Presentation Title: Interfacing Blackfin® with Audio and Video Peripherals

Presenter Name: David Katz

Chapter 1: Introduction
Chapter 2: Connecting To Audio Converters
Chapter 3: Connecting To Video Converters
Chapter 4: Digital Video Refresher
Chapter 5: Blackfin Video Interface -- PPI
Chapter 6: Connecting To Video Sources
Chapter 7: Connecting To Video Displays

Chapter 1: Introduction

Hello my name is David Katz and I'm a member of the Blackfin Applications Team at Analog Devices. Today I'm going to talk with you about interfacing audio and video devices to Blackfin processors. My goal in this presentation is to familiarize you with the basic principles behind connecting Blackfin processors to audio and video devices. You don't really need too much background knowledge for this presentation, but a basic working knowledge of audio and video would be helpful. I'll try to provide some refreshers as I go along where appropriate.

So I'll start out in this module by talking about Blackfin audio video connectivity in a very broad sense, and then I'll zone in a little more on audio converters, specifically on how Blackfin connects to those, what are the relevant peripherals, what are some examples of connections. We'll look at some block diagrams, and what are some interfacing tips and tricks. And finally I'll close the audio section by listing some collateral that you would find very useful in developing an audio system.

Then over to the video side. I'll do something very similar, first starting with a digital video refresher, but other than that proceeding along a very similar course.

So here we have a basic AV system connectivity diagram. Obviously this is just an example but it shows the Blackfin's view of the AV world. We have the processor in the middle and then depending on what we're hooking up to, we have several different devices in between the Blackfin and the end device. So for instance we have a microphone going through an audio converter, an analog to digital converter, such that the Blackfin can read the digitized stream. Similarly a digital stream from Blackfin will go through the audio D to A converter (digital to analog converter) before being sent out to the speakers. On the right hand side we have the video, the video analog to digital converter that accepts the analog stream from some analog source, such as a camera and digitizes it to Blackfin's side. This analog video converter is known as a decoder, a video decoder. Similarly on the display side we have a video D to A, or a video encoder, that accepts the digital stream from the Blackfin and converts it to analog form for display on a CRT for instance.

So why is Blackfin well suited for A/V applications? It's important to frame this discussion in some basic background. One is that Blackfin offers multimedia grade performance. This means a couple things; one is that it offers very high clock speeds, up to 600 megahertz and beyond, and this allows for real time computation that allows for video compression with high resolution formats, multiple formats being supported with video and audio streams for instance. And coupled with that is a flexible instruction set that allows very complex and efficient operations to be performed on every clock cycle. So then there's the powerful connectivity and data handling side of things where the direct memory access controller, the DMA controller, of Blackfin is completely decoupled from the processor core, which means that it can be moving data very rapidly without bothering the core while it's doing its processing. Also on the connectivity side, the peripheral mix is very important to be able to connect to these the external world. Blackfin needs and has a rich peripheral set.

Finally scalability is key. Scalability not only across applications but within applications. So dynamic power management is one of the fundamental pillars of Blackfin. It allows, within an application, for voltage and frequency scaling to the point where the application needs it such that voltage or frequency or both, can be dialed down if an application does not need that level of performance at a certain time.

Now in a global sense, the wide product portfolio of Blackfin allows everything from the lower end, let's say 300 MHz single core processor all the way up through dual core processors with each core having 600 MHz performance. So there's a huge range of performance scalability in that window, and this allows for not only feature enhancements for existing systems, but also the ability to rightsize the processor to fit the application.

Chapter 2: Connecting To Audio Converters

So let's talk now about how Blackfin connects to audio converters. There are two main types of serial interfaces on Blackfin that are relevant in the audio world, one is a low bit rate interface that is used for control and configuration of audio devices, and another is a higher bit rate peripheral that's used for the actual data interchange. So on the control and configuration front there are two peripherals in high usage there, one is the two wire interface, the TWI, and the other is the serial peripheral interface, SPI. And we'll talk about these more in a second, but in general the forward channel of these peripherals is used to configure or control an audio converter, and the reverse channel relays feedback or status information from the converter.

