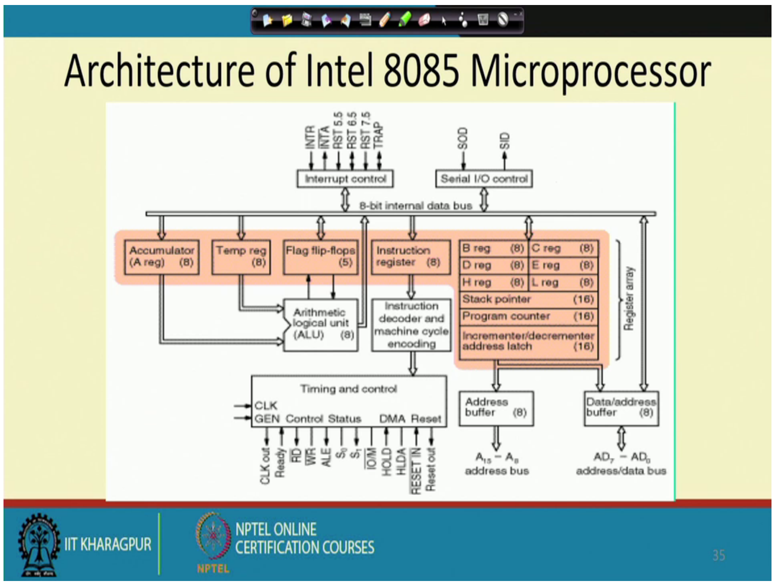
Microprocessors and Microcontrollers

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

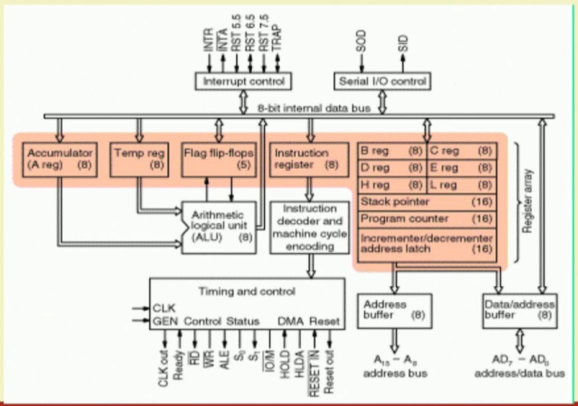
Lecture - 10
8085 Microprocessors (Contd.)

So, if we look into the internal architecture of 8085. So, if we have a look at the block diagram level view of this 8085.

(Refer Slide Time: 00:20)



Architecture of Intel 8085 Microprocessor



IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

35

Now, you see that there are some special registers, like this accumulator is a special register then this temporary register. So, this is another special register and this flag register. So, this is another special register. So, they are, they are actually in conjunction with this arithmetic logic unit. So, as we have seen that any processor, it will have an arithmetic logic unit it will have some register file consisting of a set of registers, and some controller.

So, in that light so, if we just try to look into the details of this 8085 internals. So, you will see that this ALU part. So, if you blow it up a bit, then you will find that ALU is consisting of not only the ALU, but this special register a register temporary register, and there is a set of flag register. There is a flag register which consists of a set of flip flops, this a register is 8-bit wide temporary register is also 8 bit wide. And the flag is 5 bit wide. And this ALU it does operations in 8 bit, 8 in in terms of 8 bits. And there is an internal data bus which is 8 bit wide; so all the data that are flowing through the internal internally through the processor.

So, they are A 8 bit at a time. Now let us look at it from the outside like the way the instruction comes to the processor how does it continue. So, when the instruction comes. So, the this is the instruction will come through this line this AD 7 to AD 0, when this part is we assume that it is it is configured as data bus. So, that was outside world the first instruction has arrived here. Then this instruction will go through this line. So, it will come to the instruction register.

So, the instruction reaches the instruction register, and then this instruction register the opcode part of the instruction so, it is decoded. So, instruction decoder and machine cycle encoding. So, this particular module will decode the meaning of the instruction, and it will also tell what are the operation that it has to do. So, accordingly it will instruct the timing and control logic, and it will that is this timing and control logic will get the instruction opcode from this from this instruction decoder. It will generate the clock signal, which will get the clock signal, and it will have this clock out it will it will it will have the clock signal from the outside world.

And so, it has got many other lines which are acting as input or output. Like this clock out signal so, this is basically generally it is the clock out value. So, whatever clock signal is coming in. So, that is generated as clock out, then this read bar line; so read bar line is given as control output. So, write bar line is also given as control output the arrow is missing here, but the arrow will be outside. So, wherever the arrows are missing I think this will be the outward direction. Then these, let us look into the signals that are coming into this timing and control module.

So, clock is one signal clock generator because there are that crystal will be there. So, crystal will be connected between these 2 pins the clock generator pin. Then this ready signal so, ready is a special signal. So, we will see it in detail later, but what it essentially does is that if the memory chip onto with which you are connecting the processor, if that memory chip is slow compared to the processor. So, processor after it has put the address on to the address line. So, it will expect that data to be available after sometime. If the memory is not able to give the data within that time, then it should tell the processor that I need the address for some more time.

