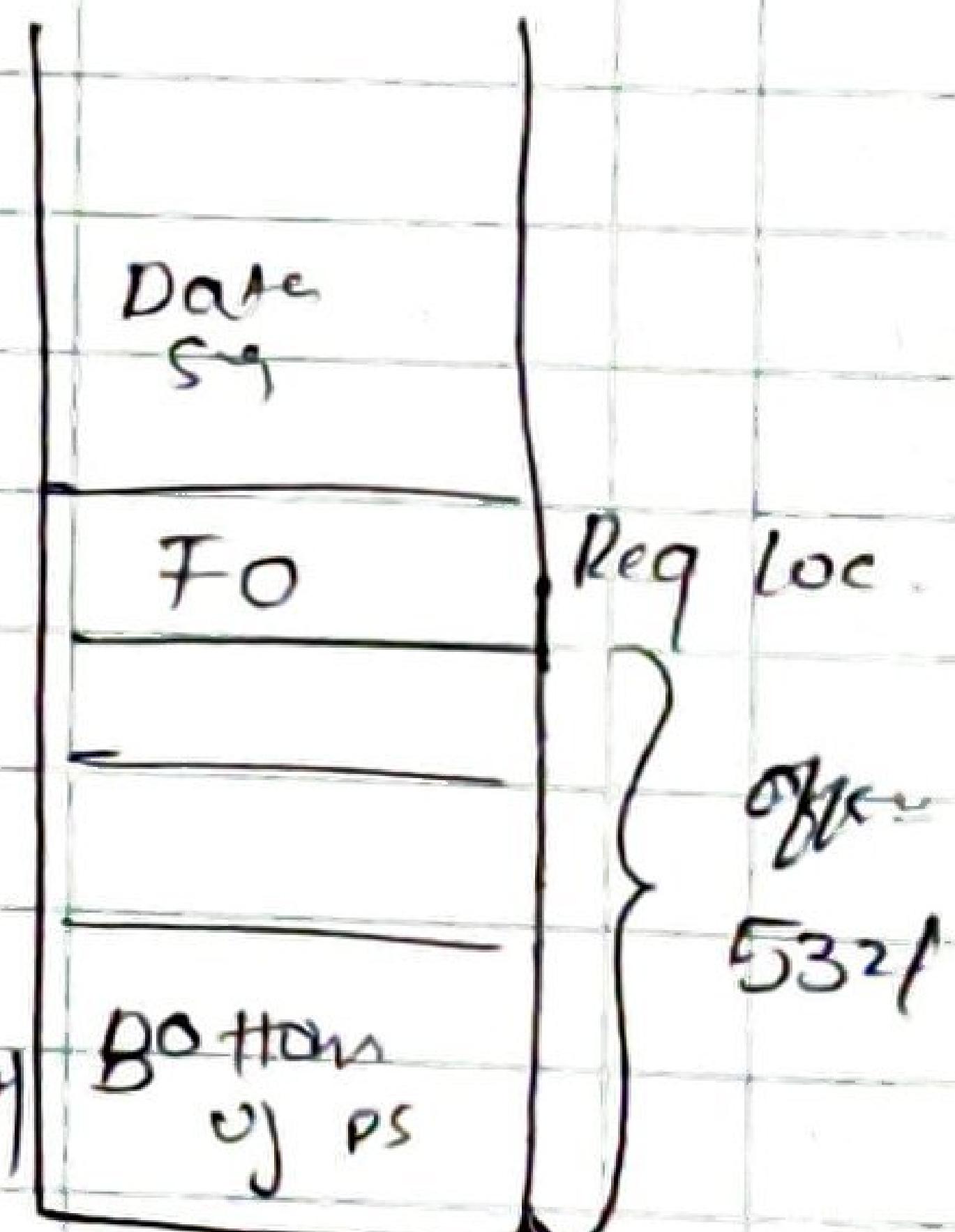
$$\text{Assume } DS = 5000H$$

$$SI = 4321H \quad ? \quad EA$$

$$BX = 1000H \quad J$$

$$4321 + 1000$$

$$EA = \cancel{5000} \underline{5321} \text{ (offset)} \quad 5000H$$



$$DS \times 10H + 4321 + 1000$$

$$5000 + 4321 + 1000$$

$$= 55321H$$

$$CH \leftarrow \underline{55321H}$$

TOTAL

Procedure and Macro

Procedure is a set of instructions stored as a separate program in the memory.

