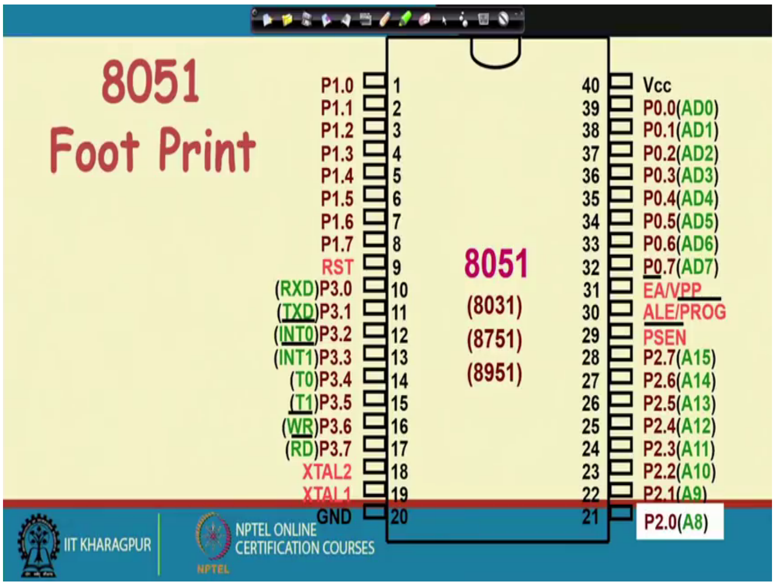
Microprocessors and Microcontrollers

Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 24
8051 Microcontroller (Contd.)

So, if you look into this pin diagram of 8051. So, this is that a pin diagram; so, 8051.

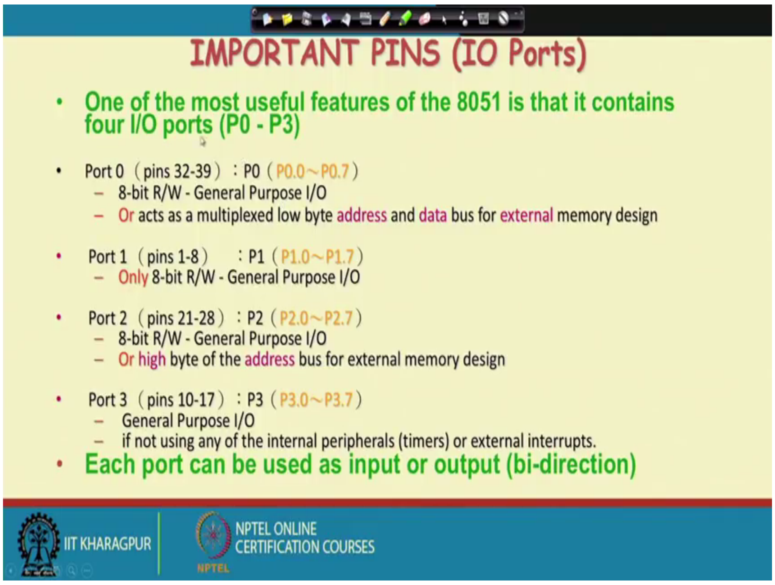
(Refer Slide Time: 00:22)



8051 Foot Print

So, it is similar for 3151 and 8951 also; so, more or less we have discuss the pins. So, we will see it in more detail in the successive slides.

(Refer Slide Time: 00:33)



IMPORTANT PINS (IO Ports)

- One of the most useful features of the 8051 is that it contains four I/O ports (P0 - P3)
- Port 0 （pins 32-39）：P0（P0.0～P0.7）
  - 8-bit R/W - General Purpose I/O
  - Or acts as a multiplexed low byte address and data bus for external memory design
- Port 1 （pins 1-8） ：P1（P1.0～P1.7）
  - Only 8-bit R/W - General Purpose I/O
- Port 2 （pins 21-28）：P2（P2.0～P2.7）
  - 8-bit R/W - General Purpose I/O
  - Or high byte of the address bus for external memory design
- Port 3 （pins 10-17）：P3（P3.0～P3.7）
  - General Purpose I/O
  - if not using any of the internal peripherals (timers) or external interrupts.
- Each port can be used as input or output (bi-direction)

The important pins are like this, first of all the most important feature of 8051 is that it has got 4 I O ports. And for that matter, if you look into any microcontrollers since the micro controllers are talking to the environment. So, this I O ports are very important parts in them and all micro controllers, they will have some I O port integrated into them.

So, in 8051 there are 4 such I O ports marked as P 0 to P 3; port 0, they occupied the pins 32 to 39 of the processor of the chip and they are called P 0. And they are also written as P0.0 to P0.7; the way it we are writing it as P0.0 to P0.7 because of the fact that this ports are bit accessible; so, you can control individual bits separately.

So, that is why we write it as P 0.0; 0.1 like that; so on the other hand. So, in case of say 8085; so, we have to write in port and then the data is 8 bits only. So, whether you use those remaining bits or not; so, that is up to the program, but the system gives; the system the 8085 processor will read the 8 bit pattern or when it is outputting it will output at 8 bit pattern.

So, this port 0 is 8 bit read write operation; so, for general purpose I O or it can act as multiplexed low byte address and data bus for the external memory design. So, if you would have the external memory, then we have to use this multiply so, this port 0 bits for the multiplexed addressed data bus; lower order address and data bus they have multiplexed like that.

Then we have got port 1 or P 1; so this P 1 so this is clear. So, P 1 does not have any other function multiplexed into it. So, if in your design if you want to use some port, the first option is to go for the port P 1 because it here no other function will be multiplexed; so, it should not harm any other system operation.

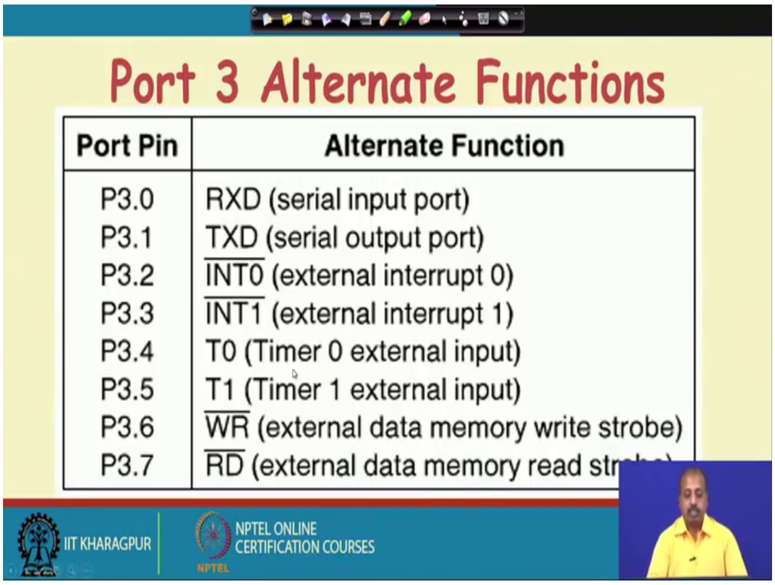
So, 8 bit read write operation and again it is general purpose I O; for port 2. So, this will occupy pins 21 to 28 and written as P2.0 to P2.7. So, again this is multi functional first of all; one function is 8 bit read write operation for general purpose input output or the it also provides the higher address bits for the external memory design.

