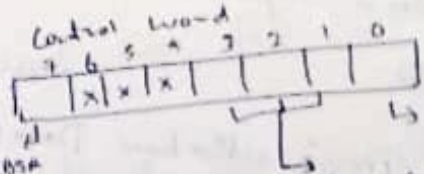
Part A

SET

①

Memory mapped I/O	I/O mapped I/O
⇒ 20 bit addresses are used to access registers which can be used for I/O operations	⇒ 8 bit / 16 bit addresses are used to access I/O ports for I/O operation
⇒ I/O can be transferred from any port to any register and vice versa	⇒ Only IN and OUT can be used to move values from Port to Accumulator
⇒ The M/IO Bar Pin will be active high	⇒ The M/IO will be active low

② To set Control word



Part-B

SSN

(11) (·) 8255 is a Microprocessor which can be used as a I/O operations aid.

(·) 8255 is a 40 Pin IC which mayorly consists of Three ports Port A, Port B, Port C

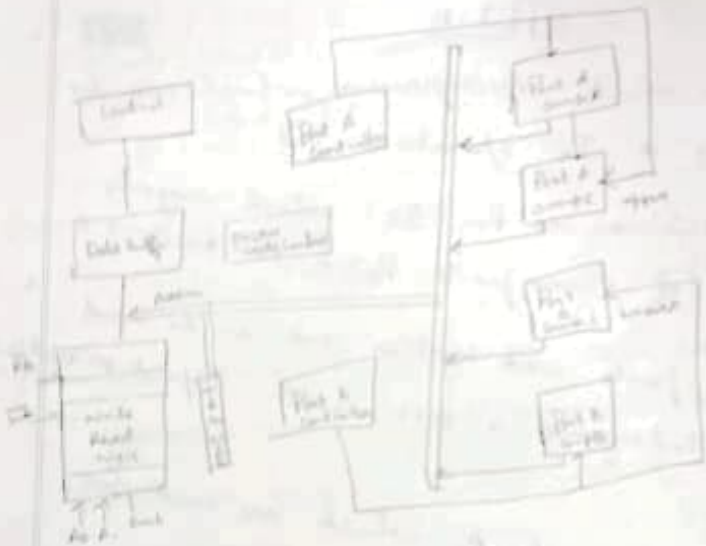
(·) All 3 ports work together and can be used when interfacing a 8086 with input or output devices

(·) It has Pin A_0, A_1 which are used for selecting ports

A_1	A_0	Port
0	0	A
0	1	B
1	0	C
1	1	CWR

(·) 8255 also has $\overline{RD}$ and $\overline{WR}$ pins which are set to active low to perform Read and write ie from Port and to Port respectively.

(·) 8255 has a reset pin which lets us to reset all the previous configurations



Different modes available in 8255

(•) Mode 0 Both Port A and Port B are considered to be two 8 bit Ports and Port C is considered to be a 2 independent ports

(•) All the ports can be set to Input or output

(•) Outputs are latched and Inputs are not latched

(•) No Handshake / Interrupt Capability

SSN

③ 2 chips of 32K × 8 ROM
4 chips of 32K × 8 RAM

ROM 1 & 2 — F0000 to FFFFF

RAM 1 & 2 — D000 to DFFFF

RAM 3 & 4 — E0000 to EFFFF

for ROM 1 & 2

	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Starting Addr:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ending Addr:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The common address are A_{19}, A_{18}, A_{17}, A_{16}, these are used for chip selection and A_{15} - A_0 will be used selection in ROM

For RAM 3 & 4

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Starting :	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
End :	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

chip selection bits are A_{19}, A_{18}, A_{17}, A_{16}

```
Segment cs
    finit
    fld x
    fld y
    fadd
    fadd ST(1)
    fst sum
End segment
```

(-) What happen in we are allocating values for variables x and y as 10 21 and 12 13 as we declared Dw for x and y it can have real numbers and integers

(-) In Code segment
we are loading x into stack, so it will take the position ST(0)

(-) Then we load y into stack which take ST(0) pushing x to ST(1)

