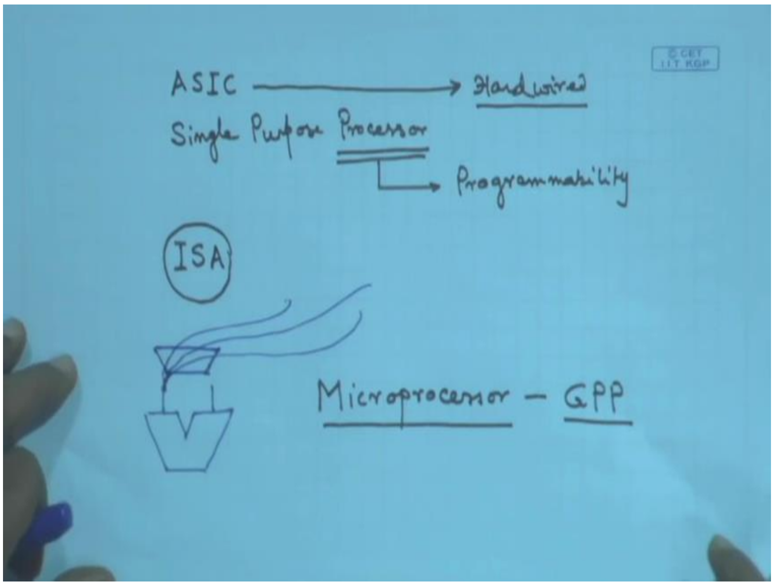
Embedded Systems Design
Prof. Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 03
General Purpose and ASIPs Processor

In the earlier class, we have seen that we can have 3 types of processors. Typically one is a general purpose processor or application specific instruction processors and application specific ICs which are purely or we can call it also as single purpose processors.

(Refer Slide Time: 00:47)

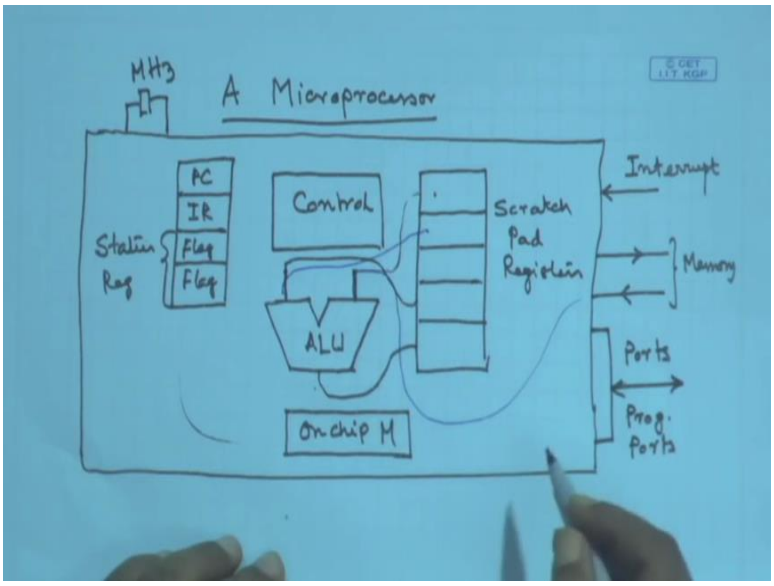


That means they can; now there is a slight difference between an ASIC and a single purpose processor. Whenever we use the term processor then there is some programmability that is being mentioned along with it. But in the case of ASIC, it is an application specific IC that has been designed for a specific purpose. So, this is completely hardwired; this is completely hardwired and I do not have any scope of programming it and changing the functionality, but for single purpose processors; there may be a little bit programmability.

Now, yesterday or in the last class; we also had talked about the instruction set architecture. The instruction set architecture is what distinguishes between the architectures of the different processors. Say Pentium will have a set of instructions and those instructions; I mean those instructions will have different features and accordingly there will be different instruction cycles; different ways in which the instructions are executed. Now if we move to ARM; another processor or some other DSP processor, in that case also we will have different instruction sets. Now a general purpose processors since it caters to a larger scope of different problems, the instruction set is quite large, on the other hand for ASICs; application specific instructions processors there is a much less.