So, this port 0 was giving us the lower order address bus and the data bus and this port 2 is giving us the higher order address bus. So, for memory connections; we have to use port 0 and port 2 pins. Then port 3; so, this is again multiplexed; so this is for so it has got many functions, so it is general purpose IO.

So, if not using any of the internal peripherals timers or external interrupts then it is ok; otherwise so they are to be; we have seen previously that they are, the read write operations; read bar write bar pins and all those things are there, so this is also multi functional.

So, as we are have seen previously like in this diagram we can see that P3.0 is received data, 3.1 transmit data; then this interrupts then T0, T1; the timers then write bar read bars, so they are also multiplexed on to these port 3. Each port can be used as input or output bidirectional and that is also it is bit controllable. So, you can to control at the bit level may be one bit is configured as input, other bit configure as output; so, that is possible.

(Refer Slide Time: 04:36)

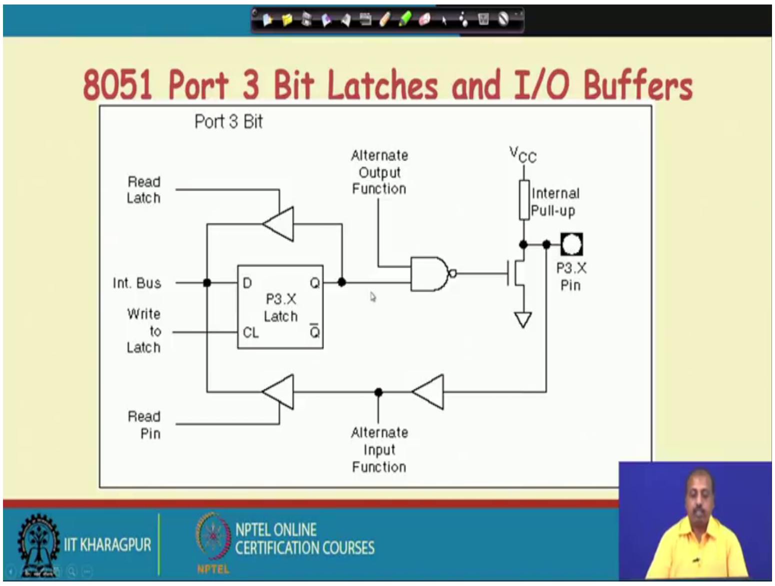


Port 3 Alternate Functions

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read str

So, this is the summary of port 3 alternate function; 3.0 is received serial input data, 3.1 is serial output data; so, this we have already discussed.

(Refer Slide Time: 04:46)



8051 Port 3 Bit Latches and I/O Buffers

Now port 3 bit latches and I O buffers; so, this the structure of a particular port bit. So, you see that every port bit; so this 3.X, so this is the pin. So, from the pin the line comes and it goes through this direct ultimately connected to the internal bus. So, it is connected like this, but in between there are many other things.

So, there is a latch here; so from the data bus whenever you are outputting some data. So, you can put into this latch; as a result until and unless you are writings you are writing something new one to this latch; this the processor need not bother about what value it has to output on to the pin.

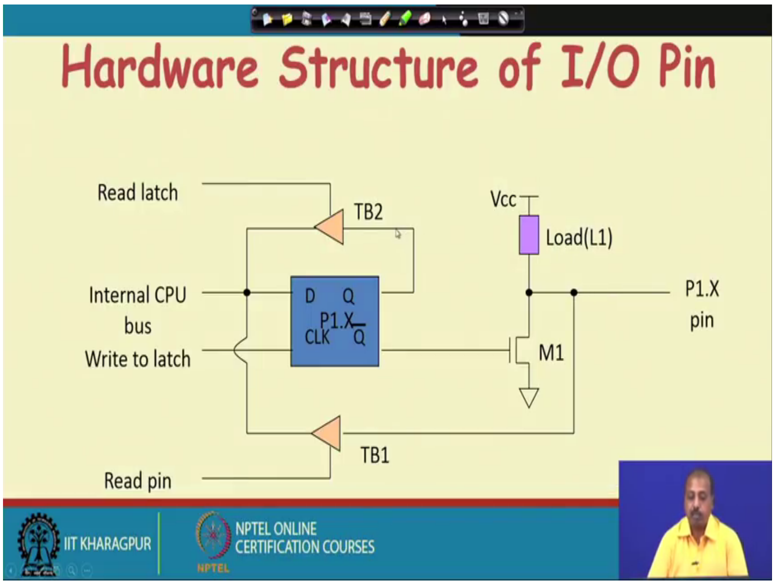
For example, for turning on one LED, the LED may be connected from here and the processor may write a one here. As a result this LED is turned on and that LED; so remains on, so even if the processor is doing something else; as long as it is not writing a new value system 0 to this latch. So, this point or this Q output will be at 1; as a result this value will be at 1.

So, if this Q quality output is 1; you can understand that this NAND gate output will be 0. So, this transistor will be off as a result; this one value will be coming here and the LED will be on here, so this way we can have this thing. Then otherwise, we can have; so when for the input part; so there can be two cases; one is you can read the content of this latch or you can read the content of this pin.

So, accordingly we will find that there are separate instructions for reading latch and reading pin. So, if you are trying to read the point; then this control has to be enabled as a result this pin content will be available on to the bus. If you are trying to read the latch, then this read latch control has to be activated. So, this will be coming here and going to the latch and of course so other things the alternate input function and alternate output functions. So, these are actually the other alternate functions that we have like providing read bar write bar etcetera or this transmit receive.

So, all those port 3 alternate functions that we have; so, they can be connected to this pins; this lines alternate functions.

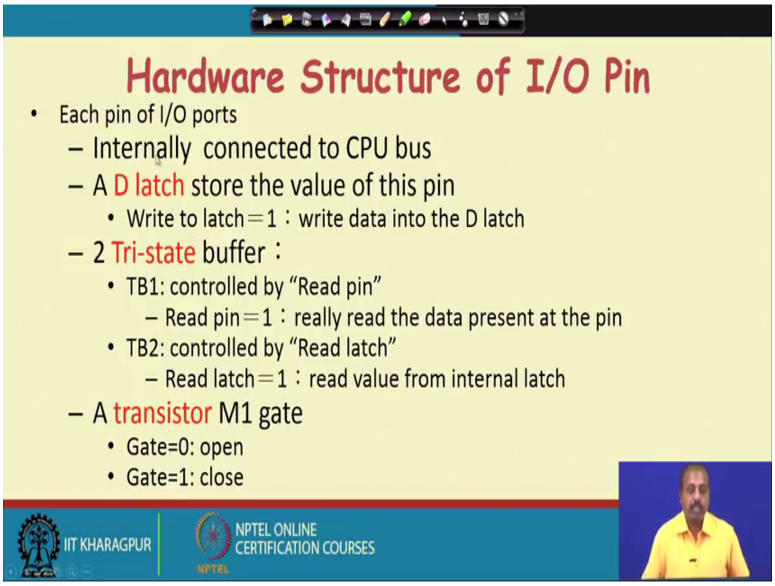
(Refer Slide Time: 07:17)



Hardware Structure of I/O Pin

So, we will see some example about the how this pin structure is going to work. So, this is the diagram that we had previously.