Procedure is used to perform specific task which can be called from main program whenever required.

Instruction used in procedure

AT CALL → call the procedure from main program.

AT RET → return back the main program.

Types of procedure

Depends on where the procedure is stored in memory

(1) Near procedure

(2) Far procedure

→ Recurrent procedure

→ Recursive procedure

[Near] In this procedure is present in same code segment where the main program is stored in memory.

- For near procedure, CALL instruction pushes only the instruction pointer (IP) register value on the stack that is no change in CS.

[Far procedure]

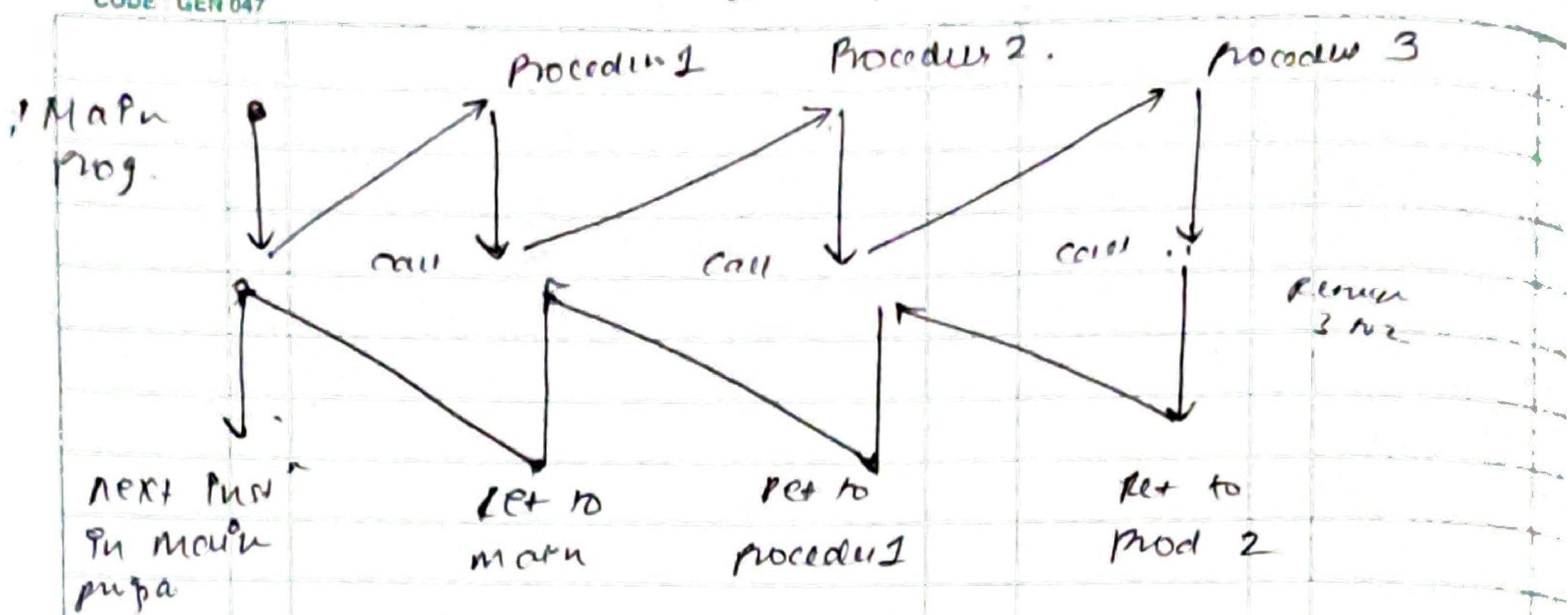
⇒ In this method, procedure is not available in same code segment.

- So call instruction pushes both IP and CS registers to call the procedure.

Recurrent procedure

In this method, the flow of program execution enters the procedure 1 from procedure 2 and then main program.

