Presentation Title: Multimedia Starter Kit

Presenter Name: George Stephan

Chapter 1: Introduction
Sub-chapter 1a: Overview
Chapter 2: Blackfin Starter Kits
Sub-chapter 2a: What is a Starter Kit?
Sub-chapter 2b: Benefits
Sub-chapter 2c: Multimedia Starter Kit
Chapter 3: Blackfin SDK
Sub-chapter 3a: What's in the SDK?
Sub-chapter 3b: Supported Devices
Sub-chapter 3c: SDK Installation
Sub-chapter 3d: SDK Root Folder
Sub-chapter 3e: SDK Folder Layout
Sub-chapter 3f: Apps Folder Layout
Sub-chapter 3g: System Serv/Dev Driver
Chapter 4: MMSK Software
Sub-chapter 4a: Algorithms
Sub-chapter 4b: Sample Applications
Chapter 5: Demo: JPEG-MJPEG
Sub-chapter 5a: Hardware Installation
Chapter 6: Conclusion
Sub-chapter 6a: Summary
Sub-chapter 6b: Additional Information

Chapter 1: Introduction

Sub-chapter 1a: Overview

Hi my name is George Stephan. I'm with the DSP and Systems Group at Analog Devices. We're here today to talk about the Multimedia Starter Kit.

About this module training before we begin, this training discusses the Analog Devices Blackfin Multimedia Starter Kit and the Blackfin Software Development Kit, the SDK, if you will. We're going to try to show you how to develop multimedia applications using regularly available hardware and software that comes bundled with the kit, and the tools from Analog Devices. For this training I'd like to recommend that you are familiar with our VisualDSP++ Tool, you have a basic knowledge of the software terminology, and have experience in embedded systems, as well as a basic knowledge of our Blackfin processors and our development boards. I'd like to make a quick note before we begin that when we talk about VisualDSP++ here, we really mean VisualDSP++ 4.5 or later. The starter kits don't work with older releases of the VisualDSP++ tool.

Now let's talk about the outline for this training module. We're going to start off with a brief overview and then I'm going to continue to talk about the Blackfin Starter Kits, specially delve into the multimedia itself, the Multimedia Starter Kit, and then talk about the Blackfin SDK which comes with every Starter Kit that we ship and then continue on to talk about the software portion of the kit, specifically in the SDK, the algorithms that come with the application. And then I'm going to provide an online demonstration, I'm going to use this development board in front of me that's going to be connected to a PC through USB connector so that I can get my content from it. I'm going to use the video monitor on my right to render some video content with a JPEG, and a Motion JPEG, and then I'm going to use the DVD player to do the reverse and capture some JPEG images or some Motion JPEG content. And then finally we'll have a brief conclusion and then I'm going to provide you with additional information for you to look into other things.

Now the Starter Kit is basically a bundle of standard Analog Devices hardware and software that's off the shelf, it's all built around Blackfin processors. The SDK which comes with the Starter Kits is a collection of industry standard algorithms and Analog Devices-provided applications, tools and utilities and such also comes bundled with the Starter Kit. The benefit of the Starter Kit is really the fact that it is the basic building block for evaluating and starting products using a Blackfin processor. It is meant to shorten your development cycle and your learning curve and getting you right to market with your product. The stuff just works – this is the take away that I would like you to go away with today – is the fact that the out of the box experience is the stuff just works, you open it up, you bring up an application, put the hardware together and run the applications, build and run.

Finally this is a mere beginning, we have Multimedia Starter Kit, Audio Starter Kit, and over time there are many more kits coming out, some of them are already in the works. The SDK will continue to grow, the SDK is available on line if you already have the hardware, but you need to check with the "read me" File to make sure that the hardware matches the application you're trying to use.

Chapter 2: Blackfin Starter Kits

Sub-chapter 2a: What is a Starter Kit?

About the Blackfin Starter Kits: What is the Starter Kit? It's a collection of ADI off the shelf hardware, it comes with the hardware with the supporting extender cards, with the SDK and any other tools you need like VisualDSP++, all designed around a Blackfin processor intended to get you up and running, and just have you focused on your application and your products.