(Refer Slide Time: 04:30)

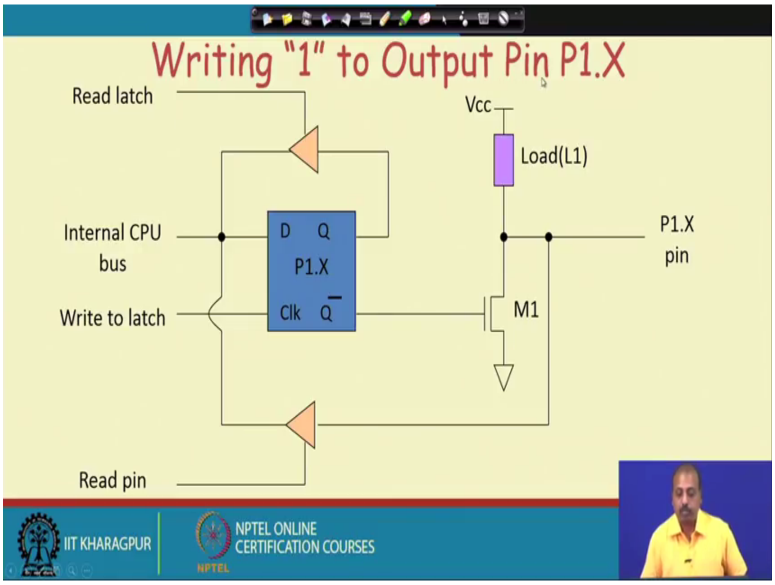


Hardware Structure of I/O Pin

- Each pin of I/O ports
  - Internally connected to CPU bus
  - A D latch store the value of this pin
    - Write to latch = 1 : write data into the D latch
  - 2 Tri-state buffer :
    - TB1: controlled by "Read pin"
      - Read pin = 1 : really read the data present at the pin
    - TB2: controlled by "Read latch"
      - Read latch = 1 : read value from internal latch
  - A transistor M1 gate
    - Gate=0: open
    - Gate=1: close

So, each pin of the I O port; so, it is internally connected to the CPU bus.

(Refer Slide Time: 07:39)

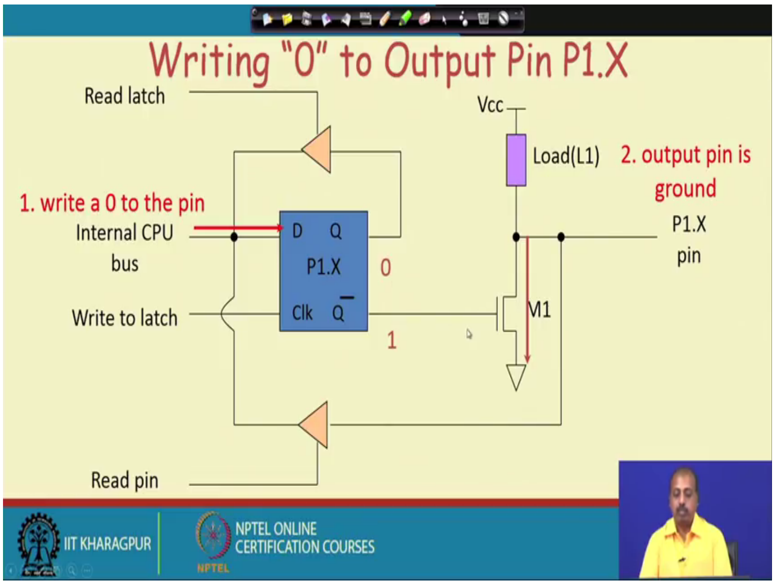


Writing "1" to Output Pin P1.X

So, we will take some example like how to write a 1 to the pin say 1.X. So, this is the port 1 and we want to write a 1 here; so how to do it? So, we can just we can write a 1 here; so, to write a 1 to this pin; so we put a 1 here.

So, this D flip flop; this D latch will be so and this right to latch, so this signal is activated. So, this D bit becomes 1 and so this bit becomes 0, this Q bar becomes 0 Q becomes 1, Q bar becomes 0 when Q bar; if Q bar is 0. So, it is connected to this transistor gate of this transistor; so, gate of the transistor getting 0, so this transistor is turned off. So, if you measure the voltage at these point, so you will get high. So, accordingly you will get a one at these point.

(Refer Slide Time: 08:30)



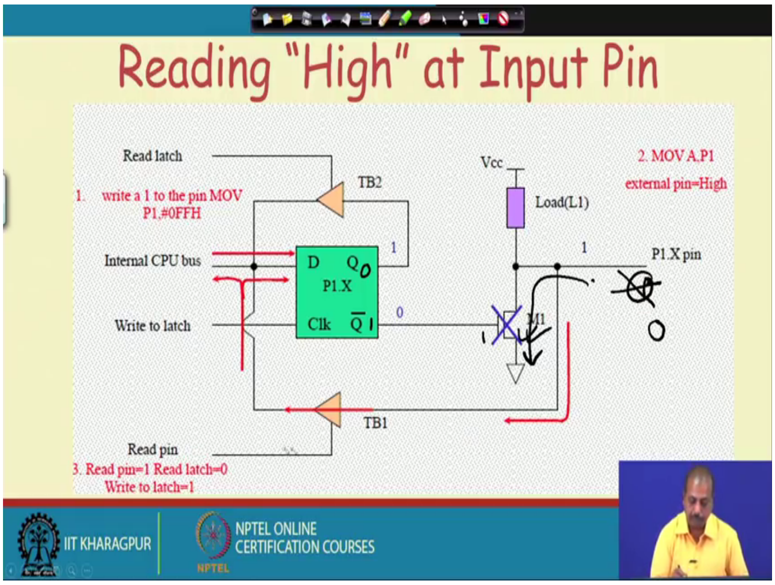
Writing "0" to Output Pin P1.X

On the other hand, if you are trying to write a 0 to the output pin. So, you write a 0 here as a result Q value becomes 0 and Q bar becomes 1, when Q bar becomes 1; then this transistor gate is 1. So, this transistor is on as a result at this point you will get the voltage of 0. Because this transistor is on; so you will get a 0 at this point, so at the pin you are getting a 0.

So, you see by writing the value of the internal bus in to this latch. So, ultimately the pin is getting affected; so, that way we can do the output operation on to the pins. So first to write 0 to the pin, so first we do this. So, 1 goes to the pin then we output pin is grounded because this gets a 0; so, this is 1. So, this becomes off; this becomes on, so this pin gets grounded.