Now, the best example of a general purpose processor, one maybe you can say a spark, for spark machines we have got different processes, for Pentium machines we have different processors, but for embedded system, the general purpose processor is represented by the microprocessor. A microprocessor; all of you are exposed to this, there are different varieties of microprocessor, different companies are coming up with micro processors. Now these microprocessors are also general purpose processors. They have got their control unit, they have got their arithmetic logic unit, and they have got relatively larger instruction set, there are ports. So typically a microprocessor; what will have a microprocessor.

(Refer Slide Time: 03:47)



A hypothetical microprocessor, we will have of course some control, it will have an arithmetic logic unit, I am assuming a simple microprocessor where we have got only one ALU, no parallel processing, there is a bunch of registers here which are also known as scratchpad registers. And there are some control registers; some control registers like the program counters, instruction register all those things are there which are required for the control. And there are some status flags some status registers these are the status registers.

Besides these there may be in the modern microprocessors, there may be some on chip memory, some on chip cache then maybe some on chip memory; on chip memory and there are so, if I now encapsulate the microprocessor then there are some interfaces for communicating with the external world like there can be interrupt lines, there can be other lines for memory interfaces going to the external memory and there can be ports for communicating with the external world. Now these ports are often programmable. So, these are often programmable ports; that means that I can program these ports either to be acting as an input or as an output. So, this is a microprocessor.

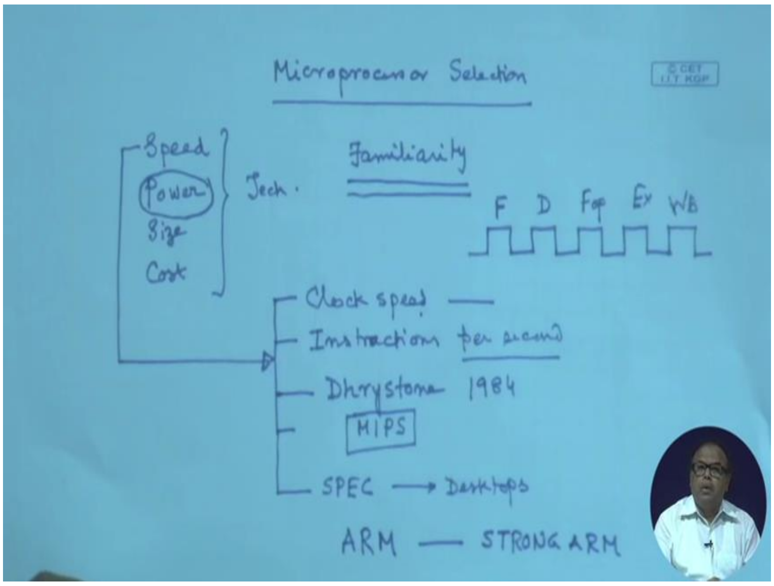
Now, this is driven by a clock, of certain mega hertz; some mega hertz. Now it is going up further to 1000s of mega hertz, but gigahertz is now a common place now. So, we are having, but by the way just the increase in the clock speed is coming to a saturation, now we have seen in the past decade that the clock speed of the microprocessor were shooting up, but now that is gradually saturating because of technology reasons now. So, this is so we have got some components inside the microprocessor which executes different instruction set.

Now where does the where does we say the instruction set architecture, now what role does this instruction set architecture play with respect to this architecture? The instruction set architecture is modeled by the way the instructions the instructions are taken from the memory and through some path data path it will come to the instruction register right and that will be decoded in the controller after the decoding different parts data can be of I mean brought in fetched or I can fetch data from the registers, this can be connected with the registers or with the other lines.

Now, all these for the different instructions, there should be different paths that will be activated during execution and that activation is being done by the controller. So, the controller, as we discussed, the controller is essentially a state machine which is in the present state depending on some inputs, it goes to the next state like that it goes on and the data that was coming, suppose I fetch the instruction from the memory. So, there must be a data path that will bring it here, see in 1 instruction, I want to fetch the data from the register, 2 registers, this one and this 1, say here and the other one from this register here and do the ALU activity and the result will go here that is one scenario.

Now, in another scenario maybe the instruction thus being executed is taking a data from the memory and is coming here, another one is coming from this register here. Now how can that be made possible? Now I have got only one ALU, I have got one ALU and there are 2 inputs, now the data can come from this path or from another path or from another path. So, how do I select which path should be activated depending on the instructions that is all of you will immediately understand. We need a multiplexer that is why if you recall in the earlier class, we save the data path consists of the computing elements, the words, the multiplexes, buffers and all this that is the data path. So, this is how a microprocessor which is the most we can consider general purpose processor is what?

