



PINS REQUIRED WITH EXTERNAL MEMORY INTERFACE.

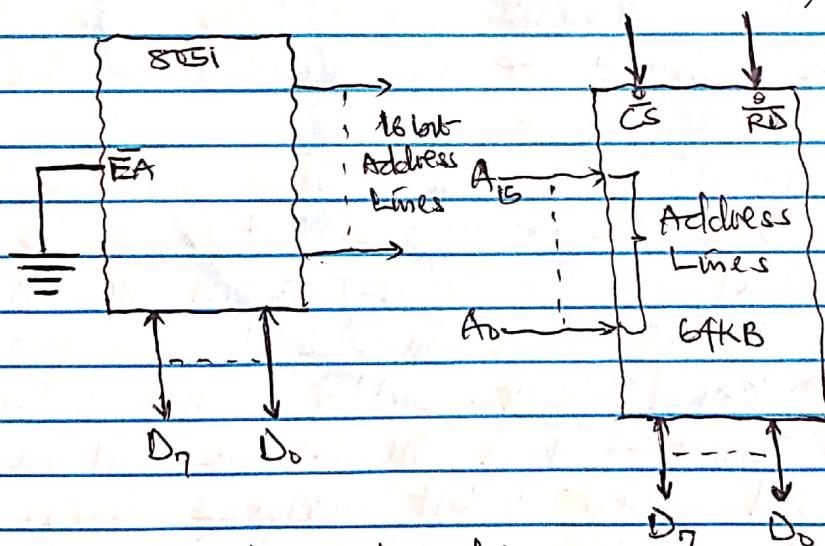
EA [External Access]: If ~~connected~~

If we have connected one external code memory then

$\bar{EA} = 0$ (GND); If External Code Memory is Connected
 $= 1$ (Vcc) ; If no external Code Memory is Connected Using only Internal Code Memory

The \bar{EA} pin must either be connected to ground (0) or connected to Vcc we cannot keep it floating.

If $\bar{EA} = 0$ is connected to ground this indicates that we have connected one code memory code memory ROM



To access 64KB Memory Locations we require 16 address lines from A₀-A₁₅

The Microcontroller should be able to generate 16 bit address lines.

16 Bit Address lines

8 Bit Data lines

24 pins

We require 24 pins to activate the address and receive data from Code Memory.

If we connect $\overline{EA} = 0$ (GND) this indicate we have connected one external code memory which is the microcontroller. In this code memory we know that there is one chip select pin, Read Line pin, 8 data lines from $D_0 - D_7$, Address Lines.

The maximum code memory is 64KB. To access 64KB key locations we require 16 bits address lines.

Now suppose we have connected this 64KB code memory the address lines will be 16 bit address line from $A_0 - A_{15}$. To access any memory location in 64KB we require 16 bit Address Lines. So, this means that from our microcontroller we must be able to generate 16 bit address and through this address the address location will be activated ~~at 16 bit~~ and from that location we have 8 bit data that is available at $D_0 - D_7$ and from this data line it has to somehow enter into the microcontroller $D_0 - D_7$ which are connected to the data lines of the code memory $D_0 - D_7$.

Basically, we require 16 Bit Address bus & 8 bit Data bus in total we require 24 pins to activate the address and receive data from memory. Now to save the number of pins in the microcontroller the designer multiplexed some address and data lines. Out of the 16-Bit address bus there are higher 8-bit address lines and lower 8-bit address lines. ($A_8 - A_{15}$) Higher 8-bit address lines, ($A_0 - A_7$) Lower 8-bit address lines.

So, what the Intel (designers) did they multiplexed 8-bit data lines to lower 8-bit address lines. In place of writing the address lines as $A_0 - A_7$ I will write as 8-bit Address Data Line $AD_7 - AD_0$ ($AD_0, AD_1, AD_2, AD_3, AD_4 \dots AD_7$). Now these 8-bit lines are multiplexed address data lines. The meaning of multiplexed address data lines is that address and data both will flow but at different time interval. Multiplexed type of signals are flowing one is address type of signal



and the other data type of signal at the same time simultaneously cannot be permitted. Address flows first and data will flow on the same line.