So, in that case it will put this ready signal a high telling that the memory is not yet ready so, the data is not yet available. So, that way this ready comes as input to this timing and control module. Then these then write was then the next input line we have is the hold line. So, hold line will be it is important with respect to a special type of operation which are known as direct memory axis or DMA. So, in DMA what happens is that there may be multiple processors connected to the same bus and they are talking to the same memory.

So, when that is the case the multiple processors are masters of the same bus. So, there will be contention like, which processor will be accessing the bus next or so, the policy is that whenever a processor wants to use the bus. So, it will send a hold signal. So, that the other processor will be releasing the bus, and it will send a hold acknowledgement telling that I am not using the bus. So, you can use it now so, this hold and hold acknowledge. So, these 2 pins are used in that purpose.

So, we will discuss it in more detail when you go into this DMA data transfer type of operation. Then this reset in bar. So, this is basically the reset pin. So, if the reset pin is pressed then this reset in bar line will be activated. So, that it will be reset, and it will generate the reset out. Then it will generate the hold will generate hold out, then this status line; so s 0 s one and I O M bar. So, that will tell you the status like as I have already said that which operation it is doing now and whether I O operation or memory operation. And the ALE signal so, ALE signal. So, this is an output. So, this will be used to de multiplex the address and data bus values from the multiplexed address data bus.

So, other than that we have got this these general-purpose registers like B register C register D E H and L so, these registers. So, they are all 8-bit registers; however, in some instructions so, you can use them as 16-bit register pair as well. So, in that case I have got register pair B, which consists of the registers B and C making it a total of 16 bit we have got register pair D, making a D making de as a pair and H, H L as a pair. So, these are all individually 8-bit register, but when you are taking 2 of them together makes a 16-bit register, then there is stack pointer.

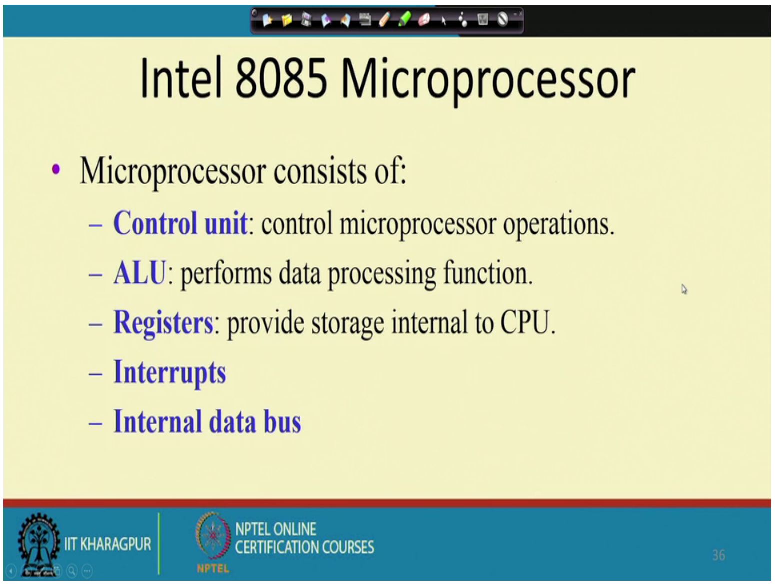
So, stack pointer we will discuss later. So, this is a special purpose register which is a 16-bit register, then there is program counter. So, program counter is actually just like in the architecture portion we have discussed. So, it will it will tell like what is the next instruction address to be accessed. So, if you reset this processor by putting a reset bar line high. What happens is that this program counter value becomes reset to 0. So, this value is reset to 0. So, as a result this next time this the processor goes into the instruction fetch cycle, this 0 value will be put through this address buffer to the address bus.

So, the memory as if the processor is now trying to access from memory location 0. So, you if your system initialization routine starts from address 0. So, then it the system will get initialized. So, that way this is the special thing. So, when this reset in bar pin is activated this program counter value becomes 0. So, that way it goes on. Saying then then there is a incrementer decrementer address latch. So, this is basically for incrementing decrementing the address part some additional circuitry will be required some latches will be required. So, that will constitute another 16 bit.

Other than these so, we have got this serial I O control which will be controlling the SID and SOD line serial input data and serial output data these 2 lines. There are interrupt control so that will model that will handle the interrupt. So, if you want to tell the processor from the outside that something extra has happened something special has happened and that has to be taken care of. For example, if this 8885 microprocessor is used for controlling the operation of a plant, then then if there is a fire detected. So, some smoke detector detect some smoke, then it can send interrupt to the processor telling that some abnormal situation has been detected. And this interrupt comes to the processor when we will see later that whatever the processor was doing that will get suspended for the time being and it will go into execution of the corresponding interrupt service routine for servicing that extraordinary situation.

So, these are the internal internals of 8085. So, this is so, this is the document released by the designers the Intel people, and as a user of the system. So, we get to know that these are the special registers; these are the registers that are there. So, which register is accessible in which instruction, and how they can be used etcetera; that will be documented in the remaining part of this discussion.