(-) Then when we add ST(1), it adds ST(1) to ST(0)

=) In ST(0), ST(0) + ST(1) is stored

SSN

(·) IX is the index register

(·) MC is a register which points to a mask

(·) CC is the channel control

(·) When we want to use 8086 along with 8089 we can,

(·) increase the processing time

(·) increase efficiency

(·) Every time an I/O instruction comes, the coprocessor 8089 is activated using OUT command in 8086

(·) The 8089 processor wakes up and it waits for a request

(·) Once the request arrives, 8089 fetches the request and processes it

(·) Once it performs the request, it sends a signal to wake up the main processor

(16) We are going to Analyze 16 eight bit numbers : if they are even

So will take an array and traverse through it and check if its even or not

(i) If even increase the count

(ii) If not go to next number

Port A 08000

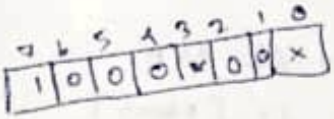
Port B : 08001

Port C 08002

control reg. 08003

Assume Port B and A, C are in mode 0

Control word will be



control word is (80)H

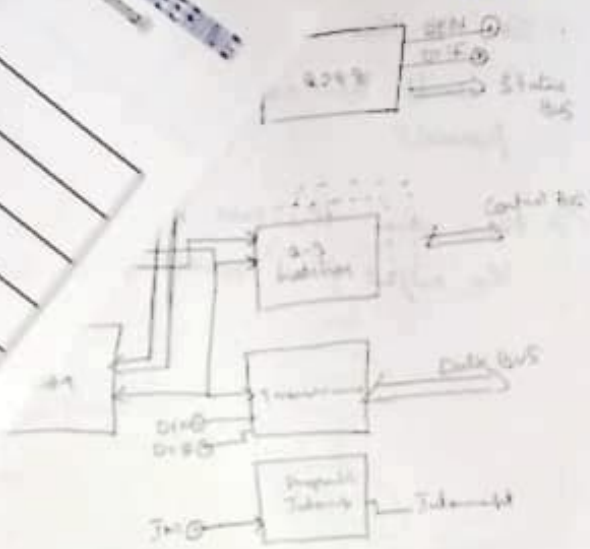
P.T.O for code and Diagram

SSN

In order to change Memory mapped I/o to I/o mapped I/o

the values of port A, B, C should be changed

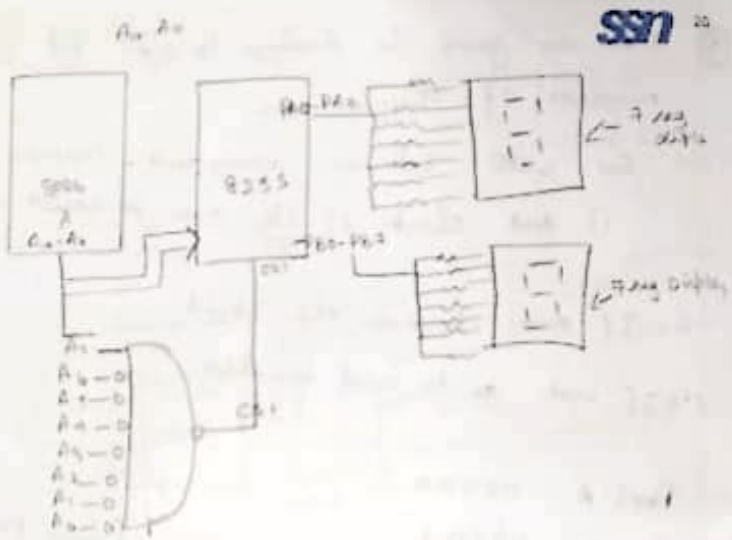
from 20 bit to 16 bit



- 8284 synchronized with same clock, Ready and Reset pin

(1) The RQ/GT can be used to get the Bus mastership

(2) The 8089 takes control of Buses when request is granted.



