MIPS Solutions for GoogleTV

This document describes MIPS processors and the GoogleTV operating system.

Document Number: MD00789
Revision 02.05
November 5, 2010

Contents

Introduction

For the better part of a decade, organizations ranging from game console makers to DVR manufacturers to cable and phone service providers have tried to stake a claim as the digital-living-room hub. To date, no one has ascended to the throne. Google is now stalking the living room with the GoogleTV platform set for deployment in late 2010, initially on only three OEM platforms.

Google has leveraged its success in other markets to bring Android forward in the market – now the number one operating system for smartphones. This has extended Google's already strong brand into the mobile Internet device space. The result of this move towards dominance in a new market significantly increased the number of users of Google's prime revenue source, its search-based businesses. Android's success comes from Google's commitment to keep the software within the open source community, and freely available to OEMs.

Given the market and application development community's acceptance of Android, Google decided to base GoogleTV on Android. In doing this, the company aims to repeat its success in the mobile phone market with an Android-based TV platform that serves as the hub for digital media and entertainment. Google is seeking to capitalize on an obvious end user need: whether consumers care about the Google brand or not, they do need a better way to organize, control, and view content that increasingly comes from multiple Internet, local network, and broadcast sources.

The stated goal of GoogleTV is to provide not only a platform that satisfies this need but also, leveraging its search capability, a platform that provides the broadest content search/organizing capability within a compelling, intuitive user interface. Google also seeks to provide an advertising platform that keys into all of the strengths of the GoogleTV platform. Therein lays the business model behind GoogleTV: Google is reaching out to gain a revenue stake in the $70B advertising market created by today's terrestrial, cable and satellite TV networks.

Ultimately the GoogleTV brand will equate to a technology platform embedded in TVs and set-top boxes along with services and applications designed for that platform. The details of GoogleTV are still under development, but we already know quite a bit about the plan. The platform will rely on the Android™ Linux-based operating system that Google has succeeded with in the smartphone market. The platform will make use of the Chrome web browser and application platform. GoogleTV will support a broad range of interactive multimedia capabilities including a host of audio and video CODECs, and hardware-independent graphics and multimedia platforms such as Adobe® Flash® Player 10.

The MIPS architecture is well positioned as a platform for GoogleTV. MIPS Technologies has dedicated engineering resources to ensure the latest releases of Android can be optimized for MIPS-based SoCs. MIPS has also worked with an entire ecosystem of software provides to make sure all the components of the software stack required for GoogleTV run and are optimized for peak performance on the MIPS architecture. In addition, MIPS has made sure that all the key development tools and associated software libraries are available for MIPS Android and hence for GoogleTV.

MIPS silicon vendors have a natural jump start on the market for GoogleTV: MIPS-based solutions currently dominate the digital home with the leading share of DTVs, set top boxes and Blu-Ray players. The MIPS architecture has become the de facto standard as an applications processor in this market. Hence, silicon vendors can leverage this market position with their consumer electronics partners to swiftly build a leadership share in the emerging GoogleTV market.

MIPS will provide extensive support for its licensees as they move to bring forward solutions for GoogleTV. Our work with the Android operating system, the software stacks needed for GoogleTV and our ongoing relationship with Google will enable our licensees to quickly enter the market as GoogleTV moves beyond initial reference platforms and into mainstream development within the digital home market.

By designing your “System-On-Chip” (SoC) to the right specifications now, you can be prepared for GoogleTV as Google opens up platform opportunities beyond its initial launch partners in the summer of 2011. The goal of this paper is to help our licensees get a jump start on moving into the GoogleTV market by providing a preliminary description of hardware requirements and recommendations for developing a SoC that will support the GoogleTV operating system.

What is GoogleTV delivering for End Users?

In May 2010, Google announced “GoogleTV”, an open platform that merges and presents multimedia sources/sources on a single, easy to use platform. This effectively combines Internet-based multimedia interaction with terrestrial and satellite television channels and is intended to present the result in a cohesive, intuitive user experience. With GoogleTV, end users can search and watch a large array of content available from a variety of sources including existing television content providers, the web-based services, their personal content libraries and even content generated from or accessed through mobile applications.

GoogleTV is built on top of Google’s successful Android operating system. It includes out-of-the-box applications for multimedia location, display and recording. The platform incorporates the Google Chrome browser as the basis of its multimedia browser. Not only can users can access any desired terrestrial/satellite televisions channels as well as Internet and cloud-based multimedia content/information. Google intends to implement sufficient tagging and taxonomy to allow users to search for information or additional content whenever they decide to supplement whatever their GoogleTV appliance is currently delivering.

