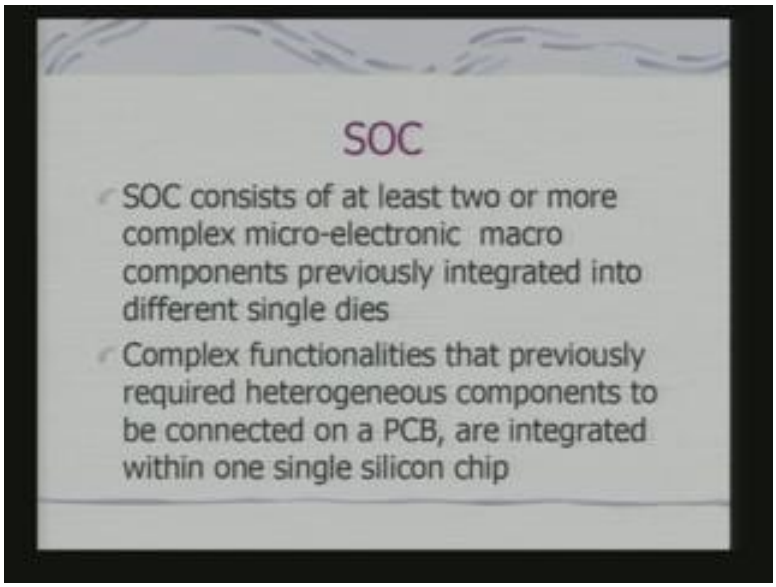
Embedded Systems
Dr. Santanu Chaudhury
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Lecture - 10
System on Chip (SOC)

In the last class, we had discussed digital signal processors. Today, we shall see how may be a digital signal processor, as well as general purpose processor can be combined. And put into a single silicon real estate, forming what we call System on Chips.

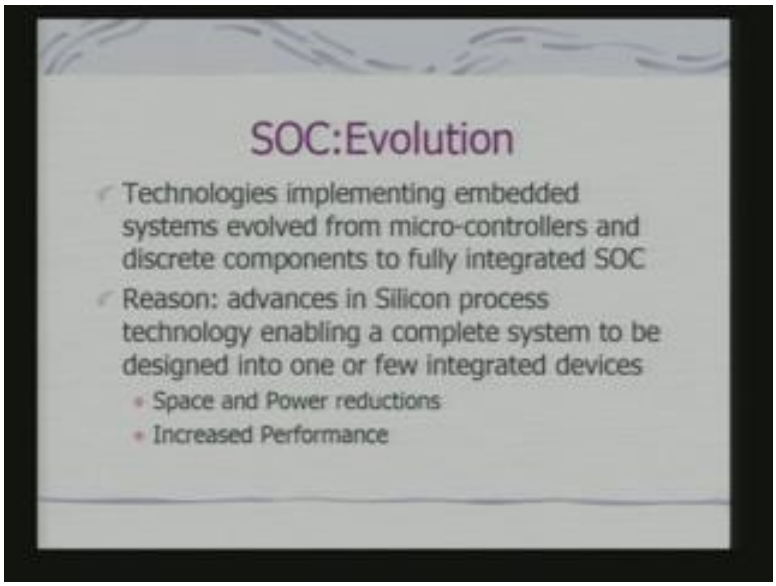
(Refer Slide Time: 01:25)



SOC are system on chip, consists of at least two or more complex. Microelectronic, macro components, which are being previously integrated into different single dies, in individual form. Complex functionalities, that previously required using heterogeneous components; and designing a PCB. And interconnecting them on a PCB. And now being integrated, within one single silicon chip.

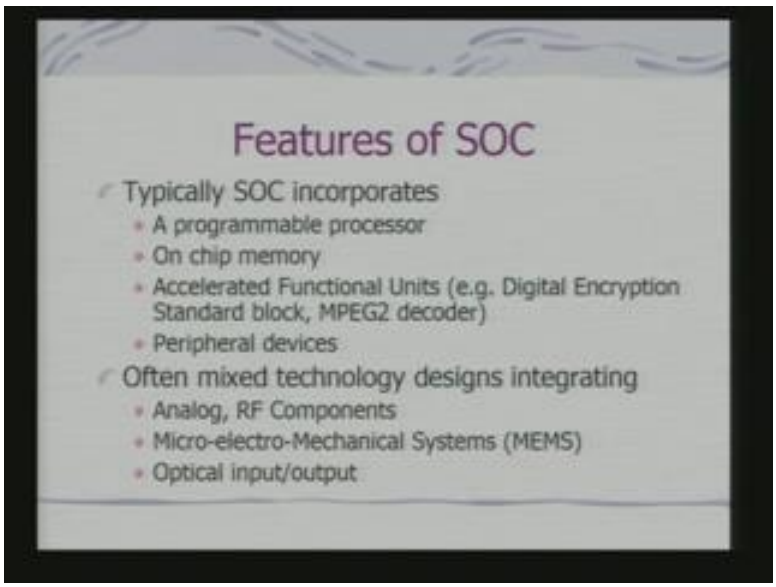
So, having multiple complex entities, hardware entities coming onto a single silicon real estate is what we shall refer to us silicon on-chip, or system on-chip not silicon on-chip, system on-chip.

(Refer Slide Time: 02:25)



The technologies implementing embedded systems, evolved from micro-controllers. And discrete components, to now become fully integrated SOC, why? Obviously, advancement in the silicon process technology, which enables us to have more number of heads. And effectively a complete system to be designed into one or few, integrated devices these results in space and power reductions, as well as better performance.

(Refer Slide Time: 03:03)

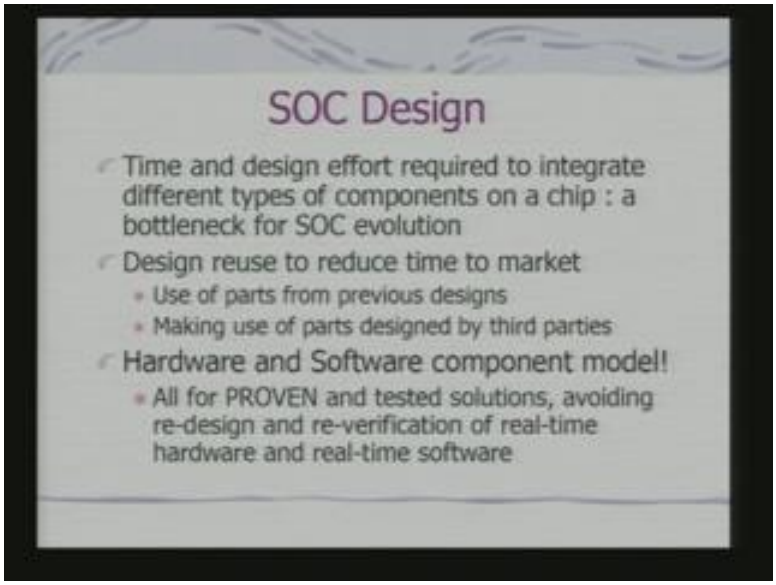


What does the features of this SOC? Typically, our SOC incorporates a programmable processor on chip memory. Some additional functional units, which are more application

specific like digital encryption, standard block or MPEG decoder. This we call accelerator functional units, on top of that there could be peripheral devices. But, this is a very typical SOC configurations, along with it we may expect other components as well.

And often what we get today, is called mixed technology designs; where on-chip itself you may have analog component. Maybe RF component, for SOC designed for communication applications. You may have microelectronic mechanical systems, because when you are using an emended system. And it has to actuate some action, you may have micro-electro-mechanical systems, on the silicon real estate itself. As well as you may have optical input or output. So, variety of components, can now get integrated and each component in itself is a complex subsystem.

(Refer Slide Time: 04:40)

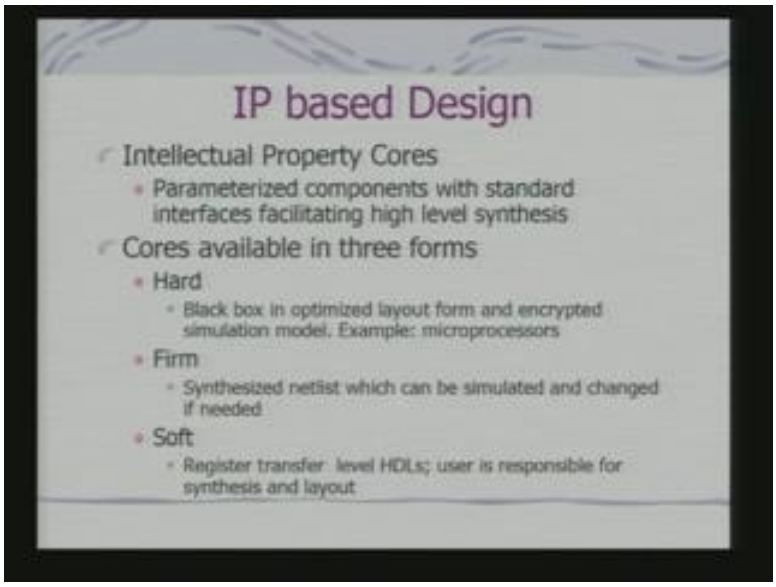


So, you can realize that, the effort required to design, such a kind of an integrate system could be pretty high. Time taken for designing such system, is also expected to be high so, how to overcome this bottleneck or difficulty. We use a principle, what is called design reuse. Make use of parts that have been previously designed, as well as tested. And also making use of parts, that is one particular manufacturer, when decide to make use of parts, designed by somebody else.

So, effectively what we have moved into is the hardware, as well as software component module classical object oriented programming in software, propagates this software component model. Now, we are talking about both hardware, as well as software

component model and what does it show, that in shows proven and tested solutions, and avoiding re-design and re-verification of real-time hardware and software components, which are to be used for a particular application. This is known as intellectual property base design.

(Refer Slide Time: 06:07)



And what you get is intellectual property cores, as the basic components of such a design. In fact, when we are talking about ARM processor; and we refer to as ARM core. So, ARM core becomes an intellectual property core. And what is essentially in that context, that interface of that reusable component has to be known, and known to whom, known to an application developer, who is designing the system on-chip. Now, this course are available in a variety of form for this purpose.

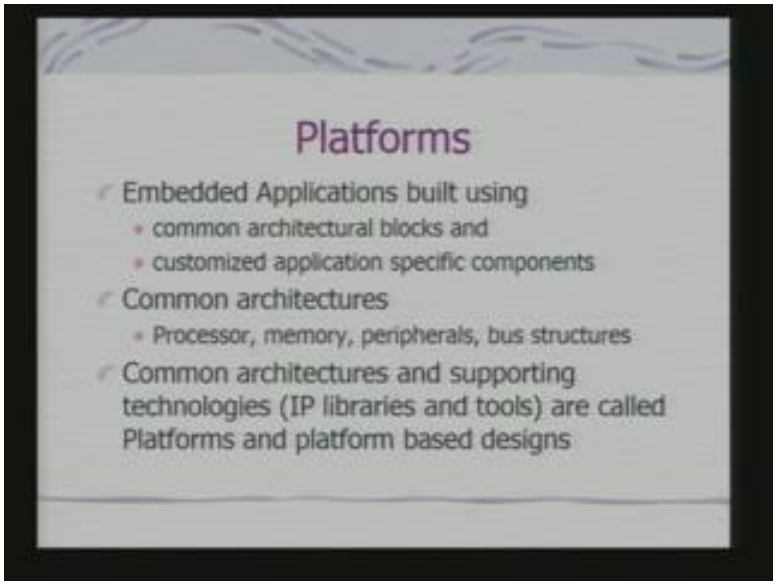
We call hard core, firm core and soft core. In a hard core, the designer of processor provide the design itself, in a black box form. In a black box form, in an optimize layout. That means, they provide in a form which can actually go on to the silicon itself, with the layout of the components being specified. They also provide an encrypted simulation module. So that, that component can be used with other components in an simulation, you actually manufacture them.

