

By
JB Fowler,
Xiangdong Fu and
Rishi Bhattacharya

Introduction

Digital video promises to revolutionize how we interact with electronic products and with each other. The list of consumer and embedded applications that can derive additional value from adding video functionality is endless. For example, users will be able to interact with video as they would image and audio files, opening the doors to a wide variety of video-based services that will provide innovative new revenue streams for service and content providers, including video-on-demand and localized advertising. Advanced surveillance systems will be able to capture video and process it in real-time, providing automated object tracking and event notification capabilities, while improving reliability of service. Video-based medical devices could guide nurses and doctors through simple and even complex procedures more effectively than text instructions by showing them what they need to do rather than telling them.

While there is no question that video increases the potential of such applications, developers have shied away from introducing video capabilities because implementing digital video has been an extremely complex process. Developers have had to spend months familiarizing themselves with multimedia standards that are constantly changing before they can even begin algorithm development. Meanwhile, existing digital video implementations are often tied to a specific platform and operating system, and developers have to make significant design investment through hand-tuning of code.

In short, implementing digital video has been a complex, time-consuming and expensive process, making it unfeasible for the majority of consumer and embedded applications.

All this changes with DaVinci™ technology.

The DaVinci™ Effect: Digital Video Without Complexity

More Than Silicon

To address the call for more capabilities and higher performance demanded by OEMs, Texas Instruments continues to build upon its leading-edge digital signal processor (DSP) technology, offering developers more performance per MHz and per dollar. However, OEMs also face challenging design barriers, such as compressed time-to-market and limited development resources. For more complex processors to be of value to developers, they must be easier to use than their predecessors. Digital video is a complex application and TI recognizes that developers need more than just world-class silicon.

Based on the DaVinci technology, TI is offering the next level of product development tools and support. Its first processors, the TMS320DM6446 and TMS320DM6443, are truly innovative DSP-based, System-on-Chips (SoC) with integrated coprocessor engines designed to specifically accelerate digital video applications. DaVinci technology, however, encompasses more than the traditional suite of development tools and applications support, as it extends to include the base software developers need to implement digital video quickly and easily.

The goal of this strategy is to bring digital video to the component level, enabling developers to introduce digital video to an application without requiring developers to program a single line of DSP code. This is possible through the use of application programming interfaces (APIs), which reduce design complexity by allowing developers to implement digital video through APIs rather than having to hand-code and optimize assembly-level code. In other words, DaVinci technology offers developers access to the latest innovative and high-performance digital video capabilities with a minimum of design investment.

What is Challenging About Digital Video?

One of the major design challenges for developers today is determining exactly what is meant by digital video. For example, digital video can be encoded in a number of different formats:

- MPEG-2
- MPEG-4
- Windows Media Video (WMV)
- DivX
- H.264
- H.263

Digital video can be stored using a variety of mechanisms:

- nonvolatile memory, such as removable Flash
- local hard drive
- remote server
- fixed media, such as DVD or VCD
- remote device, such as a camera or personal media player

Digital video can be accessed in several different ways:

- fixed file
- broadcast media
- real-time network stream
- non-real-time stream
- wired networks
- wireless networks

Content can be manipulated in multiple ways, so that it can be:

- scaled to a multiple resolutions
- transcoded to a different CODEC for playback on an otherwise non-compatible device
- stored in nonvolatile memory or on a hard drive for later playback
- analyzed using object identification and tracking techniques for real-time security applications

The ideal applications that incorporate digital video should be able to support video under many of these variations because users don't want to be limited in where and how they interact with video. In fact, such details should be transparent to users.

While users may not want to be bothered with how a particular video stream is sourced, it is critical from an implementation perspective that the digital video device understands the differences between various video sources and destinations because

each of these mechanisms has different strengths and vulnerabilities, which have a direct impact on quality and how the video stream is optimally handled. For example, video transported over a network needs to be able to accommodate jitter and latency without affecting video quality. Video sent wirelessly is vulnerable to packet loss and must also be able to dynamically adjust the data rate based on existing network conditions. Broadcast video received over an antenna must have error correction mechanisms in place because there is no means for retransmitting corrupted frames. Even the straightforward process of retrieving video from a DVD must account for scratches on the surface of the DVD that can lead to unacceptable frame freeze and endless loopbacks.

The nuances involved in each of these scenarios can become quite complex, and implementing them is not trivial. And yet, users expect each of these challenges to be overcome without question and don't consider the accomplishment of these technical feats to be a product differentiator. Today's developers don't have the bandwidth to be bogged down by all the implementation details necessary to cost-effectively accommodate each of these challenges.