(Refer Slide Time: 10:06)



Intel 8085 Microprocessor

- Microprocessor consists of:
  - **Control unit**: control microprocessor operations.
  - **ALU**: performs data processing function.
  - **Registers**: provide storage internal to CPU.
  - **Interrupts**
  - **Internal data bus**

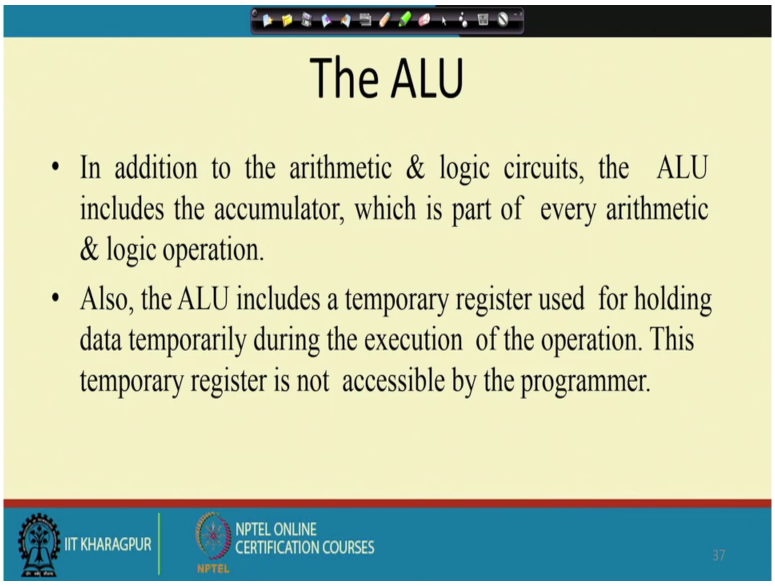
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
36

So, if we summarize; so, this microprocessor consists of control unit, ALU, registers, interrupts and internal database. So, out of that this control unit will control the total operation of the microprocessor, ALU will perform a data processing function. So, whatever the arithmetic logic operations are necessary. So, that will be done by the ALU registers. They will do they will provide the storage internal to the CPU. And so, actually this is very interesting because for so, if you want to access a particular data or if you want to do and do an operation very fast, then if the operands are in memory then while doing the operation I should bring the operands from the memory to the ALU for doing the operation.

But if they are in the CPU registers, then that is internal to the chip. So, no external access is necessary, as a result the operation will be much faster. So, if you are really looking for faster execution, then you should put the operands of these instructions into the resistor. Of course, that may not be possible because registers are limited in number. So, we have got only say B C D E and H L. So, only these 6 registers are available in the inside the CPU. So, only the most important variables in the program they can be put into those registers, and the remaining ones will be in the memory, and their operation will be slow, but this essential the critical one. So, they can be made faster.

So, these registers they provide storage internal to the CPU. Then interrupts are there and for interrupting the system and telling that something special has appended that to take care of that situation. And there is internal data bus. So, that will be connecting the modules internal to the processor.

(Refer Slide Time: 12:07)



The ALU

- In addition to the arithmetic & logic circuits, the ALU includes the accumulator, which is part of every arithmetic & logic operation.
- Also, the ALU includes a temporary register used for holding data temporarily during the execution of the operation. This temporary register is not accessible by the programmer.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

37

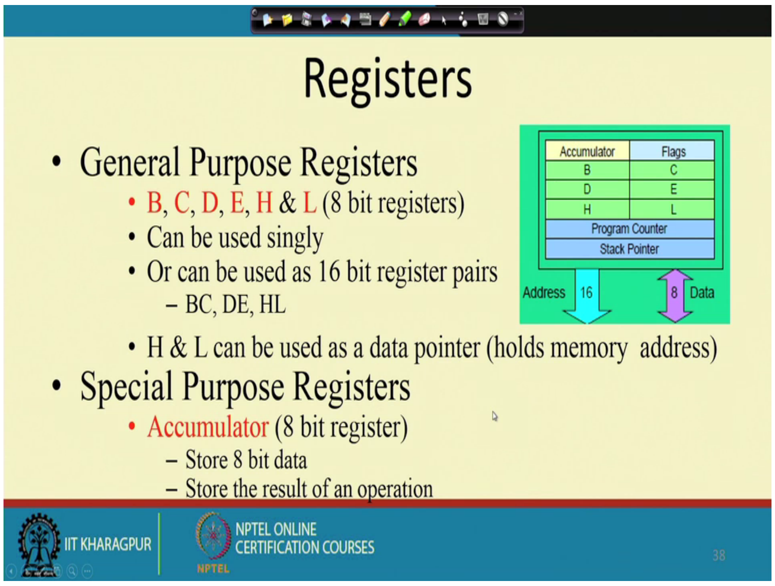
So, the ALU so, they in addition to arithmetic and logic circuits the ALU includes the accumulator which is part of every arithmetic and logic operation. I will like to highlight this point that, it is part of every logic of arithmetic and logic operation.

So, for any operation this ALU the any ALU operation this accumulator is taken as one of the operand or the first operand, and the result is also stored in the accumulator. So, it is the source operand, the first operand, and as well as the destination of these instead of these operations. So, also apart from this a accumulator. So, there is a temporary register that holds data temporarily during execution of the instruction. And this temporary register is not accessible by the programmer. So, programmer does not have any access to this temporary register; so if we just go back and in to look into this diagram like this ALU when it is doing the operation.