While GoogleTV will be delivered with a comprehensive set of multimedia content capabilities, Google will be encouraging the development of 3rd party applications for the platform. These will available through the Android Market. Applications will have access to the various multimedia services, searching capabilities, control APIs, etc. that GoogleTV. The result is expected to be one of the more complex computation environments and thus will be sensitive to overall hardware performance capabilities – which the MIPS architecture is fully capable handling, given its historical use in similar embedded platforms.

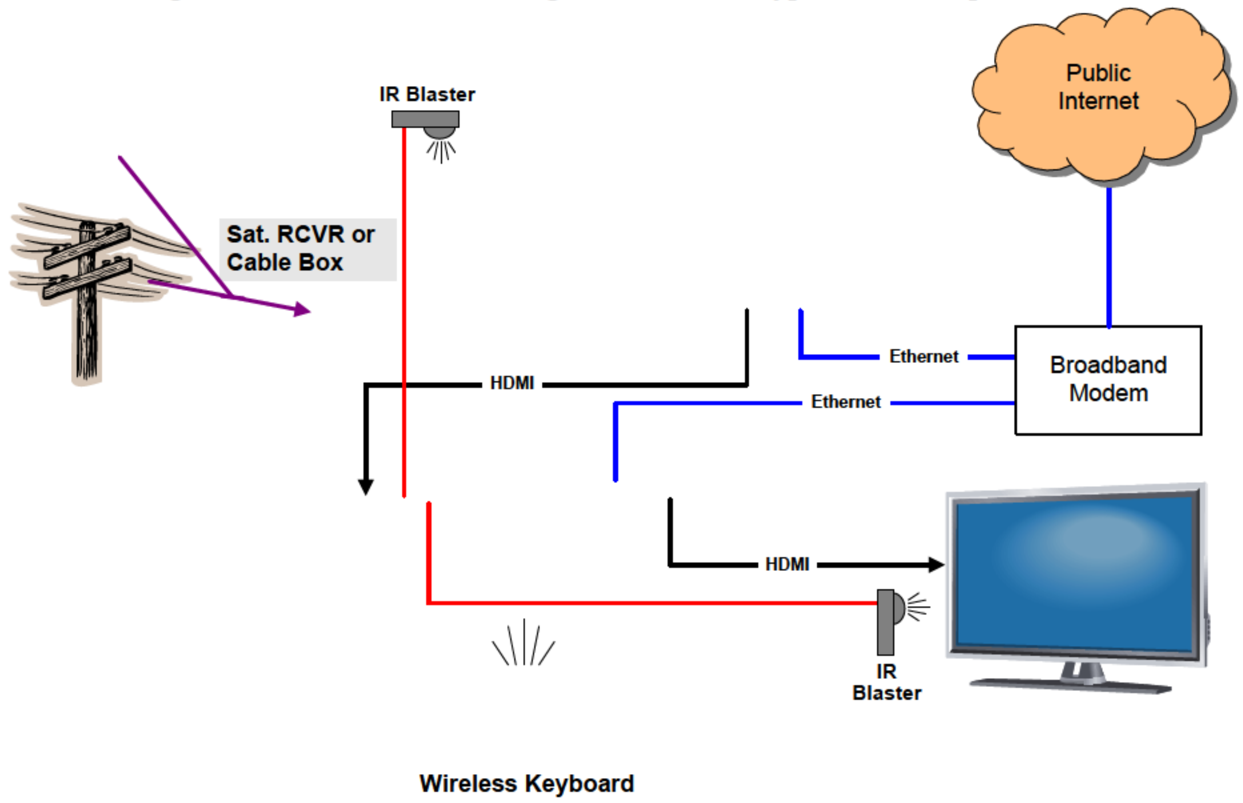
To navigate the content, GoogleTV will have an integrated search capability to help viewers easily find relevant content across over-the-air and pay-TV channel listings, DVR, and the Internet, as well as a picture-in-picture layout to access multiple windows simultaneously. GoogleTV will also have a home screen to help viewers quickly organize their favorite content and personalize their TV viewing experience. Some of these features require advanced integration with the cable or satellite hardware.

Finally, Adobe Flash Player 10.1 will be integrated directly into the Google Chrome browser on GoogleTV, enabling viewers to experience tens of millions of web pages with rich Flash content including games, animations, applications, videos, audio and more.

Later in 2010, Google will release an updated Android SDK which supports applications built for GoogleTV. In 2011, Google will release a set of TV-specific APIs for web applications, encouraging web developers to begin building unique web applications for use on television sets.

Google will make GoogleTV "open source" to help spur innovation in the industry and so that other developers can benefit from the project.