Now the SPORT (what we call it on Blackfin) is a synchronous serial port. And it's used as a data channel for audio data. It can operate at high bit rates and it's full duplex and supports both transmit and receive applications. It can do so simultaneously, and I'll talk about that more in a little bit as well. I should note that in some codecs (a codec is a combined audio A to D and D to A), in some codecs like the popular PC AC97, the SPORT also serves as the control channel for the Kodak.

So let's talk a little about SPI. The Blackfin peripheral is compatible with the Motorola SPI standard. It's a full duplex serial interface and it operates up to 33 Mbits per second, which is usually well beyond what most converters need for any type of control. It supports both master-slave and multi-master environments, and it really only consists of a three pin data communication interface. There's the MOSI, which is the output from a master that is an input to the slave device. There's the MISO, which is the input to a master from a slave, and then there's the serial clock, the SCK line. Then in order to enable the master and slave side of things, the SPI chip select input is used when the Blackfin is a slave to some other SPI device. The SPI device will select the Blackfin through that line. When the Blackfin is the master, on the other hand, it will use one of it's many SPISEL lines, one of its slave select lines, which allows it to select any number of other devices, usually a processor will have, let's say, on the order of six or seven of these slave select lines.

Now the TWI, the two-wire control interface is compatible with the I^2C interface from Phillips. And it provides a very simple and concise way to exchange control and data information between multiple devices. The Blackfin implementation supports both master and slave operation, and it also supports transmission speeds up to 400 kbits per second. The only two lines involved here are the SCL and SDA, nominally the clock and the data lines. They comprise the interface and the phase differences between those data streams determine the mode of operation that you're in at a given point in time.

The SPORT, as I mentioned, is the high speed Blackfin synchronous serial port. It features fully independent transmit and receive channels, and as we see on the right hand side of the figure each side of the interface, the transmit and the receive side, consist of four pins. So there's the primary data, the secondary data, the clock and the frame sync. And we have four for transmit, four for receive.

The secondary data line actually is synchronized to the same frame sync and the same clock that the primary is, and this allows basically double the data rate at a given time in both receive and transmit directions. So a primary data channel can operate up to 66 Mbits per second, and this doubles to 133 when you include that secondary line. The SPORT supports a very broad word length range, from three bits, all the way up to 32 bits. So this encompasses and surpasses most high precision audio applications. It also has programmable internal and external clocks and frame syncs. And for speech applications there is built in companding, both µ-law and A-law, and there's also support for a multi-channel interface, a TDM (time-division multiplexed) networked kind of interface. And finally I^2S signaling is supported. I^2S is an industry standard that was developed by Phillips for stereo audio transmission. It nominally consists of three wires and these are the serial clock, the frame sync, which in this case is the word select, the left channel or right channel, and the data. So data always transmits in most significant bit format and each SPORT can accommodate four transmit and four receive audio channels because of the fact that there's the secondary channel coupled with every primary channel and the left/right frame sync for each SPORT gives a total of four channels in each direction. So you can receive a pretty good stereo density with one or two SPORTs or even more in your system.

All right. Here's just an example diagram of how to hook up to an audio converter, in this case we're looking at an A to D, the AD1871. We have an analog signal from a microphone coming in and we're converting it with basically a pretty straightforward interface. The control channel is via SPI, so the Blackfin is the master, and selects the SPI peripheral of the 1871. And then the data port is just a straight transmit using three of the four SPORT pins in the receive direction on the Blackfin. So the Blackfin is receiving the audio data that's digitized by this converter.

Now in a very similar vain we can connect to an audio DAC, a D to A converter. Here's the 1854 connecting to the Blackfin. Again SPI interface, the Blackfin configures and controls the converter, and this time of course the data is flowing in the other direction. The SPORT sends the audio to be made in analog form, sends it over the SPORT to the 1854, the 1854 converts it and then sends it out for output on speaker or some other medium.

Before we close off audio I want to leave you with some tips and tricks for connecting to these converters. On the two wire interface it is important to remember to put pullup resistors on both SCL and SDA and this is as per the I^2C spec. Since these pins are never driven high. On SPI, this seems like kind of an obvious statement at surface level, but it's a very common mistake, we want to make sure that all MISO pins are connected to MISO pins. And same with the master out slave in, MOSI, those should be connected to MOSI. So in other words, tie all the MOSI pins in your system together, and tie all the MISO pins together. And also, of course, because the two signal names are very similar, make sure that you're not swapping the signal names and because of that their actual functionality.

