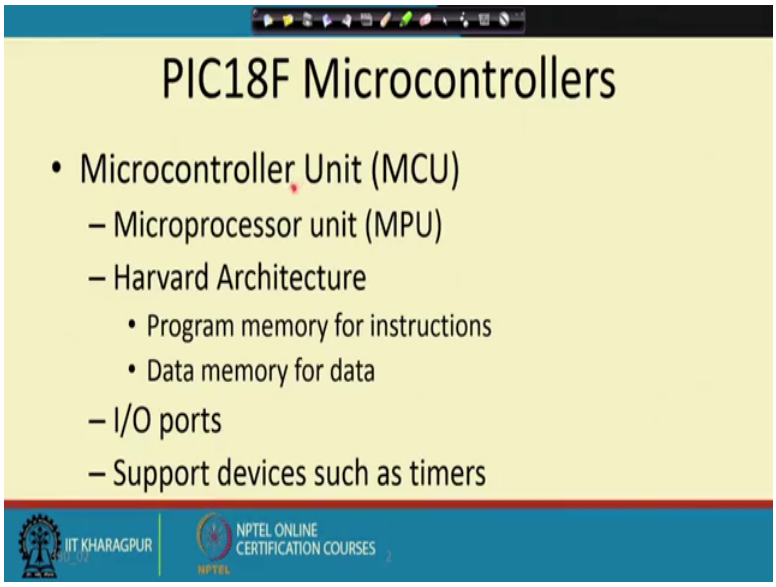
**Microprocessors and Microcontrollers**
**Prof. Santanu Chattopadhyay**
**Department of E & EC Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture – 49**
**PIC**

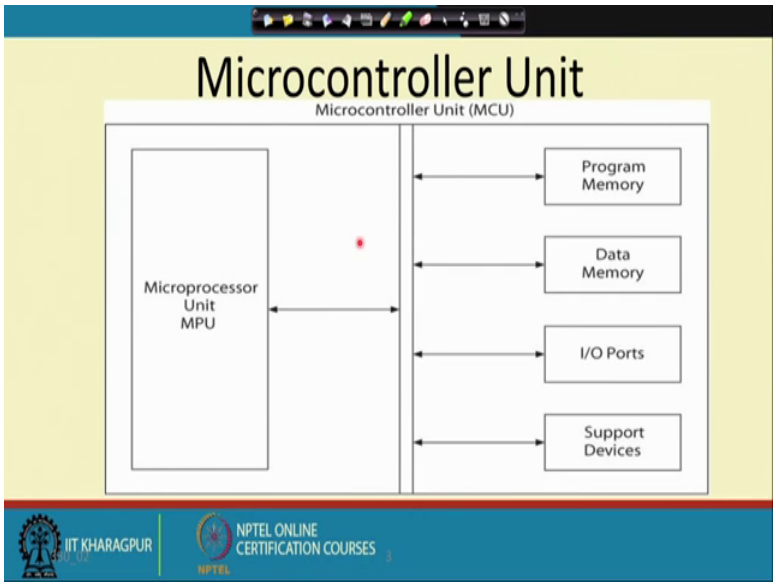
Next, we will look into the PIC microcontroller series. So, this is one of the very simplest microcontrollers that we have and it ranges over a series of such a processer so, we call it a family of this PIC microcontroller. So, you can have very simple process in my microcontroller with PIC or you can have more advanced versions so.

(Refer Slide Time: 00:44)



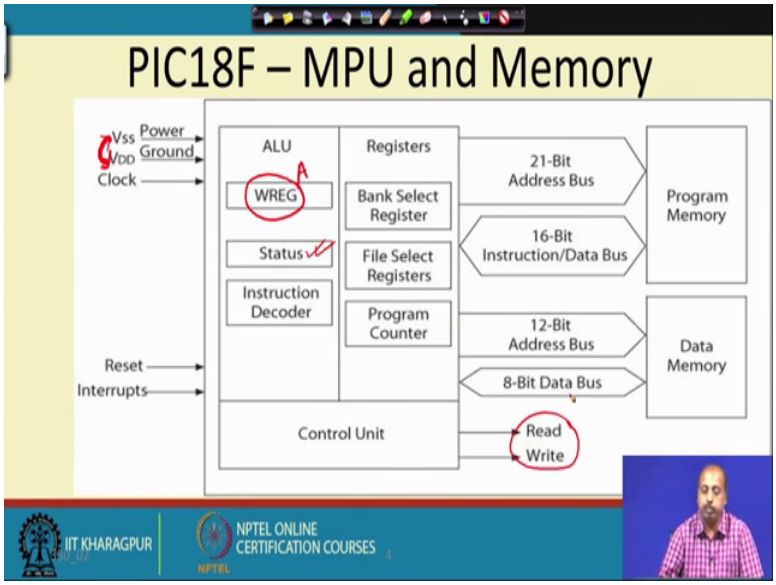

So, when you look into this PIC so, the structure is the consisting of these components there is a microcontroller unit or MCU there. So, and the microcontroller unit it consists of the microprocessor unit which will be doing the processing part it follows the Harvard architecture. So, program memory and data memory they are separate. So, we have got program memory for instruction and data memory for data access, then we have got I/O ports and it will also support device such as timers. So, 18F family of microcontroller so, they will have all these things.

(Refer Slide Time: 01:14)




So, pictorially we can see it like this the as if we have got the microprocessor unit or MPU and there are address and data buses. So, the bus so, they are so, this program memory, data memory, I/O port and support devices they are all hanging from the bus ok. So, this is the program and their address and data buses will be there from which the from the MC MPU they are connected.

(Refer Slide Time: 01:41)




Going into more detail so, this MPU you can divide it into 2 major portion, one part is the ALU part where it will be doing the arithmetic logic operation and the other part is

the register part so, it will be doing the register operation. So, this PIC18F family so, it has got 21 bit address bus that is 2MB of memory program memory and 16 bit instruction or data bus.