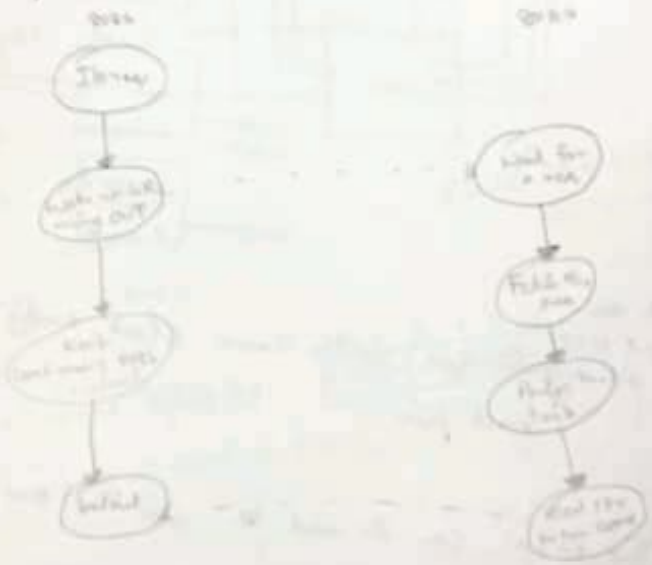
ALP:

```
MOV AL, 80 H
MOV CX, [0800 3]
MOV CX, AL
[illegible]
[illegible]
[illegible]
MOV SI, 2000
MOV CH, 10 H
MOV BL, 00
L1: MOV AX, [SI]
    DIV 02
    CMP AH, 0000
    JZ L2
    JNC L1
L2: INC BL
    JL1
MOV CX, [BL]
OUT A L, [0800 0]
OUT BL, [0800 1]
MOV AH, [CH]
MOV BH, [CL]
H
```

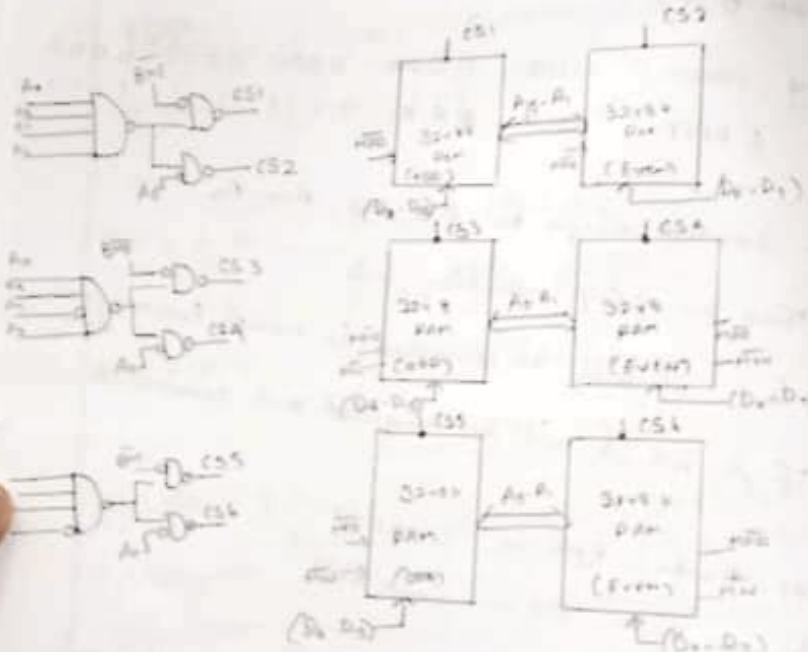
- There are multipurpose registers present which helps in storing and manipulating data
- DA is a register which is used to point to a source
- DB is a register which is used to point to destination
- BC is a register which points to a translation table
- TP is the Task pointer
- PP is the parameter pointer

(1) Once All this done, 8086 would be idle, it will be executing its 8086 instructions in parallel

(2) Then when the 8089 is done, 8086 waits and gets the output from 8089.



SSN



(9) MASM Code for 8087 addition

Assume Code Segment CS Data Segment DS

Code Segment DS

```
X   DBW  10 21
Y   DBW  12 13
```

End Segment

(4) Then we store the value stored in ST(0) to Sum.

fld x

fld Y

fadd ST(1)

fst Sum Sum ← ST(0) = X + Y.