So, it needs the value to be available at it is input and these values available at the input. They should be stable values they should not be floating around this; bus like if I am trying to add the content of a register with B register, then the B register value should be available here in a steady fashion. So, what the system will do it will transfer this B register value to this temporary register, and then it will give that signal to the ALU. As a result, the values will be added and the result will be available on to the bus which will be stored in the accumulator.

Or if it is trying to add a with some memory location content, then the memory location content will come to this data buffer through there through this bus. So, it will come to the temporary register and this timing and control module. So, it will generate the appropriate control signals for doing that the value will be loaded into this temporary register. And then the value will be added using this ALU, and stored back into the accumulator. So, this temporary register is not accessible by the programmer. Because programmer is not supposed to know in fact, what is the structure of this temporary register it is not revealed by the Intel people? So, it is just an it is just said there is something like that is there inside.

(Refer Slide Time: 14:37)



Registers

- General Purpose Registers
  - B, C, D, E, H & L (8 bit registers)
  - Can be used singly
  - Or can be used as 16 bit register pairs
    - BC, DE, HL
  - H & L can be used as a data pointer (holds memory address)
- Special Purpose Registers
  - Accumulator (8 bit register)
    - Store 8 bit data
    - Store the result of an operation

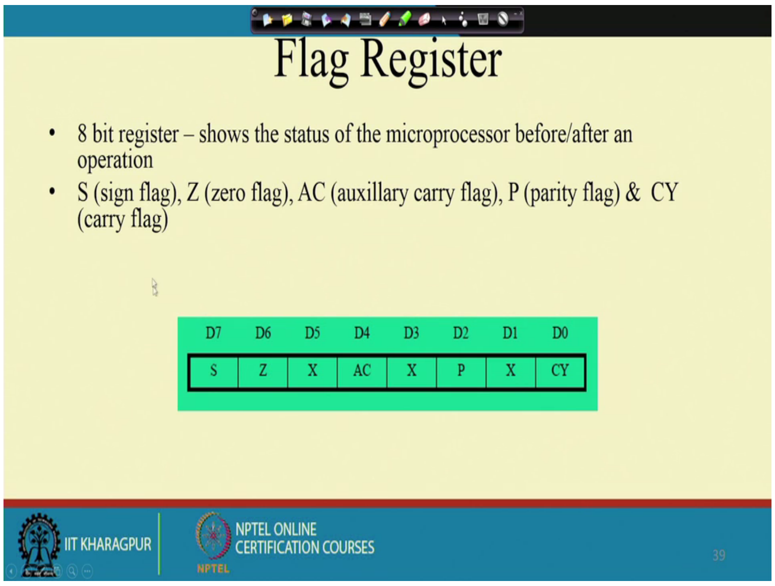
Accumulator	Flags
B	C
D	E
H	L
Program Counter	
Stack Pointer	

So, whether it is an 8 bit register or something more than that, so that is also not clearly told. So, as far as the users are concerned so, users will see these registers. The accumulator, flags, B C D E H L so, these pairs. So, out of these this B C D E H L. So, they are called general purpose register. So, the general-purpose register means, they are normally used for this ALU operation. So, all of them are 8-bit register, and can be used singly. So, you can use this bc these registers one register as a as an 8 bit register or you can use it as 16-bit pair like B C D H L 16-bit pair also you can use.

And H and L they can be used as data pointer. So, this is another very interesting thing like, say many times what happens is that we need to implement a pointer in our program. So, for pointer what is required is that the operand that we have in the instruction. That is not the actual operand, but that is an address of the operand. So, we have to access the memory location 0.82 by that particular operand to get the actual value. So, this H L pair, when it is used as this a pointer. So, this can be used to hold some memory address, and what the processor will do? It will not use H L register pair value as the operand, but it will be using the corresponding memory location as the operand.

So, this helps in implementing the pointers in high level languages. Apart from that there, are special purpose register. So, once one such special purpose register is the accumulator. So, accumulator is an 8-bit register, and it is normally it is storing their, the first operand as well as the result of the result of any operation.

(Refer Slide Time: 16:30)



Flag Register

- 8 bit register – shows the status of the microprocessor before/after an operation
- S (sign flag), Z (zero flag), AC (auxillary carry flag), P (parity flag) & CY (carry flag)

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
39

Another special register is the flag register. Now while doing this arithmetic logic operation, various exceptional situations can occur. For example, when I am storing a number; so that the number may be the number may be positive or negative.

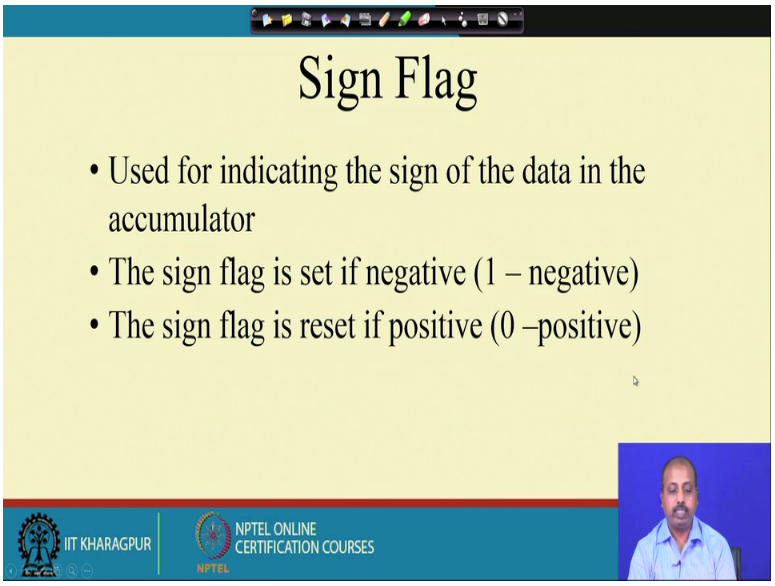
So, when storing a number in the accumulator, if the accumulator content is negative, then I there may be some decision that I need to take based on that situation. So, we need to know what is the value of this operation that has been done, in the last operation whatever operation has been done, what is the result of that the type of the result. So, if the result after doing the operation is say negative, then this particular be D 7 of the special register flag, will be set to one telling that the result was negative. And the result if the result is positive then this D 7 will be 0.