So, this is a so, you have got 2 mega 2 megabyte 2 mega locations and each location is 16 bit write and for the data memory part so, I have got 12 bit address bus and 8 bit of data bus ok. So, 12 bit address bus so, you have got 4 kilobyte of data memory and 8 bit data bus so, every data location is 8 bit.

So, this program bus and data bus so, they are separate, the address bus so, for the program memory and address bus for data memory they are separate and we have got the registers like we have got the Bank select registers, we can select the alternate register Bank we can have file select register. So, there are some special function register so, they can be selected by that and you see just like in 8051 we had the situation like the registers are also part of memory.

So, here also the same thing so, registers there for part of the memory. So, we do not have to there is no other CPU register as such so they are all part of the program memory and then there is a Bank select registered and there is a file select register. So, Bank select register is a single register that will tell which register Bank we are going to use and file select registers so, they will select they will act as pointer for the selection of register from the memory.

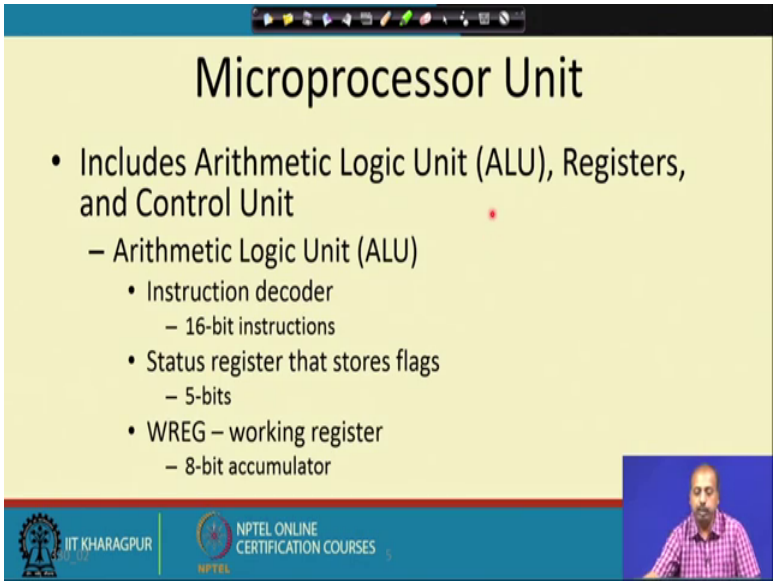
Then we have got the program counter that will be there be there, on the ALU side we have got 1 W register or working register so, this is nothing, but the accumulated. So, unlike other processes we have seen the accumulator. So, this is that A register sort of thing for it will be doing that accumulation job all the operations will be with respect to this W register. Then there is one status register.

So, that is that the status register of other processes also similar to that and one instruction decoder. So, instruction deco so, as you can understand now what is going to happen like the instructions are first coming to this instruction decoder so, instruction decoder will control all other module that are there. And then we have got this is this memory select register so, they will select the register from where the data has to be brought and they will be do you will do some operation with W register as the source as

one of the source and also the destination and then from W register you can move the values to other registers and so, on.

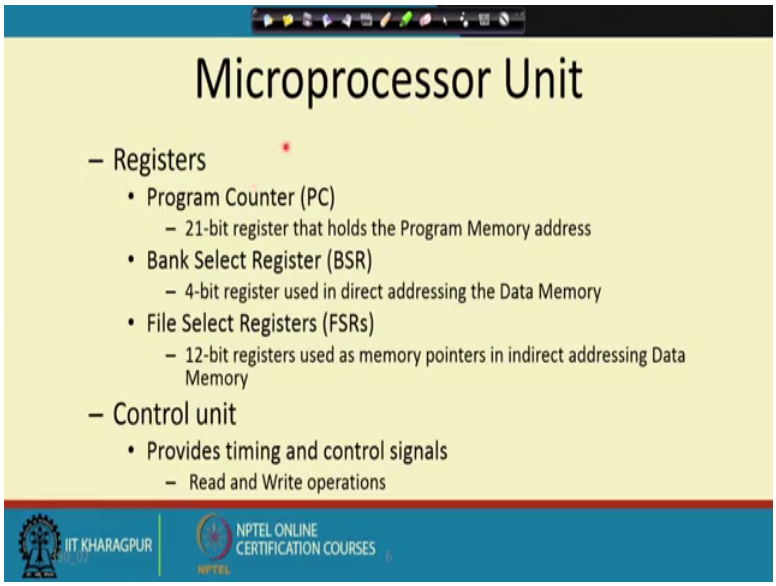
So, you will see some examples later and we have got other things like a V S S and V D D. So, they are power as a view actually there is there is a mistake here this V D D is the power line and V S S is the ground line ok, this will be just the reverse so, these 2 are just the reverse V S S and V D D. Then there is a reset pin by which we can reset the processor and there is one interrupt interrupts are there by which we can have a number of interrupt, then we have got 2 more controllers this read and write. So, they will be controlling the read write operation for the read and write signals for the memory that will be provided by that.

(Refer Slide Time: 05:30)




Now, so, if you have look into this microprocessor unit it has got this arithmetic logic unit, registers and control unit. So, arithmetic logic unit so, it has got the ins it has got arithmetic logic unit will have the instruction decoder so, all instructions are 16 bit instruction. So, you will have this instruction decoder will have a 16 bit instruction register there and this status register it stores the status flag. So, there are 5 bit status register is there and W register is called working register which is the actual the 8 bit accumulator that we have ok. So, this is similar to that accumulator register of 8051 or 8085 so, it is similar to that.