(Refer Slide Time: 11:15)



Now, for an embedded system, how do we go about selecting a microprocessor, now how do you select the microprocessor that is a very important issue when we go ahead with some embedded system design, now one thing is of course, speed; how much speed?

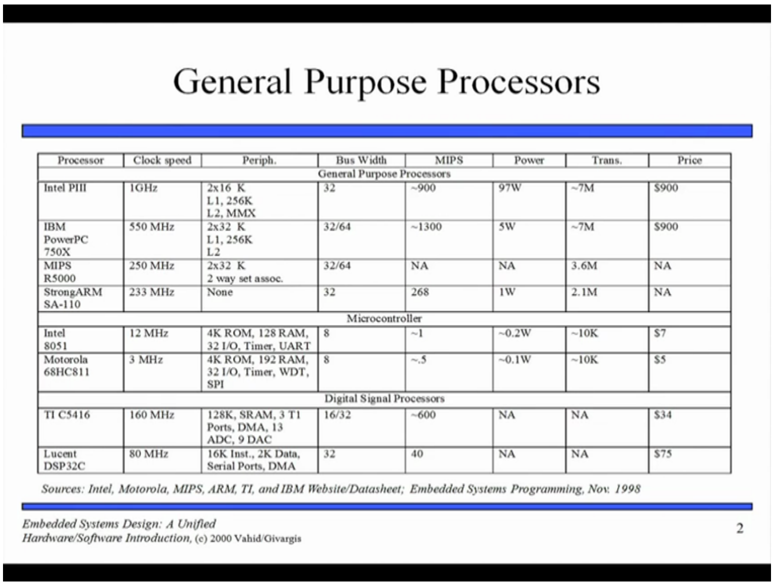
Another point is the power that is spent, the size cost; these are all the technical specs value, the other aspect that you should also look at is whether you have got some expertise, whether I am familiar with the instruction set, your familiarity, the designer’s familiarity, expertise and all those things. So, that is also an important point because as we said that the time to market is a very important thing, it is better if you can choose a microprocessor about which you are familiar, you have already done some development. So, better that can also come into your optimization decisions.

Now, when we just, I would like to touch up on how do we talk about speed? When we talk about speed, what are the factors that come with speed? one is of course, the clock speed; the faster the clock, the faster this clock is the faster the instructions will be executed that is true, but instructions per cycle may differ, different instructions may take different cycles therefore, clock speed will determine the say, this is the clock pulse and we have got, say here some fetch is being done; some decode is being done; I am doing the fetch operand and then I am doing the execute then maybe I am doing some; back here, now for some instruction I may need all these for some instruction maybe just I will not need any operand, there is an immediate data. So, I can execute that. So, the instructions cycle is very important.

Another point is in number of instructions I execute part second, now the clock speed is there, but also the instruction cycles are important. Now, therefore, we have got different benchmarks to see, how the micro processors perform, for example, just for the reference Dhrystone benchmark is a very popular; synthetic benchmark that was developed in 1984; it talks about the MIPS million instructions per second and now MIPS architecture is also another architecture in the paradigm of risc machines, there is besides Dhrystone, there is spec is a set of more realistic benchmarks, but this more tune towards the desktops and that sort of thing spec benchmarks there are other benchmarks also we will you can find that out.

Often when we look at the microprocessor, the microprocessor that will be using we look at the different parameters now power time and again I will be emphasizing on the power that it consumes if you increase the speed is as it becomes faster than the power consumption also is high, the arrangement that you will have to make for the heat dissipation will also be much more complicated.

(Refer Slide Time: 16:14)