The advantage of this is that we can use just 16 pins instead of 24 pins this is able to reduce the size of the microcontroller. The AD₁₅-AD₈ are dedicated to address bits while AD₇-AD₀ are dedicated to 8 address and data bits.

Port-D → Can be used for I/O purpose. If there is multiplexed address and data lines so the 8 pins of port-D will carry AD₇-AD₀ (P0.7, P0.6, P0.5, P0.4, P0.3 ... P0.0)

P0.7 → AD₇

P0.6 → AD₆

P0.5 → AD₅

:

:

P0.0 → AD₀

For high 8-bit address lines we use

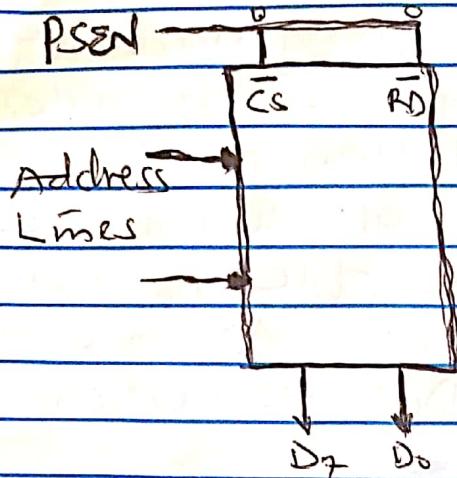
Port-2 → A₁₅-A₈.

PSEN (Program Store Enable): This pin is active only if EA is 0 (GND). This means when we connect external memory then the PSEN pin will be active generating signals from PSEN

$$\overline{\text{PSEN}} = 0 \quad (\text{If } \overline{\text{EA}} = 0)$$

If $\overline{\text{EA}} = 0$ the PSEN pin generates a zero voltage and if $\overline{\text{EA}} = 1$, PSEN will generate a 1 voltage.

PSEN is used to enable code memory and read data from code memory. Say for example we have our code memory and CS=0 and RD=0 are connected to PSEN. In this condition PSEN will generate a 0 signal (zero) signal

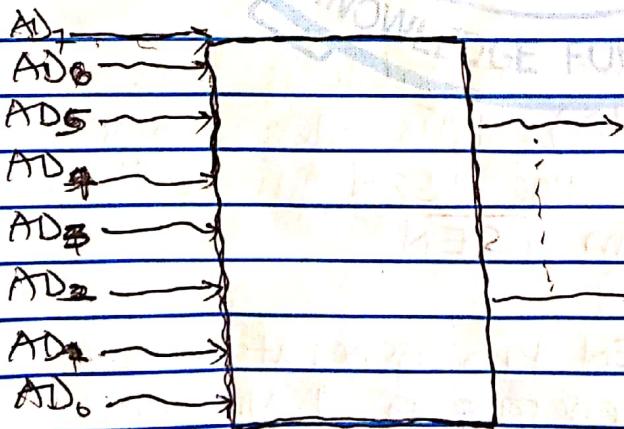


If $\bar{CS} = 0$ this external code memory will be selected.
If $\bar{RD} = 0$ means that microcontroller will read the data from this memory and this memory is already active because $\bar{CS} = 0$ and $\bar{RD} = 0$.