The following illustration shows how GoogleTV fits into a typical TV set up:



The Software for GoogleTV

GoogleTV consists of a sophisticated software platform that not only manages the underlying hardware but also presented the end user with a cohesive user interface and an extensible set of powerful applications. The overall software architecture for GoogleTV may be pictured as follows:

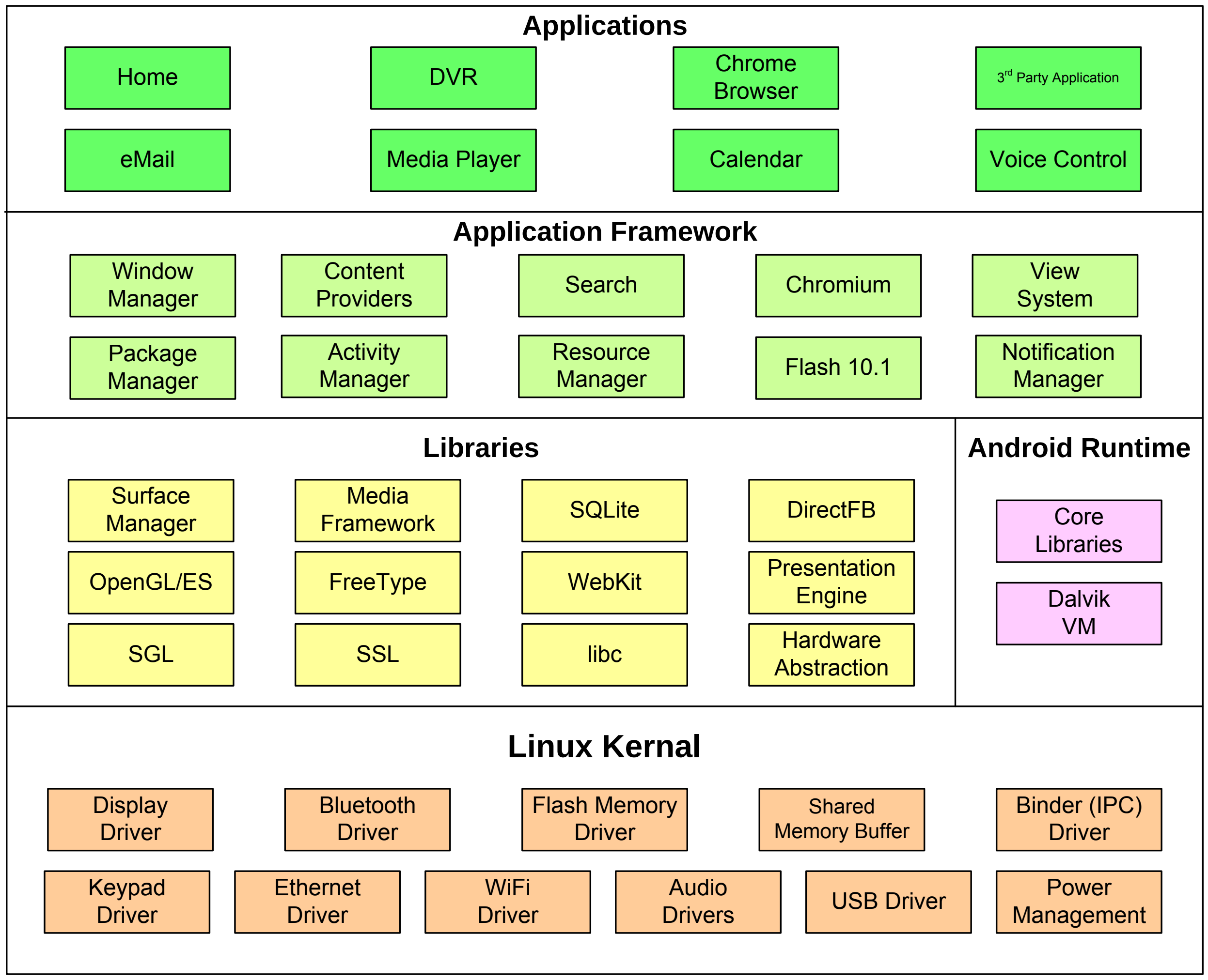


Figure 1 GoogleTV Operating Environment

We will now look more deeply into the implications and requirements of for some of GoogleTV's more critical software aspects.

Android™ Overview

Android is a software platform created by Google designed as a basis for network aware platforms such as mobile phones, tablets, set top boxes, digital televisions and other consumer information appliances. It is based on the Linux Kernel, and includes a number of software libraries written in “C” and a Java virtual machine for running application code. It typically is made available with a full suite applications written by Google and 3rd parties that complete the platform for a given environment. For example, the current release of Android for mobile phones comes complete with a software stack for handling cellular radios, making/receiving phone calls, instant messaging, managing a GPS interface, push-base email, contacts, calendar, etc. – i.e. all of the expected mobile applications.