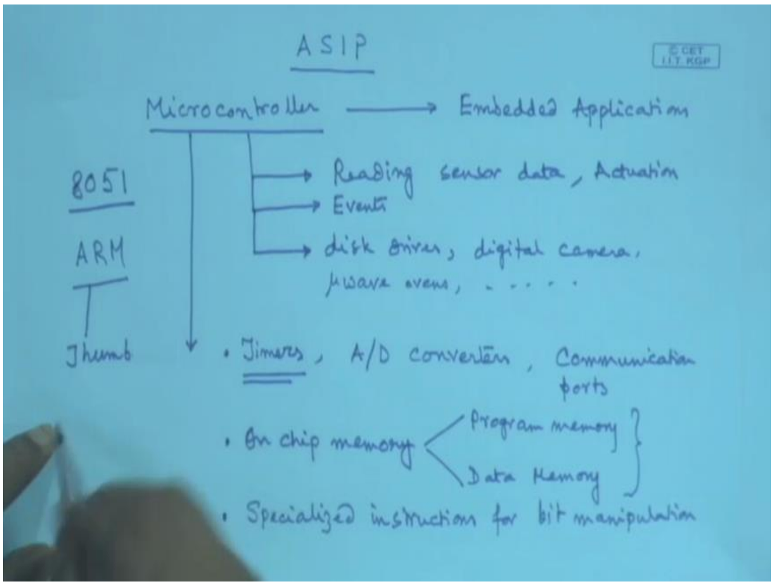


Processor	Clock speed	Periph.	Bus Width	MIPS	Power	Trans.	Price
General Purpose Processors							
Intel PIII	1GHz	2x16 K L1, 256K L2, MMX	32	~900	97W	~7M	$900
IBM PowerPC 750X	550 MHz	2x32 K L1, 256K L2	32/64	~1300	5W	~7M	$900
MIPS R5000	250 MHz	2x32 K 2 way set assoc.	32/64	NA	NA	3.6M	NA
StrongARM SA-110	233 MHz	None	32	268	1W	2.1M	NA
Microcontroller							
Intel 8051	12 MHz	4K ROM, 128 RAM, 32 I/O, Timer, UART	8	~1	~0.2W	~10K	$7
Motorola 68HC811	3 MHz	4K ROM, 192 RAM, 32 I/O, Timer, WDT, SPI	8	~.5	~0.1W	~10K	$5
Digital Signal Processors							
TI C5416	160 MHz	128K, SRAM, 3 T1 Ports, DMA, 13 ADC, 9 DAC	16/32	~600	NA	NA	$34
Lucent DSP32C	80 MHz	16K Inst., 2K Data, Serial Ports, DMA	32	40	NA	NA	$75

Now, given this; let us have a look at different general purpose processors; if you look here, we can see that Intel’s Pentium 3, the clock speed was one Giga IBM power PC 550 megahertz, MIPS machines 250 megahertz, I should have mentioned while I was discussing about microprocessor, about the strong ARM processor, these are very important processor that has found a lot of application ARM and also strong ARM.

Now, let us just restrict ourselves to microprocessors for the time being, I will come to microcontrollers and DSPs later. So, you can see that they have got certain bass width; that means how many bits of data it can handle at a time 32; 32 or 64? It can be in 2 forms, the speed number of the MIPS; how many million instructions that it can execute per second? Around 900, 1300, similarly power you can see Intel Pentium 3 consumes 97 watt whereas, IBM power PC is 5 watt, how many transistors are there? Those things are there, the price is more or less the same, so we can here is a just a snapshot of the different general purpose processors.

(Refer Slide Time: 18:04)



Next let us come to ASICs; the most common ASICs means since has a application specific instruction processes and the most common ASIC is a microcontroller.

Now, microcontroller; a microcontroller can be considered as an ASIC, what is there in the microcontroller? A microcontroller has got a microprocessor no doubt in it, but a microcontroller is essentially used for embedded applications; it has got some ways and means. So, what we do? It has got the facilities for reading sensor data, some actuation will be able to actuate data, it deals with mostly events, as the events take place outside as it is a reactive system data is present, but not in a huge amount, we have got microcontrollers in applications, one of the applications VCR is already becoming obsolete a, news item came out that last VCR has been produced so, but once VCR were very important applications video recorders and players.