Sub-chapter 2b: Benefits

Let's move on to the benefits of the Starter Kits, if you will. The main benefit is we wanted to use this kit to help shorten your learning curve and development cycle. We wanted you to be able to get up and running very quickly, get you focused on your portion of the product, without having to spend a lot of time learning how to bring up our development boards and our tools. It is a basic building block for Blackfin products. We use all the standard stuff – standard tools, standard hardware, standard platforms – and it's all bundled into one box, comes at an attractive price, the SDK is available for free both in the box and also for download. The VisualDSP tool is only a trial version, and the full version requires a purchase.

The Multimedia Starter Kit is just the beginning of these kits. We already have an Audio kit and many more other kits that are already in the works, so over time look for more of these to come out. The SDK that comes with these kits will continue to grow as more kits are created, so it will always be backward compatible to previous offerings of the kits.

Sub-chapter 2c: Multimedia Starter Kit

A little bit about the Multimedia Starter Kit because that's what we're here for today. The Multimedia Starter Kit comes in two different flavors. We have one that is specifically designed around our Blackfin 561 processor, and one that is designed around the Blackfin 533 processor. They both have the same applications, the same algorithms. From a software perspective, they're the same software bundles in there. They both have the same hardware, extender cards that make the application run. They both focus around Multimedia applications, and they both have the same tools that help you build and run the applications that we provide. The main difference is really the Blackfin evaluation board that comes with either of these kits. One comes with the BF561 EZ-KIT Lite evaluation board, the other one comes with the BF533 EZ-KIT Lite evaluation board. As far as the hardware that comes with them, they both have the Audio Video EZ-Extender card, and that's what we used to capture video and render video content. We also have a USB LAN EZ-Extender card, and that particular card is used strictly in this case for interfacing to a PC, for storing files on the PC, or retrieving content from the PC, as you'll see in our demo.

And the VisualDSP that comes with it is an evaluation version that can be purchased separately from Analog Devices. And finally, they both come with the same SDK software. The software is downloadable online – Blackfin SDK and the evaluation version of VisualDSP++.

Chapter 3: Blackfin SDK

Sub-chapter 3a: What's in the SDK?

Now let's talk about the Blackfin software development kit, the SDK itself. When we sat down to put together the Multimedia Starter Kit, we had to look at what's available and what would be attractive to put into this kit. We put together industry-standard algorithms, and this is the way it's going to go with other kits as well. We looked for things that are available in the open community out there. We have pre-built binary of all those algorithms, they're in the library directory, and if we can provide source code, we'll provide source codes. Some of the algorithms come from Analog directly. The other thing we have is a lot of fully functional non-demo applications. What I mean with that is you take an application you open the project file, you bring it up, you build it, you run it and it works. It does something, it does a function, its not just a demo piece. It's key to understand that everything runs right out of the box. The stuff just works. We provide all the source code we think is necessary, utilities, source code excerpts from various places. The only exception would be if we have IP issues and we can't provide them, we won't. We also provide binaries that we were able to use on the PC to help us in running some of the application. For example as you'll see in our demo later on we'll need something that'll help us provide a view into the PC area so we can open files and move files and be able to render content. We provide all the release notes on the applications that are in the kit. Every application will have its own "read me" file that discusses more particular things about the applications as well as any documents, data sheets, anything that is relevant to running and getting your out-of-the-box experience to be just right.

Sub-chapter 3b: Supported Devices

A little bit about the devices that we support with the Multimedia Starter Kit. As you see from this diagram we support just about everything that's available off the shelf. We try to make our application so that it doesn't require special devices. For example everybody typically has a micro phone or a line in source, video cameras, you have a PC, you can provide speakers, standard video monitors, NTSC or PAL monitors. LCD displays, CMOS censors, DVD players. This is all off the shelf you don't have to purchase anything special to use this Multimedia Kit. Not all applications require all the hardware you just saw in this diagram right here. You need to look at the "read me" file that comes with each application to understand what you need for additional hardware to run that particular application. You also need to look at the "read me" text to also know which version of the Blackfin evaluation boards you need, what's the minimum rev for the processors, minimum rev for the hardware, the extender cards as well.