So, this is how this is taking place, so this value is 0, this value is 1; so, that will be getting a 1.

(Refer Slide Time: 09:39)



Reading "High" at Input Pin

1. write a 1 to the pin MOV P1,#0FFH

2. MOV A,P1 external pin=High

3. Read pin=1 Read latch=0 Write to latch=1

On the other hand, if you are trying to read an input pin. So, when you are trying to read an input pin; so, we have to set this read pin value to 1. So, first of all, so we have to read this, we have to put this pin value to 1. So, if this pin value to 1; so, value that we have here will be coming at this point.

But there is a catch here because suppose the value that we had previously put on to this D flip flop is 0. So, D flip flop previously this latch was equal to 0; the Q was equal to 0. So, Q bar was equal to 1; as a result this transistor we had a 1 here; so, this transistor was on, so it is not off; it was on.

So, whatever voltage value that you put at the input pin; so, if you measure the voltage here you will get a 0. So, as a result that 0 will be coming to here; so, even if this input pin is set to 1; because previously this latch was set to 1. So, this value will be coming to; so, the previously this latch was value as 0 and Q bar value was 1; so, you will be getting a 0 at this point.

So, if this previously this value was 0; this value was 1. So, you are getting a 1 here; so, this transistor is on. As a result even if I want to get a 1 at this point, this 1 will get discharged by this path and this 1 will be modified to 0. So, whatever be the value that you put here; so, when you give this read pin signal here. So, you will be getting a 0 at this point, but the value that I have put there is 1.

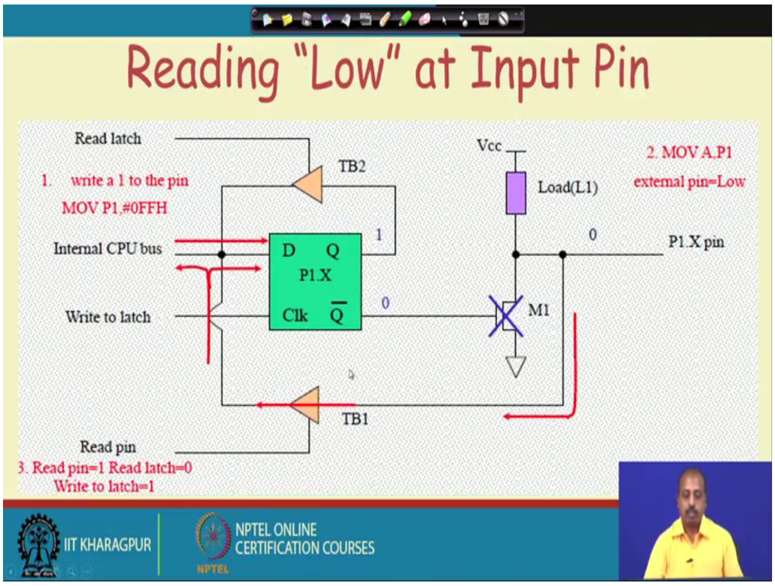
So, how to do that? So, for doing it; so, what we need to do is first write a 1 to the pin; so first we write a 1 to this pin. So, if we write a 1 then these value becomes 1.

So, this latch becomes 1; so this Q bar becomes 0, as a result this transistor is turned off. So, now this voltage value at this point cannot be discharged by this transistor. So, if you put a 1 here at the input pin, then when you are doing this operation that is reading the pin. So, this value will be coming here and it will come to the internal bus and similarly if the value that you put here is 0. So, that way that 0 will get transmitted via this catch and it will be coming faithfully to this bus.

So, make it a rule that whenever you are trying to read from an input port, first you write a 1 at that port. So, this is the rule write a 1 to the pin; so by executing the instruction move P 1 comma 0FFH. So, here all the bits of port 1; so, they will be configured as input port; input bits, so now after that you can do a read operation. So, this is the first thing that we have to do; so, when we do this move a comma P 1. So, then the external pin will become high and then you can get the value of this port on to this a register; otherwise this will not happen.

So, any point you are trying to read the content of a port; first you wrote 1 to the port. So, it may be to the bit or may be to the entire port; depending upon your application, but that has to done.

(Refer Slide Time: 13:22)



Reading "Low" at Input Pin

1. write a 1 to the pin
MOV P1,#0FFH

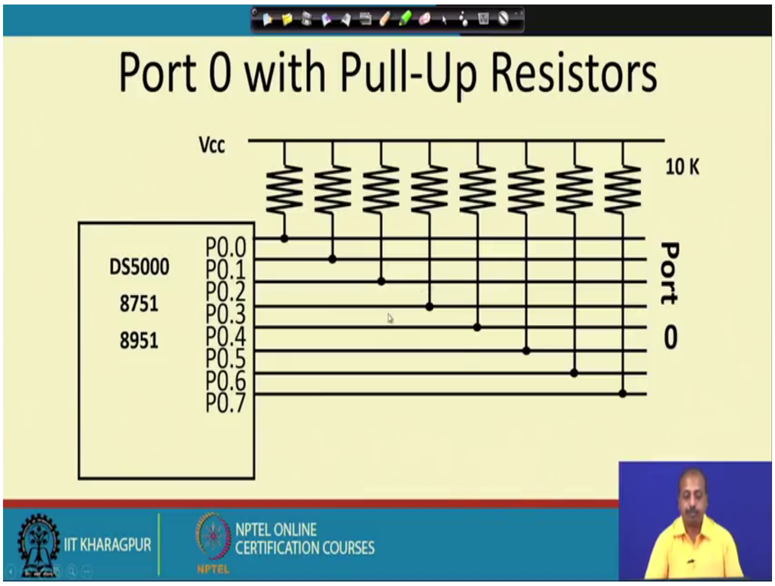
2. MOV A,P1
external pin=Low

3. Read pin=1 Read latch=0
Write to latch=1

Reading low of course, there is no problem because even if this gets discharged. So, there is no issue; like you are any way giving a 0 here. So, it does not have any issue, but still since it we cannot predict like what is the next value that we are going to get from the outside pin.

So, irrespective of whether you are reading a low or a high at the next point of time; since we do not know that in what is going to happen in future. So, it is advisable that before any in operation; before any read port operation, you do a right port with the value 0FFH. If you want to do it for the entire port and if you are connecting only a single line then or a single bit, then you set that bit to 1. So, this is the reading at for input port for the port operation we have to be a bit careful.