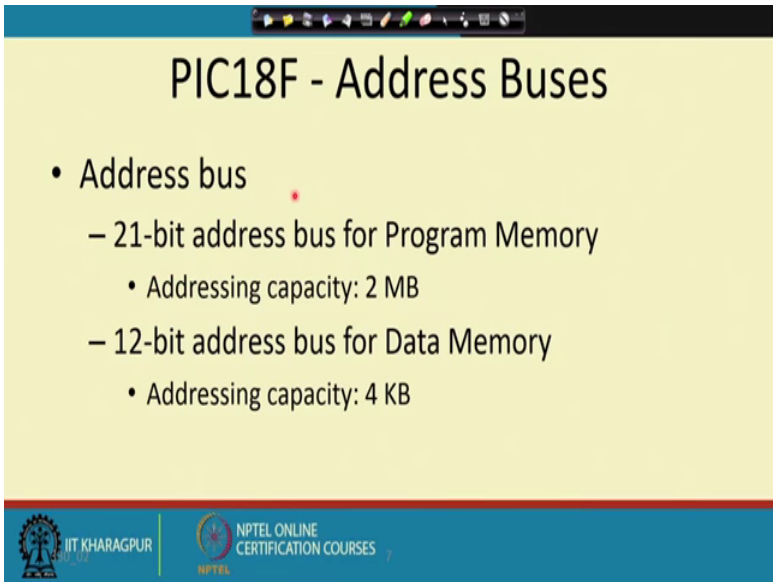
(Refer Slide Time: 06:14)




Next we have this registers microprocessor unit will have some registers like one days important register is the program counter and since the address bus is 21 bit so, program counter is also 21 bit. Then there is a Bank select register which is a 4 bit registered, it is used in direct addressing the data memory. So, you can select the data memory content you can load from the data memory into this register the working register or you can do some operation with that data register with this working register and this data memory register.

So, that way the Bank selection is done by the 4 bit registered called BSR Bank select register and there is a file select registered so, these are 12 bit registers used as memory pointers for indirect address the data memory. So, these are 12 bit registers and they will be used as pointed to the data memory ok. So, they are useful for getting that pointer so, you can in an in indirect addressing mode if you are trying to get some operand or some instruction so, they there will be using this file select register. Then there is control unit to control unit will provide timing and control signals and the read write operations will be done through this control unit.

(Refer Slide Time: 07:32)

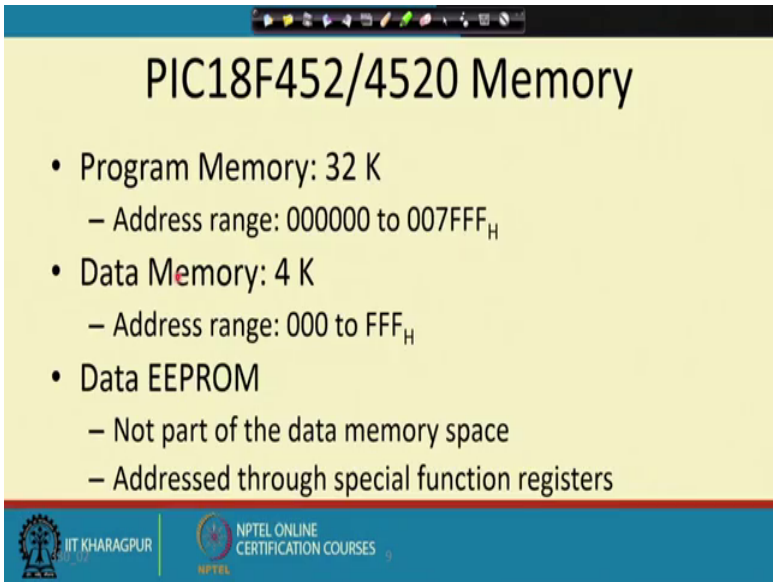



Address bus so, we have got 2 address buses, one is for the program memory and other is for the data memory, for program memory we have got 21 bit address bus. So, addressing capacity will be 2 megabyte and in case of to 2 mega in fact, this is a each location is 16 bits. So, this will be 2 mega into 2 mega word for each word is 2 byte long and then we have got this data memory which is 12 bit address bus. So, this will be 4 kilobyte because the database for data memory is 8 bit. So, even this memory is 4 kilobyte memory.

Database for 16 bit instruction database for program memory; So, when you are accessing a program memory so, program memory may be access for two different purposes, one purpose is to get the instruction and another purpose is something so, that the constants that we have. So, you can load some constant value from program memory into the processor for doing operation. So, basically the immediate operand so, when you got the immediate operand so, they are also stored in the as part of instruction.

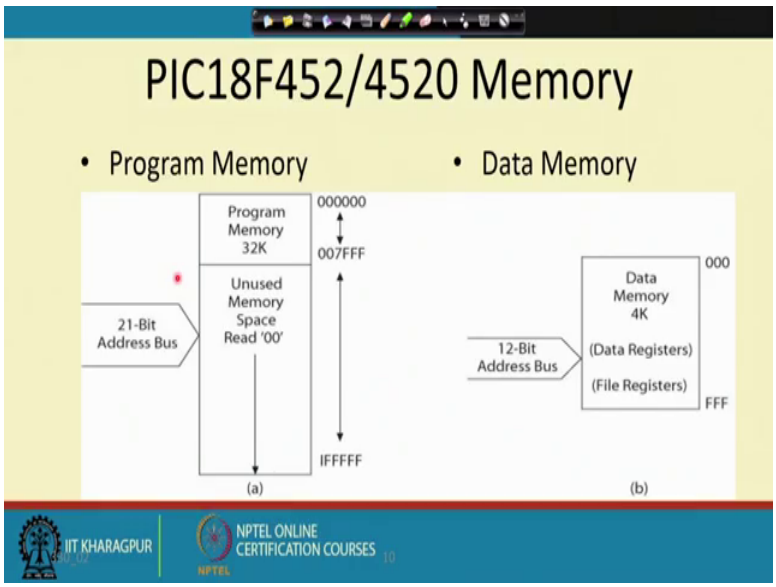
So, that way so, you can say it is a 16 bit instruction all database connecting to the program memory. So, if an instruction uses that constant data so, it will use the 16 bit constant or it can if you if it is storing the values of variables in program in the data memory, then it can use the 8 bit data bus and those variables are going to be 8 bit write and we have got the control signal read and write for the read write operation.