The Android software platform is available free from Google and the source code is available under a GPL open source license.

Android applications are intended to be entirely written in Java. Google provides a plug-in and libraries that can be used with the popular open-source Eclipse integrated development environment. Developers can use Eclipse to write Android applications and seamlessly debug them using a PC-based Android emulator. These applications are distributed as compiled to Java bytecode and run on the host hardware using the Android Dalvik Virtual Machine (VM).

Android goes to great length to ensure that applications move freely between supported hardware architectures[1].

Chrome Overview

Google Chrome is a web browser developed by Google that uses the WebKit layout engine and application framework. A large part of the GoogleTV platform is delivered using Chrome as it forms the basic Internet/web platform for GoogleTV. Chrome is also used by GoogleTV apps – many of which are really Chrome-based JavaScript applications that link the client with remote web services or websites.

Google released a large portion of Chrome's source code, including its V8 JavaScript engine, as an open source project entitled Chromium. Google Chrome aims to improve security, speed, and stability. The strength of Google Chrome is its application performance and JavaScript processing speed. The JavaScript virtual machine used by Chrome, called the V8 JavaScript engine, has features such as dynamic code generation, hidden class transitions, and precise garbage collection[2].

It comes as no surprise that the performance of Chrome and Chrome-based JavaScript is crucial to a successful GoogleTV platform. This directly translates into processor, GPU and memory requirements.

[1] Android software written to a specific platform is considered bad practice and is not encouraged. They may cause compatibility, performance and stability issues.
[2] Precise garbage collection is a form of automatic memory management used to reclaim memory occupied by objects that are no longer used by the program.

Adobe® Flash® Player 10.1 Overview

Adobe Flash is a multimedia platform used to add animation, video, and interactivity to Web pages and web services. Over 75% of all multimedia web content today is delivered using Adobe Flash. Thus, Adobe Flash is essential to GoogleTV: it is used extensively to deliver web-based video content onto the platform.

At its simplest level, Adobe Flash manipulates vector and raster graphics to provide animation of text, drawings, and still images. More importantly it supports bidirectional streaming of audio and video. Thus, Adobe Flash generates major performance requirements for GoogleTV hardware platforms in order to achieve an acceptable user experience:

In terms of streaming to the Adobe Flash client, the platform needs sufficient performance to support high frame rate full frame HD format video. At a bear minimum platforms must support a single HD stream while handling other elements of user interaction such as user generated input via mouse, keyboard, microphone, and or streaming video/still images from an attached camera without any risk of dropping events or frames. More advanced GoogleTV platforms may be streaming multiple Internet-based Flash streams while also displaying HDMI/composite video sources.

Adobe Flash contains an Object-oriented language called ActionScript. The key features include support for hardware acceleration that improves graphics performance on a range of devices supported under the Open Screen Project.

Hardware acceleration capabilities fall into two categories:

1. H.264 video hardware decoding on supported devices.
2. Hardware (GPU) graphics rendering on supported devices.

H.264 video is decoded in Flash Player 10.1 by a GPU or media accelerator. Some hardware decoders can accelerate multiple H.264 streams concurrently. The amount of acceleration depends on the specifications of the decoder. If a hardware decoder cannot handle an additional stream, Adobe Flash Player will decode that stream in software.

There are many different options for encoding H.264, and support for specific formats varies with different hardware decoders. However, in general, digital video decoders support H.264 baseline, main, and high profiles. These profiles are defined as follows:

- Baseline Profile (BP)
  Primarily for low-cost applications that require additional data loss robustness, this profile is used in some videoconferencing and mobile applications. This profile includes support for loss robustness (or for other purposes such as low-delay multi-point video stream compositing).
- Main Profile (MP)
  This profile is used for standard-definition digital TV broadcasts that use the MPEG-4 format as defined in the DVB standard. It is not, however, used for high-definition television broadcasts.

- High Profile (HiP)
  The primary profile for broadcast and disc storage applications, particularly for high-definition television applications (for example, this is the profile adopted by the Blu-ray disc storage format and the DVB HDTV broadcast service).

In addition to hardware H.264 decoding, devices that support access to OpenGL ES 2.0 can use Adobe Flash Player 10.1 to render all graphical elements (except Pixel Bender shaders) using GPU hardware to increase graphics performance. GPU mode is set automatically when running content on devices for the best possible performance.

User Interface Overview

GoogleTV expects its hardware platform to support a true 3 dimensional user interface. It stacks items on the screen in 3D with transparency. Interface elements are often 3D in nature and may be animated in 3D.