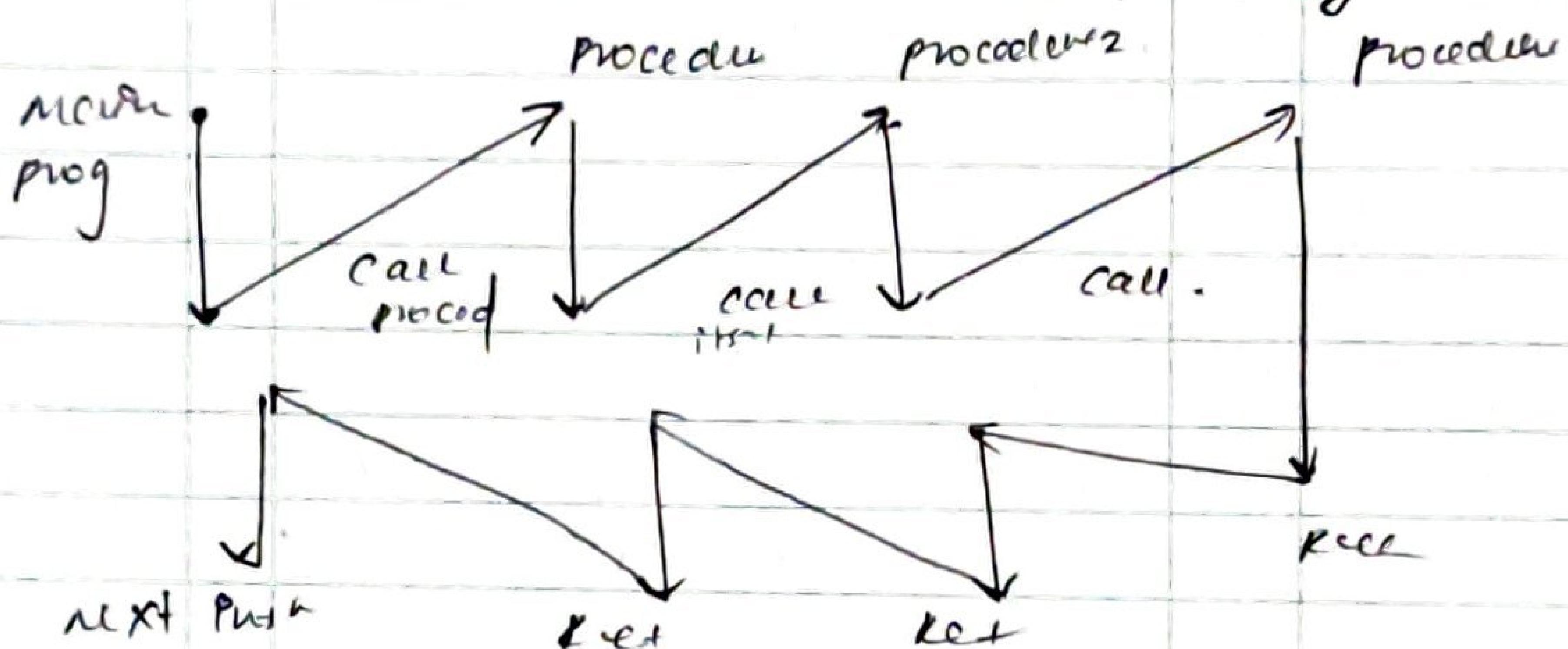
It is also called nested procedure.



* Recursive procedure

If f is procedure which calls itself.

same procedure can be called repeatedly.



If f is used to simplify the complex data structures.

Recursion depth 'n' → defines no of times procedure can be executed

Passing parameters in procedure:-

Procedure requires some data or address variables from the main program for its processing.

The data or address variables can be passed to the procedure using any one of following

- Using registers
- Using memory
- Using pointers
- Using stack.

TOTAL

Advantages

Allow to save memory space.

Program development becomes easy
debugging of error become easy
because size of program.

Disadvantage:-

- call and RET push^n are always required to put main procedure.
- require extra time to link procedure and return.

MACRO

Macro is a sequence of push^n that is written within the macro assembly directives.

- machine code can be generated each time macro is called.
- Parameters are passed as a part of statement which call the macro ~~prog~~.

Format for execution macro

```

INIT Macro      → Define macro
≡ }             Body of macro
ENDM           → End of macro
  
```

Advantages

call and RET push^n are not used in macro.

Program written with macro is more readable.

Execution time less because of no linking and return.
finding error or debugging or etc.

Dis.

For large group of push^n macro cannot be preferred.