Sub-chapter 3c: SDK Installation

Let's move on to the SDK installation. I wanted to cover the SDK installation briefly because this is relatively new as a CD and there's a couple of things that are worth mentioning. The SDK does come on a CD and it's also downloadable online – you can go to our ADI website and I'll provide additional information so you can download it directly if you already have the hardware. Double-check with your "read me" file with each application to make sure that the revs match up with the minimum requirement for running a particular application. Since we provide algorithms and things that aren't owned by Analog Devices, we have several license agreements that you have to read and click through if you want to proceed. You won't be able to install it otherwise – even though some of this stuff comes from the open community out there, licensing has to be adhered to – the Motion JPEG and JPEG components, also the Ogg audio components, Vorbis and Speex, and SRGP components to review.

Sub-chapter 3d: SDK Root Folder

Let's talk a little bit about the layout and where it sits on your hard disk in your development PC. Essentially this requires a VisualDSP++ be installed on the PC and VisualDSP++ typically installs in the program files under Analog Devices\VisualDSP 4.5, that's typically the default so I'm going to assume we're going with the default installation. So the Blackfin SDK sits relative to that, it's also inside the program files under Analog Devices, however it has its own sub-directory called the Blackfin SDK. It typically will have a release number to it, for example, 1.00 or 1.1, whatever the case may be. All the applications, algorithms, everything that comes with the SDK is essentially relative to the root of the Blackfin SDK which is this directory that I'm highlighting over here. That's important for our applications because the project files come up and they know how to build everything relative to that root directory. Anything that's coming or being built or linked to, that is provided by the VisualDSP, that's taken care of and usually handled when VisualDSP++ comes up – it sets the environments for you accordingly. In the future when we release newer versions of Blackfin SDKs, the Blackfin SDK will continue to grow, it will still support the old applications and things to do with them. However, we will not install into the same folder. You will have the option to over ride the existing folder, but as a default, we will install to the same area, program files under Analog Devices, but it'll be a Blackfin SDK with a newer rev number attached to it so that it's distinct from the previous installation so you don't lose work you've previously done.

Sub-chapter 3e: SDK Folder Layout

Let's draw our attention to the layout of the software and utilities and stuff that comes with the SDK and how they fit inside the folder on your development PC. As you see from this diagram Blackfin SDK has its own folder under Analog Devices inside the program files. However everything we shipped on this CD has a place or folder that it belongs to inside the Blackfin SDK. Let me start off with the Algorithms directory. What we have there is any algorithm that comes with source code we'll create a project file that you're able to build and create a library for it and it all belongs inside the algorithms directory. Anything that doesn't come with source code will not be there. All the algorithms will have a pre-built binary version of the libraries, and all those will end up being placed inside the Bin directory.

The BFinUtils directory will contain any common utility source code like the ITU 656 for example that's being used by several applications so all those functions will be placed inside the BFinUtils into its own source file.

We have a Documentation directory. What goes into the documents directory is release notes, all the data sheets that we provide specifically for Analog Devices components, as well as any other components that we are able to ship inside the SDK. One point I need to make is on the "read me" text – don't look for them in the documents directory and that's because they belong with each application. Each application has its own unique "read me" text and you should go to the applications directory and read about the application.

In the Include directory is any common includes that are used by more than one app or things that are used by the algorithms and the applications as well. For example, any interface include files for the Ogg Vorbis example would be found in the Include directory right here. Anything that's unique to Ogg Vorbis and has nothing to do with the application would end up going into the algorithm directory where the rest of the source code exists.

As far as the Lib directory, in there what we have is all our pre-built binary, specifically encoders, decoders and things of that nature. So if you're looking for the Ogg Vorbis library that one will be available there, the MJPEG (Motion JPEG) or JPEG library, it will also exist in the Lib directory.

Let me skip ahead and talk about the Tools directory a little bit. The Tools directory contains drivers that we need on the PC, for the example we're going to use today in our demonstration requires that it uses USB interfaces with the PC, that interface needs a driver that you install on the PC side, the host side. That driver exists here in the Tools directory. So do other things similar in nature. Source code also for PC executables will exist there.

