8086 & 8087 interface

Q. Why do we connect these two µPs?

A: (1) 8087 is a numerical data processor.

⇒ i.e. designed to perform complex arithmetic operations: such as:

√, ~~floating point numbers~~, log, trigonometry.

→ it has instruction set works fabulously.

⇒ all things that are available in a scientific calculator.

(2) 8087 works with very large numbers which is 80 bit.

(3) Both 8087 & 8086 works simultaneously i.e. multiprocessing, so you get high performance.

(4) Also, it works with floating point numbers, since integer part is not enough for operations like log, square root etc, we need floating point numbers also.

At first, you need to identify all the chips.

→ 8284 clk generator = provides clk to the system

→ 8282 Latch = capture the address lines

→ 8286 TRx = capture the data lines

→ 8288 Bus controller = generates the control signals.

→ 8259 PIC = used to increase the num. (new entry) of interrupts.

Any device which wants to interrupt, will not have a direct connection to µp.

① → so, devices interrupts through $IR_0 - IR_7$

② → Then, 8259 PIC will send the signal to µp's INTR line. (not NMI) NMI is vector. i.e. it can run only one ISR. INTR is non-vector, so. it can be any ISR.

(3) Then. up will ask the vector number via 8288 bus controller..

→ ~~who~~ here. who wants to interrupt up?

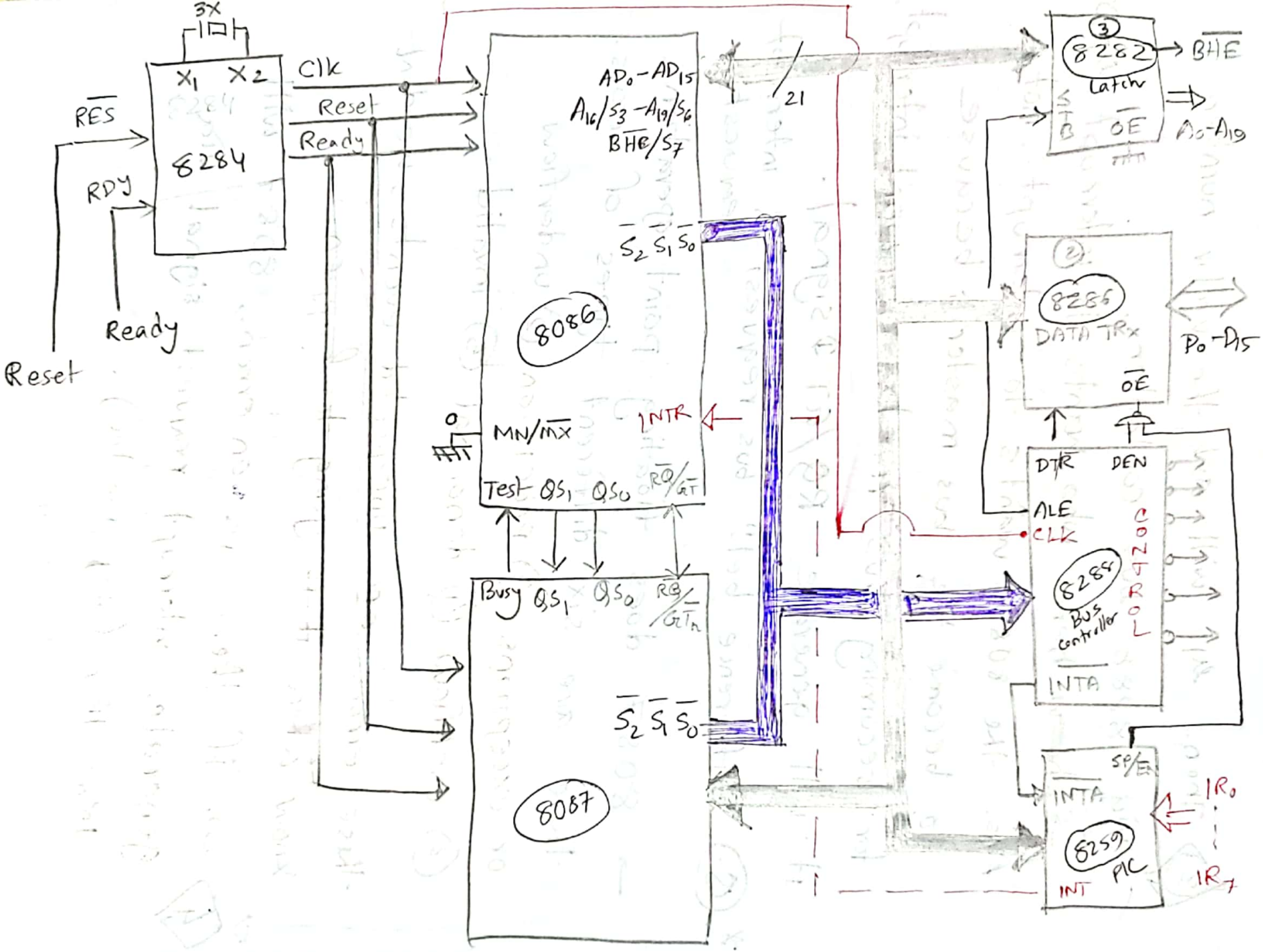
Ans: the 8087 wants to interrupt. not to become the bus master., because for becoming bus master; it won't interrupt, it will generate RQ/GT ↕ signal.

* (8.) difference bet^n bus request & interrupt request.

→ 8087 does floating point operations there are six different types of errors. or exceptions. (1) precision (2) underflow (3) overflow (4) denormal (5) invalid

these errors are unexpected events., we dont know when it is going to happen.

(4) So, if there is an error., 8087 will generate interrupt request signal via $IR_0 - IR_7$ (any one line).



1. Clk pulses

① Clk → is used triggered for changing the current state. (used for every operation)

② READY → is used to synchronize µp with slow devices.

③ RESET → is used to reset the system.

→ 8087 also a µp. & hence also needs triggers, so it needs the same clock & reset.

→ When 8087 is the bus master & it is interacting with circuit, it needs to know whether the device is ready or not.

2. Buses

Now, you need to generate 3 buses.

→ Add.
→ data.
→ control.

☐ In any operation, 1st we give address, then we give data.

☐ For interrupt occurs

→ 8087 gets an error, nearest signal immediately to 8259 PIC via (IR_0-IR_7) lines

→ 8259 get the nearest & send the signal from INT to INTR of µP.

→ Then, 8086 µP acknowledges via the bus controller $\overline{INTA}$ twice. by $\overline{S_2}\,\overline{S_1}\,\overline{S_0}$ = 000 (INTA operation)

At first, it ask the vector number

At 2nd, it gets the vector number or ISR address. so, 000 will be generated twice.

→ When, 8259 PIC sends the vector number to µP, 8286 TRx, might get confused whether it is data or address.

→ µP after getting vector number will multiply by (4) for executing Interrupt Service Routine.

→ So, 8259 PIC deactivates (simultaneously) the TRx via $\overline{SP}/\overline{EN}$ pin.

$\overline{SP}/\overline{EN}$ = Slave program / Enable (its a dual purpuse sign

→ here, we don't need ~~SP~~. we need $\overline{EN}$.

→ So, via Enable 8259 deactivates the buses of

⇒ 8286 ; as a result. $D_0 - D_{16}$ also cannot interfer

⇒ clk of 8288 says at which T state, operation needs to be executed.

⇒ $\overline{S_2}$ $\overline{S_1}$ $\overline{S_0}$ status signal tells which control signal has to be generated.