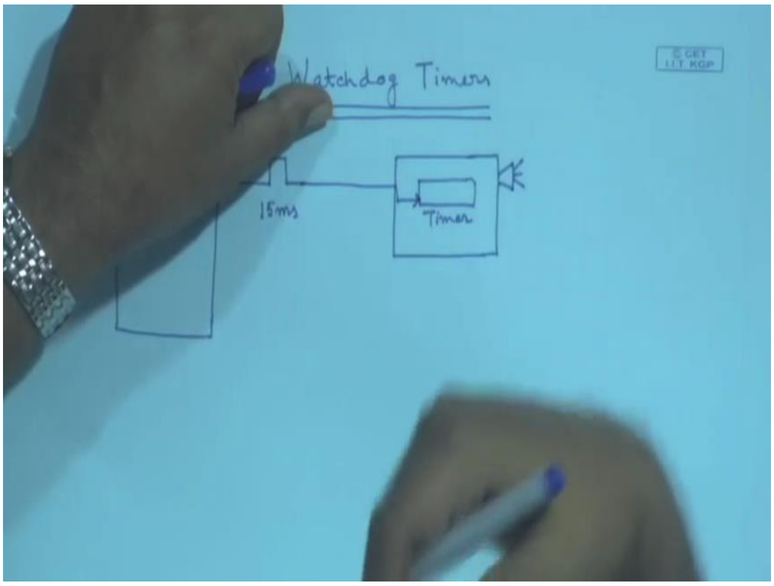
But now say for example, in disc drives in digital camera, even in microwave ovens, washing machine, and all those things have got microcontrollers, now how does the microcontroller differentiate itself with the microcontroller? Microprocessor, a microcontroller will have along with we have seen the diagram of a microprocessor here, but in addition to that for example, the instruction set of a microcontroller will be much less.

So, the control circuitry will be simpler, the data path will also consequently become simpler, but it has got in addition some devices which are very important for embedded system development like timers A to D converters communication ports and there is a register space of course, it also has got some on chip memory and they usually; we have got there some program memory and data memory as we will see in another variety of ASICs that sometimes, we will have; will deviate from the typical von Neumann architecture and we can have program memory and data memory separately.

And it allows you to have direct access to several internal pins, many of the chips pins, the programmer can directly access them and there is in the instruction set although I said the instruction set is simpler, but there are some specialized instructions for bit manipulation and other low level computations. So, these are something which are not there in a typical microprocessor we have to if I select a microprocessor. Then I have to select an AD converter, I will to have a separate timer and all those things, now in this case these are all built in. So, a timer all of you know; a timer can act as a timer or as the counter.

So, what does the timer do? A timer is usually set with a preset time and at each clock, it counts down and when it becomes 0, it gives a pulse, a counter on the other hand is reset to 0 and at every clock pulse it increases counts up and at any point of time, I can find out what is the time elapsed.

(Refer Slide Time: 24:02)

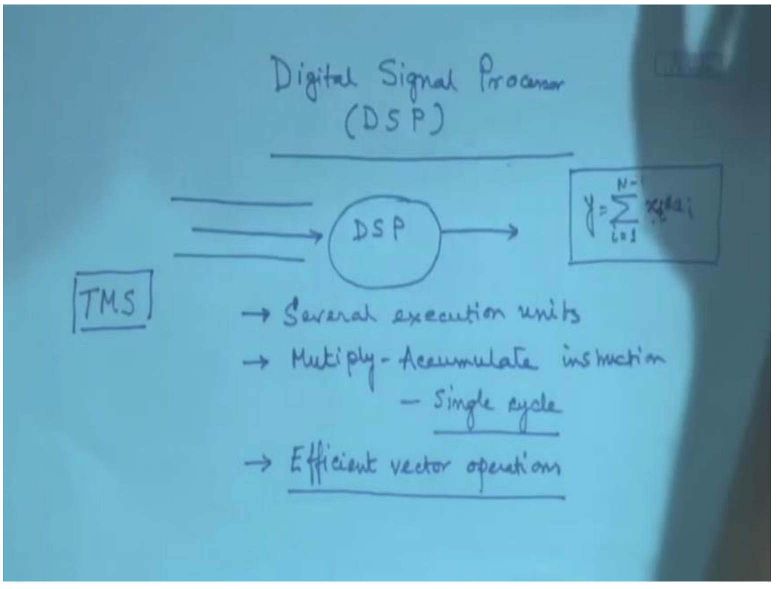


Now, one in the context of timer one very interesting embedded application requires often watchdog timers what are watchdog timers watchdog timers are as if a watchdog a timer is watching out for an event to occur say for example, there is a machine and I want to monitor the health of this machine some machine M and as a signature of its health, it will send a pulse maybe every 15 second, every 15 milliseconds a pulse comes at every milliseconds and there is a watchdog, I have just deployed a small system which will set an alarm, if there is an health hazard which will set an alarm if there is a health hazard otherwise it will not do anything.