In the SPORT domain just a couple of tips, one is that if you're operating in multi channel mode, and you happen to be in a mode where you're mastering the clocks and the frame syncs, don't connect your transmit frame sync to your receive frame sync because in this mode the transmit frame sync actually is framing the active TX data and it's like the transmit data valid pin. So this is a very common mistake and it's bubbled up to the surface on the SPORT as one of the main things to watch out for. Also make sure you use proper termination for your clock and your frame sync signals. Of course this is dependent on the speeds you're operating at and board layout and things of that nature.

Some additional collateral for audio development that you definitely would want to look into, and I'll give links for this at the end of the presentation, but from a software side the Visual DSP++ tool suite includes many peripheral drivers including the two wire interface, SPI, SPORT and others, that allow you, via a standard API, to configure and control those interfaces. Also there are complete audio device drivers included under VisualDSP, and this provides a standard API for audio converters. Many devices are supported, some of them are listed here and more are being added continually.

Also the EZ-KIT, the Blackfin EZ-KITs and EZ-Extender cards that plug on top of the EZ-KIT, there are many code examples that come with those platforms, and they show how to integrate these peripheral and these converter device drivers and they also provide really good examples of data flows and how to configure your system when you have a lot going on and they provide a very useful framework.

On the hardware side, of course, we have the aforementioned the EZ-KITs and Extender Cards, these come with onboard audio converters along with the code to drive them. This allows for very quick system prototyping and also it comes with reference schematics and the bill of materials associated with those hardware boards. So you can get a really nice jump on system development.

And finally I just want to mention the Visual Audio® algorithm development tool which really can help to streamline design of audio systems.

Chapter 3: Connecting To Video Converters

Let's cross over to the video side now, we'll see a lot of similarities but of course video being more complicated in some respects, there will be some additional information. And I'd like to first start with talking about a system video flow and then very briefly review some digital video basics. Not at all meaning to be comprehensive, but just trying to frame our discussion. Then I'll talk about the parallel peripheral interface, the PPI which is Blackfin's high speed converter port, which is also Blackfin's video port. Then we'll talk about connecting up to some sources and displays and a look at a video system diagram, an example diagram.

So the Blackfin's view of the world, this slide represents how a sample application on Blackfin might view the outside world. We have on the right side the Blackfin processor just running some sort of, maybe it's compression or formatting or image enhancement, some kind of algorithm. But what it really cares about is getting the right inputs, and sending the right outputs to these video sources and displays. Video sources generically come in two types, there's the analog source and there's the digital source. The digital source, such as a digital CMOS sensor can hook up directly to Blackfin because it's just a digital output stream from the sensor that can hook straight into the PPI on the Blackfin. An analog camera source first has to pass through a video decoder, and basically it's just an A to D converter, which digitizes the stream for input into the Blackfin. Something like a CCD sensor will be, it'll pass also through an analog to digital converter called an AFE, or an analog front end.

Now on the bottom, in the video display realm the Blackfin can natively send a digital stream out to something like a TFT-LCD display which accepts digital input. Also if it's going to something like a CRT or some kind of TV or analog display, then it first must pass through a hardware encoder, a video encoder, which is a D to A converter.

Chapter 4: Digital Video Refresher

All right. So I'll just give a very brief refresher on digital video. The ITU recommendation for digital television is BT.601. That was the recommendation that specified from a broadcast perspective, how to encode digital television signals. It supports a couple of color spaces, the most obvious one is RGB, red, green and blue, which is very intuitive, but there's a lot of correlation between the channels and so it's not really the best one to use for compression. There are three color values per pixel, there's a red value, a blue and a green. And the format -- when you say RGB888 format this implies 8 bit each for the red component, the green component and the blue component. If we say RGB666 this means that there are six bits each of those per pixel and 565 means there are 5 for red and blue, but there are 6 for green. And we'll talk more about that in our LCD discussion.