(Refer Slide Time: 14:17)

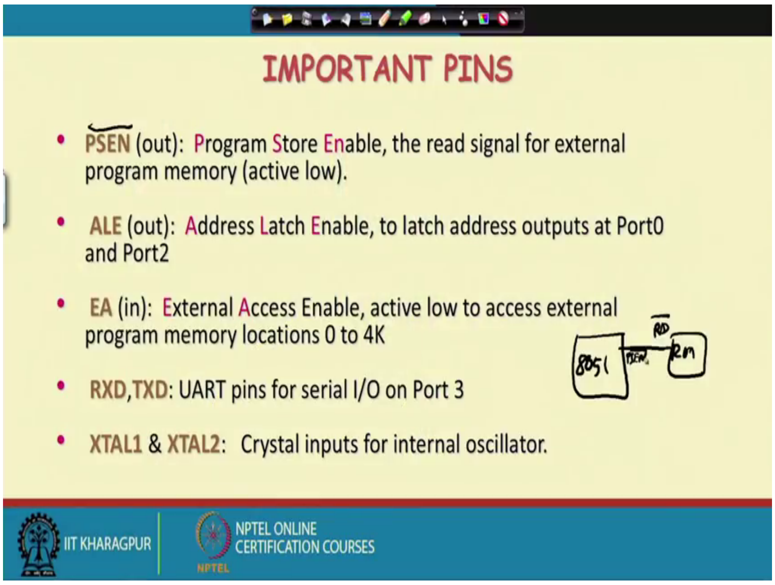


Port 0 with Pull-Up Resistors

Port 0; so, it has got some additional feature, so this is basically an open drain port. So, you need to connect these external resistances to port 0; so, 10 kilo resistances are to be connected. So, they are actually open collector type of connection or open drain type of connection. So, you should connect these resistances externally; so, this is helpful because if you want to drive some high load from port 0; so you can do that. Otherwise if you are doing it for from other ports; so, this the drive current is limited.

So, you cannot drive a high current through that; so, if you want a high drive current. So, you can have this you can connect that device to port 0 and through this pull up resistors and as a result you can put you can put high current onto those ports; over those devices.

(Refer Slide Time: 15:13)



IMPORTANT PINS

- PSEN (out): Program Store Enable, the read signal for external program memory (active low).
- ALE (out): Address Latch Enable, to latch address outputs at Port0 and Port2
- EA (in): External Access Enable, active low to access external program memory locations 0 to 4K
- RXD,TXD: UART pins for serial I/O on Port 3
- XTAL1 & XTAL2: Crystal inputs for internal oscillator.

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Next important pin that we have is the PSEN or the program store enable. This is the read signal for the external program memory and it is active low. So, we will generally call it as PSEN bar; telling that this is a active low signal. So, this is similar to the wrong interface; so, if this is user 8051 and you have connected the external ROM here. So, for the ROM; I will have that read bar line. So, that read bar line it should be connected to the PSEN bar line of the 8051.

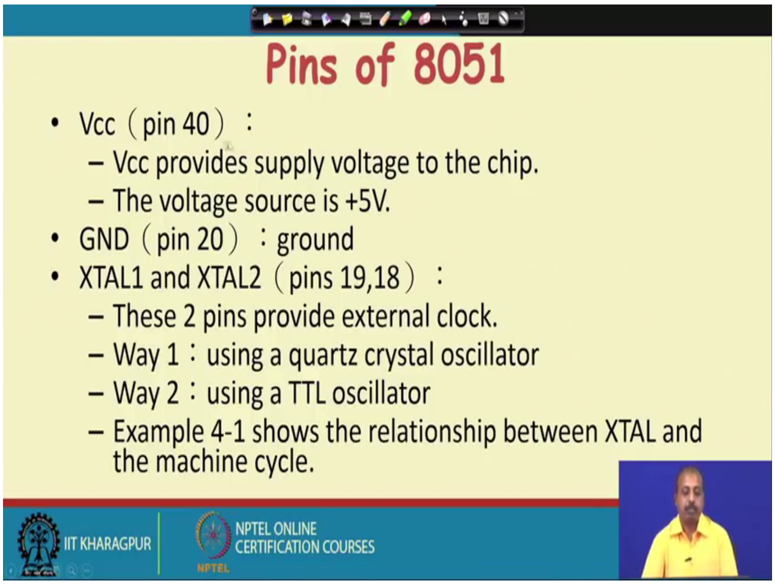
Then so, that is there; then we have got the ALE. So, address latch enable to latch the address outputs of port 0 and port 2. So, this is port 2 is of course, no problem because port 2 is the higher order address bus and that is fixed, but this lower order address bus. So, it contains multiplexed with the data bus; so, just like 8085.

So, we have got these multiplexing; so, here also we can have a multiplexing of this; lower order address bus; demultiplixing of the lower order address bus by putting some external latch. Then this external access EA; so, this is again an active low signal. So, this is also EA bar; so, external access enable.

So, if you want to access external memory then this a bar line should be made low. So, and if you are trying to access only internal memory then the EA bar line will be made high. So, we will come back to this again later.

Then there is receive data and transit data; so, these are the universal asynchronous receiver transmitter operation that 8051 has integrated with it. So, these are the pins on the serial I O on port 3; so, TXD and RXD pins. Then we have got the crystals for a crystal oscillator you have got the XTAL pins; then we have got Vcc.

(Refer Slide Time: 17:32)

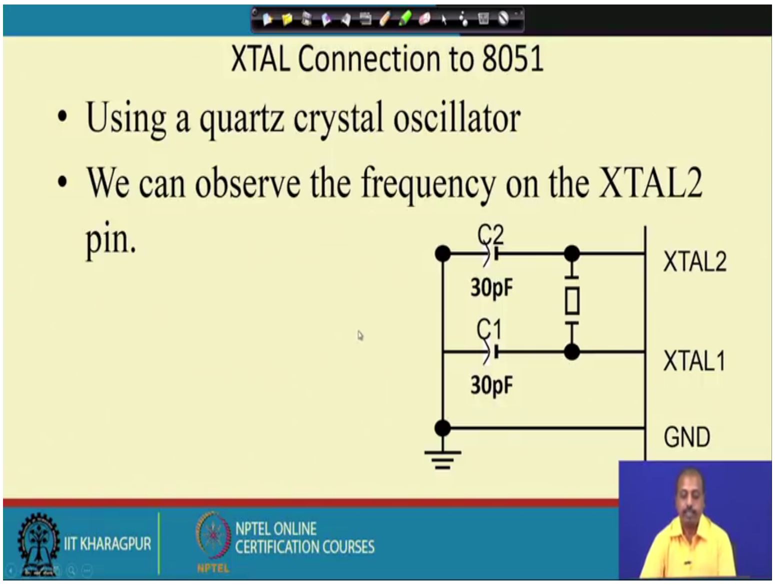


Pins of 8051

- Vcc（pin 40）：
  - Vcc provides supply voltage to the chip.
  - The voltage source is +5V.
- GND（pin 20）：ground
- XTAL1 and XTAL2（pins 19,18）：
  - These 2 pins provide external clock.
  - Way 1：using a quartz crystal oscillator
  - Way 2：using a TTL oscillator
  - Example 4-1 shows the relationship between XTAL and the machine cycle.

So, Vcc is the supply voltage which is plus 5 volt, then we have got the ground pin 20; which is the ground pin. Then the crystals we have got Xtal 1 and Xtal 2, so they are providing external clock. So, you can use a quartz crystal oscillator or you can use a TTL oscillator. So, you can connect some crystal oscillator here.