How I will implement that there is a so this is a small system within which I will have a timer, have all of you got the problem that I want to ensure that this is healthy and the way of knowing whether it is healthy or not is every 15 millisecond I must get a pulse from here. So, what should this timer be programmed to this can be put this should be programmed and this timer should have 15 millisecond or say sixteen millisecond whatever you program and this pulse well reset or set the timer again to 15 millisecond.

Suppose it starts with 15 millisecond and it is counting down 14, 13, 12 like that if it comes to 0 and this pulse does not arrive then the alarm will be set otherwise if this is healthy then this will come and it will again rejuvenate this very simple application. All of us; we have got a pulse rate and so we can also monitor that in that way, but it is the question of so it is a watchdog timer, these are very important application of timers in embedded system where we watch out for an event to occur, we are expecting some event to occur and whether that event is occurring or not. So, timers AD converter communication ports, all these things are involved in microcontroller.

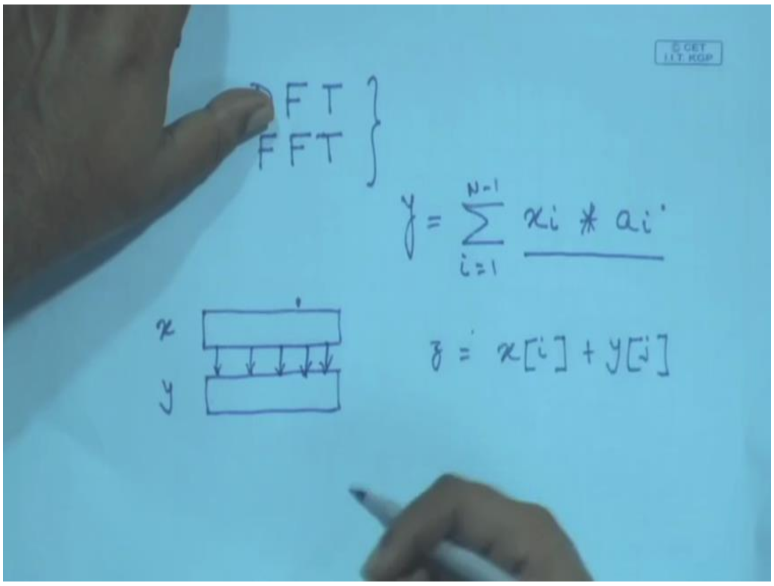
(Refer Slide Time: 27:42)



The other very important ASIC is a digital signal processor or DSPs. Now a DSP is used for as the name implies is used for signal processor, say often we have we need very fast processing, we have got streaming data, data is coming continually and we have got a signal processor which will have to process the data on the fly. For example, it can be a cell phone cell phones voice filter of a cell phone or a digital TV data is coming, I have to process that and display that music synthesizers other things. So, consequently a DSP needs fast execution.

So, unlike a typical standard microcontroller features got one processor 1, ALU one all those things here in a DSP, we often have multiple I mean different execution unit I would say several execution unit one very important instruction that is there in its instruction set is multiply accumulate instruction now one common operation in any DSP is a very common operation is among other things you will have something where you will compute something like this some data maybe x i and maybe some other things a i, say there is a multiplication here and we will go on doing this. So, multiply and accumulate multiply and add multiply and add it is not these is the multiplication x i multiplied by a i and that is being added continually.

(Refer Slide Time: 30:47)



Now, this sort of this type of operation is present in different DSP applications like say Discrete Fourier Transform, Fast Fourier Transform, these are very standard signal processing applications, all of these have got operations of this form or a j, it could be 2, it could be 2 summations etcetera.

Now, therefore, in a typical microprocessor, how do you implement this? We require a multiplication and that multiplication may require if I do it without any fast multiplier this multiplication will require more number of cycles there I add it again I multiply it add it multiply store add multiply store add or I will have to have very fast multiplier which will multiply and then adding there by I am saving some time, but in a DSP, there are multiply accumulate instructions which are single cycle, single instruction cycle specially designed and there are other instructions also besides a single cycle this thing.