Thus, even though a GoogleTV platform primarily displays what are 2D video and sill image content elements, it also must be capable of handling advanced 3D user interface graphics – including making a 3D user interface appear as a semitransparent layer over a video stream or other application generated window.

The entire interface can be set up to run in a variety of interfaces and aspect ratios. The user gets to set up what works best for his/her display device. All GoogleTV devices need to support full 1080p resolution devices in 16x9 aspect ratio.

The interface itself consists of menu buttons and either icons or thumbnails representing available content or functions. GoogleTV leverages the multitasking, multithreaded capabilities of the underlying Android platform. By clicking on an interface element, a user launches the associated Android application and multiple applications can be active at any given moment.

The “back” and “home” functions generally take the user back through application windows/states or, in the case of the “home” functionality, to the top level GoogleTV homepage as last viewed by the user, in which case the most recently viewed application is still visible and executing under a semitransparent home screen.

In the initial Android platforms, switching to another application generally kills the most recently running application, particularly if each application needs to display a video stream. As platform performance improves, this restriction may change as the underlying Android platform is capable of keep multiple applications active in parallel.

There are also applications which the user may start and relegate to background processing – such as the DVR functionality. These do not impinge on the user interface but do impact performance requirements.

The bottom line is that GoogleTV requires 3D capable hardware and benefits from shading support and other functional enhancements in the GPU.

The Hardware for GoogleTV

Overall Hardware Requirements

The differences between a traditional digital video based consumer electronics product and one that is prepared for "GoogleTV" lie in the hardware and software necessary to unify multimedia viewing, associated applications and generic web browsing. The key to success is providing a user experience that never fails, never stutters and presents a coherent environment which does not overwhelm or confuse the user. Since GoogleTV is intended for consumer electronics platform, the implication is that a GoogleTV has to run with PC/MAC like performance but with the reliability and ease of use of an appliance.

The starting point for implementing a GoogleTV platform is the typical platform used today for smart/digital TV. The following block diagram shows an example smart TV architecture.

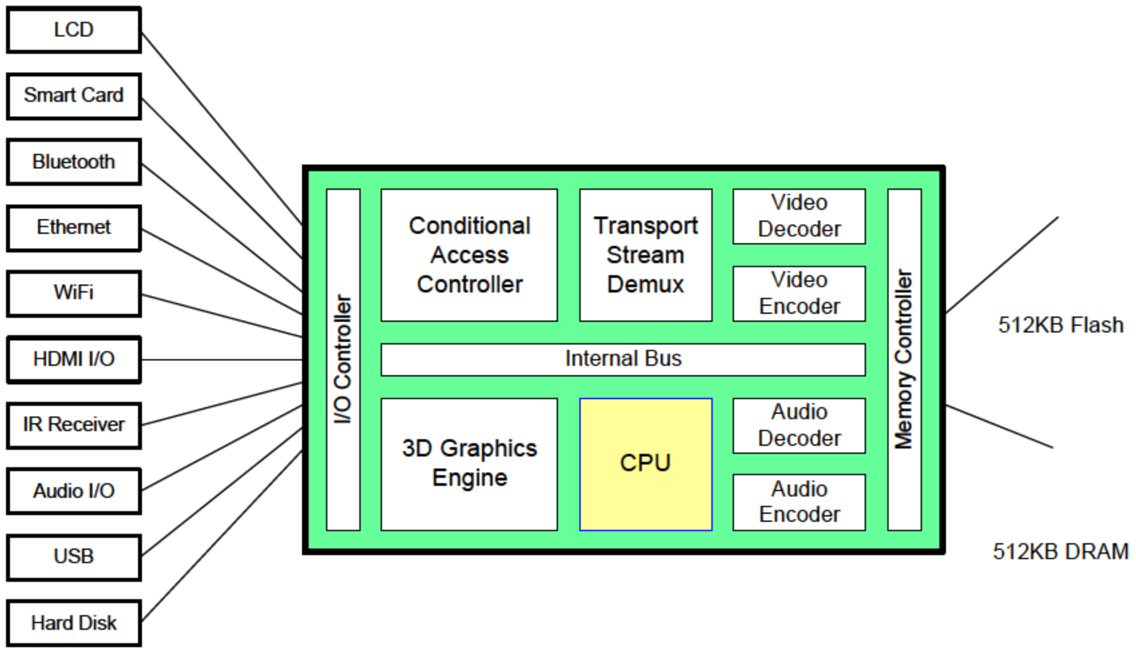


Figure 2 Smart TV Hardware Requirements

These architectures need to be reviewed and modified based on the performance and connectivity requirements of GoogleTV.