(Refer Slide Time: 17:57)

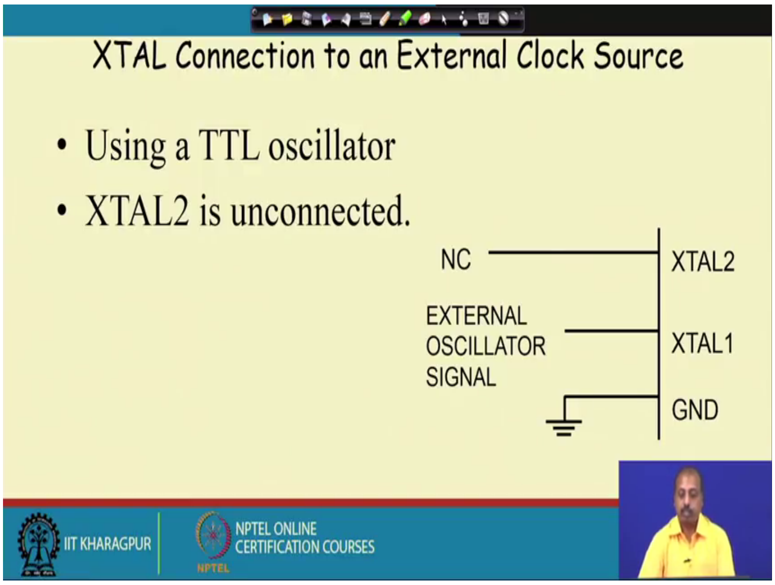


XTAL Connection to 8051

- Using a quartz crystal oscillator
- We can observe the frequency on the XTAL2 pin.

And get the frequency.

(Refer Slide Time: 18:01)



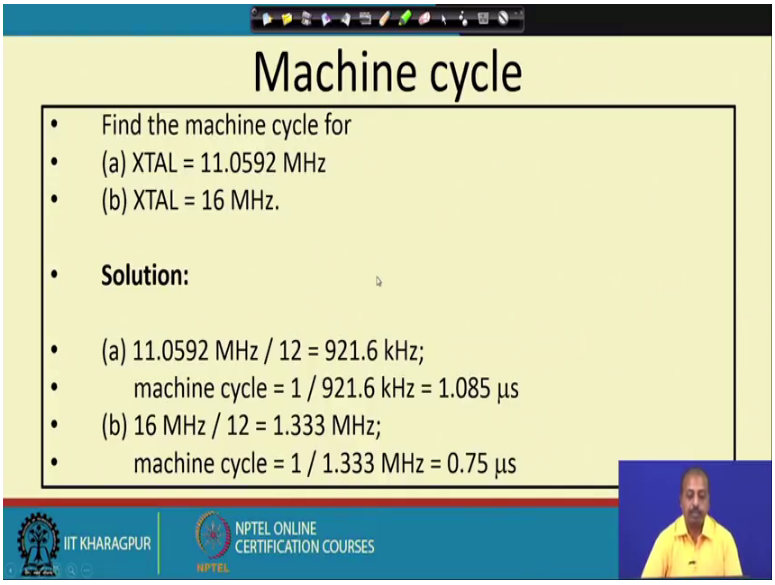
XTAL Connection to an External Clock Source

- Using a TTL oscillator
- XTAL2 is unconnected.

NC
XTAL2
EXTERNAL OSCILLATOR SIGNAL
XTAL1
GND
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, it can also be a crystal TTL based oscillators; so, that can also be connected. So, for external source, in that case we do not connect anything to the Xtal 2, but in Xtal 1 we connect the oscillator output and this is grounded.

(Refer Slide Time: 18:22)



Machine cycle

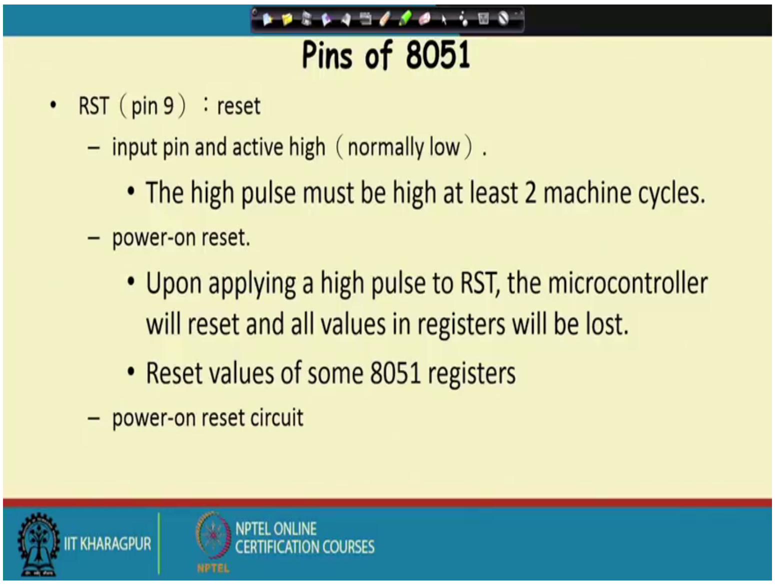
- Find the machine cycle for
- (a) XTAL = 11.0592 MHz
- (b) XTAL = 16 MHz.
- **Solution:**
- (a) 11.0592 MHz / 12 = 921.6 kHz;
- machine cycle = 1 / 921.6 kHz = 1.085 µs
- (b) 16 MHz / 12 = 1.333 MHz;
- machine cycle = 1 / 1.333 MHz = 0.75 µs

So, we can find out the machine cycle for 8051. So, it is like this that the crystal oscillator may be say if in one case it is say 11.0592 Mega Hertz; then one machine cycle is; to get a, so the internal clock that it has is 11.0592 divided by 12 that is 921.6 kilo hertz and so one machine cycle is 1 upon 921.6 kilo hertz that is 1.085 micro second.

So, in this case if the crystal is 11.0592 megahertz; if it is 16 megahertz by the same operation. So, it is internally it is divided by 12; so you get 1.333 megahertz. So, machine cycle becomes 0.75 micro second; so this division of 12 is fixed. So, whatever crystal we connect, whatever frequency it is generated; it is divided by 12 to get the internal machine cycle frequency.

Then we have got some more pins RST or reset pin. So, it is active input pin and it is active high; so normally it is active low, but we must hold it for two machine cycles for getting it active sensed. So, that is the reset signal and also this is also the power on is there.