(Refer Slide Time: 09:12)



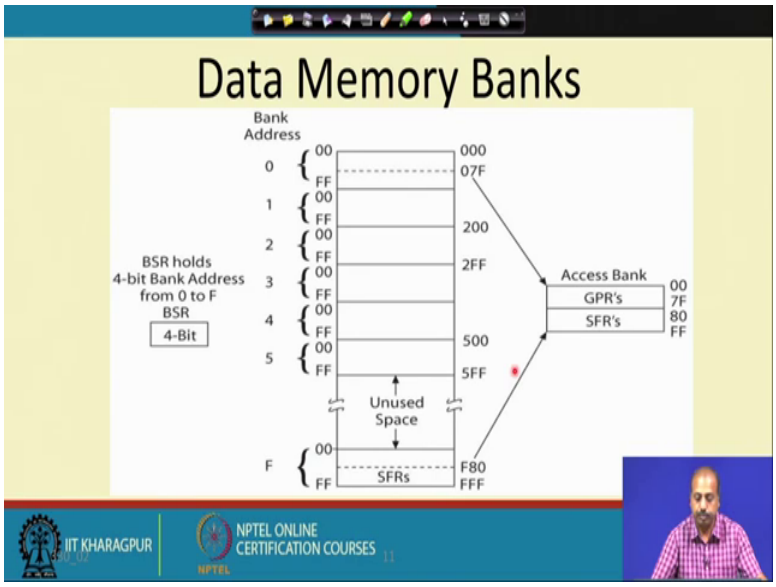

So, 18F family there are many by variant. So, there is 452 or 4520 so, this is a slightly improved version, where the program memory size is 32 kilo and then it is address range is a all 0 to 7FFF and then the data memory size is increased. So, it is going to be 4 kilobyte, address ranges 000 to FFF H and there is another data EEPROM. So, that is not part of data memory space and this is accessed by special function registers that SFRS, we said that there are the Banks select registers and file select register. So, file select register will be using the special function registers for this purpose.

(Refer Slide Time: 09:54)



So, this is the structured so program memory will it will be having a 21 bit address bus, out of that this part will be useful so, 32K will only receive this only 32K. So, this part of the memory is unused. So, from 800 to 1FFFF this part so, this part is unused so, it has got only 32K program memory. On the other hand the data memory so, it has got 12 bit address bus and their that that data memory is consists 4 kilobyte and then it has got data registers and the file registers so, both are there.

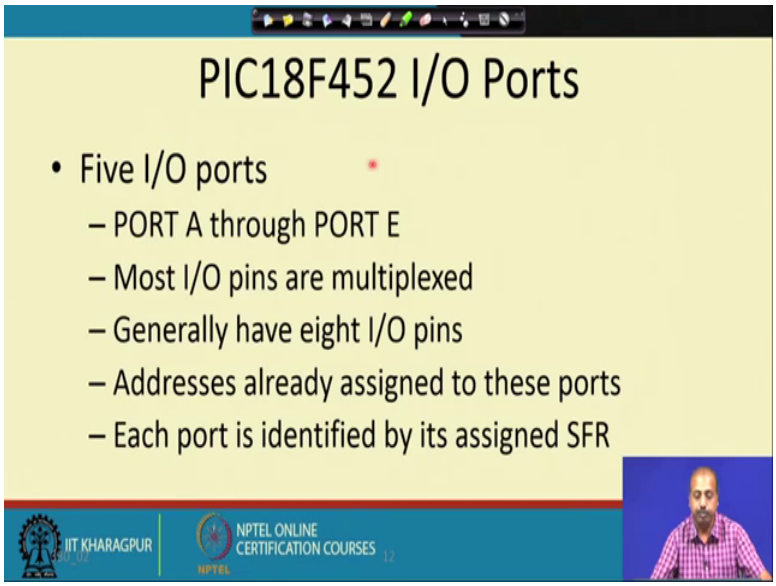
(Refer Slide Time: 10:32)




So, data memory is divided into Banks as I was telling there is a Bank select register. So, Bank select register is a 4 bit register so, it can contain the Bank number 0 to F. So, you have got this 0 to F Bank select and then each of this Banks it will have some general purpose registers and some special function registers. So, this is all purpose register at 00 to 7F ok.

So, when is the general purpose registers can be there. So, that is 128 there are 128 general purpose registers in this location and there is another 128 special function registers. So, each block of 256 data RAM locations data memory locations is divided into registers, out of which 128 at the general purpose and 120 at the special function registers and this you can select this you can select the proper Bank by choosing this BSR bits ok. Setting this BSR bits properly so, you can tell Bank 0 to Bank F so, all the 16 Banks can be selected.

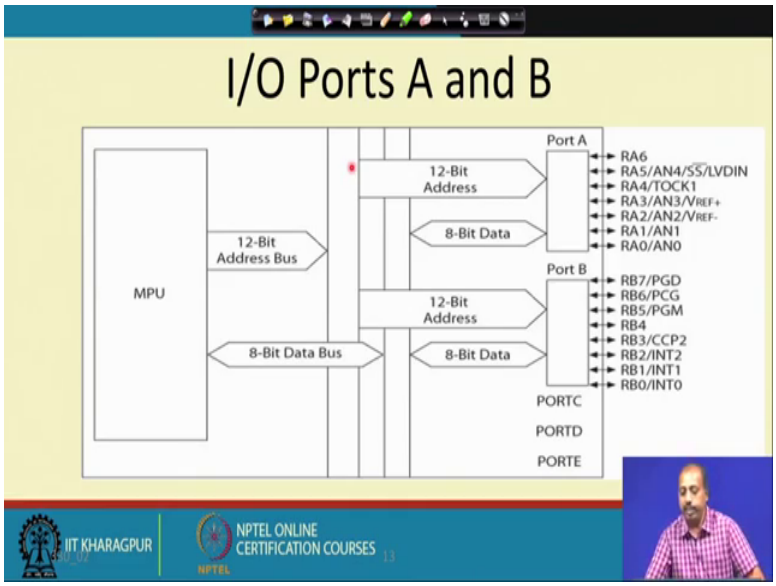
(Refer Slide Time: 11:42)




I/O port so, there are a number of I/O port, PORT A through PORT E so, A, B, C, D, E there 5 such I/O port and most of the I/O pins are multiplexed. So, they are multifunctional just like a 8051 ports they are multifunctional here also the pins are going to be multifunctional. Generally they have 8 input output pins, individual ports they have got 8 input output pin and this addresses are already assigned to this port. So, that address are fix PORT A to port is there at this is at fix and each port is identified by it is special function registers just like your 8051 you have seen that using the memory so, we have got locations corresponding to the ports P0, P1, P2, P3.