The other variant of the core is call firm core, where you really do not provide the design at the layout level but, provide at the net list out. That means, at the level of connections between the constituent would block maybe at the gate level connections. So, what is the

advantage that, it can be simulated definitely. Because, you have got a net list connection and specification have the component level. It can be also reuse in size with necessary changes.

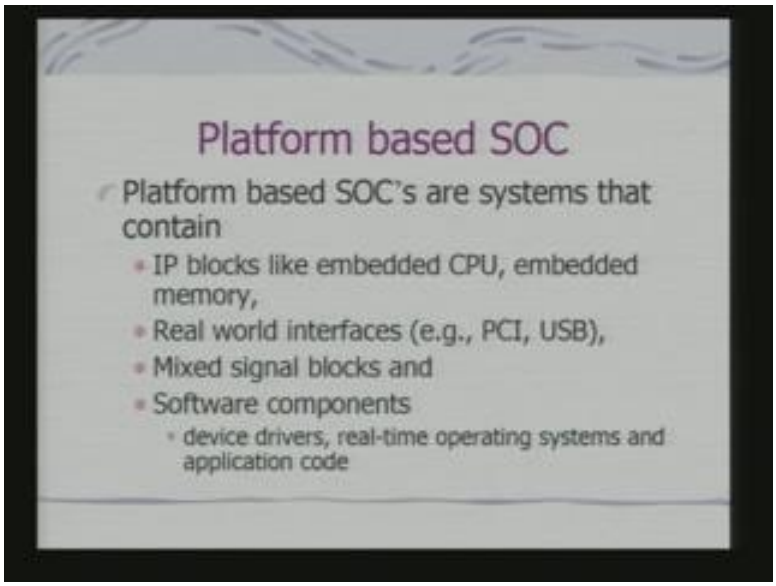
So, the visibility of the design is more, compare to that of hard core. The other one is soft core, where you actually specify the design, in terms of register transfer logic. That typical hardware description language is used for describing the design. So, here the user is responsible for the complete synthesis layout, technology mapping and final production of the chip. So, this becomes a software. So, software cores are therefore, can be changed to the maximum extend as per the users requirement. Now, this gives the concept of what are called, platforms.

(Refer Slide Time: 09:04)



Embedded applications are built using common architectural blocks; and customized application specific components. And common architectural blocks, are basically processors memory, peripherals, bus structures. Now, this common architectures and this supporting technologies, which are intellectual property libraries. And tools to use this common architectures are called platforms. And platform based designs, because using this platforms other systems can be designed. So, platforms are not just the hardware.

(Refer Slide Time: 09:51)

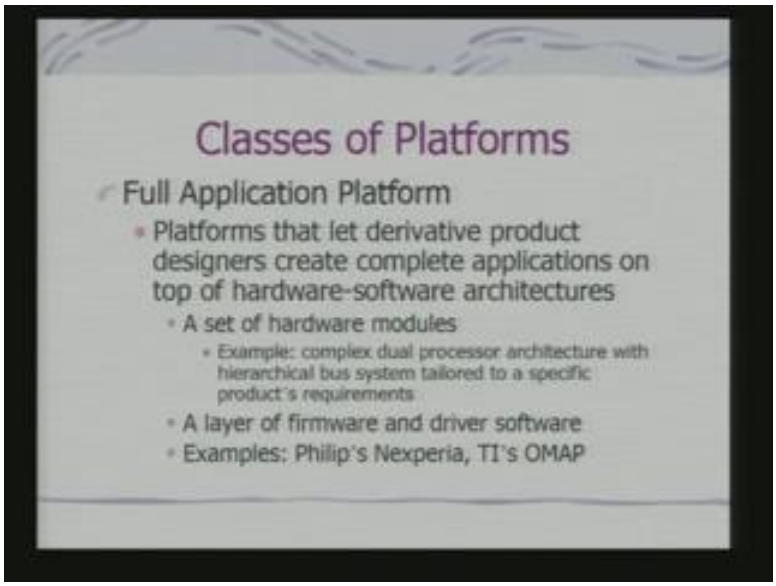


So, what we are talking about, a platform consist of IP blocks. Like embedded CPU, embedded memory. And when it comes in a form of a hardware, you may not know the details of it. But, you can also use the layers and the features provided on the platform, for developing the application. So, a platform can come with an embedded CPU, embedded memory, real word interfaces, mixed signal blocks and software components the tools to use all this elements, which has been put into the platform.

The software component can be device drivers, real-time operating systems and application code. So, you should clearly distinguish between two aspects. One aspects is, I can take a core and intellectual property core in some form. Either hard form or in a soft form, add-on my design and generate SOC for my application. Now, this basic components also can be packaged, in terms of hardware and software elements. And can be delivered to a customer, for developing it is application.

Here, is not doing really the silicon compilation of the complete system. Is using the components provided, but the manufacturer, which includes both hardware, as well as software elements and then, designing the system. And this is what is platform and platform based SOC.

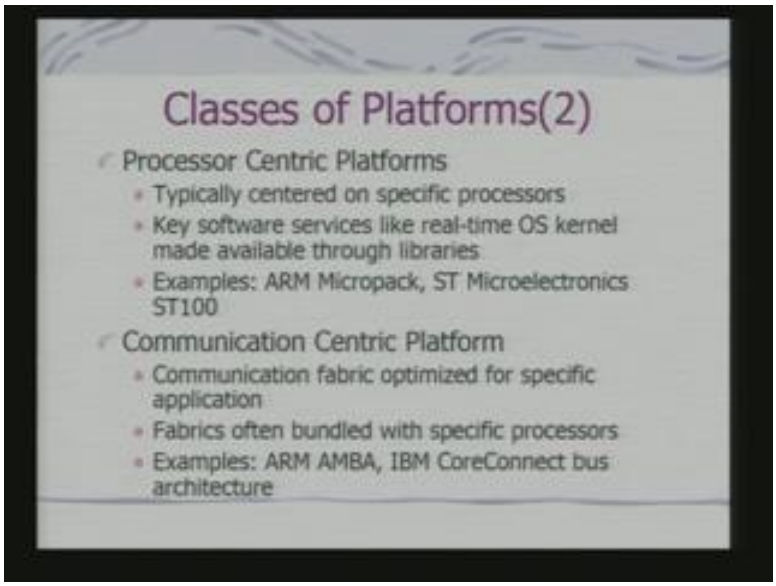
(Refer Slide Time: 11:41)



There are different classes of platforms full application platform. Platforms that let derivate product designers create complete applications on top of hardware-software architectures, that I was talking about. So, it can come with the set of hardware modules. It maybe a complex dual processor architecture, with some kind of an hierarchical bus system tailored to a specific product's requirements. And use that hardware, a layer or firmware and driver software as provided.

There are various examples, Philip's Nexperia, TI's OMAP, they are examples of what we call, full application platform. So, here the application development effectively means, developing the software and maybe some add-on hardware, on top of the platform that has been provided in there are processors centric platforms.

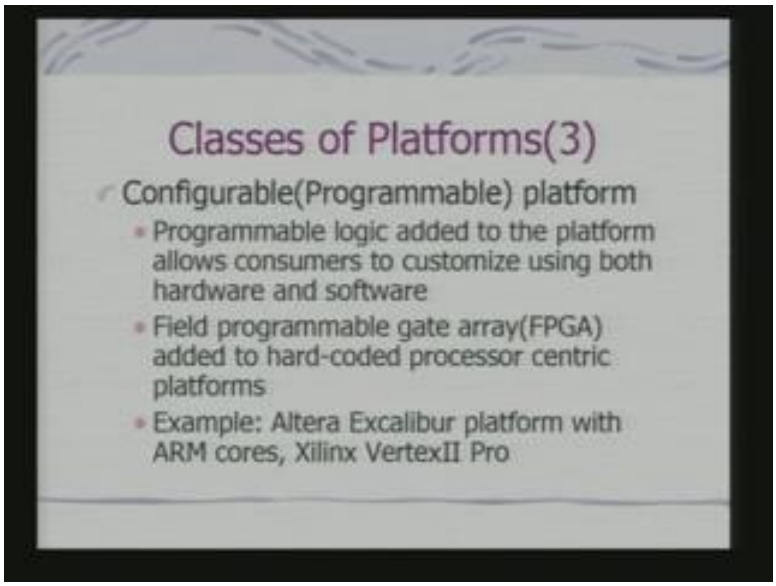
(Refer Slide Time: 12:43)



They are typically centered on specific processors. And they provide along with the processors, the key software services the real-time OS kernel, which could be supported on that platform. So, examples are what we call ARM micropack, ST microelectronics, provides ST 100. These are all examples of processors centric platforms. There are at the same time communication centric platform; where the basic objective of the platform to provide the communication fabric, optimized for specific applications.

We are already seen ARM's AMBA bus. So, ARM's AMBA bus, as well as maybe a norm processor. Integrated onto a single system on-chip, can provide an example of communication centric platform. So that, you can exploit that bus, to add-on additional components, have communication between them depending on the application that you are looking at.

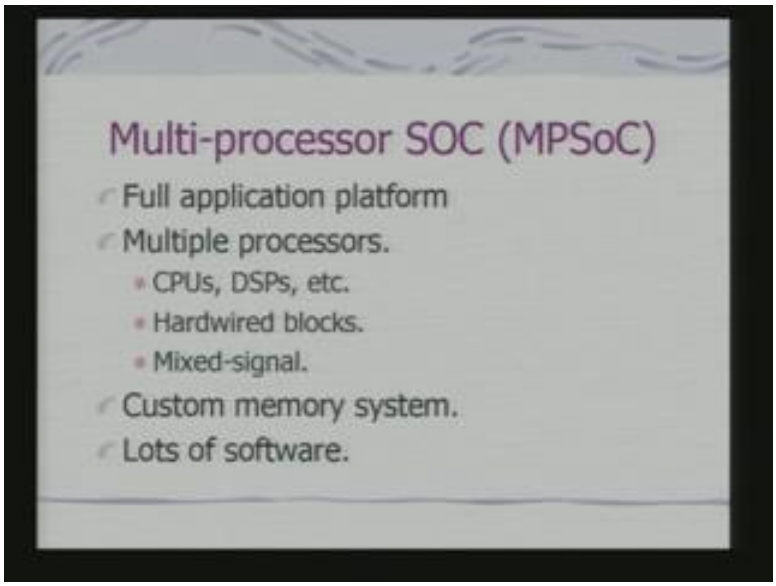
(Refer Slide Time: 13:50)



Then we can have configurable platforms. Here, the programmable logic is added to the platform, which allows consumers to customize using both hardware and software. So, you have got field programmable gate array, added to hard-coded processor centric platforms. So, when you talked about processor centric platform, it was primarily the processor. And some additional components, with a software services bundled with that processors centric platform.

Now, here we have got additional field programmable gate array logic or any of the form of configurable logic. So, you can customize, in terms of software as well as hardware. There are examples, Altera Excalibur platform with ARM cores. That means, we ARM core, it provides additional FPGA blocks. Xilinx VertexII Pro, which is built around power PC. So, these are all examples of configurable platforms. And these links are to concept of what is call, configurable computing or reconfigurable computing. We shall come to that point towards end of this class.