The last operation maybe it has resulted in the value being total is 0. Maybe we have subtracted to same values and as a result the result has become 0. So, in that case this D 6 bit will be set to 0, set to one telling that the 0 flag the 0 the result has been 0. So, this particular bit will become 1 telling that the result is 0. Otherwise the z bit will be equal to 0. Then this D 5 is not used similarly this D 3 D 5 D 3 and D 1. These 3 bits are not used by the processor designers.

So, we have got a carry bit which is the C Y the carry bit. So, when we are doing some addition or subtraction operation by the ALU. So, it can generate a carry if the carry is generated then this D 0 bit will be equal to 1 telling that a carry has been generated. Otherwise this bit will be 0. Then this bit D 2 is the parity bit. So, telling that whether the result that we have is having an even number of ones or not. If it is not then this p bit will be 1. If it is yes then this p bit will be 0.

Then we have got another bit D 4 which is the auxiliary carry, telling that it is the it is there the after doing the addition. So, after 4 bit whether there was a carry generated or not.

(Refer Slide Time: 18:55)



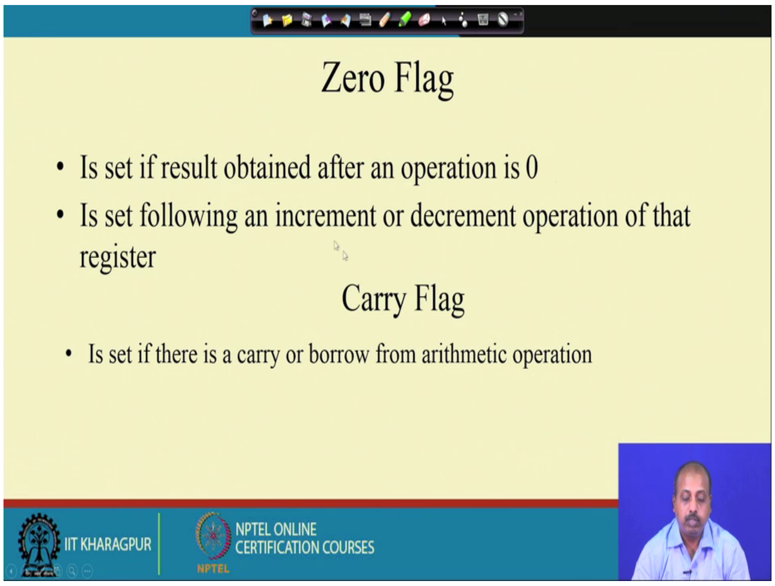
Sign Flag

- Used for indicating the sign of the data in the accumulator
- The sign flag is set if negative (1 – negative)
- The sign flag is reset if positive (0 –positive)

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

So, this will be detailed in the next few slides. So, first one is the sign flag as I was telling that it is used for indicating the sign of the data in the accumulator. It may be negative it may be positive. So, if it is negative then it is 0 it is 1. And if it is positive the bit is 0.

(Refer Slide Time: 19:11)



Zero Flag

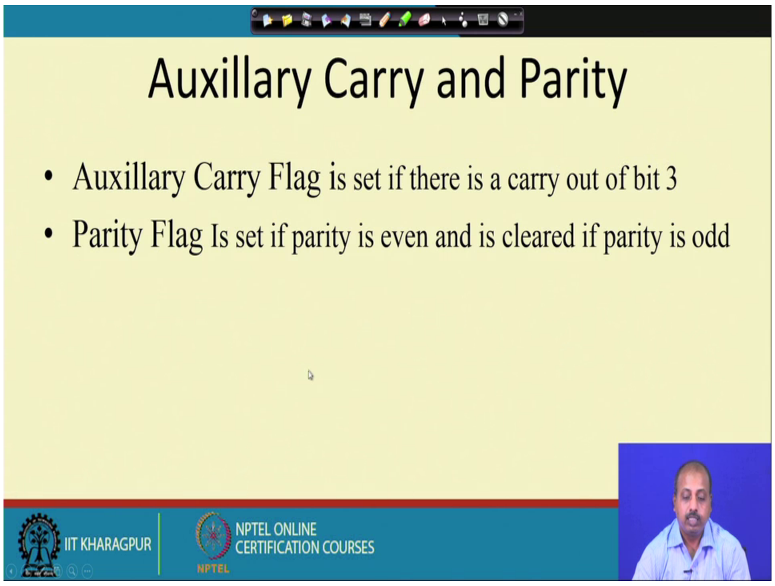
- Is set if result obtained after an operation is 0
- Is set following an increment or decrement operation of that register

Carry Flag

- Is set if there is a carry or borrow from arithmetic operation

We have got 0 flag if the result obtained after an operation is 0, then the 0 flag will be set is set following the increment decrement operation of that register. So, if it is some a register operation is done increment register decrement register or some addition and subtraction A 0 flag may be set. So, carry flag will be set if there is a carry or borrow from the arithmetic operation. So, for subtraction it is a borrow for a addition it is a carry. So, if something is carry or borrow bit is generated then this carry flag will be set.