So, here also for each of these ports A through E there are corresponding memory location so, which will call special function registers ok. So, just like you can have to this move instruction in 8051 by which we can control the data movement to individual ports and they are actually mapped onto some memory locations. So, here also the same thing so, they are mapped onto memory locations and then data and location and then it is each port is identified by some special function register.

(Refer Slide Time: 12:59)



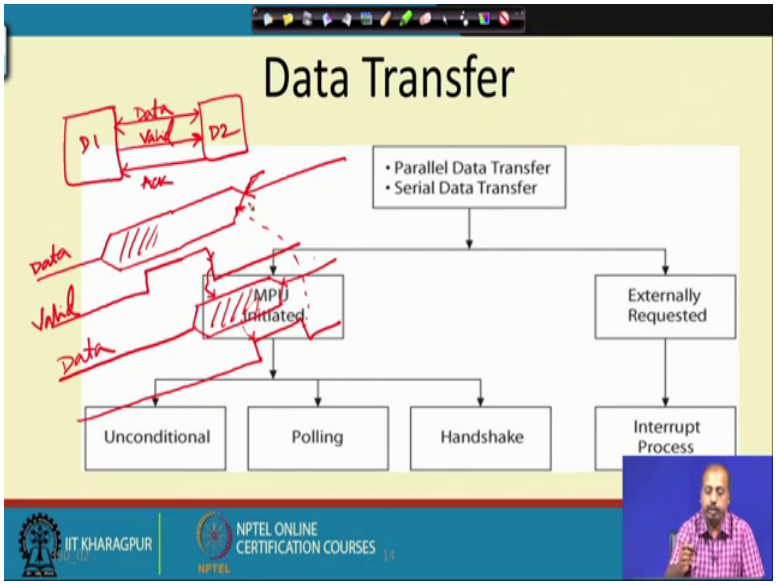

So, this is the idea so, I have got this PORT A, POPRT B, PORT C, PORT D, PORT E like that and then this from the processor we have got for the data memory 12 bit address bus and 8 bit data bus. So, this they are connected to 12 bit address so, that will select some PORT A and this 12 bit address it will also select PORT B there will be naturally this address are fixed ok. So, if you put a pa particular address the PORT A will get selected or PORT B will get selected and you see that these are the alternate that we have so, here it is not very clear like what do they mean. So, basically like this life of these are actually this there actually this port lines RA 0 to through RA 6. So, these are the port lines.

So, it is a here this is 7 lines as shown and it has got some alternate function analog 0, analog 1 and analog 2, analog 3. So, these are also used for so, you can connect some ABC through this and there whenever you are if you are doing some ABC analog to digital conversion then this analog channels can be connected there.

So, if there are 4 analog channels that can be connected for analog to digital conversion and this is V REF plus V REF minus so, this is used for the AD, ADC and then this is the some other pins like a time this is for timer clock and I think something like that. So, then there is a program these are some other control bit so, you can the you can find out their meaning from the manual, but they are.

So, what I want to show is that all these pins they there multiplex they are not a absolutely dedicated for individual put. Then this INT 0, INT 1, INT 2 so, these are the these are for interrupts so, that that way just like 8051. So, we have got the pins multiplex the port pins multiplex so, here also the same thing we have got this port pins multiplexed.

(Refer Slide Time: 14:58)




So, if you talk about the data transfer so, there can be parallel data transfer or serial data transfer. So, both these data transfer technique so, you to you can you can divide it into 2 categories, one category is the transfer is initiated by the microprocessor unit or the transfer is initiated by the external device.

So, either of them can be utilized so, when it is microprocessor initiated it may be un conditionals so, un conditional means the processor will just put the data onto the data bus for the port for the device and it is assumed that the device will accept it. So, there is no other control over this whole transfer process or there can be pulling. So, this processor will pull the device to see whether the device is ready to send or receive data. So, if it is (Refer Time: 15:51) if it is ready then only the transfer will take place, that is the pulling mode of transport or there can be handshake.

So, when the processor is ready processor is ready to send some data so, it will send the it will put the data onto the and data bus and it will put some valid signal. So, between 2 module say if we are looking into the handshake mode of data transfer so, it is like this.

So, if this is a device 1 and this is a device 2 then there is this is at this is a data line that is connected. And so, there are some more control at least 2 controls the necessary one is called valid the data valid and the other is called the acknowledgement.

And the transfer takes place like this that the initial D1 it will put that data onto the database so, this is the data it is better shown in this fashion. So, this data is normally shown the valid data is normally shown like this that suppose at this point from this point onwards the data is valid, data has been put on to the data bus. So, this is a data line here I have got the data.