(Refer Slide Time: 19:40)

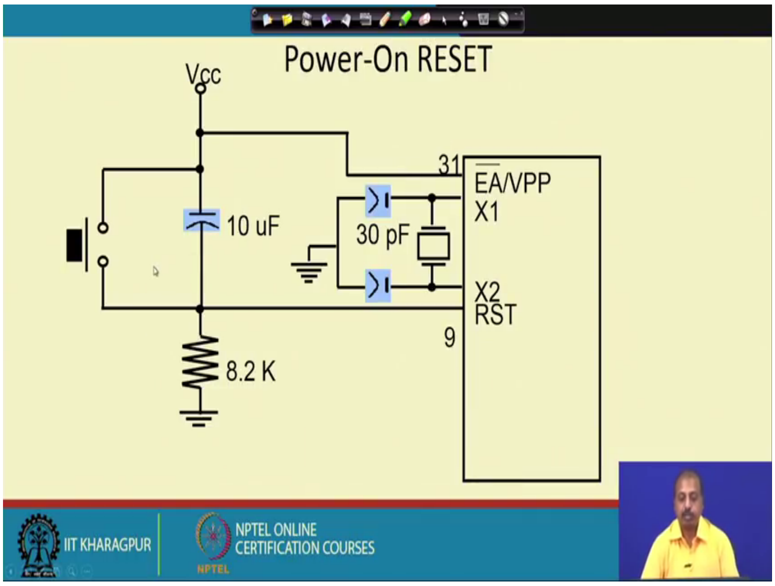


Pins of 8051

- RST (pin 9) : reset
  - input pin and active high (normally low) .
    - The high pulse must be high at least 2 machine cycles.
  - power-on reset.
    - Upon applying a high pulse to RST, the microcontroller will reset and all values in registers will be lost.
    - Reset values of some 8051 registers
  - power-on reset circuit

So, when the power on the system; so it will be a reset; so, if you apply a high pulse to the reset line then the micro controller will reset. And the values of all the CPU registers will be lost, so that they some of the values will be reset. So, we can have some power on reset circuitry for doing this operation.

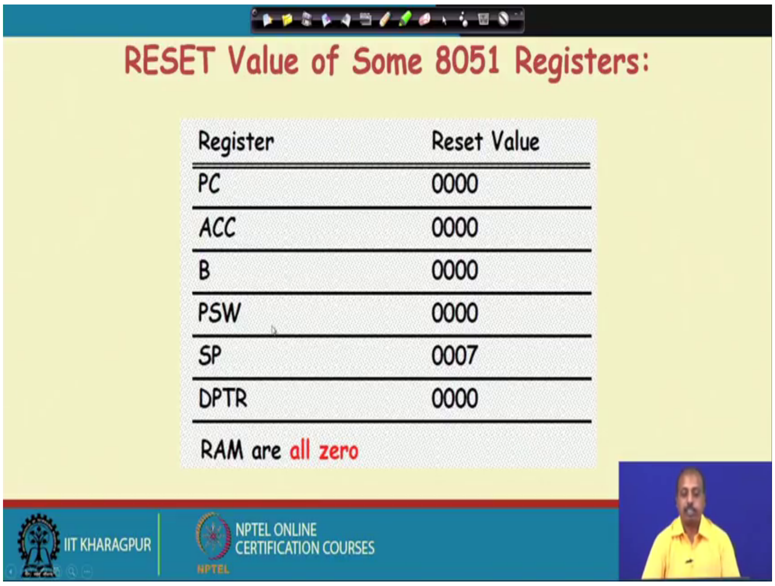
(Refer Slide Time: 20:10)



Power-On RESET

So, this is the power on reset circuitry; so, this is when this button is placed. So, this is the manual reset, so if you manually reset it then this capacitor will get charged. Then accordingly we can; so this capacitor will sorry; this will get discharge and there will be one pulse coming here. So, that will be giving as reset signal or when it is switched on; then also the capacitor slowly gets charged and accordingly we get this pin this power on part.

(Refer Slide Time: 20:45)



RESET Value of Some 8051 Registers:

Register	Reset Value
PC	0000
ACC	0000
B	0000
PSW	0000
SP	0007
DPTR	0000

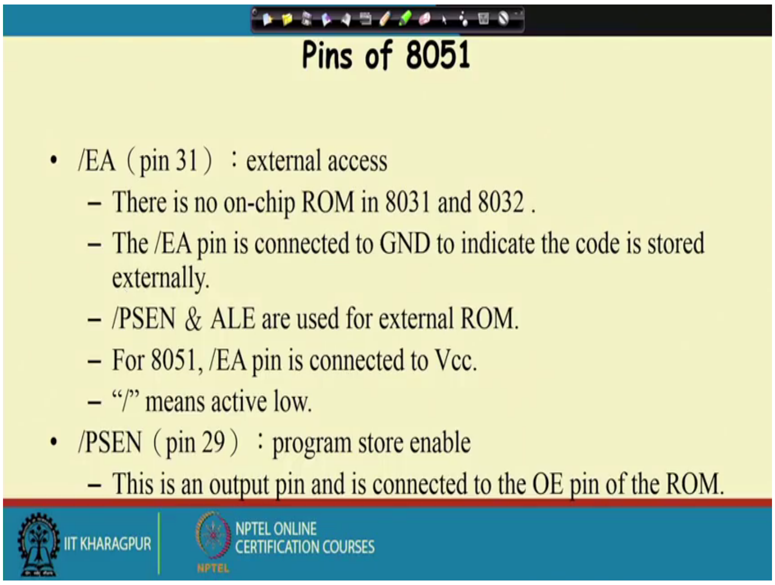
RAM are all zero

Some of the values of this resistors that we have in 8051 for the on resetting the program counter will get 0, accumulator resistor is there which also become 0, B resistor will become 0. So, all this registers PSW they become 0; the stack pointer is 7 and these DPTR register. So, this will become all 0 and the entire ram content will become cleared.

So, this is the special say; we will look into this registers once we go into the register file of 8051. So, we will see them, but their contents will be like this when the reset is done. And interestingly the ram which is the 128 byte ram that we have in 8051 or for that matter the other processors, other family of processors; so, there the ram content will also be cleared.

So, that will ensure that the scratch pad is clear; so ram is basically a part of the processor now we can say compared to the general system, where the over the content of the ram is not predictable; it can contain any garbage, but here the ram contents are cleared by the processor; so, it is all 0.

(Refer Slide Time: 22:05)



Pins of 8051

- /EA (pin 31) : external access
  - There is no on-chip ROM in 8031 and 8032 .
  - The /EA pin is connected to GND to indicate the code is stored externally.
  - /PSEN & ALE are used for external ROM.
  - For 8051, /EA pin is connected to Vcc.
  - "/" means active low.
- /PSEN (pin 29) : program store enable
  - This is an output pin and is connected to the OE pin of the ROM.