The Power of APIs

This is where the power of abstraction through application programming interfaces (APIs) comes into play. APIs enable developers to focus on application development without having to spend a great deal of time thinking about specific implementation issues by abstracting implementation details from application use. For example, from an application standpoint, all that a developer cares about is accessing the video stream. Through an API, this can be accomplished with a function call as simple as `GetVideo()`. In this case, all of the complex details involved in retrieving the next frame of video are encapsulated within the function `GetVideo()`.

Note that none of the hard work is actually avoided. Certainly, the first task `GetVideo()` must accomplish identifying where the video is being sourced and then diving into all of those complex implementation details. What an API does is mask complexity at the application layer. In this way, an application can access a video stream from a great variety of sources, despite their differences in implementation, without having to make any major changes to the application code.

This is one of the key value propositions of DaVinci™ technology and the integrated components. By supplying the necessary software and API infrastructure, DaVinci APIs enable developers to begin to approach digital video as simply as a function call. Developers no longer need to concern themselves with the details of CODEC implementation. Rather than spending months of development time designing algorithms and then

optimizing them for a particular platform and application, developers can utilize production-ready drivers and CODECs and shift their focus and development resources to application-level software where their true value-add lies.

APIs also provide critical insurance for OEMs towards protecting their code investment. As OEMs successfully bring products to market and gain market share, they will want to expand their product lines across various price points and out to new applications altogether. Ideally, they need to be able to leverage their existing application code base in these new products.

APIs enable a smooth transition across products by increasing the portability of code. By separating actual hardware implementation details from their use in applications, developers are able to port application code to new platforms and operating systems more easily; instead of having to rewrite an entire application, porting is primarily concentrated around the drivers behind the APIs.

Not only can developers reach market faster in their initial and subsequent designs, they are able to effortlessly keep up-to-date with ever-changing and evolving multimedia formats. TI and its partners will ensure that DaVinci™ drivers and CODECs are based on the most current standards, giving developers the confidence that their designs have the greatest competitive edge by being able to support the latest and most popular multimedia formats.

Again, because the implementation details of these formats are encapsulated under the APIs in low-level drivers, developers do not need to concern themselves with them. In fact, developers never need to learn to program a DSP or understand the intricacies of the hardware-based video acceleration engines of DaVinci devices to take full advantage of their high-performance capabilities.

Building Upon Linux

One key factor DaVinci technology brings to simplifying implementation of digital video is building on existing standards and processes that are already in place. Instead of requiring developers to become familiar with an entirely new set of proprietary APIs, TI has based DaVinci technology on Linux drivers that are already familiar to Linux developers. Currently, DaVinci technology supports Linux 2.6.10 based upon the MontaVista Professional Edition from MontaVista, an industry-leader for Linux software, support and development tools.

From an interconnect perspective, standard communications operations, such as open, close, read and write can be performed using standard API calls. This enables developers to access a wide range of peripherals and video sources in a simple and straightforward

manner which shields them from having to develop these interfaces from scratch. Additionally, for storage-based peripherals such as NAND-, CF-, MMC- and ATA-based devices, developers can access data via the file system interface, further simplifying digital video development.

The Linux-based video drivers for DaVinci™ technology give developers complete access and control to video functionality without requiring developers to become digital video or DSP experts. Through APIs, developers are able to configure the hardware-based video engines to match the specific needs of a particular application without involving complicated programming. For example, developers can set parameters to adjust image size, bits-per-pixel, refresh rate and color map palette, as well as manage frame buffers and DaVinci processors' built-in on screen display (OSD) capabilities (which enables developers to overlay text and graphics over video as well as multiple video streams for picture-in-picture functionality). The audio drivers for DaVinci technology also provide a wide range of capabilities including record and playback, multiple sample rates (8 KHz, 16 KHz, 22 KHz, 44.1 KHz, 48 KHz and 96 KHz), among others.

Again, the complex details behind implementing each of these features are handled in low-level drivers that are accessed through common Linux APIs and thus are completely transparent to developers. The computational resources of the TMS320DM644x devices are also implemented in an optimal fashion without any programming required by developers. This includes complex operations such as optimized utilization of DSP resources and hardware-based acceleration engines, use of the Enhanced Direct Memory Access peripheral (EDMA), in chained mode for more efficient data transfers, and packet processing in Interrupt versus Tasklet modes to flexibly meet different application requirements.

Real-World Video

The following are a series of examples showing just how straightforward and simple implementing digital video is with DaVinci technology.