(Refer Slide Time: 19:46)



Auxillary Carry and Parity

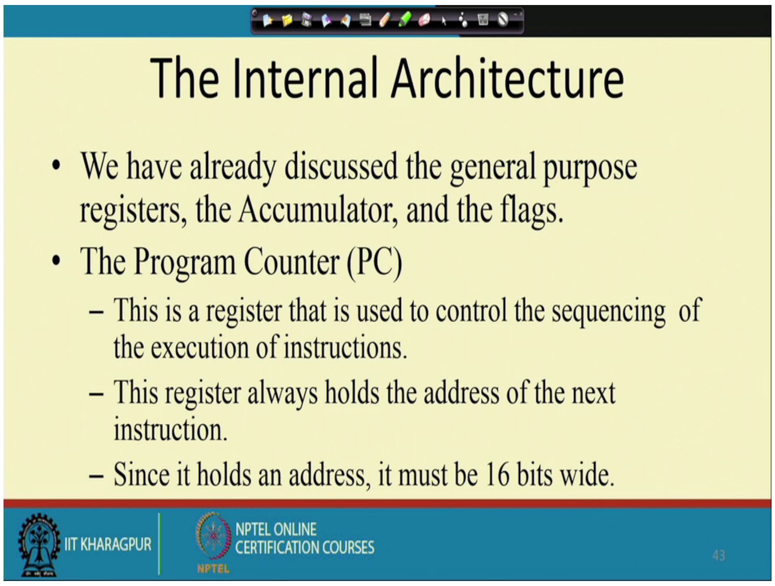
- Auxillary Carry Flag is set if there is a carry out of bit 3
- Parity Flag Is set if parity is even and is cleared if parity is odd

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Then there is auxiliary carry. So, auxiliary carry is set if the carry is there is from bit 3 there is a carry out.

So, from bit 3 so, if they all the result is 8-bit result. So, from a bit number 3 to bit number 4 if there was a carry generated, then these auxiliary carry will be set. So, this is many, many a time this is necessary particularly when we are doing operation with some binary coded decimal format, then this auxiliary carry bit is necessary. So, it is kept for that purpose and this parity flag. So, it is set if parity is even and is cleared if parity is odd.

(Refer Slide Time: 20:35)



The Internal Architecture

- We have already discussed the general purpose registers, the Accumulator, and the flags.
- The Program Counter (PC)
  - This is a register that is used to control the sequencing of the execution of instructions.
  - This register always holds the address of the next instruction.
  - Since it holds an address, it must be 16 bits wide.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

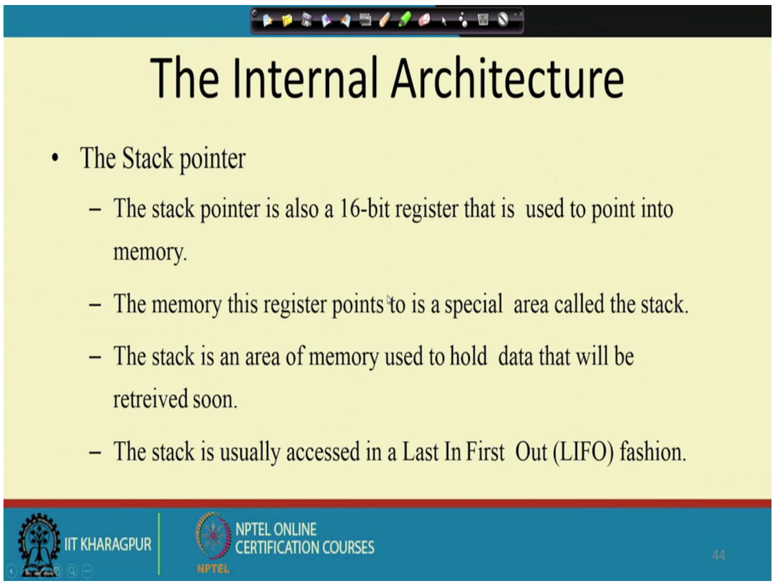
43

So, parity is even means if the number of ones in it in the accumulator is 1, then it is then it is set to 1, and if it is it is odd then it is set to 0. Now apart from these general-purpose registers like we have got special registers like accumulator and flag. The other register that we have is the program counter. So, this is a very important register, because this is used to control the sequencing of execution of instruction. Like which instruction will be executed next by the processor is decided solely by the content of this register.

So, it holds because it holds the address of the next instruction. So since, it is holding the address and address is 64 kilo 60 the address can be up to 64 kilobyte size. So, it can be in the 64-kilobyte range. So, it has to be 60 16 bit wide. So, that way I can have this this program counter register. So, each it will be doing sequencing operation.