The range of performance demanded of a GoogleTV platform is not the only difference from past DTV hardware requirements. It is also necessary to examine characteristics of the higher levels of software running – i.e. the browser and the applications. These need to perform flawlessly: no hesitation, no dropped frames – in short, no user visible issues in viewing/listening experience while maintaining a highly available interactive capability that can be delivered immediately the user requests such activity.

This means the hardware – either through raw single/multicore processor power or with hardware assist – must deliver a number of flawless user experiences and reliable connectivity, for example:

- Video recording with simultaneous display of the same or different video content
- Multiple video streams visible on screen at once (low end: two; high end: six)
- Responsive web browsing
- Flawless Flash 10 performance
- Fast application activation – which means excellent Java performance
- Context saving so that users may move back and forth between applications
- Fast performing 3D graphics for the user interface
- Fast startup from power off
- Low power consumption in standby mode
- Support for all the typical connectivity standards in the Digital Home:
    - 802.11 wired and wireless network connectivity
    - HDMI
    - Potentially legacy video standards such as S-Video and Composite Video/Audio
    - USB to extend platforms that do not possess sufficient flash memory for adequate DVR functionality

CPU

In previous generation (i.e. non-GoogleTV) digital video products the CPU may have run a small operating system and was primarily used to manage a 2D user interface, generate on screen displays and other “light duty” miscellaneous tasks.

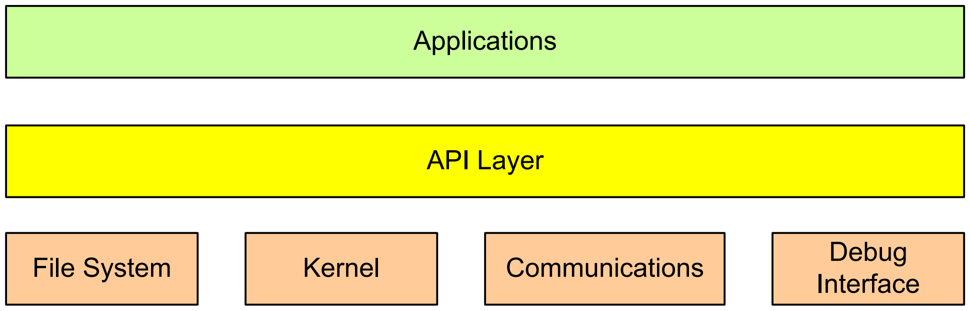


Figure 3: Traditional Digital Video Operation Environment

With smart TV products such as GoogleTV, the CPU is running a much more capable operating system and set of software stacks. For GoogleTV this consists of Google's Android operating system, essentially a Linux Kernel, a java virtual machine, several libraries for application development, an application framework and several default applications. In addition to the base Android components, GoogleTV adds the Chrome browser and several standard applications specifically designed for the TV user experience.

These applications include light applications, such as a wrapper for Amazon's Unboxed video streaming service, and more capable applications, such as a complete NetFlix client, media player/DVR and Twitter Client. This provides a significant workload for the main processor and, as GoogleTV development matures, opens the possibility for hardware accelerators, including graphics processing, transcoding and CODECs.

Another significant difference between traditional digital TV (DTV) products and GoogleTV based products is the dynamic and large variation in potential CPU workload.

Traditional DTV product are designed around relatively constant performance demands with little variance in load due to user interaction with the device. In addition, the software for typical DTVs is determined by the manufacturer and therefore it's range of platform workload was well characterized, allowing processor, memory and any additional components to be exactly sized to the meet those requirements. The worst-case system workload for traditional DTVs was typically based on the limited functions the DTV. It was coupled with a limited set of 2D user interface capabilities, with only a subset of which would be enabled or visible based on the central activity of the DTV at any given moment in time.

GoogleTV based platforms inherently possess a much wider range of demand for system performance. For example, the worst case CPU workload may occur when a user is viewing internet streaming video content where there is no native hardware decoder available in the platform and so requiring the CPU to perform software decode. At the same time, the user may be viewing broadcast TV and/or running a Java application that is helping the user gather information on one or both video stream – or even managing two way communication between the user and a remote user, or group of users.

This example is intended to show the largely un-predictable nature of the workload that is typical of a GoogleTV platform. None of the above example would be in any way unexpected for this platform or unreasonable to expect. The underlying Android operating system is POSIX compliant and supports multi-core and multi-threaded applications. Android also supports symmetric multiprocessing allowing GoogleTV platforms to fully leverage multithreaded, SMP capable processor cores.