Finally I want to talk about the Multimedia directory. This folder contains application software, specifically the source code for all the applications that we provide in the starter kit. New applications that come in future kits, for example if we provided applications on audio, only there will be a directory under Blackfin SDK which will be called "Audio" and in it we'll have audio-specific applications. Look for more of these applications directories to come into existence over time as we release newer versions of the SDK.

Sub-chapter 3f: Apps Folder Layout

One of the things I want to touch on before I move out of this layout is inside the application directory whether it's multimedia audio or whatever else it may be, we have broken it down and we have a sub folder in there, as you see in this diagram the Video In/LCD Out application. This particular application is able to run on a Blackfin 533 and a 561 development board so essentially for each one of these processors and development boards, we have a unique folder for it. It will contain the project file used for the 533 or the 561, any "read me" text specific to the 533 will go there, and the 533 directory right here, and for the 561 will go into the 561. Any LDF files or anything that's unique for the processor goes into the processor-specific folder. All the other source code that comes with this application, for example, Main.c that is common to both versions of the application, will reside inside the "VideoInLCDOut" folder at the same level as these project-specific sub folders. The debug or the release directory, those are the two directories that will contain the object files as we build. When you run the VisualDSP tool it will create those as you select your configuration and it will place them also inside this particular application, in our case "VideoInLCDOut", you'll see another folder here that will be called Release or Debug, whatever your configuration selection was.

Sub-chapter 3g: System Serv/Dev Driver

I want to briefly mention the system services and device drivers and how they fit into the big picture. All our applications that we build and provide in our SDK will use the system services and the device drivers. They are key to better performance, they have common functionalities for embedded system applications. They're simple, they're efficient, they provide easy access, and they provide a common API for our services and the drivers for the components that we use in our development boards. When we look at the source code inside our application you'll see that we use services and drivers that aren't part of our source code, they're shipped with the VisualDSP and they're there to be used.

Chapter 4: MMSK Software

Sub-chapter 4a: Algorithms

Let's turn our attention to the Multimedia Kit Software. Let me start by talking about the algorithms and what's shipped on the Multimedia Starter Kit, SDK portion. What we have here is a bunch of algorithms that are either available to the public community or things we have developed at Analog Devices. So what we have is a JPEG and Motion JPEG encoder, decoder, a codec if you will. We use it in several of our applications, we provide it without sources in this case, however, we do have the pre-build binaries both for debug and release versions for your development. We also provide Ogg components specifically of Vorbis and Speex. They are both encoders as well as decoders. We use them in applications that are shipped as part of this kit. We also have SRGP component and edge detection and other things that over time will go into this kit. But today this is what we have. There will be many more kits that come out and with them will be more algorithms and applications.

Sub-chapter 4b: Sample Applications

Let's draw our attention to the applications at this point. I can't cover all the applications that go into this kit, however, I'm going to talk about a sampling of some of the applications we provide within the SDK. For example, we have stream Edge Detection, something that takes a video input whether from a DVD player or a video camera, we detect an edge and then we put it out on a video monitor. We also have JPEG and Motion JPEG applications. Just like the edge detection, it takes in a video input from either of the devices and puts it onto a video monitor. As well, it takes in files that already have JPEG or MJPEG content on a PC and also decodes them and renders them to a video monitor. We have an application that handles CMOS sensors, same concept. We capture from CMOS sensors and we also display it on LCD display or video display. We have a video capture application, same concept – capture from a video source, either a video camera or DVD player, and render to a video monitor or an LCD display. You use a switch to switch between them. We have Vorbis Player, this plays Ogg Vorbis content that comes from your development PC. We have
Speex encoder and decoder, again this is the Ogg Speex this takes in a line in or a micro phone speech input and then encodes it and decodes it back to a speaker. We have an SRGP example as well and a bunch of other things.