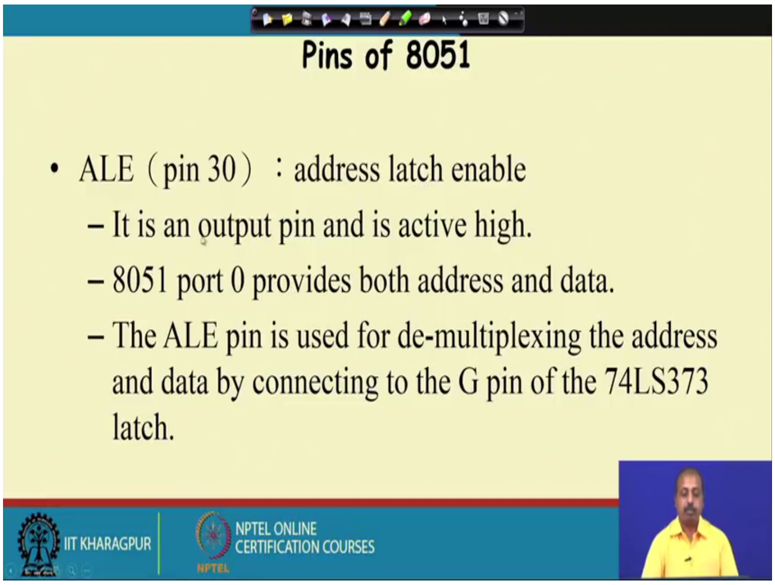
Then this EA bar pin; so, or external access. So, there is no one chip ROM for 8031 and 32. So, for them the we can have this EA bar pin connected to 0 because there is no ROM chip present so it has to be external memory and so in those cases. So, it is connected to ground to indicate that the code is a stored externally and this PSEN bar and ALE, so they are external. So, they are used for this external ROM access.

So, PSEN bar will be as you have said that PSEN bar will be connected to the read bar line of the ROM; and the ALE is for de multiplexing the lower order address bus and data bus. For 8051 a bar pin is connected to Vcc; so because 8051 has got internal ROM. So, when the system comes up; so this EA bar pin. So, this program will be started from the internal ROM; so that way this EA bar pin should be high.

So, that it does not go to access the external memory immediately. And this PSEN bar is the program store enable, so again this we have already said, this connected to this OE bar. So, OE bar is the output enable bar which is same as the read bar pin that we are talking about.

So, in case of ROM; so, instead of telling that it is a read bar. So, we normally the pins are normally identified as output enable or OE bar.

(Refer Slide Time: 23:38)



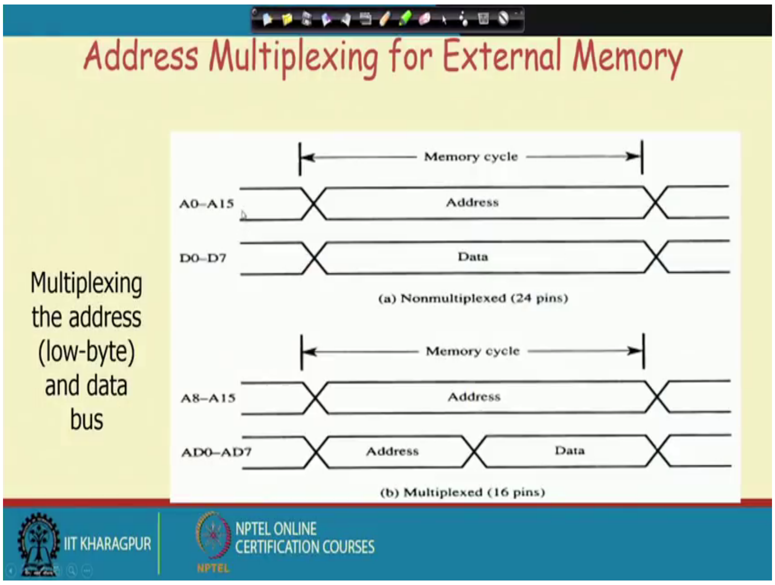
Pins of 8051

- ALE (pin 30) : address latch enable
  - It is an output pin and is active high.
  - 8051 port 0 provides both address and data.
  - The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 latch.

ALE is the address latch enable; so, just like 8085. So, this is there so it is an output pin and active high. So, it provides port provides both address and data and ALE is used for de multiplexing the address bus. Just like the de multiplexing we did for 8085; so, typically a latch like 74373 can be used and this 74373 has got a G pin which is actually the latch pin you can say.

So, this ALE can be connected to G; so, when this G is high then whatever is in the input, so that value is latched on to the 74373 latch. So, this is the multiplexing for external memory; so this is the A 0 to A 15 address and D 0 to D 7; so, this is the non multiplexed access.

(Refer Slide Time: 24:28)



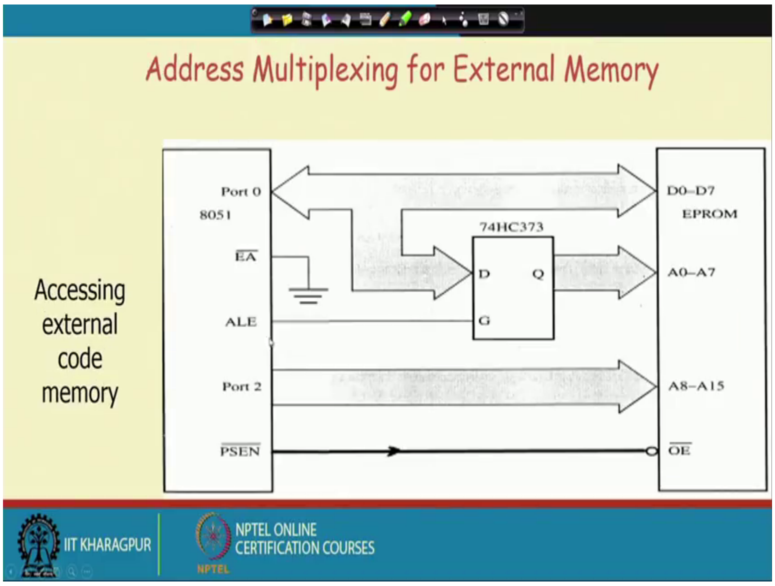
Address Multiplexing for External Memory

Multiplexing the address (low-byte) and data bus

So, total 24 bit; so when the multiplexed access this AD 0 to AD 7; so they are; so, A 0 to A 7 and D 0 to D 7 are multiplexed. So, in the first part you provide the address for the lower order address byte and in the second part, we get the data part.

So, that way the number of pins reduce from 24 to 16.

(Refer Slide Time: 24:58)



Address Multiplexing for External Memory

Accessing external code memory

So, as you are telling that this port 0; the higher order address bits. Port 2 has got the higher order address bits; A to A 15, port 0; the data bus will be coming connected. So, data bus is connected D 0 to D 7 and the address bus lines A 0 to A 7 are multiplexed with D 0 to D 7. So, they are also available here; so, for that it is connected to this 373 and this ALE signals, so it is connected to the G pin of 373.

So, in the initially when 8051 puts the address onto the address bus; the higher order address bus contains the higher order bits, lower order address bus contains the bits A 0 to A 7. ALE signal is given, so the value gets latched on to this 373 latch and after that this ROM will; this PSEN bar line is connected to the y bar line of the ROM. So, as a result the ROM will put the data on to the data bus and this they will come to again the port 0 of 8051; which is acting as the data bus for the 8051; the external data bus for the 8051.

So, in this way just like 8085; so, here also we have got multiplexing of address and data bus and it can be connected through the latch.