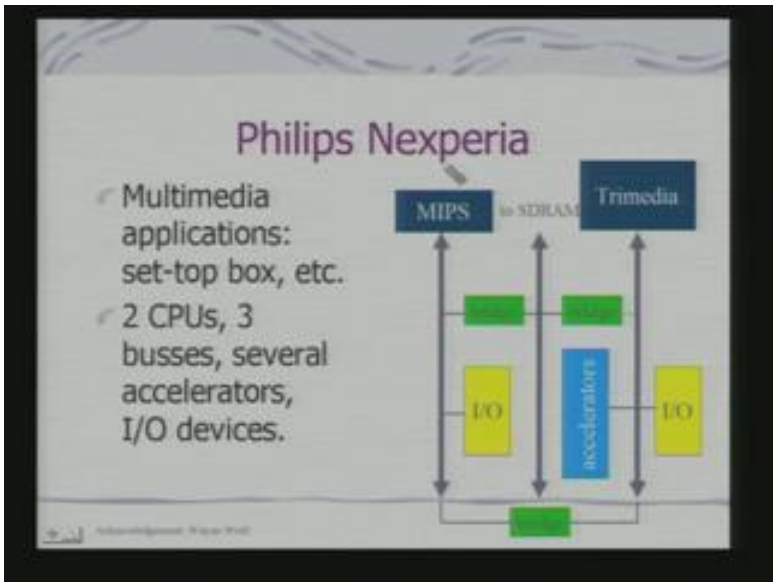
(Refer Slide Time: 15:13)



This multiprocessors SOC or MPSoC, they are typically full application platforms. In fact, other way round majority of your full application platforms are nothing but, multiprocessor SOC. So, they have got a CPU and a DSP. These are many minimum and common requirement. Some of the special hardware blocks, the hardware blocks designed from special purpose functions. And there will be mixed signal elements, that means you are for analog interfacing.

So, you have got analog, as well as digital interfacing elements customized memory system and lots of software on top of this hardware. So that, application developments gets facilitated. So, examples I had already started with this refer to Philips Nexperia.

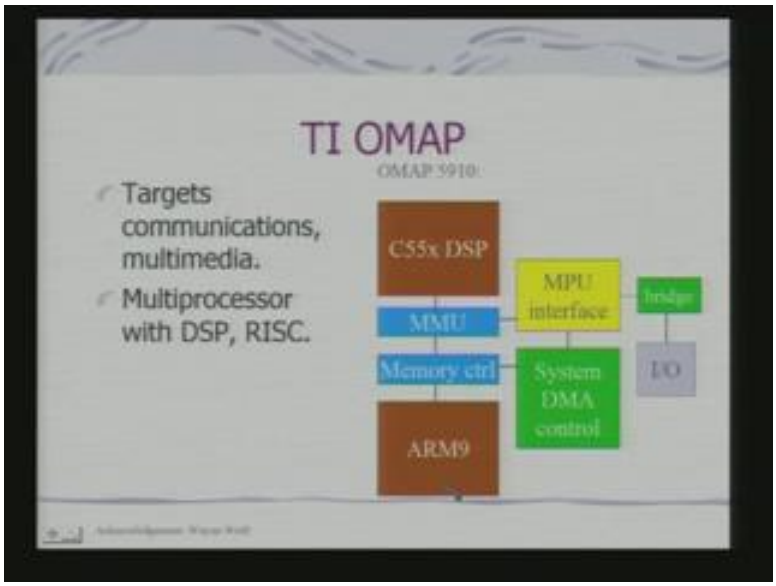
(Refer Slide Time: 16:07)



So, here we have talked about Trimedia CPU Philips and MIPS processor, which is an example of a RISC processor. These two code set together and you have got other accelerators, which are meant for accelerating some special functions. And these are the individual buses and buses to SDRAM and the bridge, between the buses of the individual processors. So, this is a kind of a multiple processor system.

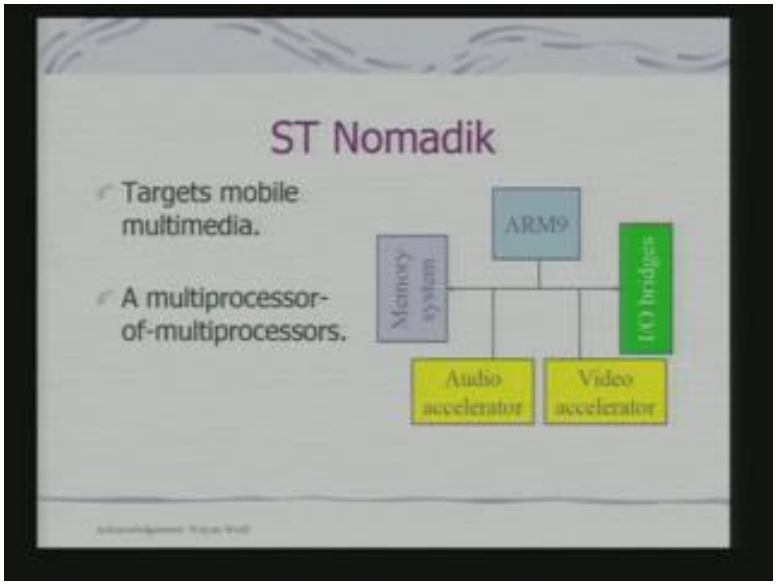
And since Trimedia, was primarily designed for media processing, like video, an image processing tasks this SOC, that is application platform was designed by Philips particularly for applications like set-top boxes.

(Refer Slide Time: 17:00)



TI OMAP has got C55x DSP, we have studied that already as well as ARM9, these two. And there are variants, this is we are looking at one particular OMAP. And they have been used together for multimedia, as well as... In fact, handled portable communication terminal applications.

(Refer Slide Time: 17:26)

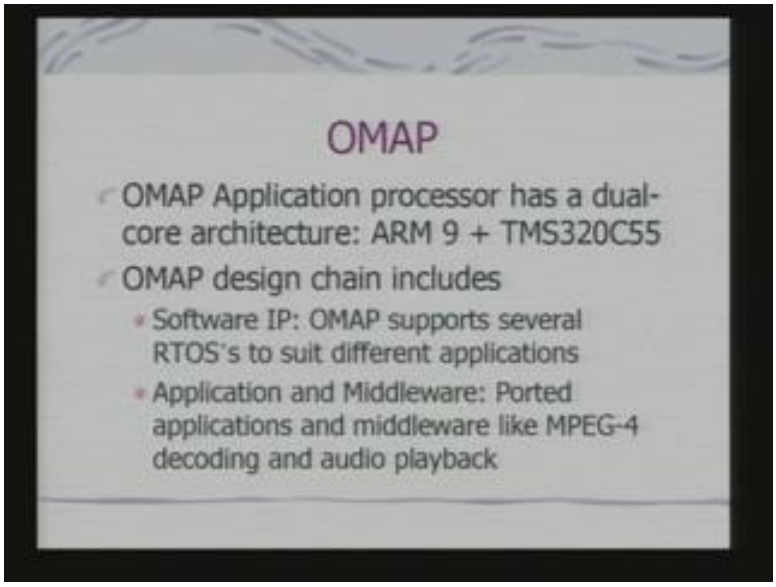


This is an example from ST. Now, this is very interesting, because it just not a target mobile multimedia but, it is not just two processor, it is a multiprocessors of multiprocessors. So, you can see the complexity to which the design can go up, when an

SOC is being designed for applications. So, here the audio accelerator and video accelerators, are collection of heterogeneous multiprocessors. And they are actually used along with the ARM9 core to provide your platform.

So, please keep in mind this platforms are… Platform is what this kind of hardware blocks, along with the software layer to use this blocks efficiently. So, now we shall look at one of them OMAP, Open Multimedia Applications Platform from TI, in more detail to understand the features of applications platform.

(Refer Slide Time: 18:28)

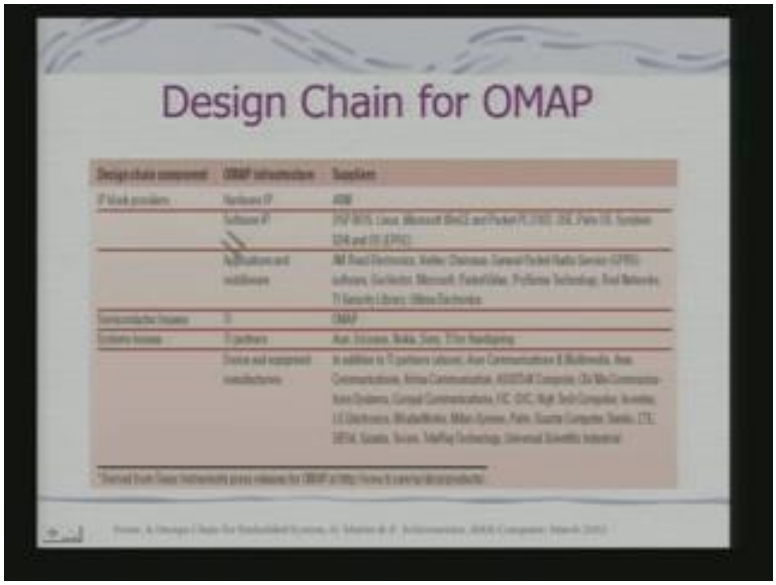


OMAP application processor, we have already seen has a dual core architecture. ARM9, as well as C55 DSP. And it is design chain includes software IP, there is OMAP supports several real-time OS's, to suit different applications. So, you can use that OS's to program, these SOC for application development, also application and middleware. There are ported applications and middleware, like MPEG-4 decoding audio playback. MPEG-4 is a video decoding, a video standard.

And it is being used for video decoding standards and audio playbacks. So, you have got this applications, these are ported onto it and these application themselves can be use subsequently. So, when you are talking about a platform, therefore hardware, the basic OS. And these OS is built with a kind of a software architecture, ((Refer Time: 19:34)) for these associates. Because, you got to ((Refer Time: 19:37)) of software architecture, for this associates.

Simply because, you are using two processors and how they use this two processors, that becomes a critical issue. And when there are more processors, the problem complexity increases. So, you need proper software tools, to use this processor for your application development. So, always this platforms, application oriented platforms come with hardware, as well as the software bundled together.

(Refer Slide Time: 20:10)

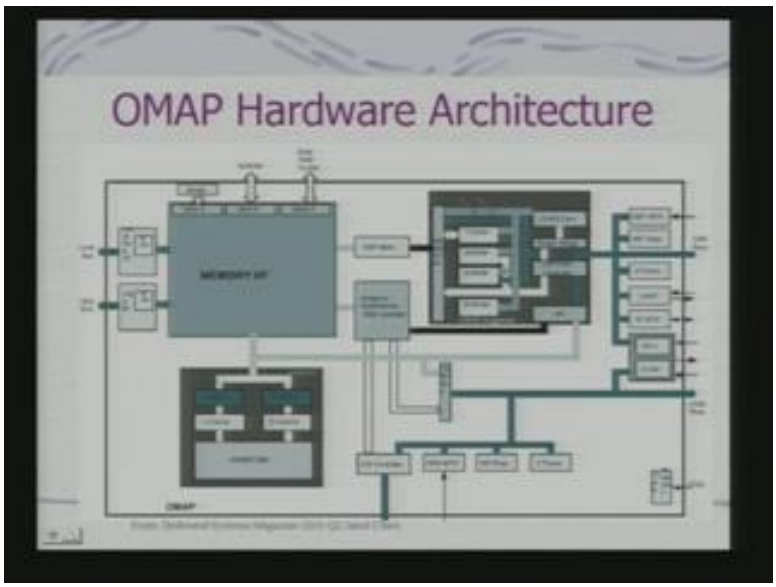


In fact, this is an example, and showing from for the TI OMAP. And you can see that these has, this is a process by which the features have been integrated for multiple vendors. So, you have got the hardware IP, which is coming from ARM and your software IP, which comes from a number of this OS vendors. You have got Linux, Microsoft Linux is a public domain, which has been mapped on to it. You have got Microsoft, you have got palm OS, Symbian.