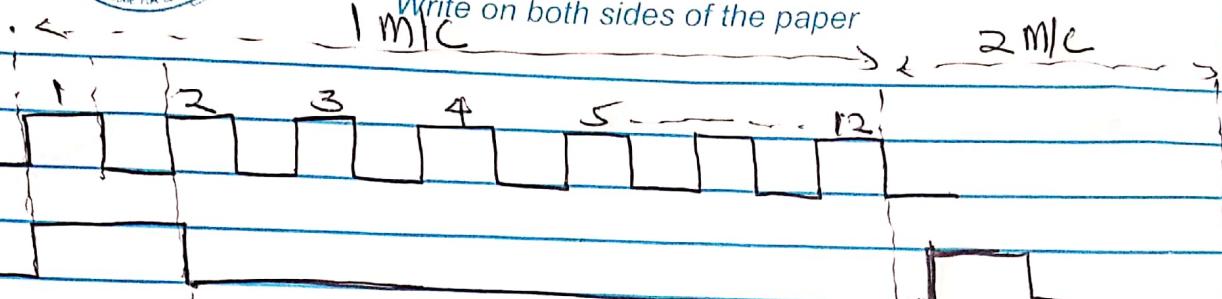
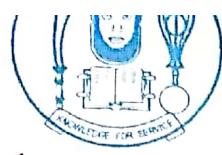
Now depending upon the value of the address lines the particular location will be activated between \bar{CS} and \bar{RD} and the content of that location which is 8-bit data can enter into the microcontroller.

ALE (Address Latch Enable)

ALE is used to demultiplex the address data lines.



We will generate 8 different lines for addressing and the 8 lines of $AD_7 - AD_0$ will be used for data. To demultiplex we require ALE signal.