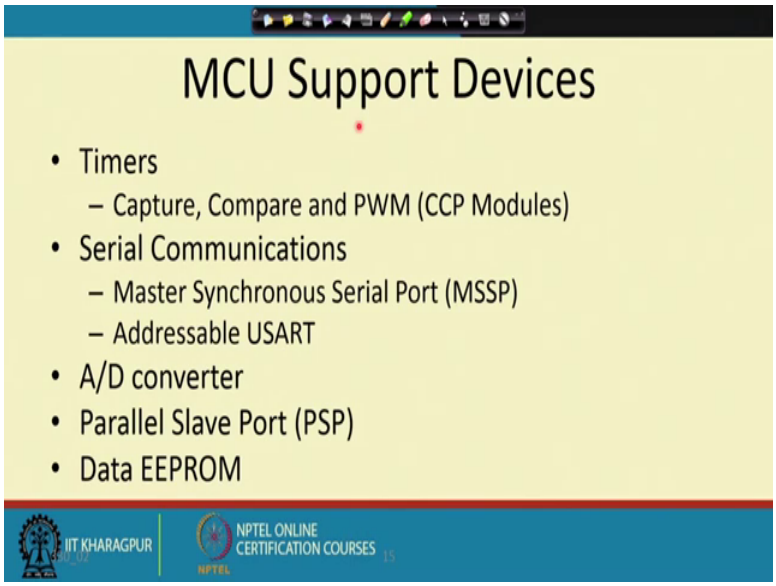
Then after putting the data this it will put the valid signal so, valid signal is activated. So, when they saw this the data will be there as long as the valid signal is there after sometime this valid signal is taken out and then when they were on this while it is falling down so, at that time so, this is the valid signal the data is received by the receiver. So, this data will be received by the receiver. So, data will be received by the receiver receiving device like this.

So, this is happened by this so, this falling edge so, the value will be it will be triggered in this data receive. So, this is data at the receiver end. So, the value is data is received here and after that data has been received so, it will be so, it will be going into it will be putting that acknowledgement signal. So, this acknowledgement signal is put some where here so, this acknowledgement signal is given.

So, this is basically given that data has been rate so, it will be giving this acknowledgement signal. And upon getting this acknowledgement signal so, this data is taken of the data is taken off from the bars. So, this acknowledgement signal when it is given this may be trigger in the removal of data from the data line. So, that way there will be proper handshake between the 2 devices for the data transfer.

So, we can have this type of handshaking mode of data transfer that is more conflicts and if it is activated by external device then this is an internet so, when the device is ready to do the transfer. So, it will give an interrupt to the MPU and then this the MPU will start the data transfer process.

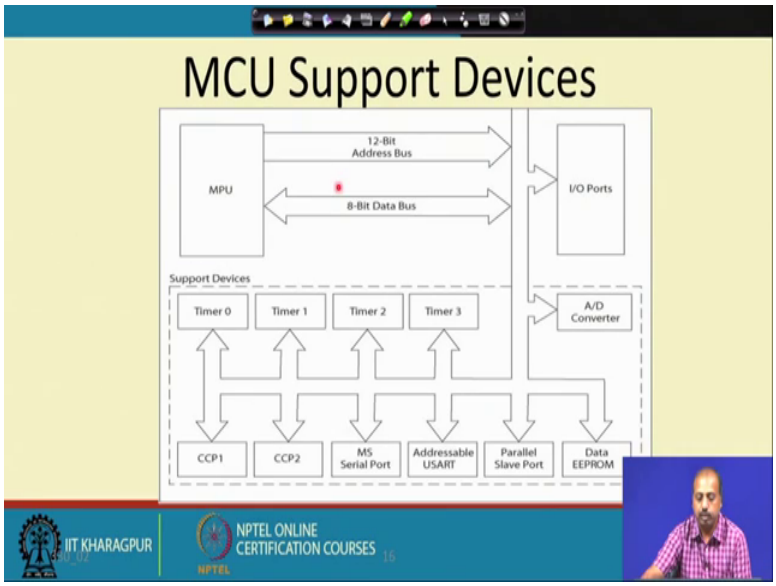
(Refer Slide Time: 19:06)




So, there are a number of devices that are supported by this processor like this a timer so, it will capture compare and this pulse with modulation so, this will be all this can be done by this timer. Then there are serial communication so, that a done by this Master Synchronous Serial Port MSSP and also have one user.

So, this Synchronous Serial Port so, this is for synchronous communication and is user this is universal synchronous, asynchronous receiver transmitter. So, that is for normally it is for this asynchronous transmission. Then we can have A/D converter you can a Parallel Slave Port or PSP. So, or we can have this data EEPROM. So, these are the various types of divide devices that you have with a PIC microcontroller. So, you can if your devices support any of these mode or if you if you if you need all this things you can connect to the device, like if you have to have some analogue signal. So, you can connect to this A/D channel and A/D converter is there so, which will be taking care of that.

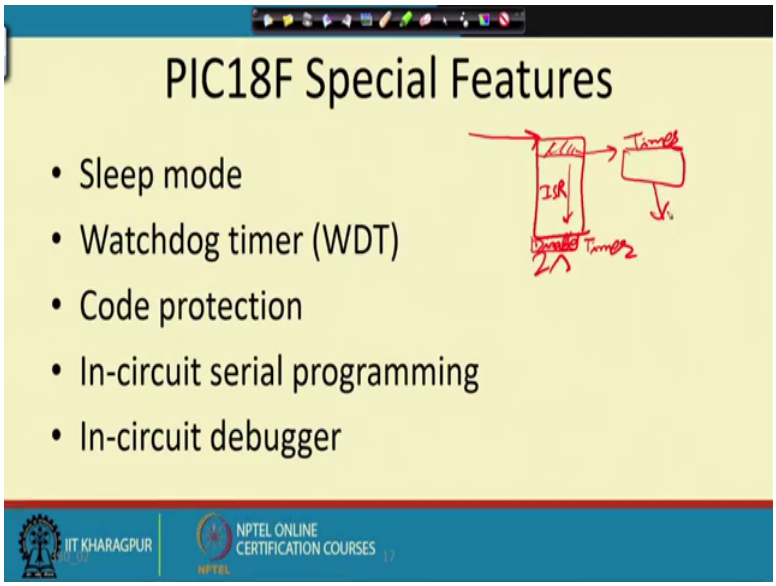
(Refer Slide Time: 20:20)




So, this is the thing that I was talking about that this microprocessor unit. So, it has got this I/O ports and this that data memory is there and the data memory part so, it will support this support devices.