This is one of the key design concerns for silicon vendors and OEMs looking to produce competitive platforms for GoogleTV. The very nature of the variable workload and extensive interweaving of network aware applications with video capable applications drives much higher overall system performance requirements. Those in turn push down onto the processor core.

Choosing a processor architecture that is already fully capable of supporting multithreaded applications and SMP configurations is essential for longevity in the GoogleTV market.

GoogleTV has entered the market with two V1 products that are clearly at the low end of the target capabilities. While products do handle the user interface requirements, they are clearly limited in terms of their capability to handle concurrent multimedia streams, simultaneous applications and web browsing. This underscores the requirement that platform/silicon designers need to chose a processor architecture that scales across the range of potential GoogleTV implementations and also can be integrated with all the various components that are needed for connectivity, graphics, multimedia decode/transcode and connectivity.

MIPS architecture is the industry standard for the digital TV and set top box market world wide. MIPS-based silicon vendors already have an advantage with respect to GoogleTV: they have access to the well integrated components needed for GoogleTV and are using an architecture roadmap that hits all the performance points needed for GoogleTV – a range of single core implementation choices and multicore capability for SMP implementations.

As a starting point, MIPS advises silicon vendors to consider the following configurations when formulating their designs:

- Entry level GoogleTV Platform: flawless user experience, limited concurrency
  - A MIPS32 74Kf or a dual core1004Kf processor
    - 2 hardware threads, or Virtual Processing Elements (VPE)
    - Includes Floating Point processing Unit running at least at half the processor clock speed
    - 32KB L1 instruction and data caches
    - Minimum of 32 entry TLB, 64 recommended
  - 256KB L2 cache
  - CPU clock speed: 1 GHz (74Kf) or 800 MHz (dual core 1004Kf) or greater
- High end GoogleTV Platform: flawless user experience, parallel video streaming, application execution and web browsing
  - At least a 2 core MIPS32 1074Kf processor
    - Includes Floating Point Processing unit running at the full processor clock speed
    - 32KB L1 instruction and data caches
    - 64 entry TLB
  - 256KB L2 cache per CPU (i.e. 512KB for a 2 CPU SoC)
  - CPU clock speed of 1 GHz, or greater[3]

[3] 1 GHz based on requirements for web browsing experience that is similar to performance on mainstream Windows PC's

Graphics

To support GoogleTV and the Android Kernel, developers must implement a mandatory 3D graphics engine that is compliant with the OpenGL ES 2.0 specification.

The OpenGL ES 2.0 graphics engine is necessary in a GoogleTV based product for the following purposes:

- The rich 3D user interface
- Internet browsing
- Adobe Flash 10.x
- Flash based 3D games
- Alpha blending

Display Frame Buffer

1080p resolution is 1,920 horizontal pixels by 1,080 vertical lines for a total of 2,073,600 pixels. If a color depth of 4 bytes per pixel is used, then the total amount of memory required that is shared between the CPU and graphics engine frame buffer is 8,294,400 bytes or roughly 8.3 Mbytes.

ATSC televisions have a maximum frame rate of 60Hz. This means that 8.3 Mbytes of data are sent to the display 60 times a second. For a 32-bit wide memory bus, best case scenario is to transfer 4 bytes with each cycle so the data rate for sending video to the display is:

$$\frac{(.3MBytes * 60Hz)}{32bitwidebus} = 124.5M\sec$$

This is only the transfer requirement for the frame buffer. There will also be graphics engine memory accesses for pre-rendering, 3D texture maps, etc. The video de-compressor will require its own memory space as well.

Careful consideration must be given to the design of the SoC memory controller. Using the fastest possible external DRAM that the SoC budget can accommodate ensures that the end product will provide a high quality user experience.

This may sound excessive. However, in addition to rendering live/recorded video streams, GoogleTV relies heavily on browser functionality and its browser takes full advantage of 3D graphics controllers for faster page rendering. In order to attain a stutter free user experience, this implies not only does the frame buffer need to be as large as possible but also the memory controller needs to support the fastest transfer rates possible.

This situation is compounded by the fact that any video content sourced from the Internet will, for the foreseeable future, be using Adobe Flash. Flash 10 is the basic capability any GoogleTV implementation. Access to Flash-based content comes through the browser – further intensifying the work load placed on the graphics subsystem frame buffer and memory controller.

Based on these software requirements, the recommended minimum performance levels for OpenGL ES 2.0 graphics engine:

>250 Mpixels/sec
>10M triangles/sec

OpenGL ES 2.0 combines a version of the OpenGL Shading Language for programming vertex and fragment shaders.