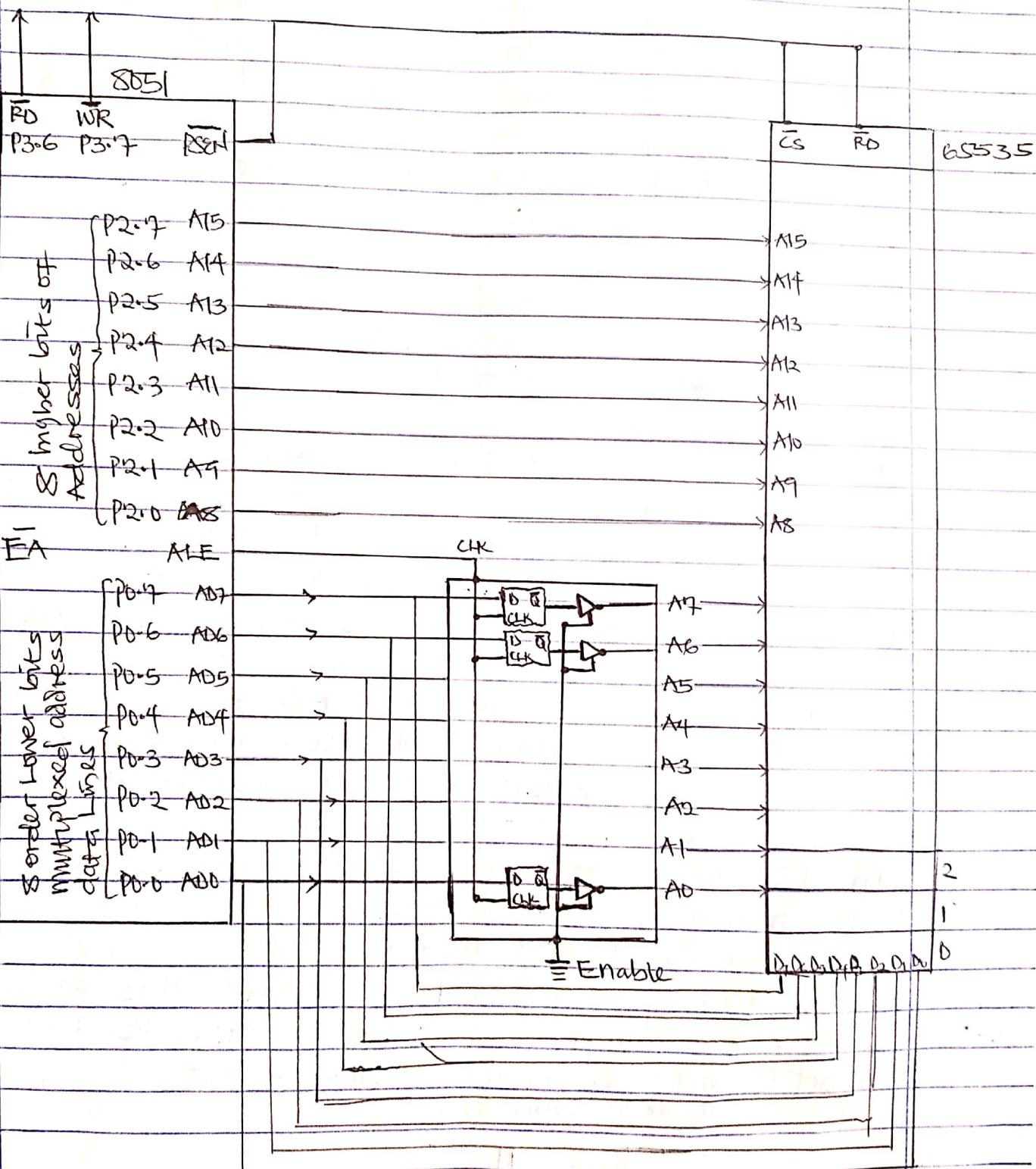
$$1 \text{ M/C} = 12 \text{ T-state}$$

In the first T-state ALE will go high and remain Low till the remaining T-state.

The high T-state in the Machine cycle is used to enable the Latch and by enabling the address Latch we will be able to demultiplex the address lines from multiple Address Data lines.

We also have

\overline{WR} (P3.6) These signals are used to access data from
 \overline{RD} (P3.7) external memory.

INTERFACING EXTERNAL MEMORY WITH THE MICROCONTROLLER
PIN DIAGRAM.

Suppose we have a microcontroller 8051 with the address data lines of AD₀ - AD₇ available for P0.0 - P0.7 8-order lower bit and address lines of A₈ - A₁₅ available.

for 8-order higher bits of Address. Since we are interfacing with external memory \bar{EA} is connected to ground (0), ALE and PSEN signal will be used for interfacing, RD and WR on port 3.7 and port 3.6 respectively will be used for interfacing. These are the signals or ports required for interfacing.

Now to demultiplex this data line we will use flip-flop or latch for demultiplexing the address data lines. This flip-flop contains 8-D flip-flop.

AD₀-AD₇ are connected to each of the D flip-flop. All the clock inputs are connected together and activated by the clock at the top of the flip-flop circuit when clock signal is applied.

ALE signal is connected to CLK, Enable is connected to ground which is 0 meaning all tri-state are in active mode that will be inverted. The ROM memory has 16 address lines starting from A₀-A₇, A₈-A₁₅. A₁₅ is connected to A₁₅ in code memory and A₈ is connected to A₈ in code memory all others are also connected.

In this memory we have CS & RD connected to PSEN there are 8-data lines in the code memory from D₀-D₇ from the microcontroller AD₇ is connected to D₇, AD₆ to D₆ and so on.

In the PC register of the microcontroller. When the microcontroller is put ON it will wake up from the address location 0000H then the controller will check the position of EA pin. If it's ground this means the controller will transfer the value of the PC lower 8-bit to port 0 and the higher 8-bits to port 2

Higher 8 bit. Lower 8 bit
 $\begin{array}{c} \text{D} \ 0 \ \text{D} \ 0 \ \text{H} \\ \text{P}2 \quad \text{P}0 \end{array}$

Port 0 contains the lower order address then port 2 contains the higher order address. The program code will start at 0H in the external code memory since EA is 0. The lower order 8-bit and higher order 8-bits are



UNIVERSITY OF BENIN

Question.....
Write on both side of the paper

Do not write
in this
Margin

Converted to binary and place on the Address Data lines as (0,0,0,0,0,0,0,0) and Address lines (0,0,0,0,0,0,0,0).

Data RAM pin WR, RD are in P3.6 and P3.7 respectively. PSEN in microcontroller is for External code memory while RD and WR in the microcontroller are for external data RAM.

Question.

Interface 8051 microcontroller with 32KB of ROM and 4KB of RAM.