So, you have got this Timer 0, Timer 1, Timer 2, Timer 3. So, there are 4 timers there is 1 A/D converter and it has it is a multi channel A/D converter as we have seen, there are 4 channels why this and all this A/D converter can be connected. Then we have got this other devices CCP1, 2 then this Serial Port User, Parallel Slave Port, Data EEPROM. So, all this can be connected oh so, they are they are they are this devices are supported. So, if you want to this devices you can use them.

(Refer Slide Time: 21:05)




Special features so, it has got a sleep mode. So, in which in which the processor will be we can make the processor to go into sleep mode. So, it will the power consumption will be low, then there is a watchdog timer mode so, it will be doing some. So, watchdog timer is it is for if some deadline miss is going to occur you can detect that situation. So, it is like this that we have got suppose the we have got some interrupt and that if that interrupt occur so, these interrupt service routine has to be executed.

Now, if this interrupt service routine is not complete within some pre defined time. For example, for some critical operation maybe we say that this interrupt service routine should be over within 2 second. Now, if that inter service routine is not over by within 2 second then it maybe take it is taken as a failure for the system and some higher level activity has to be triggered. For example, if there is a way smoke detectors so, that detects some fire and then this fire extinguisher are not activated within a very small amount of time.

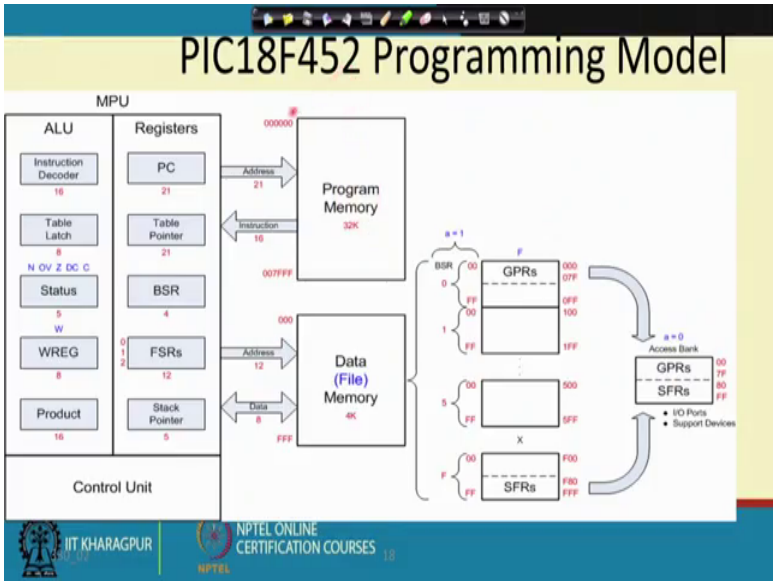
So, we need to take some high level action maybe we have to inform this fire brigade about this emergency condition and all that. So, that way so, there will be some fix time unit by which this ISR should be over. So, to do that so; what we do at the beginning of the ISR apart from writing this ISR code so, we also start a timer. So, you also started timer.

And then either this ISR in the end of the ISR we disable that timer we disable the timer at the end. So, if the ISR is over within 2 second then this timer will get disabled so, it will not be going to affect the system operation or if this is ISR is not over within 2 second then this timer will overflow it will time out and it will generate another interrupt that will be treat a higher level interrupt.

So, this a we can have this watchdog timer so, this type of timer they are one time timers and they are called watchdog timers. Then we have got code protection so, I can protect the code for access so, they are so, we can have some address range and protection around that. Then in circuit serial programming so, that is possible. So, you have so, we can have so, you can we while the circuit is in operation so, we can serially transfer the some programs to the power to the microcontroller and there is in circuit debugger.

So, this will be this will be doing that debug operation debug job so, this will be doing the debug job and there the debugger will be useful for (Refer Time: 23:57) finding problems with the system next.

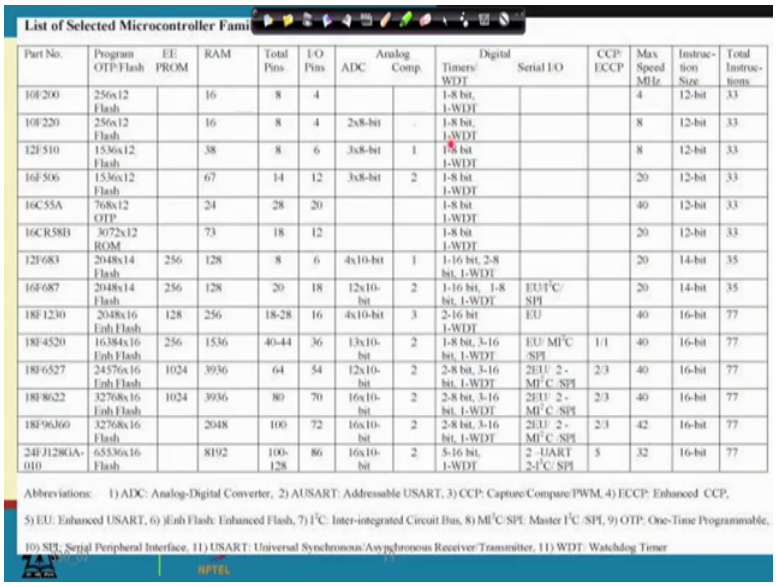
(Refer Slide Time: 24:05)




So, this is that programming model so, as a user of the system so, you can find out that we have got this MPU that has got this ALU and registers so, ALU has got instruction register. Then the table latch, then this we have the status flag 5 bits flag, then that the working register and there is a product register which is for multiplication result and we have got this the registers program counter, then this table pointers.

So, they will be pointing to this various location for indirect access, then this bit select register, then this a file select register there are 3 register FSR1, 2 and 3, they can be used for selecting that. So, if you have got these BSR using 0 so, it will be selecting this Bank 0, BSR1 will selecting Bank 1 like that and again this Bank 0 so, it is divided into G GPRS and SFRS. So, general functions registers and I/O special function registers so, plus there are I/O sub codes and other support devices that are there.