Symbian is the OS targeted for cell applications. So, all these source of software, IP intellectual property from OMAP. There has been an number of contributors for application and the middleware and TI has actually contributed, the other hardware IP the DSP, and does a manufacturing. And it has been taken up, where number of device equipment manufacturers. So, these are some set, which is being shown and you will find many of this consumer electronic partners. As well as cell phone manufactures, have picked up OMAP for developments.

So, the whole idea here is, in the context of embedded systems, you are not looking at hardware or software in isolation because, it is an integrated whole and you are using components, supplied by different vendors. And providing an environment to an application developer, to develop his customized application on top of this platforms. Let us look at OMAP hardware architecture.

(Refer Slide Time: 21:49)



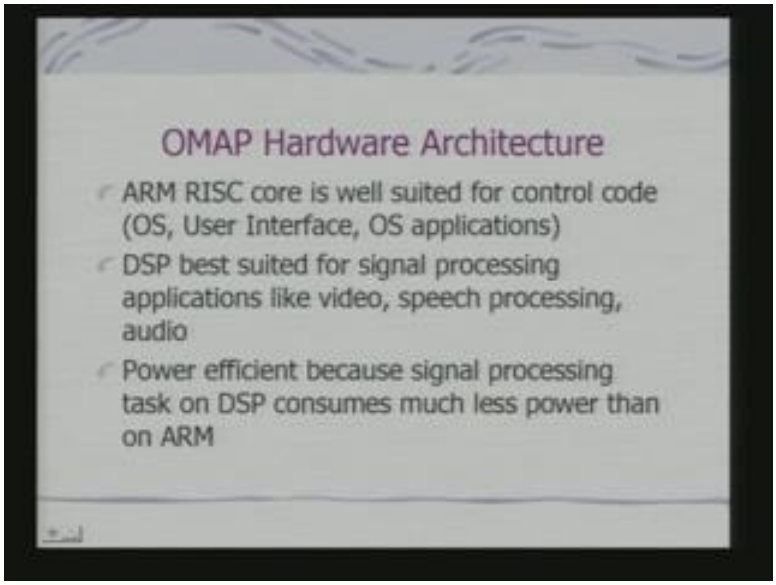
It consists of two basic components, this is your DSP component. And this is your odd component. This is the memory interface, you can see that the same memory interface is provided to both the processors. And each of this processors have got their O1 on board RAM. Because, I already talked about this, single access RAM, dual access RAMs, instruction cache, which is already there within the DSP core. In case of ARM core, in fact, ARM9 is a hardware architecture based processor.

So, you have got an instruction cache and data cache, which is separately. And then, it has got it is own MMU's, memory management units. So, through this their using this processor is using the memory interface, to access external memory. This is also using this DSP MMU, through this memory interface to access external memory. You have got a DMA controller. And DMA controllers, provides the direct access. So, the other element, which provides direct access to the memory is via DMA controller.

It has got a number of this interfaces, interfaces for peripheral devices it has timers and watchdog timer. Watchdog timer is important for embedded applications, that there as

well. So, this is what we are found, that there are two processor cores. They have their own memory management scheme. They have there, on what is typically call private memory. And they also have a mechanism to access global memory. They have got a set of peripherals, by which they can interface with external devices. They can individual interface, they can also connect to external devices via sheared interfaces.

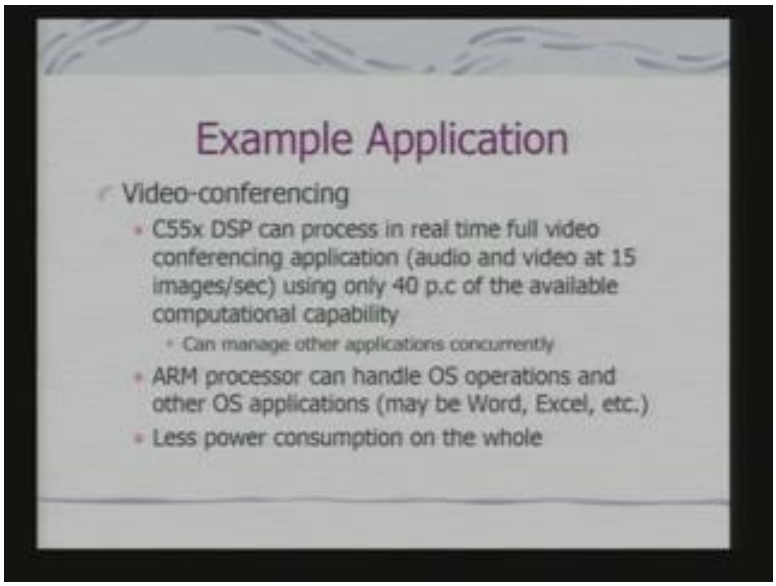
(Refer Slide Time: 24:07)



So, how does these two processors, really coordinate. ARM RISC core is well suited for what is called, control code. OS, user interface, as well as OS applications. And DSP is best suited for signal processing applications, like videos, speech processing and audio. But, obviously you should not that ARM also has got various instructions, which facilitate signal processing as well. But, what is being emphases here is the management of the control code, which is expected to be the primary task for ARM.

And what is being found also through benchmark, to instead is, that this combination. That is a mapping signal processing tasks on to DSP. And OS on to ARM is power efficient, because signal processing task on DSP consumes, much less power than on ARM. And in fact, if required you can use ARM signal processing capability.

(Refer Slide Time: 25:19)



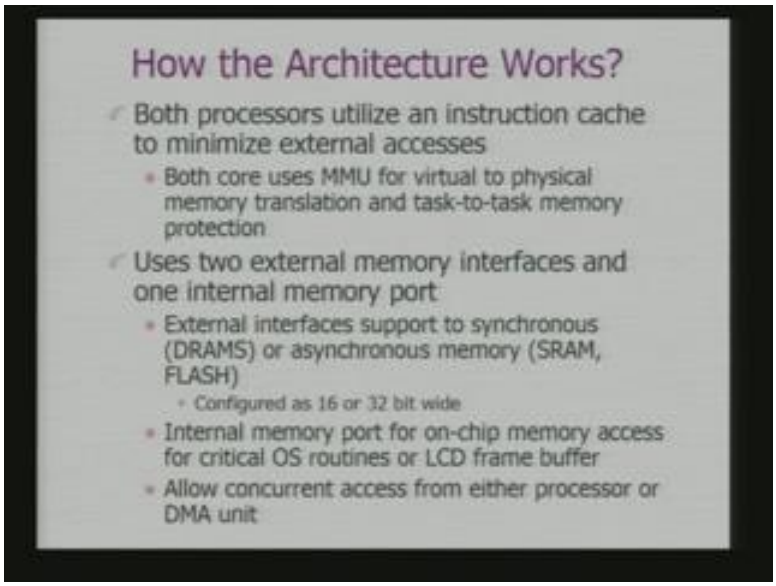
So, let us look at the example application scenario, video conferencing. So, in a video conferencing, you expect to send image, as well as audio and that too at a particular frame rate typically, expectation is 15 frames per second. So, you really do not have a jar key vision and you have, your faces images at regular intervals. So, these imposes are real-time constraint on the processing.

So, S55 the DSP, can process in real-time full video conferencing application using only 40 percent of the available computational capability. And it has been also found, through benchmarking studies. And therefore, it has got about 60 percent of the computational capability available, for managing other services concurrently. Now, ARM can handle OS operations, while video conferencing on, as well as OS applications.

If I am having windows C on it, it may support word excel etcetera. So, what a user can do, user can have a video conferencing, as well as can use OS applications simultaneously. Because, of availability of two processor cores on SOC. And you can consider this as a, typical application could be like a palm top or a communicator with the camera. So, that with or using the camera, you can have a video conferencing, or video telephonic going on. And at the same time, you can use the OS space services.

And less power consumption is on the whole. Because, the DSP power consumption is much less, for this signal processing tasks.

(Refer Slide Time: 27:16)



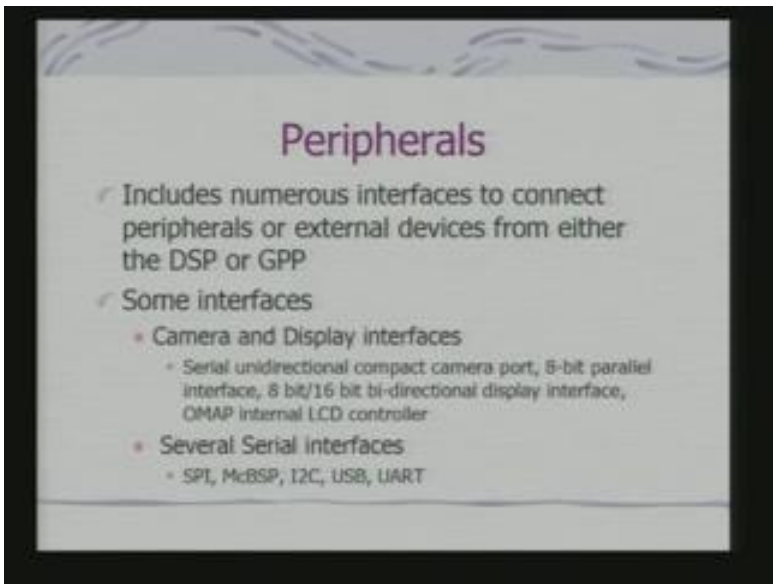
Now, how this architecture really works. Both processors, utilize an instruction cache to minimize external accesses. That is what we have already seen, in the architecture diagram. And they both use MMU, memory management unit, for virtual to physical memory translation and task-to-task memory production. Because, there tasks these processors will have distinct task. And there task need to be protected against violations of access.

And what is important is, since each of the news MMU, they minimized or optimized and external memory accesses. And external memory accesses, are typically power hand ((Refer Time: 28:02)). So, if I minimize those accesses, then I make the whole system as well power efficient. Now, how does it access external memory, it uses two external memory interfaces and one internal memory port. External interfaces support synchronous or asynchronous memory.

Synchronous memory, like say dynamic RAM's. And asynchronous memories, which could be flash or static RAM. And actually the program will be resident on static RAM or Flash. And there is an internal memory port, for on-chip memory access. Your critical OS routines or expected to be loaded on the chip itself. As well as the LCD frame buffer. In fact, if it is targeted for your mobile applications, there would be a requirement of an display panel.

And in the display panel, the data which is to be display panel or to be stored in a frame buffer, which is again in a memory area, where you have a mapping from each pixel on the display to a memory location in the frame buffer. And that would be very frequently accessed. Because, they are you actually have updates to be displayed to the user. So, that also forms part of your internal memory. This internal memory allows concurrent access, from either processor or DMA unit. So, from peripherals, you can have fast transfer of the data on to the memory.

(Refer Slide Time: 29:46)



A number of peripherals can be interfaced, a number of interfaces are supported. We shall look at some of this interfaces because, they are interesting, in terms of the target applications. It provides camera and display interfaces, because if I am targeting for this kind of embedded portable terminals. Then, they will typically have a camera connected to it. And so that camera interface, forms a part or an integral component of this SOC.