ROM:

No of address lines n

Memory capacity = 2^n

$$32\text{ KB} = 2^n$$

$$1\text{ KB} = 1024\text{ B}$$

$$2^n = 32 \times 1024$$

$$2^n = 32768$$

$$n = 15$$

RAM:

No of address lines n;

$2^n = \text{Size(memory)}$

$$2^n = 4\text{ KB}$$

$$2^n = 4 \times 1024$$

$$2^n = 4096$$

$$n = 12$$

15 address lines are required for connecting the ROM for memory capacity of 32KB

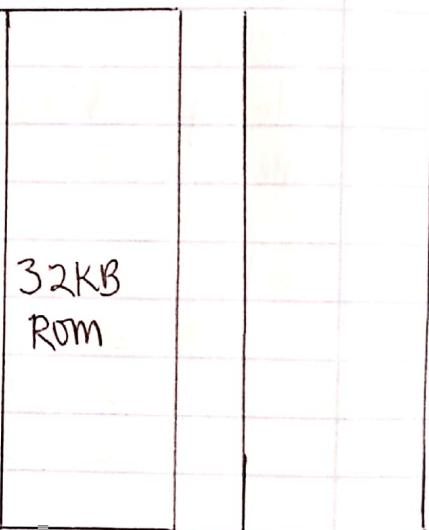
12 address lines are required for connecting the RAM for memory capacity of 4KB.