For example, consider a home media gateway designed to enable users to playback video from a variety of sources – PC connected over Ethernet, camera connected over USB 2.0 and file storage on a hard drive connected over ATA. Because DaVinci technology is based on standard Linux drivers, once a video stream is opened, the application code for servicing video data is as simple as a file transfer in all of these cases:

Example Pseudo Code for HDD to Display of H.264 Video:

```

InitPeripherals()
    InitEthernet();
    InitUSB();
    InitATA();
    InitDisplay();
    InitH264Decoder();
OpenCodec()
    OpenH264Channel();
StartDecode(ATA.Address, H264.Channel, Display.Address);

```

That's all the code it takes to access video data from any source. As important, the same code can be used in all of these cases, making application code flexible and versatile.

Such flexibility is also important when determining the destination for digital video. Consider a set-top box that connects directly to a home theater monitor or PC screen, or can download content to a personal media player. Each of these displays has a different resolution and may not support the original format in which the video was received. Again, once the video stream has been opened and configured, the application code is straightforward:

Example Pseudo Code to Change Display Size:

```

InitPeripherals()

// Display driver is passed arguments to move away from
// default values
InitDisplay(QCIF | CIF | D1 | 720p | 1080i);

```

Advanced video features are also simple to implement because of DaVinci™ technology's use of industry-accepted APIs. Time-shifting, for example, enables viewers to temporarily pause playback of a real-time video source while they get a drink from the kitchen or answer the phone. Effectively, time-shifting requires that the real-time video is stored to disk rather than displayed immediately so that the user doesn't miss any of the program. When the user resumes playback, the device must continue to store the incoming video to disk while it plays back the video it has already stored (i.e., simultaneous encoding and decoding of video). As before, the application code is simple and intuitive:

Example Pseudo Code for Pause and Resume Video

```
[Action] Pause Button is pressed
OpenCodec()
    OpenH264Channel;

//Start compressing and storing Incoming data from Ethernet
to HDD
StartEncode(Ethernet.Address, H264.EncodeChannel,
ATA.Address);

...

[Action] Resume Button is pressed
StartDecode(ATA.Address, H264.DecodeChannel,
Display.Address);
```

Programmable Flexibility

The foundation of DaVinci's flexibility comes from its programmable architecture. The TMS320DM644x, for example, is a dual-core ARM[®] plus DSP architecture. Developers program their application code using established and reliable open source ARM tools. No coding is necessary for the DSP as it is loaded with "DSP-executable" code that the developer accesses transparently through APIs in the application code.

The advantages of a programmable architecture are substantial. While ASICs provide a complete implementation, they can only be used in a single product. Additionally, if the specifications of the product ever change, the ASIC must be redesigned, which is an expensive and time-consuming proposition. Alternatively, while FPGAs provide enough programmable processing capacity, they offer a far from complete implementation and actually introduce significant design inefficiencies since developers must work in a second and independent hardware development environment.

DaVinci™ technology provides the best of both approaches. Because the architecture is programmable for both the ARM and DSP cores, developers have the ability through code and driver updates to keep systems current and optimized. Because TI and its partners supply the necessary API code and drivers, the digital video implementation is complete as well.

With the programmability of DaVinci™ processors, the IP developed around a single application can be reused. With an ASIC, separate hardware is required for each CODEC the system supports. Due to the programmability of the DM644x, the same hardware resources can be dynamically reconfigured to support whatever CODECs are currently required, thus conserving and maximizing hardware resources to keep system cost as low as possible.

The DaVinci Effect

With the accessibility of cost-effective silicon, digital video will quickly become a product-differentiating feature across the consumer and embedded electronics markets. To stay competitive, developers will need to be able to introduce digital video capabilities to their designs as quickly and in as efficient a manner as possible.

While leading-edge silicon based on SoC technology is an essential ingredient in bringing digital video to market, without the right software tools and comprehensive, integrated software, developers will find themselves dedicating significant design resources to the overwhelming task of developing basic digital video functionality instead of focusing on adding their own value at the application level.

Through high-level Linux support, industry-standard APIs and production-ready low-level drivers, DaVinci technology eliminates the complexity of designing digital video systems and offers a complete approach to digital video, providing the necessary computational capabilities without unnecessary complexity. With its programmable dual-core architecture, devices based on the DM644x processors can be easily designed and kept up-to-date, enabling developers to create reliable, portable application code quickly and without having to spend months hand-optimizing CODEC implementations.

DaVinci technology promises to change the way developers think about digital video. No longer is video a major undertaking that involves months designing an entire subsystem from scratch. With DaVinci technology, digital video has become a matter of integrating yet another system component, pure and simple.

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

DaVinci is a trademark of Texas Instruments. ARM is a registered trademark of ARM Limited.

All other trademarks are the property of their respective owners.

B091905

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265