Chapter 5: Demo: JPEG-MJPEG

Sub-chapter 5a: Hardware Installation

Let's turn our attention to the online demo portion of this training module. Today what I want to demonstrate is an application that uses a JPEG and a Motion JPEG encoder and decoder. Essentially we're going to take content that already resides on a PC and we're going to decode them, both JPEG and then later MJPEG to a video monitor. Another piece of this application is that we're going to use this DVD player to capture images and put them into files on a PC, or capture portions of the video clip and also store it to the PC, and then we'll try to play them out on this video monitor after we're done.

Every application comes with its own "read me" file. Before you begin any application read that file, in it we describe what development boards it supports, the rev of the processors, any switch configurations, jumper settings, the extender hardware used for this demonstration, and equipment needed, like a video monitor or video source. Let's begin with our application display, let me change screens and bring up a VisualDSP session.

Right at the beginning when you open up VisualDSP, we're going to go and find our project, so we'll go to either project or a project group, and in this particular example it happens to be a project group. Let me back up a little bit and say that this is inside the program files under Analog Devices Blackfin SDK and now we have everything that comes with the SDK. In this case we're using a multimedia application and we're going to use a JPEG and MJPEG application. Since I have a BF561 development board, I am going to open up the Blackfin 561 project file, let me close this for a sec, and as you see, the first thing I want to do is open up my "read me" text file so that I can read a little more about this application and what I need to do to configure the hardware. So looking at it, you'll see that this one is built for the 561 development platform, it tells you about the hardware that we use. Another important piece of information here is, if you recall, I talked about the system services and the device drivers, this portion of the "read me" file tells you which systems services we're using and drivers and things of that nature. Furthermore it gives you an overview of what this application does, a little bit about how it runs. Then I'm going to skip down and get to the portion of it that talks about the configuration of the hardware itself.

So today I'm going to run this demo here. For it I need the EZ-KIT Lite, the BF561 EZ-KIT Lite board and I need also the USB LAN EZ-Extender card to connect to the PC. I'm going to use the onboard video output for this demo. The switches have to be set as specified right here, I've already preset my switches for this demonstration. Similarly let me turn this board over again and I'm going to show you that there's also a couple of switches here for the USB LAN EZ-Extender and as you see right here, we also tell you what you have to do to configure those as well.

I've got the USB hooked up to my USB LAN card, this wire goes to the PC, that's how we're going to bring in the files, and I've got my video output connector and video input connector – incidentally this stuff also exists inside the "read me" file and it tells you where to connect them and how to connect them. Let me scroll down and show you that. It tells you which connector to use, what the label is, so everything is all set up, now we're ready to run the application.

These applications run without any emulators, but for this demo I'm using an emulator that's hooked up to my USB on my development board. I'm going to select a session by clicking on the session menu and go to a pre-selected session which I happen to have already from before and that's my USB ICE here. If I didn't have one, all I'd have to do is select the "live via debug" agent and this example will run just as easily as that, using the debug agent. If my selection list does not have a session of my choosing then of course I can create a new one. And you should probably review the BOLD training that comes with VisualDSP++ and it'll probably talk about how to select sessions and things like that.

Now that my session is selected, I'm going to click on "Rebuild All" button here on the tool bar, while this is building, you can see in the "Build output" panel you'll see every module that it's building, you're going to see any error codes that occur, any build errors or link errors will also be displayed. This looks like it built successfully obviously because it's asking us to load the project at this point, and we're going to go ahead and select "yes" to load our built binary into the memory of the development board. Let me resize the windows a little bit and close the panel so that I can talk about the source code a little bit. As you know the 561 has two cores, our core B in this case runs in an idle loop the entire time so we're not really using core B in this particular application, so I'm going to change my focus to core A where this demo is running. Looking inside the main function of this application, you'll see at first I call a function called "ezInit." This essentially sets up the cores on my Blackfin processor, clocks and things of that nature. And next as you see right here, it will configure my cache, there's some data and some instructions that I want to put into cache, of course, these are data that have to do with decoding or encoding the content that I'm going to be working with, the buffers of the content I'm going to be working with today. The next thing I'm going to do after I configure my memory, my cache, I'm going to configure my memory as well and finally I'm going to install my system services so that I can use my services and not have to worry about how to do additional work that is provided by the services. Before I begin, I will configure my buffer that's going to be used to pass data over the USB line between the host and the Blackfin board and target board. And then we're going to install the driver and get the application working.