So, I hope you understand that if you are trying to jump from one location to another. So, suppose I am at present executing instruction at memory location thousand. And the next instruction I want to execute has to be from location 3000. Then what I need to do essentially is that this program counter value somehow it should be loaded with the value 3000. So, that is the thing to be done. So, this is taken care there. So, this is the done this is the responsibility of the program counter.

(Refer Slide Time: 22:04)



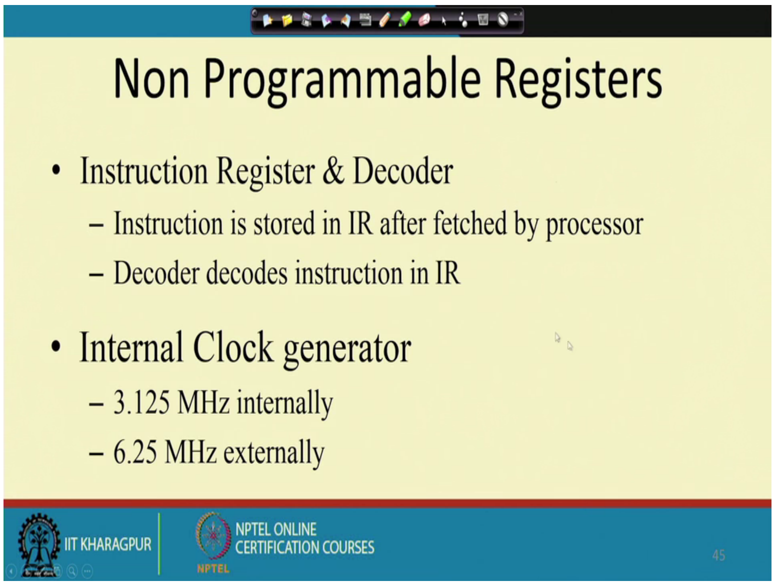
The Internal Architecture

- The Stack pointer
  - The stack pointer is also a 16-bit register that is used to point into memory.
  - The memory this register points to is a special area called the stack.
  - The stack is an area of memory used to hold data that will be retreived soon.
  - The stack is usually accessed in a Last In First Out (LIFO) fashion.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES 44

Another very interesting register that is there is the stack pointer. So, this is also a 16-bit register, and it points to a particular position in the memory. And this memory this register points to is called a special area called stack. So, this is this is an area of memory to hold the data that will be retrieved soon. And this is called this is goes in the last in first out fashion. So, we will come back to this stack later for the time being we just understand that this stack pointer is another 16-bit register, that points to a special portion of memory called stack.

(Refer Slide Time: 22:46)



Non Programmable Registers

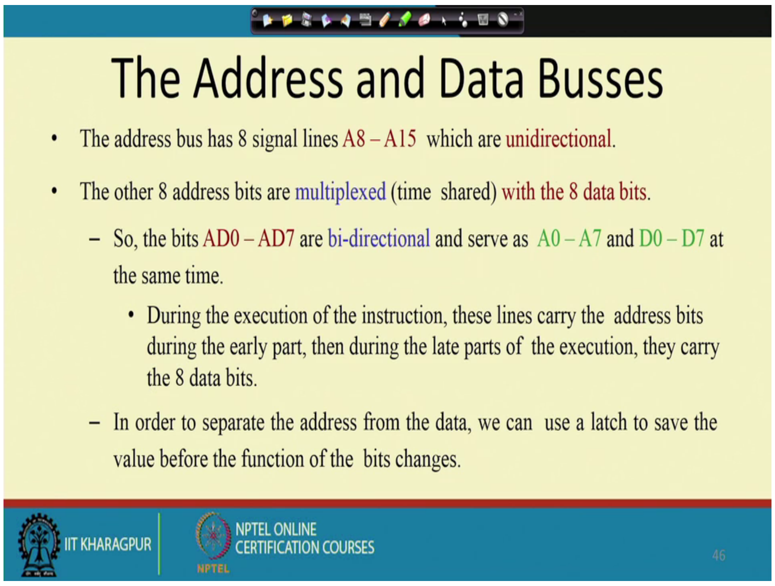
- Instruction Register & Decoder
  - Instruction is stored in IR after fetched by processor
  - Decoder decodes instruction in IR
- Internal Clock generator
  - 3.125 MHz internally
  - 6.25 MHz externally

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES 45

So, we will come to this stack pointer later. There are some other non-programmable registers. One is the instruction register and decoder. So, the instruction which comes from the outside world by so, the processor puts the address of next instruction on to the address bus, and the memory responds with the next instruction. So, next instruction when it comes to the data bus buffer. So, it is directed towards the instruction register.

So, the instruction is stored in the instruction register, after being faced by the processor. And the decoder will decode the instruction in the instruction register. So, instruction registers output will go to a decoder and the decoder will decode that instruction. And there are internal clock generator. So, internally it is 3.125 megahertz, and externally it is 6.25 megahertz.