(Refer Slide Time: 25:16)



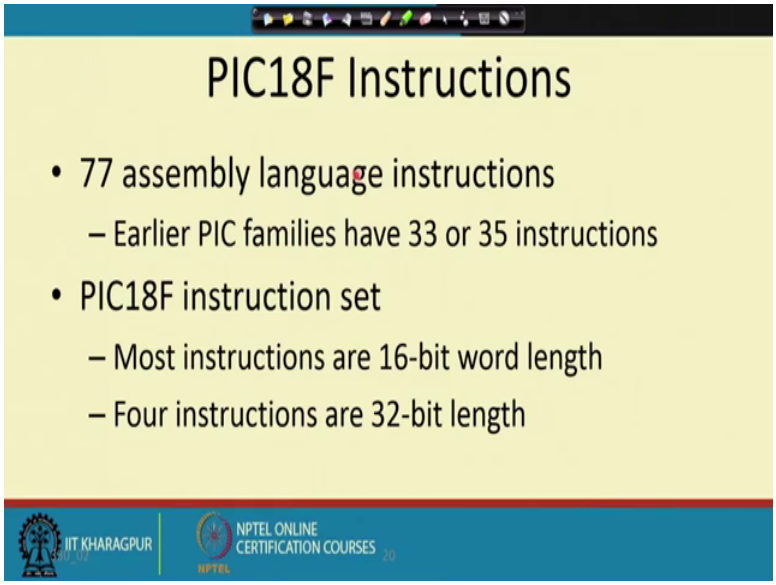
List of Selected Microcontroller Fami

| Part No. | Program OTP/Flash | EE PROM | RAM | Total Pins | I/O Pins | Analog ADC | Analog Comp. | Digital Timers/ WDT | Digital Serial I/O | CCP/ ECCP | Max Speed MHz | Instruc-tion Size | Total Instruc-tions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10F200 | 256x12 Flash | | 16 | 8 | 4 | | | 1-8 bit, 1-WDT | | | 4 | 12-bit | 33 |
| 10F220 | 256x12 Flash | | 16 | 8 | 4 | 2x8-bit | | 1-8 bit, 1-WDT | | | 8 | 12-bit | 33 |
| 12F510 | 1536x12 Flash | | 38 | 8 | 6 | 3x8-bit | 1 | 1-8 bit 1-WDT | | | 8 | 12-bit | 33 |
| 16F506 | 1536x12 Flash | | 67 | 14 | 12 | 3x8-bit | 2 | 1-8 bit 1-WDT | | | 20 | 12-bit | 33 |
| 16C55A | 768x12 OTP | | 24 | 28 | 20 | | | 1-8 bit 1-WDT | | | 40 | 12-bit | 33 |
| 16CR58B | 3072x12 ROM | | 73 | 18 | 12 | | | 1-8 bit 1-WDT | | | 20 | 12-bit | 33 |
| 12F683 | 2048x14 Flash | 256 | 128 | 8 | 6 | 4x10-bit | 1 | 1-16 bit, 2-8 bit, 1-WDT | | | 20 | 14-bit | 35 |
| 16F687 | 2048x14 Flash | 256 | 128 | 20 | 18 | 12x10-bit | 2 | 1-16 bit, 1-8 bit, 1-WDT | EUSART/ SPI | | 20 | 14-bit | 35 |
| 18F1230 | 2048x16 Enh Flash | 128 | 256 | 18-28 | 16 | 4x10-bit | 3 | 2-16 bit 1-WDT | EU | | 40 | 16-bit | 77 |
| 18F4520 | 16384x16 Enh Flash | 256 | 1536 | 40-44 | 36 | 13x10-bit | 2 | 1-8 bit, 3-16 bit, 1-WDT | EU/ MI2C /SPI | 1/1 | 40 | 16-bit | 77 |
| 18F6527 | 24576x16 Enh Flash | 1024 | 3936 | 64 | 54 | 12x10-bit | 2 | 2-8 bit, 3-16 bit, 1-WDT | 2EU/ 2 - MI2C /SPI | 2/3 | 40 | 16-bit | 77 |
| 18F8622 | 32768x16 Enh Flash | 1024 | 3936 | 80 | 70 | 16x10-bit | 2 | 2-8 bit, 3-16 bit, 1-WDT | 2EU/ 2 - MI2C /SPI | 2/3 | 40 | 16-bit | 77 |
| 18F96J60 | 32768x16 Flash | | 2048 | 100 | 72 | 16x10-bit | 2 | 2-8 bit, 3-16 bit, 1-WDT | 2EU/ 2 - MI2C /SPI | 2/3 | 42 | 16-bit | 77 |
| 24FJ128GA-010 | 65536x16 Flash | | 8192 | 100-128 | 86 | 16x10-bit | 2 | 5-16 bit, 1-WDT | 2 -UART 2-I2C/ SPI | 5 | 32 | 16-bit | 77 |

Abbreviations: 1) ADC: Analog-Digital Converter, 2) AUSART: Addressable USART, 3) CCP: Capture/Compare/PWM, 4) ECCP: Enhanced CCP, 5) EU: Enhanced USART, 6) Enh Flash: Enhanced Flash, 7) I2C: Inter-integrated Circuit Bus, 8) MI2C/SPI: Master I2C /SPI, 9) OTP: One-Time Programmable, 10) SPI: Serial Peripheral Interface, 11) USART: Universal Synchronous/Asynchronous Receiver/Transmitter, 11) WDT: Watchdog Timer

So, this is a list of different PIC controllers that we have so, their capacities and things.

(Refer Slide Time: 25:24)



So, if you look into this instruction so, there are 77 assembly language instructions. So, the previous versions it had 33 or 35 instructions, but this new version it has got more number of instructions that way this PIC has become more powerful.

So, compared to other processes the numbers of instructions are much less, but we will see that this is quite powerful for doing the microcontroller applications.