CODECs

Modern consumer electronics devices that display video content all use a dedicated hardware decoder to handle 1080p high definition content. Using software CODECs running on the main application processor only works for extremely small screen areas (for example, contemporary smartphone displays). Such software-based decoding would constitute and inefficient use of a main application processor and would be impacted by other demands on the application processor such as dealing with network connectivity. GoogleTV is no different from any other set top box or digital TV platform and thus dedicated hardware is recommended to implement the various CODECs needed for the variety of content accessible from GoogleTV.

There are several CODECs required for GoogleTV. These requirements are independent of television specific standards (i.e.: ATSC, DVB, etc) which it assumes that the SoC manufacturer has complete familiarity with. The CODEC requirements are necessary to provide the same Internet browsing experience on the TV as a user would have on a PC. It is up to the SoC manufacturer to decide if they want to implement the CODECs in hardware or software. If some CODECs are to be implemented in software, then it is very important to ensure that there is enough computing power available to meet the minimum performance levels set by the various standards bodies.

The following table is a summary of the CODECs commonly used by a majority of media content found on the Internet:

CODEC	Description
H.264[4]	Web & Adobe Flash Video - 1080p, 60fps
VP6	Adobe Flash Video - 1080p, 60fps
VP8	HTML5 Video
Sorenson Video	Adobe Flash Video
Ogg Theora	Open source video format used in Google Chrome and YouTube
Ogg Vorbis	Open source audio format used in Google Chrome and YouTube
RealVideo	RealVideo is a proprietary video format developed by RealNetworks
RealAudio	RealAudio is a proprietary audio format developed by RealNetworks
WMV	Windows Media Video format
MOV	Apple QuickTime movie format
MJPEG	Motion JPEG video format
AVI	Audio Video interleave format used by Microsoft
FLV	Adobe – FLV1 video CODEC, MP3 audio
JPEG	Compression format for still images
PNG	Patent free still image compression format
GIF	Graphics interchange format for still image compression
WMA	Windows Media Audio
MP3/MP2/MPA	MPEG audio formats
MIDI	MIDI music synthesis, requires synthesis engine
WAV	Audio for Windows format
AAC	Advanced Audio Coding
G.729 (optional)	VOIP audio conferencing format

Table 1 Commonly Available CODECs and Their Applications

Memory - DRAM and Flash

Given the need to handle multiple video streams and concurrent application execution/web browsing, MIPS recommends at least 1 GBytes of DDR3 memory for entry level GoogleTV implementations. In higher end systems designers should consider at least doubling this memory capacity to 2 GBytes.

Flash memory is also a major requirement for GoogleTV platforms. MIPS recommends a minimum of 256 Mbytes for entry level configurations and 512 Mbytes for high end configurations of system flash memory for storing operating system and user applications for GoogleTV. Flash memory or rotating magnetic storage may also be needed for DVR functionalities. Entry level systems should plan on an additional 5 GBytes of flash memory for storing media content. High end systems should consider increasing this substantially as DVR workspace is crucial for most users. In addition, USB ports should be made available to allow for supplementary storage and also an easy path for users to load their own content onto the platform for viewing or uploading.

[4] Hardware acceleration required for high definition TV viewing experience

The external memory interface is the single most significant point of performance limitation. In a system where the CPU, graphics processor, compression/de-compression engines all share the same memory, it is absolutely critical that the memory interface be designed for the widest and fastest throughput that the SoC can afford.

Given how critical graphics performance is to GoogleTV, SoC and system level designers should check with their graphics engine supplier for specific details on the memory requirements and system bus requirements.

Conclusion

As with Android-base mobile phones, we expect an explosion of GoogleTV devices to follow the initial 2010 launch, starting in 2011.

GoogleTV clearly aligns with MIPS' vision of the future connected television and entertainment experience. MIPS shares and supports Google's commitment to the architecture-neutral, open source nature of its Android-based initiatives such as GoogleTV. MIPS will ensure that our licensees have access to the best available software ecosystem for Android-based solutions.

MIPS licensees lead in providing solutions the digital home today. There is no doubt that a great numbers of future GoogleTV systems will be based on MIPS. Giving careful consideration to the recommended performance requirements for GoogleTV stated in this white paper will yield highly attractive SoCs that combined with appropriate graphics, memory, and connectivity components will ensure an outstanding GoogleTV experience.

Interested in creating a solution for GoogleTV? Then contact MIPS Technologies Inc. at www.mips.com

References

Google I/O Conference 2010
http://code.google.com/events/io/2010/

Porting Android to a new device
by Peter McDermott
2008-12-04
http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Porting-Android-to-a-new-device/

The Khronos Group
http://www.khronos.org/opengles/2_X/

MIPS Technologies, Inc
http://www.mips.com/android
http://www.mipsandroid.org