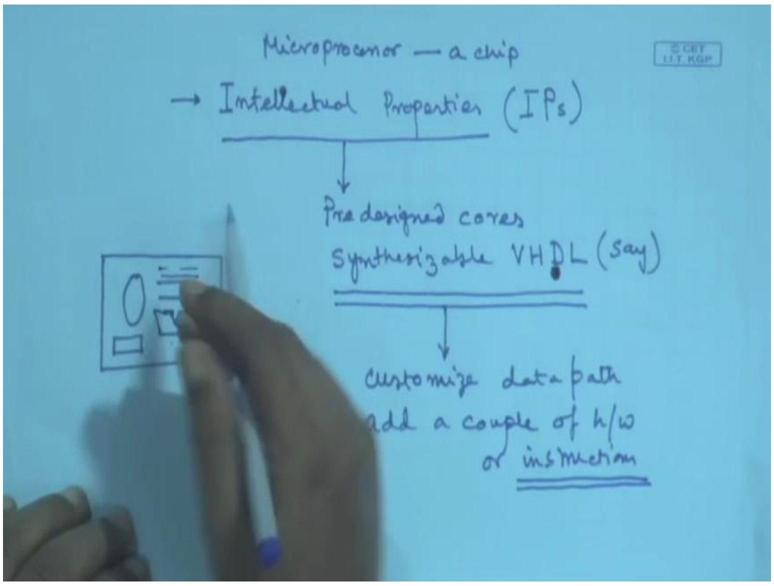
There are scopes of efficient vector operations like for example, efficient vector operators operations for example, it is often the case that the 2 arrays one is say x, another is y and a lot of operations can be say x i plus y j and we repeat it for say from i to a from one to n or something like that. So, this addition so, there will be special instructions where this can be added in a much faster manner parallel addition and those.

So, those sorts of operations will see the instruction set of some typical DSP processors. So, I would suggest that all of you try to look at some Texas instrument; TMS series. TMS series, DSP processors and look at their instruction set for microcontrollers you can see the typical say for microcontrollers you can start with simple one, I would suggest that all of you look at eight 0 five one is a microcontroller which is Intel's microcontroller look at ARM and strong ARM microcontroller.

And study their instruction set and compare it with the microprocessor you can take any microprocessor if you take the simplest possible microprocessor 80, 85 with all of you must have done you can see that and along with that you compare the instruction set of 8 0 5 1, what is added there? What are the extra instructions? What are not there? Try to make a list for ARM, you will find there are 2 different modes in which ARM processor can work one is a thumbnail mode; thumbnail mode and there is a normal mode. So, it will it has got a special thumb mode where you have got you can work it in 32 bits and you can make it much less power consuming. So, look at study these and also for DSP, you study some microprocessors of Texas instruments and you will see how the instruction sets vary that. So, that is an example of an ASIC.

Now, that trend is for more and more even more customized ASICs, we are trying for even more customized ASICs earlier we had microprocessor as chips.

(Refer Slide Time: 36:03)



Now, today we increasingly we are actually getting now, what we call earlier microprocessor was a chip and we use to buy microprocessor a chip now we are buying intellectual properties or IPs what is meant by this we know intellectual properties means you invent something and you apply for patent that is his intellectual property here although the name is saying, it means you are getting some predesigned cores for example, is a nothing, but some synthesizable I will just explain it VHDL say; VHDL say.

For example, I have designed a processor or a part of the processor and I have designed it I have tested it now that description I am offering to others as a synthesizable form of VHDL code, VHDL is VLSI hardware description language. So, once you have that program, you have got that program it is synthesizable. So, you can directly put it in your tool no not using your intellect. So, much of for the design because its already designed you just put it in your design tool and you are getting that processor or a part of the processor may be a fast multiplier all those things are coming up as and those are the core base design or that is why is a course pre design course available as synthesizable VHDL. Now this much is offered to you, now you can customize you take this you can say for example, customize the data path you can customize the data path of this you can add a couple of hardwires, hardwire or instructions, what do you mean by you can add a couple of instructions.

What is meant by adding a couple of instructions in this context, you are you have got something and that that has got some controller some data path with multiplexers and everything is there, some elements are there and there are some there is some multiplier, there is some adders. Now you want to add 1 instruction, you find that for my job, I need to add 1 instruction or 2 instructions that will make it even faster, now had it come as a chip, you could not have broken that chip and work it further, but here you are getting a VHDL code.

Therefore, you can add just as a software program, it is a function as if now you can add lines to the function so that new you can add to the data path which will be implementing your desired instruction and then synthesize it is it clear that is how that is a trend now of taking this IPs and further upgrading them for further design. So, this is the IP based design and this is a trend by which we can very fast go towards single purpose processors.

Next day we will be talking about single purpose processors in much more detail.