Now the most popular color space supported and the one that BT.601 prefers is the YCbCr color space. Where Y equals the luma component and the Cr or Cb equals the chroma component. These signals are generated via RGB in a mathematical relationship and they are highly uncorrelated, which means that they provide much better compression characteristics then RGB information does. And that's one of the key reasons that this color space is so popular, is recommended by BT.601.

4:2:2 YCbCr specifically is what's recommended for the broadcast realm. That means that the chroma is subsampled such that there's one luma and one chroma value, either a Cr or a Cb per pixel. BT.601 allows for 8 bit or 10 bit quantization of color space components. And the NTSC and the PAL broadcast formats which are really the end game of BT.601, this recommendation allowed NTSC and PAL to standardize such that each of them has the exact same number of active pixels on a line, 720 active pixels. And then the different frame refresh rates are just normalized by including extra lines for PAL because it only has 50 fields per second refresh rate, where as NTSC has 60 fields per second.

The basic timing signals associated with digital video are Hsync, or horizontal sync, which demarcates the start of active video on each row going left to right on a video frame. Vsync is the vertical sync signal and that defines from top to bottom the start of a new video frame. Field is used only really in interlaced video and that denotes which field between odd and even, which field of a video frame is currently being displayed. For progressive scan systems it's not really used, not really important. And finally there's a clock which is a data clock for each pixel component.

Now moving from BT.601 the next layer that was implemented for digital video, the next recommendation from the ITU was BT.656. This basically took 601 and it defined the physical interfaces and data streams that were necessary to implement the 601 recommendation. So it has bit parallel and bit serial modes, we are only concerned with the bit parallel mode here. For broadcast NTSC and PAL it specifies a 27 MHz nominal clock rate, and 8 or 10 data lines depending on the resolution of the system. The great thing about 656 is that all those synchronization signals I just talked about are embedded in the data stream. So all you really need is the data stream and the clock. And the signals in a straightforward manner for 656, H stands for Hsync, V stands for Vsync and F stands for field. And we'll talk more about that in a moment.

656 does support both interlaced and progressive formats.

So here is a representation of a video frame both for NTSC and PAL under the 656 recommendation. Basically this shows that the horizontal blanking region is demarcated by the H bit, so EAV or End of Active Video is when H equals 1. The transition to H equals 0 is the SAV, the Start of Active Video. Likewise when V changes from 1 to 0 it denotes a new region, blanking versus active video. And then the field, Field 1 and Field 2 are also denoted by the change between 0 and 1 on the F line. These control codes are sent in the data stream like I mentioned, along with the data. So what you'll see if you were to take a snap shot of the data stream is a few leading bytes, (we're looking at an 8 bit case of 8 bit video components). "FF 00 00" -- that's the preamble associated with a control code. It lets the end device know that a control code is coming. And then the "AB" in this picture is really just the pertinent control code that tells if H is changing, if V or F is changing and there are some checksum bits in there for error correction as well. Then if we're at the start of a line, you'll see a long sequence of horizontal blanking, which is denoted by the "80 10 80 10" sequence. And then there's another preamble code that tells the system that H=0, the SAV (the start of active video) is coming. Then there will be the whole video line, the 720 active pixels which equates to 1440 bytes. And then we're off to the start of the next line with a new end of active video sequence.

Chapter 5: Blackfin Video Interface -- PPI

So I had mentioned that the PPI on Blackfin is the video port that handles this information. PPI stands for the parallel peripheral interface. And it is a video interface however it generically supports high speed parallel converter interfaces as well. And in the generic sense it is a high speed parallel port. But it has some nice video friendly features as well. It's pretty straightforward interface, it consists of 16 data lines, PPI15 through 0 in this picture. Up to three frame syncs, PPI_FS1, _FS2, and _FS3 and then the clock, PPI_CLK. This clock is always sourced externally to the PPI, but the frame syncs and the data, depending on the mode you're operating in, can be either inputs or outputs. I should mention as we'll see on some subsequent slides, in the 656 modes of operation these frame sync lines are not required because everything is embedded in the data stream.