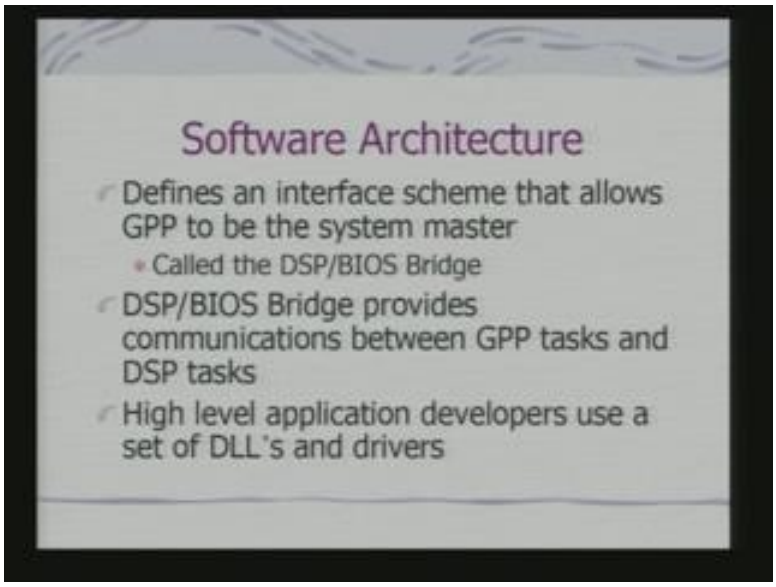
It has got basically two camera interfaces. A serial unidirectional, what are call compact camera port, as well as 8-bit parallel interface. Also it is supports bidirectional display interfaces, and has on-chip LCD controller. So, that the LCD display can be managed. Now, you can realize that additional components, have been put on to the chip. By the designer, that is TI keeping in mind the target application.

Also there are several serial interfaces, SPI, McBSP, 12C, USB, UART. Now, these interfaces, in fact MAC DSP, we have already seen. And through this various devices

can be now connected to OMAP. In fact, if you are thinking, in terms of using your OMAP processor, this one. As a core processor in a cell phone for data and voice communication, you will require a modem.

So, modem can be interfaced, using this serial interfaces. Even you can have an additional coded, like say an audio coded. And which can be connected via serial interface. There are other interfaces as well, but we are not going into them in details.

(Refer Slide Time: 31:53)

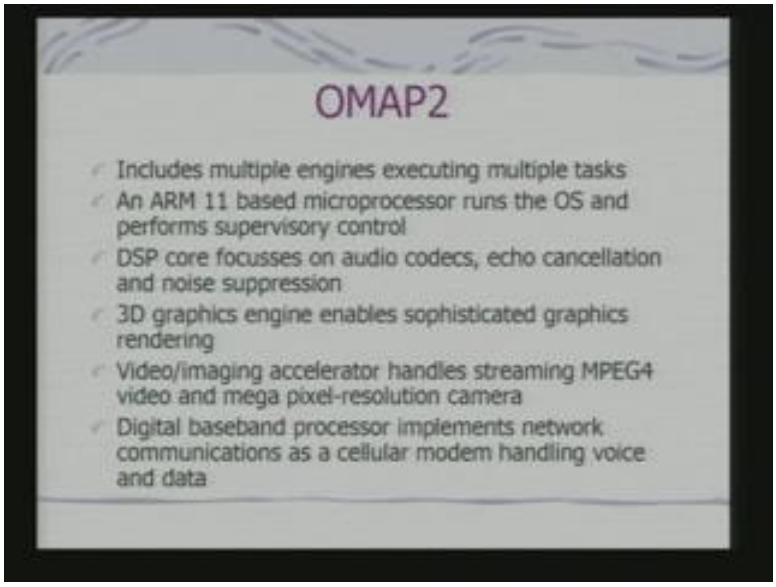


So, what is your software architecture this becomes a key thing, because somebody, some manufacturer can design a complex hardware SOC. But, if the proper software interface or the software architecture is not defined. Then, using this SOC is becomes extremely difficult. And that is precisely the reason, the point I was making all through that platforms essentially mean, packaging of both hardware and software together.

So, software architecture for OMAP, defines an interface scheme. That allows the general purpose processor to be the system manager. And these interface scheme is called DSP BIOS bridge. And these DSP BIOS bridge provides communication between GPP tasks and DSP tasks. And these can be used in a way, that you adjust using a unit-processor system. So, the basic objective is of using this DSP BIOS is, that externally for an application program.

Or it would appear, that he is just programming on to a single CPU. And internally the DSP BIOS is taking care of mapping sub-tasks to the DSP's. So, high level application developers use a set of dynamic link load libraries, and drivers. And have the field that there using a single processor and enforcement of this architecture, why I am showing this, is because you can see the complexity. That can be added today, onto a single chip with the advancement of process technology. The idea one, we talked about primarily as a processor core DSP and ARM.

(Refer Slide Time: 33:54)



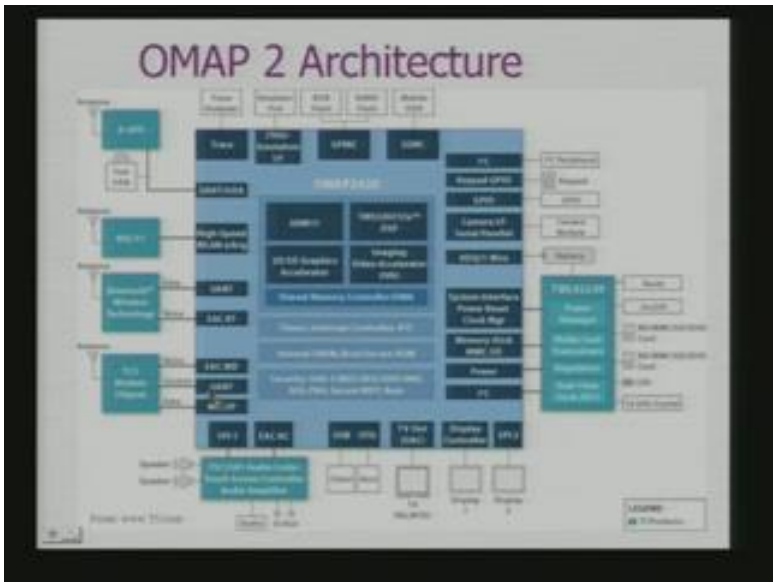
Now, we have got multiple engines, executing multiple tasks integrated on the same chip, that is what is OMAP 2. Here, you have ARM11 based microprocessor, which is advanced version from ARM9. DSP core focuses on audio codes, echo cancellation and noise suppression. These are primarily important for caring out telephonic conversation. There is on top of that, a 3D graphic engine which is again basically a processor to enables sophisticated graphics rendering.

Also there is a video imaging accelerator, that handles streaming MPEG-4 video; and mega pixel resolution camera. So, this video handling and the camera handling is no longer with the DSP, DSP concentrating completely on the audio task. Also there would be a digital based time processor, which implements network communication, as a cellular modem handling voice and data.

In fact, this is primarily targeted for cellular phones. And what today people are talking about this smart phone. That is a phone which has got all elements of personal intermittent as well. So, in fact you can understand the 3D graphics engine, would enable highly sophisticated MMS service. That is the display service on the phone. Video imaging and accelerator would enable you to use the camera. And the digital based on processor, enables you to use the phone, as a voice and data modem.

And why DSP core focuses on audio, because you can consider, you are even cell phone as an MP3 player. So, you can have MP3 data which can be played. So, that whole task goes to the DSP core.

(Refer Slide Time: 35:58)

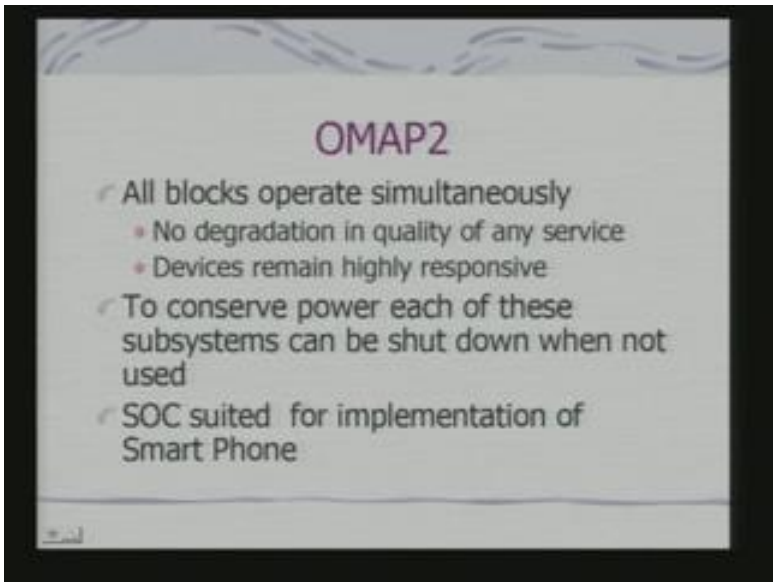


So, the architecture now becomes much more complex. In fact, you have got the ARM11, this is a core processor. If you are talking about multiprocessor, this becomes a core processor. You have got ARM11, you have got C55x, you have got 2D, 3D graphics accelerator, you have got video accelerator. So, these four processing elements are have been integrated together, along with various other features.

In fact, there is a keypad GPI, general purpose, IO to interface a key pad. A cameras serial panel interface, that same parallel real interface that, I have talked about earlier. It has got again a security block, that is if I want to use some kind of an encryption service. So, for that hardware has also been provided and we have got the URT and this here, we are showing a modem chipset.

So, this modem chipset can be used, through this ports. I was talking about the serial ports, pager modem chips it can be used. So, these are the serial interfaces, which are provided for chipset. And this other, these interfaces provides for the actual playing of the sound. See you can find this to be a pretty complex and elaborate processors. So, in fact what has been done is all the functionalities, which are required as been built into a single chip.

(Refer Slide Time: 37:34)



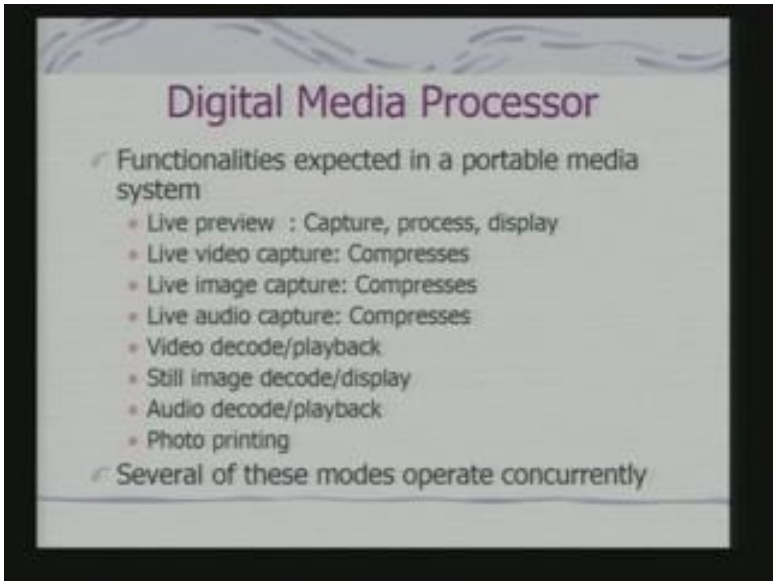
So, all blocks here operate simultaneously. So, there is obviously, no degradation in quality of any service. Why the degradation would come in, if that service is being sheared on a single processor. So, depending on the load, there would be time allotment on this different services can vary. A simple example is, if the processor is loaded. Then, they are maybe situation, where are decoder or video decoder, when meets a deadline. That means, a frame may not be decoded at a 40 millisecond interval and you may get a some jump well the video is being displayed.