Explain different data transfer mode of 8251

CODE: QPN 047
Explain DMA data transfer modes in 8251

Pentium Architecture

DMAC Architecture.
Direct memory access counter

- A single 8257 DMAC has 4 DMA channels (0-3)
- Four input / output devices are connected on these DMA channel (one for each)
- In the default priority mode, channel 0 is the highest and 3 is the lowest priority.
- Each channel has four components:
 - Address Register
 - DRQ
 - Counter Register
 - DACK

a) Address register (16 bit)

- It is used to store 16 bit memory address for DMA transfer.
- MP initializes the register with starting address of DMA.
- Through each byte is transferred the address get increment or decremented depending on mode.

b) Count register (16 bit)

- It is used to store 14 bit count of DMA transfer.
- The remaining higher two bits are used to define the mode of DMA operation.
- MP initializes the register with 14 bit count ($N-1$).
- Through each byte is transferred the count get decremented.
- This repeats till the count become 0.
- The higher two bits are used to select mode.

3 different modes:- DMA verify

DMA read

DMA write.

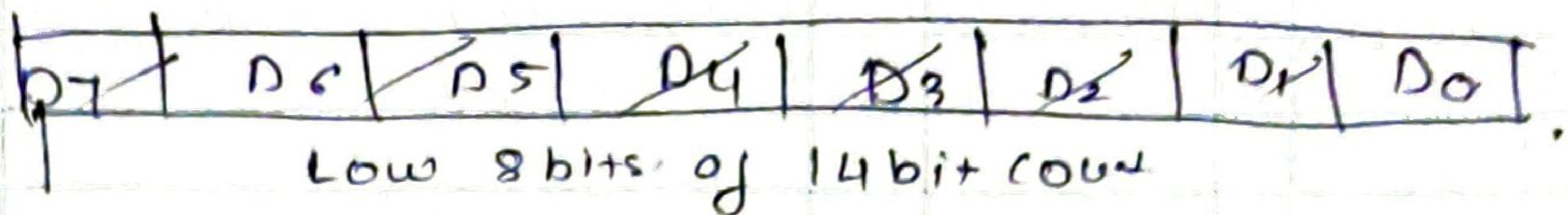
T ₁	T ₀	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
14 bit binary count															

TOTAL

DMA Read:

In this mode, DMAC becomes the bus master, it transfers data from memory to I/O.

hence in every transfer, the signals produced are MEMR and IOW.



mode	BPTS	mode select
0	0	DMA verify cycle
0	1	DMA write cycle
1	0	DMA read cycle
1	1	No action

DMA write

In this mode, when DMAC becomes the bus master, it transfers data from I/O to memory.

hence in every transfer the signals produced are IOW and MEMW.

DMA verify

In this mode, when DMAC becomes the bus master, it does not actually transfer any data.

This mode is just used to verify DMA process.

It will not produce any R/W signals to perform data transfer.

DREQ

I/O devices give this signal to DMAC to request DMA transfer.

DACK

It is given by DMAC to I/O device, indicating DMA transfer is being performed.

TOTAL

Priority Resolver

Priority is needed when several DMA channels get request from I/O devices.

PRIORITY RESOLVER decides which channel will be serviced first and which one will become pending.

There are 2 priority - Fixed Priority
Rotating Priority

Fixed Priority

- This is default mode.
- Channel 0 is the highest priority and channel 3 is lowest.
- Fixed priority cause Domination.
- If channel 0 and 1 keep requesting all the time channel 2 and 3 will starve this is called Domination.
- To avoid this we use Rotating priority.

Rotating Priority

- Here, once the channel is serviced it becomes lowest priority.
- All channel below itself by one position in priority order.
- As priority move in circular manner, it is called rotating priority.
- It gives every channel a fair chance.

HITn	CH0	CH1	Nyq
	CH1	C2	
	CH2	C3	
	Low	CH3	CH0

Read/write logic

It mainly provide Read and write signal as well as chip select signal.

The read and write signals are connected to TOE, TOW.
It also have A3...A0 address bus.

Control logic and mode set register

It generates the internal control signals for DMAC.

It also provide external signal such as HREQ, HLDN, AEN, ASTB, etc.

Data Bus Buffer

It connects external data bus of system with internal bus of DMAC.