So the PPI is bi-directional, it's half duplex so you're either doing a video input or a video output sequence at a given time. And we support the bit parallel recommendation for ITU 656. The clock and frame syncs have some signal polarity programmability, and the PPI has a lot of bandwidth saving features, for instance you can choose to receive only certain parts of the 656 input video frame. You could receive only the active video and ignore the blanking -- by ignore I mean that that blanking information does not filter through to the DMA engine, so the DMA bandwidth, the data movement inside the part is saved because you're not reading the blanking information into the part. Now you could if you wanted only look at the blanking information, this might be interesting in some kind of interactive TV applications or closed captioning, etc., where you're interested in the blanking intervals of the video frame. Or of course you can read in the whole video frame (blanking plus active video).

You can also optionally ignore the second field of an interlaced stream. This can be useful in some applications where data quality or video quality is not as important as the compression that you can achieve by doing this, by using only Field 1 of every frame. You can also skip even or odd data elements, so you can read in only the luma or only the chroma of a 4:2:2 input stream. And the whole PPI peripheral works hand in hand with the concept of the 2 dimensional DMA engine that Blackfin has, which allows you to specify an arbitrary region of memory and send it out to the PPI without having to worry about parts that you do not want to send out. You can specify an arbitrary rectangular region of an image, let's say, and send it out the PPI.

Chapter 6: Connecting To Video Sources

Okay let's look at connecting up to some video sources. You'll see there's not too much difference from what we've seen before in the general sense, we have the control channel here where we're looking at a CMOS image sensor. We have a control channel that in this diagram is through the two wire interface that connects to the sensor's I^2C bus. And that's used for configuration of the device where the actual data flows straight into the 8 bit PPI in this case. There's no reason that this couldn't be a 10 bit or 12, or even 16 bit application, depending on the resolution of the sensor, of the imager that you're connecting to. But this diagram shows an 8 bit connection.

The imager in this case usually will provide the pixel clock as well as some kind of framing. The horizontal sync will demarcate the valid line region and the Vsync will be a "frame valid" type of signal. And of course many sensors also support the BT.656 standard in which case these synchronization signals are not needed.

Now while we're on the subject of CMOS image sensors I should mention that the Blackfin EZ-Extender Cards and the EZ-KITs support connection to a wide range of CMOS sensors from these leading vendors: Micron, Omnivision, Kodak and others. As an example you see here Micron makes a CMOS imager headboard that provides a common interface that can connect to a whole family of sensors. So this is a great development as far as connectivity on CMOS imagers.

Now if we're in the analog realm, if we're taking a signal from an analog camcorder we'll need to use a video decoder first, such as the ADV7183B. The analog signal goes into the video decoder, again the Blackfin configures the part via I^2C, the TWI interface, and will get an 8 bit digital data stream from the 7183 straight through to the PPI, and in this case the video decoder is clocking the Blackfin. It provides a line locked clock for the Blackfin PPI.

Chapter 7: Connecting To Video Displays

Okay, so that's the source side. If we look at displays we have a pretty parallel situation here with the video DAC. If we have video information, a frame stored in memory that we're planning on sending out to a display, if it's an analog based display we first need this video encoder such as the ADV7174 or 7179. We configure via I^2C, we send the data via the PPI data bus, and if needed, we send these frame sync signals via the PPI frame syncs. Again this particular encoder does not need them. Many video encoders do not need them because they adhere to the 656 recommendation. In this case the oscillator is external to both the encoder and the Blackfin device, whereas in the decoder case there was that line locked clock that directly sourced the PPI clock. Now TFT, or Thin Film Transistor, LCD panels are kind of different from what we've looked at so far in some ways. But in some ways they're very similar. If we look at the middle of this picture we see again the same kinds of things we've seen before for connecting to a video device, we have the horizontal and vertical syncs, the data sampling clock, and in this case again the clock is external to both the PPI and the LCD panel. So that's the signaling layer, the synchronization layer. On the bottom we have the data bus, now here almost always when we're connecting to TFT-LCD panels we'll have the whole 16 bit data bus being used because most of these TFT panels are 18 bits and beyond. So I'll talk more about that in a minute, but essentially we have 16 bits from the PPI connecting to the 18 bit panel. And I'll get into a little more of the reasoning in a second.

The PWM timer block that you see up on top is an optional connection that many times these TFT-LCDs for reason of cost or footprint will not integrate a timing controller, which means that you buy the TFT module and then you buy a separate timing ASIC to go along with it, which is added cost into the system. In these cases where the TFT module does not support the timing controller directly, the Blackfin can many times supply these signals through some extra timers and it would be on the order of what you see here, a signal to, say, start sampling, a signal to denote the scan direction. And then maybe a clock for gate drivers.