Once we run, we're going to be inside this "while" loop, there's going to be commands happening from the PC side, I'll show you the PC host app that I'm going to be running with this application, and it's going to provide commands that are going to be picked inside this "while" loop and while in here it's going to go and execute the task being requested from the PC host, the application that we're going to be running on the PC side. For the most part, it's going to be doing a lot of work inside the USB I/O start where once it knows whether you want to decode or encode, it's going to go find the files or create a file and then pass data back and forth decoding and encoding or whatever the case may be and then either displaying or saving. We'll talk about it as we're doing things. That's essentially the overall execution for this application.

Now let me hit the Multiprocessor Run key and get the application going at this point. As you see from this pane down here both cores are running, a quick reminder that core B is running in idle and core A is doing all the work. Now I'm going to switch tasks and go into a DOS prompt window and that's where I'm going to run the Blackfin host application that's running on the PC so I can begin to do work with this application.

The host application is called PC2BF.exe, so let me type that. When it comes up, hopefully it'll detect our connected target, and in this case it does, it does see a Blackfin USB LAN EZ-Extender device. It sees that there is a target out there and it tells me that I'm using the onboard video codec as I specified in my build options. So let me now start running this application. The first thing I'm going to try to do is decode some JPEG files that are available on the PC in one of my folders. So I'm going to hit "1" as this application instructs me to do. And as you see, it's started to talk to the USB device and let me turn your attention to the video monitor and it's showing you the first file it's supposed to decode and display to the screen. Now these files are listed into a list file that contains a list of JPEG images to decode and display. So I'm just going to follow the instructions on the screen which essentially says that I should hit the Return key when I'm done viewing and I want to display the next JPEG image.

So let's continue on. I'm going to show you a series of four JPEG image files, and then when it runs through the entire list, it's going to come back and give you the menu to do the next portion of this demo. So as you see, we had four files stored on the PC, we used the USB LAN card that we're connected to here which is hooked up to our development board, and then we used the Blackfin 561 to decode all that content and display it to the video monitor that's attached to our development board. So you see this is the bundle, this is the Multimedia Starter Kit.

Let's change tack and let's do some Motion JPEG now. It's the same concept, those files reside on a PC, we have a list that lists them all, it is going to open the file, read the content of the file and then take in each Motion JPEG file and decode it on the Blackfin processor and then render it to the video monitor that's attached to my right. So let's begin. So let me hit "2" here which is going to essentially get us to render some Motion JPEG file contents. Turn your attention to the screen. This is a simple Motion JPEG that's rendering right now, and again I think it's a series of four files that we tell it to bring over the USB and render to the video monitor. And they're being played in the resolution that they're captured in, that's why the video clip shows this small. This is the same thing as the JPEG portion, we have the files, they're on the PC, we use the USB LAN extender card to bring in the content and then we decode it on the Blackfin 561 and then render it to the video on my right.

The next thing I want to demonstrate is capturing images or clips from a video source. To my right I have a DVD player which is what I'm going to use to capture images and later I'll do the Motion JPEG portion of the capture. I'm going to hit "3" to indicate I'm going to capture JPEG files, we're playing whatever is in the DVD player and now we're capturing. Essentially as I'm being instructed by this application, the Switch 6 on the EZ-KIT development board is the switch I'm going to use to indicate that I want to capture a particular image. And I think it's going to allow me to capture a series of four images. So I'm going to go ahead and hit the Switch 6 which is this switch right here on the board, and it already captured the first one, I'm going to hit Return key to go to the next file, and again I'm going to hit the Switch 6 again and we're going to go through a series of four of those. Now that we're done, those files are stored on the PC, and I'm going to switch context to the Explorer, and you'll see that the files are again inside the SDK folder, if you recall which is inside Analog Devices Blackfin Multimedia where the application is running. The application is the JPEG-MJPEG, and in here, as you recall, we are running the Blackfin 561. We have a folder that we created called the Images, and in here we have a JPEG folder that contains the newly captured JPEG images and you'll see that they're called JPEG Out 1, 2, 3 and 4.