(Refer Slide Time: 23:45)



The Address and Data Busses

- The address bus has 8 signal lines A8 – A15 which are unidirectional.
- The other 8 address bits are multiplexed (time shared) with the 8 data bits.
  - So, the bits AD0 – AD7 are bi-directional and serve as A0 – A7 and D0 – D7 at the same time.
    - During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits.
  - In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

46

So, the crystal that is connected is of 6.25 megahertz, and internal clock that is generated is 3.125 megahertz.

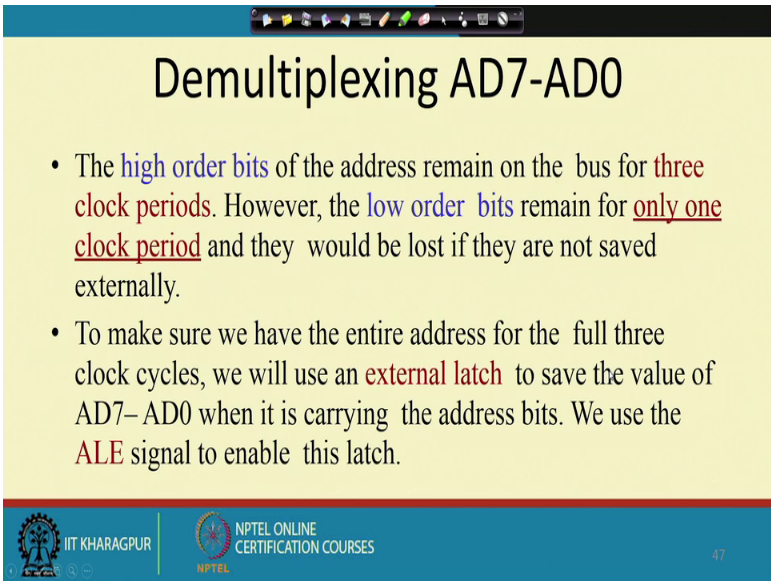
Looking at the address and data buses, the address bus has 8 signal lines A 2 A 15 which are unidirectional, and the other 8 bit so, they are multiplexed back; that is time shared with eight-bit data bus. So, this lines A 0 to A 7. So, they are actually the lower order address bus. So, they are act they are multiplexed with the data bus. So, data bus is 8 bit only so, D 0 to D 7. So, it is multiplexed with that. So, the idea is that when this address bus value is necessary. So, data bus value is not necessary at that point and if it is necessary we have to take care of that, but assuming that they are not used simultaneously. So, we can multiplex their use.

So, that is the way that is the basic idea of multiplexing, when I have got 2 items multiplexed on a line then A 2 items are not used simultaneously. So, I can I can use that for one at a time. So, in a multiplexed fashion, but we will see that it is not always true particularly for 8085; it is not that this address and data buses are not needed. Simultaneously, because while you are trying to read from memory. So, this you have the memory needs that the address be stable, when it is putting the data on to the data where the address bus value should also be stable.

So, it is not a true multiplexing in that sense, but as far as the processor is concerned. So, it is multiplexing in the sense that the same sets of lines are being used for address bus and data bus; so during the execution of the instructions. So, these lines carry the address bits during the early part, and then during the late part of execution carry the data bus data bits. So, in at the beginning they will carry the address bits towards the end it will carry the data bits. In order to separate address from data we can use a latch we will see that is what I was talking about that this is the definition of early part and later part. So, that is a bit fuzzy.

So, where when this where is the boundary so that is not very clear. So, as a result to be on the safe side as a designer; so what is required to do is externally we differentiate this address bus and data bus, this this we do ADE multiplexing of this address bus and data bus. So, that the operations are done properly.

(Refer Slide Time: 26:21)



Demultiplexing AD7-AD0

- The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally.
- To make sure we have the entire address for the full three clock cycles, we will use an external latch to save the value of AD7– AD0 when it is carrying the address bits. We use the ALE signal to enable this latch.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES 47

So, this brings us to the concept of de multiplexing the address and data bus. So, this higher order address bits of the address they will remain on the bus.

So, we have no problem with the lines A 8 to A 15. So, there is no problem with those the lines, because they remain in the bus for 3 clock periods. So, we will see later that when 8085 is trying to access memory. So, it needs 3 clock cycles starting from the point at which the address is put onto the address bus till the memory will put the data on to the data bus and the value comes to the processor. So, that takes 3 clock cycles. So, in that the in the first clock cycle this higher order address bus will have the higher order the address bits A 0 to a sorry A 8 to A 15.

And the lower order address bus it will have the value A 0 to A 7. And this A 8 to A 15. So, that will be holding for all the 3 clock cycles, but this A 0 to A 7 they are held for only one clock period. And naturally you need to latch them externally or save them externally. So, that they are not lost in the remaining 2 cycles, to make sure that we have got entire address for the full 3 clock cycles. So, we need to use some external latch to save the value of this AD 0 to 7 for carrying the address bits.

So, this is done by means of the ALE signal. So, that is so what is done at the at the at the clock cycle and the at the first clock cycle this address, this on the line AD 0 to AD 7 the address bits will be put that is the bits A 0 to A 7 will be put. And this ALE signal is generated by the processor. So, externally we can use this ALE signal to latch these values A 0 to A 7 into some temporary register. And from there the it can the temporary register even when the ALE signal is taken off the value remain latched there, and this temporary register it remain last in that register. And from there we can drive the address lines for the memory.