Okay some tips and tricks to help connect up to video sources and displays, first of all as you kind of gleaned from my general tone here I think, I'm a big fan of BT.656 and I think it should be used whenever possible because it eliminates a lot of timing incongruities and inconsistencies that can crop up in an application. So when you have the choice, go for 656. Also pay close attention to the default converter settings for whatever A to D or D to A you're using because oftentimes you'll find that you might be operating in a mode that you didn't expect, and a lot of times you'll be able to use these converters directly without programming through an I^2C or SPI interface, but a lot of times you'll also want to verify through the feedback channel of I^2C that you do have the device configured to the appropriate settings.

Also it's very important to make sure in whatever you're doing the clock source is as clean as possible. These clock rates are pretty high, they're in the several MHz to tens of MHz depending on your application. And a clean clock can make a huge difference in the success and the stability of an application from a system viewpoint.

And finally as I was mentioning on the last slide, when you hook the Blackfin to an LCD panel, a 666 LCD panel, 6 bits of R, G and B for a total of 18 bits. The Blackfin connects as a RGB565 device, but when you do this, don't ground the least significant bit of the red and the blue channel. This is a common, I won't call it a mistake, but it's a common practice that is not ideal in terms of the dynamic range that you're representing. Instead at the panel side, you can tie the least bit of red to the most significant bit of red and do the same for blue, tie the least significant bit to the most significant bit. And when you do this, this ensures that you'll get a full dynamic range from the very lowest to very highest value on each of those channels. Whereas with green you're connecting 6 bits, all 6 bits get passed through on the green channel because green is considered the most visually important channel of the three.

So there's lots of additional collateral for video development. On the software side again VisualDSP++ has those peripheral drivers for the PPI as well as for the TWI, for the timers, for all these different things that might be involved in connecting up a video system. And there are lots of complete video device drivers. Not only on the A to D and D to A side of things as you see with the first couple of bullets, but also with CMOS sensors and LCD displays. So these are the kinds of things that are being increased all the time. We are really growing our device drivers collateral here and it will really help in rapid prototyping of some of these video systems.

Again the EZ-KIT and Extender Card code examples are invaluable, especially for video where frameworks make a big difference, where data movement and memory accesses are very important in order to guarantee correct system operation. On the hardware side again the EZ-KITs and the EZ-Extender cards are available. These include both CMOS sensor and LCD interfaces, and naturally the reference schematics and the bill of materials are available for those.

Just to whet your appetite for another one of our online learning sessions, this shows a video system example where we have in this case a DVD coming in as an analog input, it passes straight through to a TV through a video encoder, so basically, it's decoded to digital and then sent out on the upper right side of your screen to a video encoder just for display on the monitor. But then the real interesting thing that's being done is, the digital stream is being passed into a Blackfin processor, encoded, compressed, sent out USB and then stored in a file on a computer hard drive.

So in conclusion the Blackfin processor architecture is very well suited to multimedia system designs, and the Blackfins themselves, the peripherals that Blackfin has for these functions allow a lot of versatility and a lot of good connectivity options to audio and video devices. And there is a lot of collateral available to speed your system design. I've tried to allude to a lot of this earlier and here I'll just provide some links that will be helpful in terms of VisualDSP, the tools suite, the EZ-KITs and Extender Cards, the application notes where you'll find things such as more information on hooking up to some of these sensors and displays as well as good system design practices for board layout and some additional tips and tricks. Of course there's the BOLD Training modules. The Visual Audio algorithm development tool I mentioned earlier, and an embedded media processing book which talks a lot about system flows, resource partitioning on systems and how to hook up in embedded multimedia frameworks -- how to hook these up to Blackfin from a whole systems perspective. And please feel free also to click on the 'Ask a Question button', or send an email as you see below to processor.support@analog.com.

So this concludes this training session about connecting Blackfin processors to audio and video devices. I hope you found it to be useful and that you'll take advantage of some of the great collateral we have to get your Blackfin system up and running smoothly. Thank you.

End of recording.