Part-C

(10) (a) 8089 is an 80 pin microprocessor which can be used to facilitate I/O operations

(b) When a CPU has to perform an I/O operation it has to stop all the executing operations and do I/O

(c) Instead of that we use an additional which takes of I/O o/p, so that in the mean time the main processor can continue its execution.

For RAM D^3A_4

Starting : E0000 1110 0000 0000 0000 0000

End E FFFF 1110 dd'd: : : : : : : : : : : :

Same common chips are A_{16}, A_{15}, A_{14}, A_{13}

There are chip selection bits

if A_i is 1 it will be connected to RAM in normal form

if A_i is 0 it will be negated and connected

For odd Banks, BHE is connected

for even Banks A_0 is Connected

For ROMs only $\overline{RD}$ is connected to the component

But for a RAM both $\overline{RD}$ and $\overline{WR}$ are connected

(-) All the components share the same Data Bus. (D_0-D_7) for even Banks

and (D_8-D_{15}) for odd Banks

Mode 1 Handshake Capability

(·) Port A and Port B are considered to be 2 independent input or output ports.

(·) 3 Bits of Port C is allocated to each Port A and Port B for handshaking

(·) The remaining 2 bits of C can be used for I/O operations

(·) In this mode Value of Port C in control word can be don't care because they will be used for handshaking

Mode 2 Bidirectional Operations

(·) Port A and B are assumed to be 2 independent I/O ports

(·) 5 bits of C is allocated to A for Bidirectional flow

(·) 3 bits of C can be used by Port B for handshake or even can be used as I/O port when Port B is in mode 0.

(·) Port A can be used as both input and output device.

Mode	Ports that can be in
0	Port A, Port B, Port C
1	Port A, Port B
2	Port A.

(7) The 'Test' pin of 8087 is connected to SSN Busy Pin of 8087

→ When 8087 is given data with the execution of an instruction, it makes the Busy Pin active high

This will make the Test pin deactivates the Test Pin

(8) MOV AL, 80
OUT control-reg-add-of-8255, AL → ①
MOV AL, 0F4
OUT Port C, AL → ②
MOV AL, 07
OUT Control-reg-of-8255, AL → ③

So after executing snippet 1, all the ports will be resetted and will be input ports

After snippet two, Port C will be assigned value of 1

But again after the three snippet 3, the ports are resetted and C will be set to input Port

SSN

Final status 'A' C will be set to
input Port

(b) Closely coupled config

(1) The main processor and the coprocessor both work in the same clk, RESET, READY bus

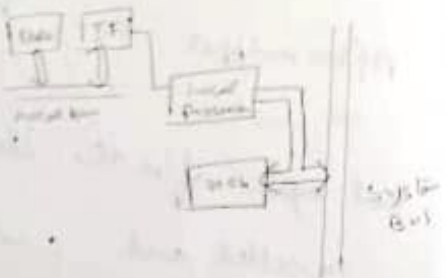
(2) Both main processor and coprocessor fetch their own instructions and execute

(3) Both processors use the same address and data Bus

Loosely coupled config

(1) The main processor and the coprocessor have their own set of buses

(2) Both / many processors fetch their instruction from their respective Buses and execute and finally share the same system Bus with main Processor



SSN

(A)

To set PC7 : Pins are 111

Control word : | 0 | x | x | x | 1 | 1 | 1 | 1 |

Value : $(0F)_H$

---

(3) The Bus controller in Maximum mode in 8086 is responsible

In Max Mode Status pin S_0, S_1, S_2 will provide

the necessary values to function

S_0	S_1	S_2	
0	0	0	INTA
0	0	1	I/O Read
0	1	0	I/O write
0	1	1	HLT
1	0	0	Mem Read or Inst fetch
1	0	1	Mem write Read
1	1	0	Mem write
1	1	1	Passive

The Bus controller is responsible to produce

ALE and $\overline{DEN}$

Only when $\overline{DEN}$ is active low, Data can be

written or read.