In order for me to show these things I'm going to add these file names into my list of files so that when I rerun this on the command prompt, when I select to display my captured images, I'll be able to show you them directly on the video monitor I have next to me. So let me go to the file which is this one right here, and as you see, I added at least three of the files into this JPEG list text file and now I'm going to go back to my command prompt, and I'm going to select option "1" which is the menu option for rendering JPEG images and hopefully it'll play our captured images. As you see to my right, these are the images that I just captured from the video clip. Pretty nice isn't it?

Now let's try to capture some Motion JPEG content. It's similar to the capturing of JPEG except this time I'm going to start the DVD player and this time when I hit 4, it's going to set itself up and then it's going to tell me to press and release Switch 6 again to start the recording, which I will do. And when I'm done recording, or when I think I've captured enough, I will have to push Switch 6 again to stop the recording. So let's do this a couple of times. So we've just captured the first video clip which is going to be called MJPEG Out 1 and we'll have similar things for 2, 3 and 4. So let me try it again, pushing Switch 6, and then pushing Switch 6 again to stop. And that's Number 2 captured. And now we're going to do the same thing for File 3, and let's do one more I think. So now that we've captured all those files, let's see if you can render them, I'm going to do the same thing I did with the JPEG captured images, I'm going to go and update my list so that this application when I hit menu option two so that it can play out my captured Motion JPEG files. So here we go and there's the file list, I've added a couple of these files in there so we can play them – AV Out 1, 2, 3 and 4. Let's go back to our DOS prompt, and let's select two which essentially plays out what's in the list. And as you see to my right this is captured directly from the DVD player, another one, and let's do one more, it's pretty good looking video isn't it? And that's that, that's essentially the end of this demonstration.

And let me summarize a little bit what we tried to do here is show you that we can take already existing content from a PC, read them, decode them, and then display them on an attached video monitor. Similarly we were able to capture video input and encode them and put them to a PC file. As you see, we used boards that come already bundled into our Multimedia Starter Kit and it was software we just brought up, built and ran.

Chapter 6: Conclusion

Sub-chapter 6a: Summary

Let me go back to my slides now. In conclusion what I'd like to repeat here is the fact that our starter kits is a bundle of readily available off-the-shelf hardware, software tools, applications, all bundled together, built around the Blackfin processors. In it we also provide you a Blackfin SDK which contains source code, release notes, documents, data sheets, algorithms, utilities anything you need to bring up application and just run and work with the Blackfin development boards. The benefit is the fact that they are a basic building block. They're meant to shorten your learning curve to get you started fast, and just get you to a product, to your end product right away. It let's you focus on things that you need to focus on and not worry about how to use our development boards and our extender cards and our tools.

A key point I wanted to mention is that the out-of-box experience should be the "stuff just works." We want you to just be able to open up, put the hardware, open up an application, look at the "read me" text, configure the hardware, bring up the application, build it and run it and it just works. And finally looking forward from here, there's going to many more kits coming out, this is just the beginning. We have multimedia, we have audio and there's many more coming. The Blackfin SDK is going to continue growing as it is today, it's going to support the older things that have come before it, and it is available online for a download if you already have the hardware. But you need to read the "read me" text to make sure that your hardware is the minimum required for our particular application.

Finally, I wanted to provide some additional information so that you can look into other things we provide. If you want to find out more about Blackfin Start Kits, we have a link which you should go to – it has other starter kits and more info on them. We have a link for the Blackfin SDK, a link for the VisualDSP 4.5 or later. We also have a BOLD training for VisualDSP, we have a link for the system services and the device drivers, and we have BOLD training modules for those as well, I recommend them highly. We have our schematics and layout, they're also available on line, and we have other BOLDs like I've already mentioned.

Sub-chapter 6a: Additional Information

This brings me to the conclusion of this module presentation. If you have any additional questions, please click on the 'Ask a Question' button and thanks for your time.