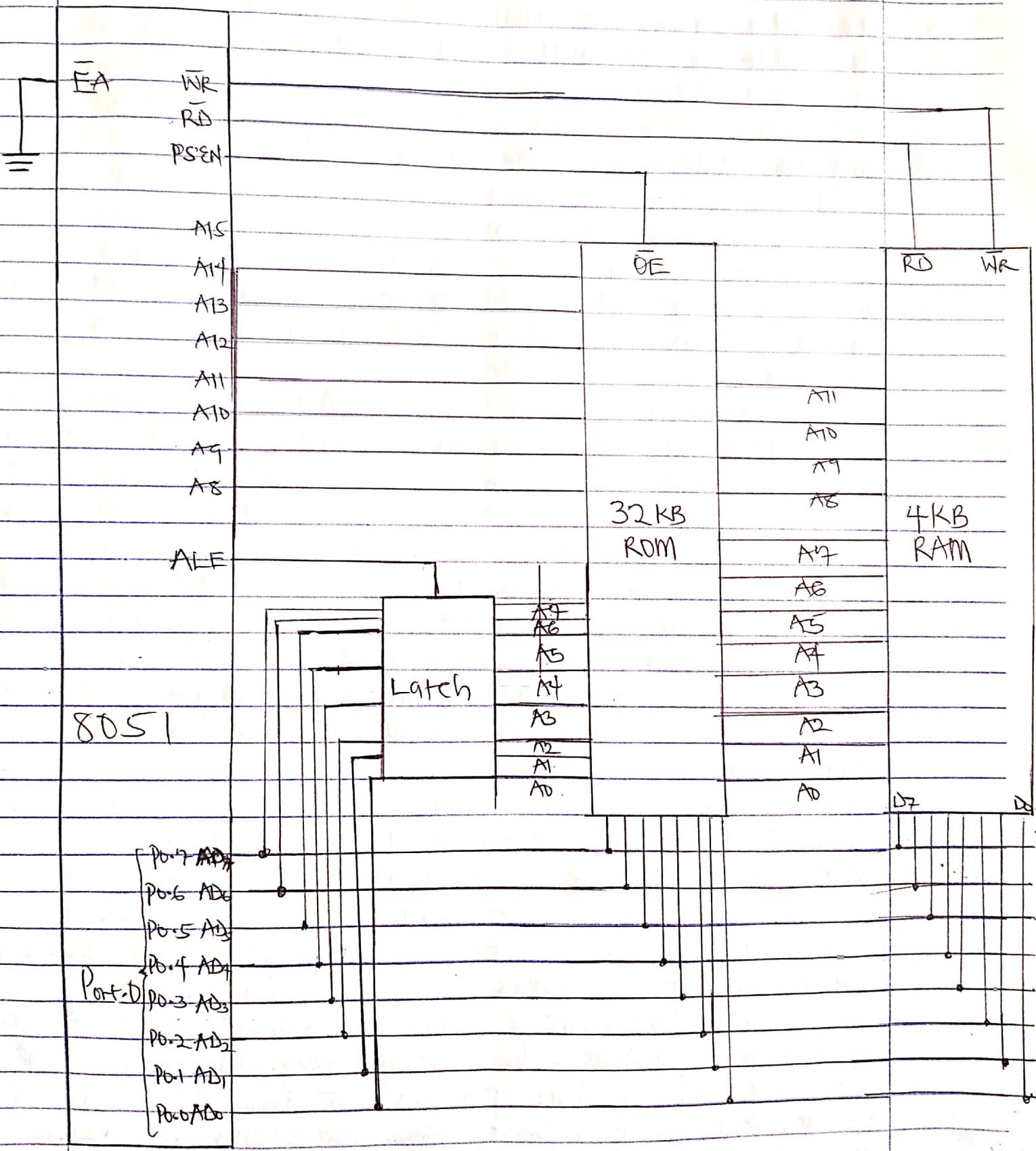
UNIVERSITY OF BENIN

Question
Write on both side of the paper

\bar{WR}
 \bar{RD}
 PS EN
 A15
 A14
 A13
 A12
 A11
 A10
 A9
 A8
 ALE

Port. D
 {
 PD-7 AD₇
 PD-6 AD₆
 PD-5 AD₅
 PD-4 AD₄
 PD-3 AD₃
 PD-2 AD₂
 PD-1 AD₁
 PD-0 AD₀





In order to interface with external memory the EA pin is to be connected to ground (GND) Logic 0. We know that the lower order address as well as the data lines are multiplexed, so we need to separate the address as well as the data lines. So in case of port D the address and data lines are multiplexed P0.0 - P0.7 (A₀-A₇) that is the lower order address, and the data lines are multiplexed.

The horizontal lines carries addresses as well as data lines from the 8051 microcontrollers. The next is how to separate the address and the data for that we need to use Latch and to the Latch ALE (Address Latch Enable) connected. When ALE=0 address data is sent but when ALE is 1 Address only is sent.

Now we need to connect the address data lines which are multiplexed to the Latch. So, when the addresses are enabled all the address lines are passed from the Latch (A₀-A₇). Then the block of the ROM is drawn which is 32KB and 15 address lines are required and so from the lower order 8-bit address lines are connected to the ROM (A₀-A₇). We need more address lines and we get it from port 2 to complete the 15 address lines (A₈-A₁₄).

In ROM we cannot write anything in ROM into it but to just read so we need to enable the read operation PSEN (Program Store Enable) this has to be connected to the output of the ROM OE (Output Enable) so that the chip gets enabled. So, when PSEN sends signals when it is low whatever data is present in that address is made available here in the dataline (D₀-D₇). So this completes the connection to ROM. Since the capacity of ROM is 32KB we need to connect 15 address lines all the lines are connected i.e. lower order address lines are connected (A₀-A₈) and the remaining address lines comes from higher order address lines then through ROM you can just read the data hence program store enable pin of the microcontroller needs to be connected to the output pin of the ROM.



The capacity of the RAM is 4KB so it requires just 12 address lines. The 8 lower order address lines ($A_0 - A_7$) are connected to the RAM and the high order from port.2 in the microcontroller ($A_8 - A_{11}$) are connected because of the 12 address lines.

In RAM \bar{RD} and \bar{WR} address lines are possible. To perform a write operation the \bar{WR} address line of the microcontroller has to be connected to the write (\bar{WR}) address line of the RAM. So, when we perform a write operation addresses to are sent to the address lines and whatever data that needs to be sent is passed to the data lines. We know that the Latch is used to separate the address and data lines.

In RAM read \bar{RD} operation is possible. The Read pin of RAM has to be connected to the Read \bar{RD} pins of the microcontroller. To perform read operations first addresses are sent and the Logic 0 is sent to the read pin and whatever data is present in the particular address will be sent to the data line and it will be received at port.0 of the 8051 microcontroller.