But, since all these are concurrent elements, there is no degradation in the quality of service. And devices, which are connected to this elements, remains highly responsible. It is not that you have to work till, this device generates an interrupt and the processor services the interrupt coming from the device. For that to conserve power, each of this subsystems can be shutdown, when not used.

Because, all this subsystems may not be really used at the same point in time and this SOC is, is suited for implementation of what are called smart phones. And it is being used by a number of vendors, like Nokia and others. And even TI says, that they would not supply, this chip it is not being used in a large value. In fact, it is being a kind of an exclusive agreements with the manufactures of the devices, with the actual semiconductor form.

So, this is a kind of a platform, which is targeted first mark phones, where you can have communications, telephonic communications. You can even have video phone, you can have FM radio, when implemented over the, you can have an MP3 player. You can have DVD interface built into it; and also obviously, you can have a camera and multimedia pixel camera to take snaps. So, it is a kind of a personal entertainment station, which can be built around entertainment. As well as communication station, which can be built around this SOC.

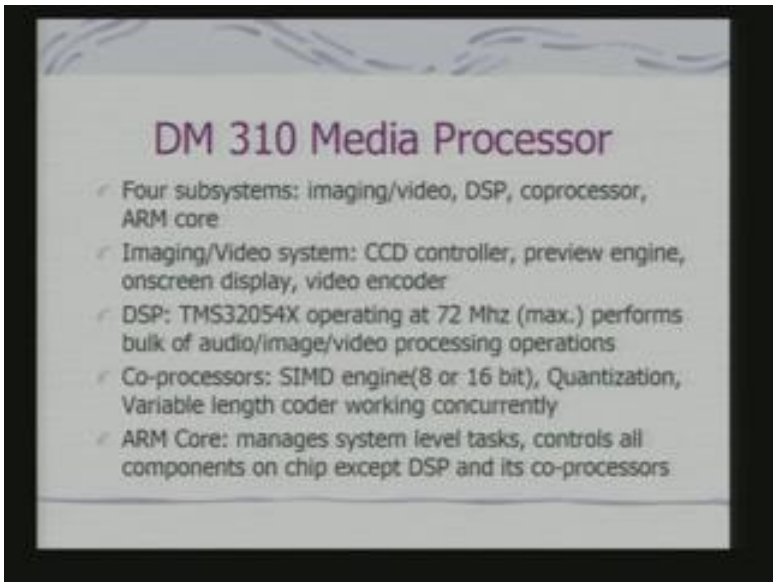
(Refer Slide Time: 40:03)



In fact, the another example, towards this direction is digital media processor, which are called portable digital media processors for providing all this services in a standalone modes. So, what we say that, you have device a chip, which can support video. As well as image capture, audio capture, video playback, audio playback, printing interfaces. So, it is just not a camera, just think that if you have such a SOC, you can be a devices which will not be just a camera.

But, it can be a audio recorder, MP3 player, as well as through which you can print images and everything put together. And in fact, these kind of commercial interest, actually drives this manufactures to design to visualize and design this kind of multiprocessor application platforms they have divide, provide in the platform, and others will be building systems around it. So, here it is an example of what I have state it is the functionalities required. And several modes would operate concurrently.

(Refer Slide Time: 41:14)



And taking this into account, there is a DM310 media processor, which is again an SOC a multiprocessor SOC. And it has got four subsystems, imaging and video, DSP a set of coprocessor, as well as ARM core. And in fact, that ARM core is being used in a very extensive fashion. So, you have this imagine and video system, has got a CCD controller. I hope you remember that, we had talked about CCD as a basic imagine sensor.

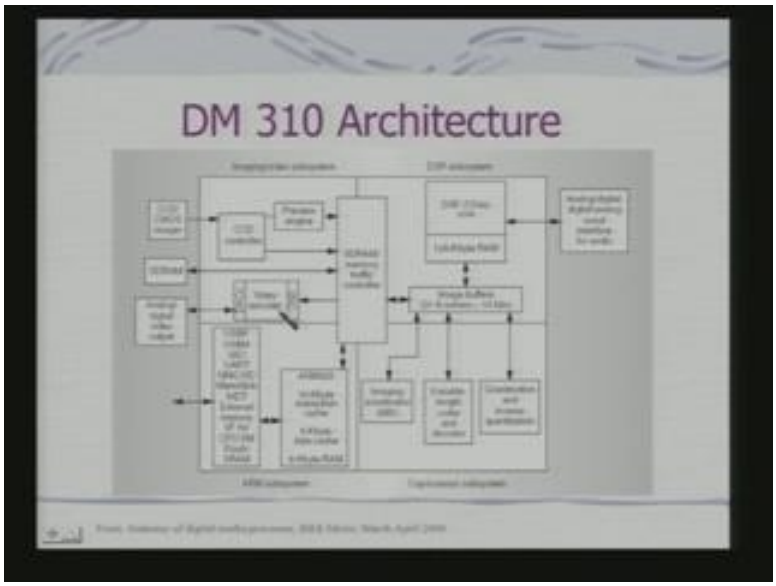
Now, the CCD controller it is just not a, if you compare this with that of your OMAP interface with the device. OMAP has providing a simple a serial, or a parallel data interface. Here, we are talking about the CCD control. That means, you can actually control the sensor itself. So, that sensor control logic, now goes into the SOC. In the case of OMAP, what is expected, that your camera is providing the data. And you have got the interface to get the data.

Here, how to extract the data from the CCD sensor itself. And how to manage the CCD sensor that hardware, becomes part of the SOC. So, it has got 54X. And the other

interesting part is along with this, it is got coprocessors. Co-processor is an SIMD engine, you all understand. And by now know that, why SMID engines are important for this kind of media processing applications.

A special engine for doing quantization why? Because for compression tasks, quantization is a very important step. As well as variable length coder. A very common example of variable length coded is a Huffman coded. We generates the cores, which are a variable length. So, you have a targeted coprocessor for this purpose, that means these are all dedicated hardware for this tasks. And on top of that you have got an ARM core, which managing your system level tasks just like your OMAP.

(Refer Slide Time: 43:16)



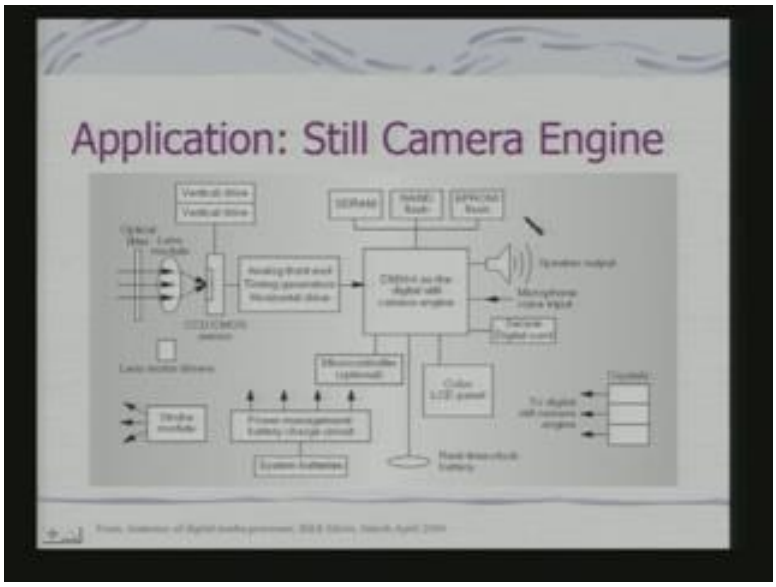
So, the architecture looks something like this. You have got your DSP core, you have got your ARM core. And the DSP core makes use of this coprocessors; and this is your image buffer. So, to this image buffer, which is on board memory, which is also being used by the DSP. As well as this coprocessors, which is imaging accelerator, variable length coder, quantization, in inverse quantization bocks and they manage it.

And this is the basic ST RAM for other purposes. And this is the separate coder, which is primarily looking at the video application. Now, here you will find that the whole approach has been to provide, a kind of a digital media processing capability. And you can understand, the slight difference between the representation with regard to the

OMAP. What is the basic difference? The basic difference is, OMAP is targeted for communication, as well as this kind of media processing application.

It is more targeted towards media processing. So, you have got actually a CCD controller, kind of blocks, in appropriated into SOC. And coprocessors for each and every tasks, related to media processing. And we can see now, how a complete embedded system can be built around it.

(Refer Slide Time: 44:48)

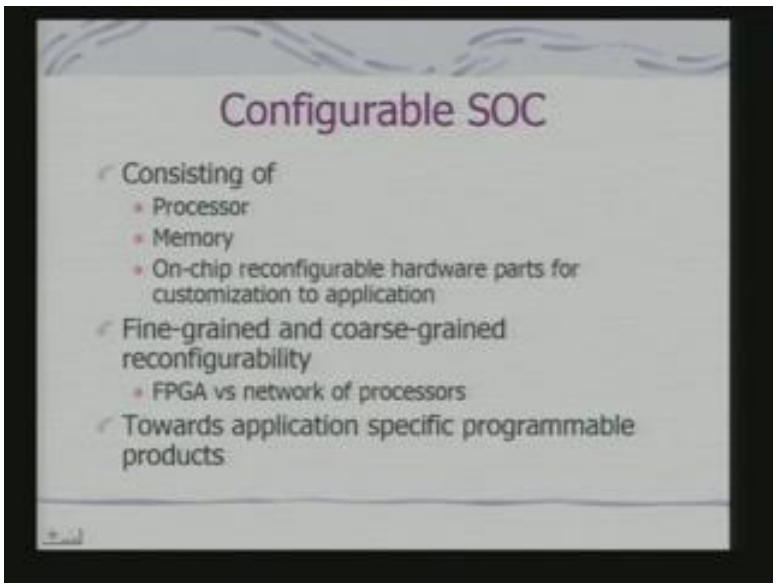


So, simple example is still camera engine. It is not just a still camera can be built around it. The still camera can have all source of other functionalities, related to that. But, this is just an example to illustrate, how this simple SOC can be used to built a complete system, a complete appliance. So, here you have got the CCD sensor, analog front end timing generators and horizontal drive. And that gets interface to your CCD controller, which is sitting on the chip itself.

You can have an additional microcontroller, which maybe optional. You have the power management circuit. These are basically your memory, this flash or where the programs will be stored. You have got a speaker output, microphone voice input, secure digital card. You can also ensure a kind of a secured access. Color LCD panel, which would be interface to see, what is the image being capture.

So, you can have a image capture, you can even annotate the image with your speech. You can say that, I am taking this image on such and such state; and such and such place. So, you can therefore, find that using this SOC, you are built a complete appliance. And this is what is a typical example of an a complete embedded system built around, this SOC. Next, we shall look at reconfigurable platforms.

(Refer Slide Time: 46:19)

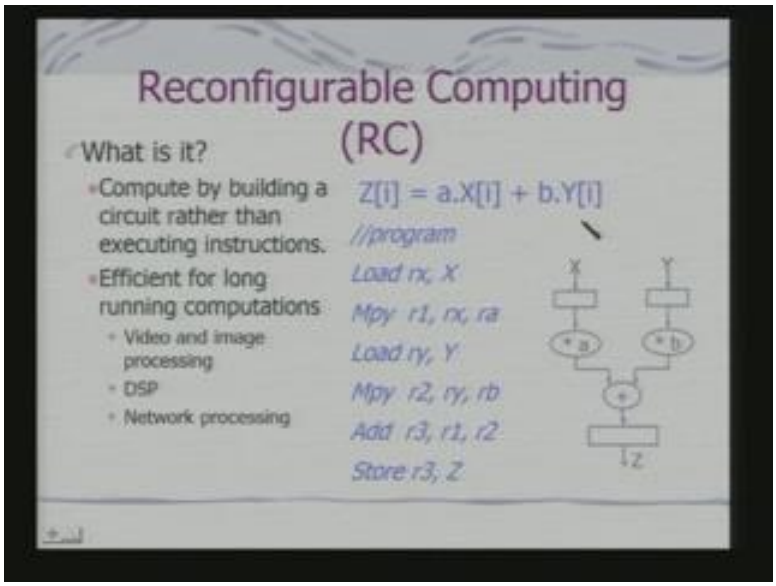


What are reconfigurable platforms, reconfigurable SOC is, typically consist of processor memory as well as on-chip reconfigurable hardware parts, for customization to application. These could be fine-grained or coarse-grained, when it is a fine-grained you have got FPGS. The excrement of the coarse-grained, there can be a network of processors, a multiple processor and this processors can be configured for your work.

And what we say, that it is a moment towards application specific programmable products. If I look at ASIP, which has an applications specific in rendered circuit there the whole design is targeted for an application. And there the focus is not re-programmability of the chip. Here, we are talking about an application no doubt but, at the same time keeping in mind the re-programmability.

So that means, depending on maybe change in the protocols. It should be possible from it user same hardware and reprogram it, and make the plans. I did not freshly design and ASIP. So, what is really reconfigurable computing.

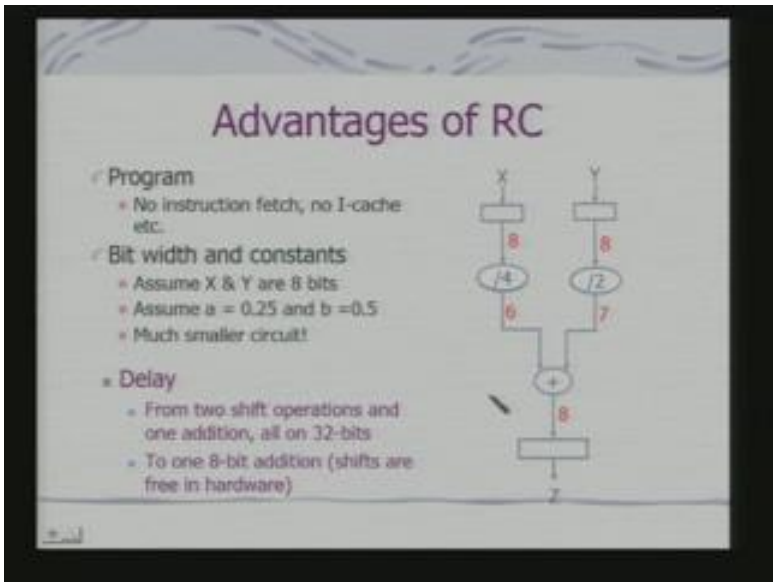
(Refer Slide Time: 47:37)



So, what we say, that compute by building a circuit, rather than executing instructions. Take for example, a simple multiplication here. I am multiplying the two elements by constant a and b. And I am trying to compute the result. Now, what is the interesting here is, this is the set of assembly language instruction. So, it is a multiply load, you can understand this will be used for computing this function.

So, if I have the ability to generate a circuit, corresponding to this statement then what I am really doing is, a reconfigurable computing. And how to generate this circuit obviously, I can know the operations. How do you implements the circuit, if I have got an FPGA block, I cam program the FPGA block, to realize up these particular function.

(Refer Slide Time: 48:34)

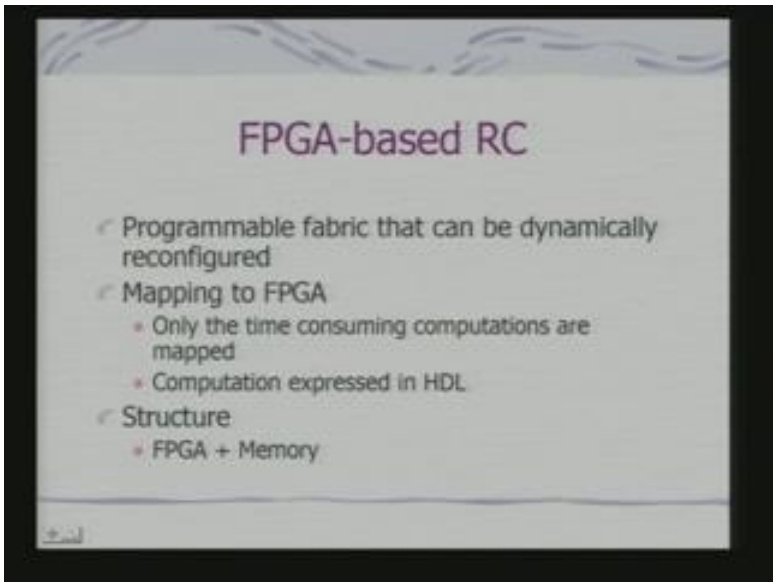


What is the advantage of this, we did not have, did not have an instruction, fetch no instruction cache require because, that is going into the hardware itself. Bus with an constants are typically issues, because in many cases I really do not need 32-bit bus transfers. Because, say if in this example, if X and Y are 8-bits and I have a 0.25 and b 0.5. I can have a very simple circuit.

So, what I have got here is, this multiplication and addition. At this 0.5 and 0.4, 0.25is division by 4 and division by 2 it nothing but, shift operations. Nothing but, shift operations and I need to do addition. So, I have got a very simple circuit and what I say is that, the delay from two shift operations and one additions all on 32-bits. These happens when, when you are using the instructions translating that operation into a set of instructions, to this kind of a circuit.

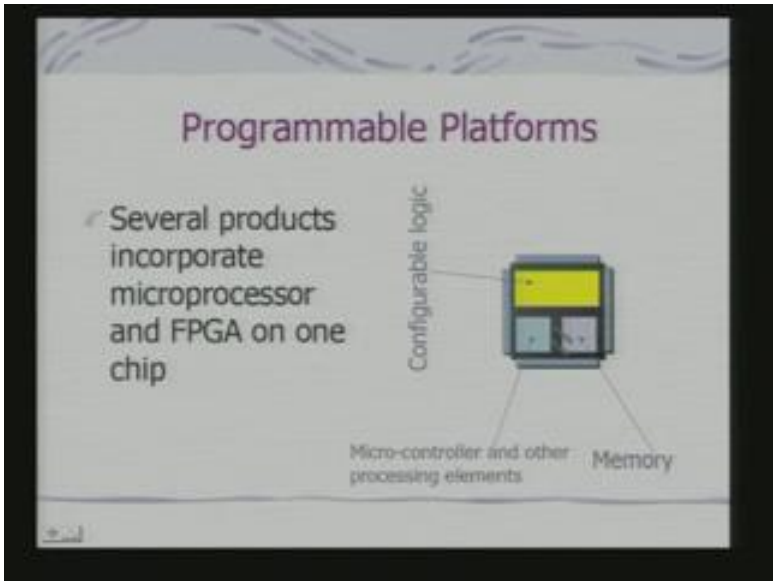
So, what you have got two 1-bit addition. And shifts are almost free in hardware. Because, if I am using a barrel shift kind of a circuit. And buildings small such circuits using FPGA blocks, then I can do shift in almost no type.

(Refer Slide Time: 49:47)



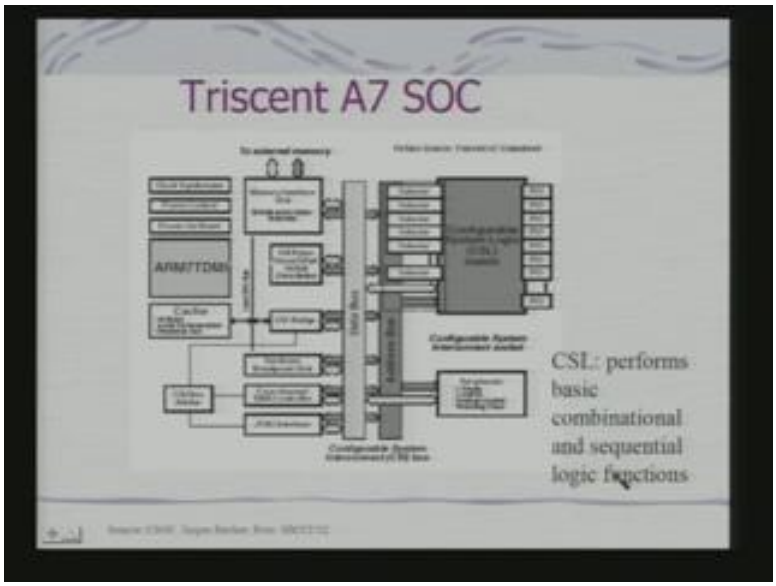
So, what we have got now, what are called FPGA based RC, that is programmable fabric, that can be dynamically reconfigured. Any of two mapped therefore, this code to FPGA, link on figure the FPGA. And then, do the computation. So, you have got a programmability not in just software, but as well in hardware.

(Refer Slide Time: 50:13)



So, the typical structure is that I have got a configurable logic I have got microcontroller as well as the memory. And these logic you can used for programming the hardware.

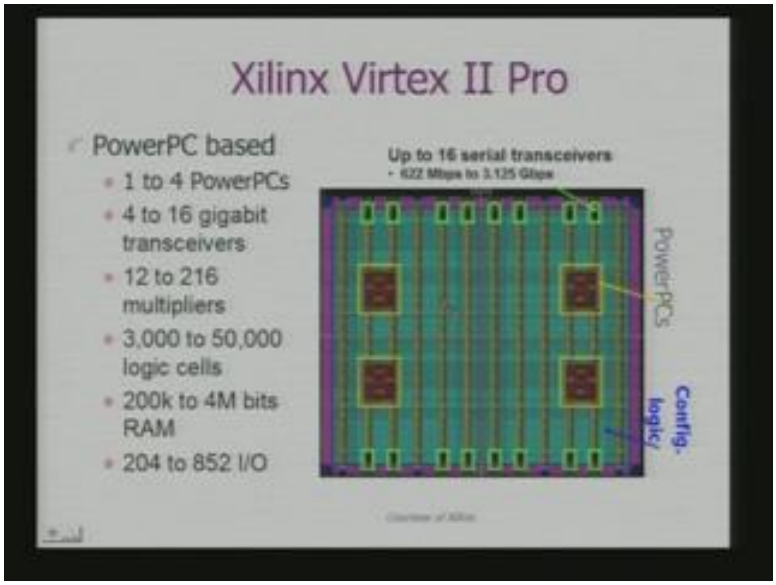
(Refer Slide Time: 50:27)



A simple example is this Triscent A7 SOC, which is again using your ARM7. That we are already studied and this is your configuration system logic block. And these are units which perform basic combinational and sequential logic functions. So, collection of gates, they can be configured to realize different function. And I can use a set of search gate to realize these functions together.

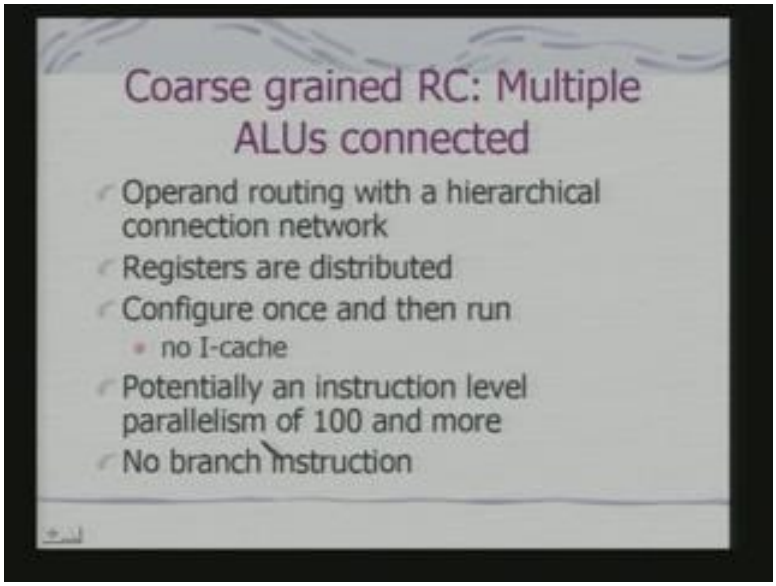
So, I have got simple computation repeated if computation mapped on to in a dedicated hardware. And these hardware can be design depending on the application that we are looking at. And in fact, there is there is this signal interface. So, that can be changed from one application to another application.

(Refer Slide Time: 51:17)



This is an example of this kind of programmable platform that I was talking about Xilinx Virtex II Pro. It has got 4 power PC processor cores sitting on the same SOC along with the FPGA logic. To add on the hardware and it has got serial transceivers to transfer the data.

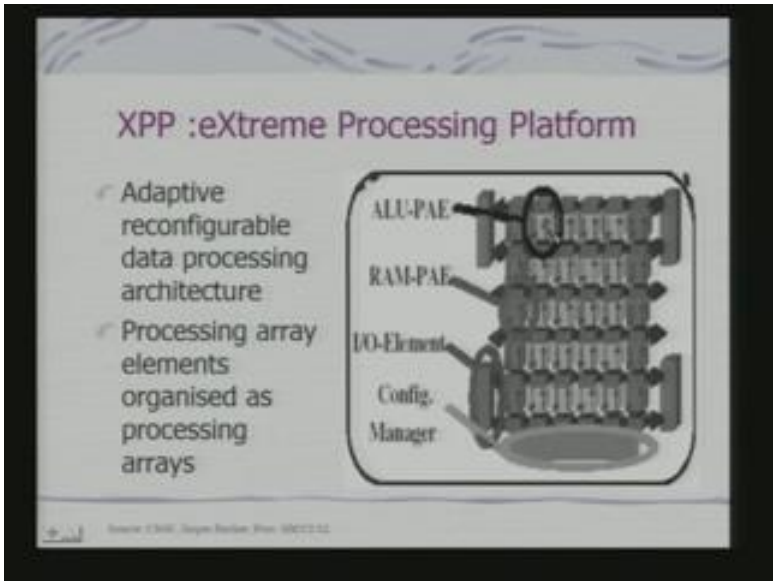
(Refer Slide Time: 51:40)



Then we can have coarse-grained re-configurability we have got simple logic gates putting together logic gates to realize simple logic functions. From there you can have a set of ALU's and you know configure this ALU's and the set of registers to do some tasks. So, it becomes an example of a coarse-grained re-configurability.

So, potentially an instruction level parallelism of more than hundred you can get. Because, if you remember VLIW you have got multiple processing units. Now, since you have got, s, many ALU's and registers which you can configure and network. Then you can actually have maybe of the order of 100 ALU's, 100 parallel operations being implemented.

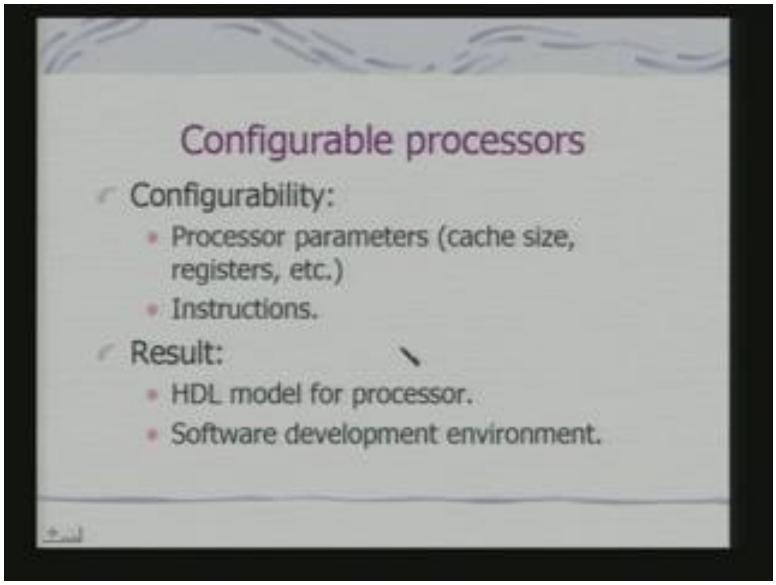
(Refer Slide Time: 52:29)



This is an example of extreme processing platform. I have just shown the extreme processing platform part, this could be interface to the again a microcontroller core by a semiconductor designer. So, what it has got? It has got n array of ALU's, these are all ALU's an array of ALU's, each ALU has got its own RAM.

And it has got a set of IO elements. And you have got a configuration manager, where you actually put in the code to say how this ALU's will be configured for a particular computation tasks. And the data for those ALU's will be in this RAM. So, you can see the extend of re-configurability that you have got now. And you have using a processor core and your reconfiguring the hardware depending on the application.

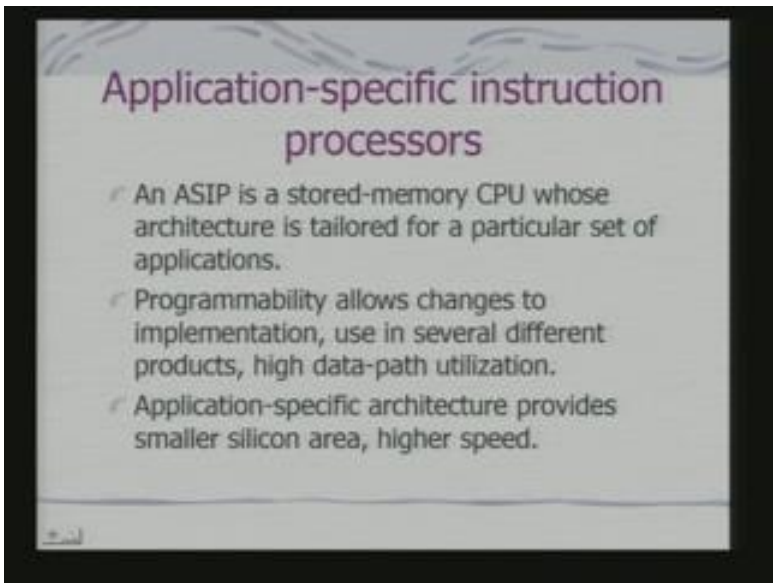
(Refer Slide Time: 53:23)



The other end is also configurable processors. The previous example what we said we have a processor as well as FPGA. We can now configure this FPGA or this ALU's depending on the applications. Here, we are looking at configurable processors. So, processors can source can be configured before you actually go into a design. So, processor, parameters, registers etcetera can be configure.

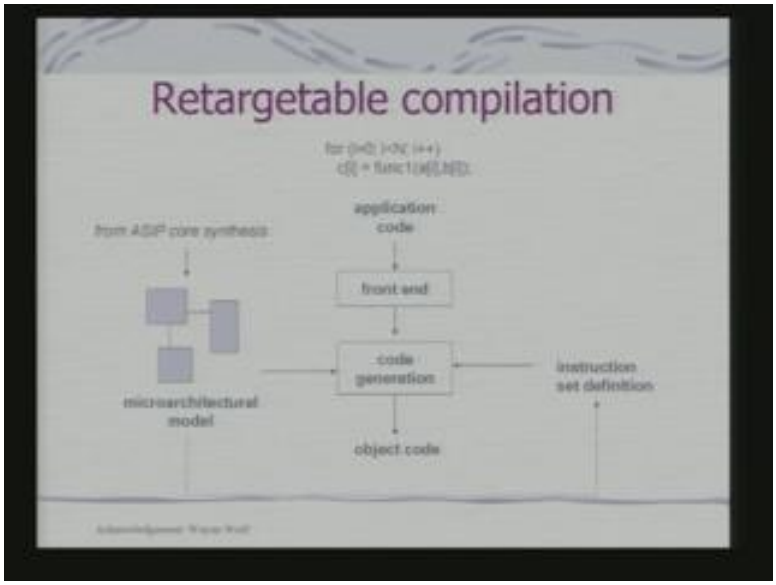
So, what we have you start with an HDL model of processor and you have got a software development environments. And this is what are called Application Specific Instruction Processors or ASIP.

(Refer Slide Time: 53:57)



So, programmability allow changes to implementation using several different product and high data path utilization. So; that means, you are configuring, the ability to configure the architecture of a processor depending on the applications. Not the major part, but at least the registers, the caches eyes and those elements. But, the key problem in this case remains.

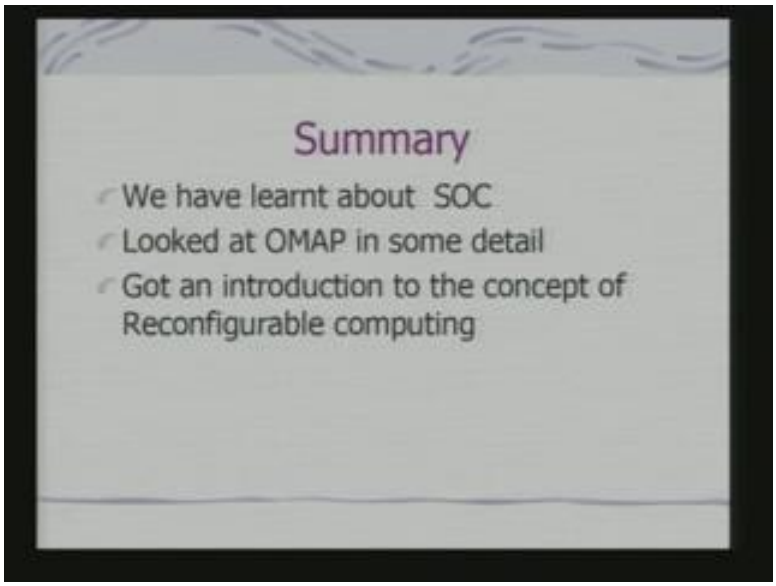
(Refer Slide Time: 54:21)



How will a software tool behave? Will you design a software tool for each an every variant of the processor that you are producing. So, you have concept of what is call retargetable compilation. So, that retargetable compilation takes as input the architectural modal, as well as the language specification and that is used in the code generation block.

To generate the code for the processor, that you are trying to synthesize. So, you can see here I I try to show how this configurability has implication in terms of software technology as well.

(Refer Slide Time: 55:01)



So, what we have to the look that is SOC is of this SOC we have reviewed OMAP in some detail and got an introduction to the concept of reconfigurable computing. And this is what we are discussed today is basically a reflection of the current trend of embedded system hardware.