Introduction

The digital security and surveillance industry is rapidly adopting networked intelligent solutions designed to manage and analyze multiple streams of high-definition IP video. System elements include network video recorders (NVRs) designed to provide robust video processing and management of multiple HD video streams, content analysis through sophisticated video analytics, and high-throughput.

This white paper provides an in-depth discussion of the design and implementation of network video recorder (NVR) products for IP video digital security and surveillance applications, including recommendations on hardware selection and software architecture.

Application and media processing enhancements in 4th generation Intel® Core™ processors and Intel® Xeon® processors enable developers of NVR products to optimize performance throughout the video processing and analysis pipeline. Functional modules within the NVR benefit from hardware-accelerated graphics, enabled by the powerful video encoding and decoding performance and parallel processing capabilities of Intel® HD Graphics. This paper includes a description of common hardware and software design issues and provides recommended approaches for optimizing the performance of the processor’s integrated graphics processing unit (GPU) and acceleration of decoding, post-processing, and video analysis in NVR products.
System Analysis and Selection

System Analysis

The network video recorder (NVR) performs three essential functions:

- **Decoding:** Transforming the code stream filmed by IP camera or local video files into original image data in YUV format.

- **Post-processing:** Converting images from YUV format to RGB format, in addition to advanced functions including noise reduction, sharpening and image scaling.

- **Video analysis:** Processing of scene content for object detection, recognition motion tracking, tamper detection, and other factors, following decoding and conversion.

Traditionally, NVR products process decoding, post-processing and video analysis functions through the CPU. To take full advantage of the powerful computing features of previous generations of Intel® processors, manufacturers have adopted Intel’s advanced processor instruction sets, including Intel® Streaming SIMD Extensions (Intel® SSE) and Intel® Advanced Vector Extensions (Intel® AVX), in addition to software including the Intel® Integrated Performance Primitives (Intel® IPP) software development library. While practical, this approach consumes significant processor and memory resources, constraining system resources available for video analysis and limiting the number of multi-channel parallel streams that can be processed by the system.

With the introduction of 4th generation Intel Core processors and the latest Intel Xeon processors, the NVR platform can be designed to optimize performance and minimize processing overhead throughout the NVR’s video pipeline. Functional modules benefit from graphics processing unit (GPU) hardware acceleration enabled by Intel® HD Graphics technology, including powerful video encoding and decoding performance and parallel processing capabilities.

Hardware Selection

High-end computing performance is essential in modern NVR products designed for video processing and analysis, and this requirement can be met by 4th generation Intel® Core™ i7 processors and Intel® Xeon® processors with Intel HD Graphics.
Performance
4th generation Intel® Core™
microarchitecture based on
22nm manufacturing process
Intel® AVX 2.0 instruction set
for improved integer/matrix
calculations

Power-efficiency
Ultra Low Power (U-Series)
one-chip processor with
15 watt thermal design
power (TDP)
Configurable power levels

Connectivity
Flexible I/O (USB/SATA/PCIe)
Intel® wireless and GbE

Graphics & Media
Integrated Intel® HD and Intel®
Iris™ Graphics for significant
performance enhancements
Choice of GT3, GT2, or GT1
graphics cores

Hardware-based Security
BIOS/firmware protection
Intel® AES-NI for faster
encryption
Faster boot/resume times

Figure 2. Performance, graphics,
connectivity and security enhancements
of 4th generation Intel® Core™ processors.

With the first new microarchitecture
based on 22nm manufacturing process
technology, 4th generation Intel Core
processors provide significant overall
performance improvements, compared to
previous Intel Core processors. As shown
in Figure 2, 4th generation Intel Core
processors include advanced
technologies for multimedia processing,
parallel computing, encryption algorithm
processing, and reduced platform power
consumption.

The 4th generation Intel Core
processor platform adds Intel® Iris™ and
Iris Pro™ Graphics with GT3 processing to
complement the earlier-generation GT1/
GT2 specifications. The main difference
between the specifications lies in the
number of execution units (EUs) available,
with the video coding and decoding units
remaining the same. Operations
including video post-processing,
encoding, and parallel computing with
the GPU benefit from the participation of
additional EUs, and processor models with
GT2 or GT3 processing can be selected for
these application scenarios. If the main job
of GPU is video decoding, selecting
processors with GT1 graphics may be the
most efficient option.

Figure 3. GT3 speed video transcode
performance improvements in 4th
generation Intel® Core™ processors with
integrated Intel® Iris™ 5100 Graphics and
Intel® Iris Pro™ 5200 Graphics.
The GPU driver runs on the CPU, and processor utilization will vary in multi-channel decoding applications due to the coordination GPUs of the same size with CPUs of different frequencies. In addition, the performances of GPUs of the same size will vary due to their individual frequency characteristics. As shown in Table 1, in Intel HD 4600 GPUs using GT2, the CPU performance of Intel Xeon E3-1225/1275 processors and Intel Core i7 processors will differ. The highest frequencies supported by the GPUs are also different. The final test results also reflect these changes.

<table>
<thead>
<tr>
<th>GPU MODEL</th>
<th>CPU MODEL</th>
<th>CPU FREQUENCY</th>
<th>THE HIGHEST CPU FREQUENCY</th>
<th>CPU CACHE</th>
<th>GPU FREQUENCY</th>
<th>THE HIGHEST GPU FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® HD</td>
<td>E3-1225V3</td>
<td>3.2 GHz</td>
<td>3.6 GHz</td>
<td>8 MHz</td>
<td>350 MHz</td>
<td>1.2 GHz</td>
</tr>
<tr>
<td>Graphics</td>
<td>E3-1275V3</td>
<td>3.5 GHz</td>
<td>3.9 GHz</td>
<td>8 MHz</td>
<td>350 MHz</td>
<td>1.25 GHz</td>
</tr>
<tr>
<td>4600</td>
<td>i7-4700EQ</td>
<td>2.4 GHz</td>
<td>3.4 GHz</td>
<td>6 MHz</td>
<td>400 MHz</td>
<td>1.0 GHz</td>
</tr>
</tbody>
</table>

Table 1. Test platform differences in 4th generation Intel® Core™ platforms with Intel HD Graphics 4600 (GT2).
H.264 Video Decode Performance: 1080p 30fps

Figure 5. 1080p multi-channel decoding results.
Main Profile, Constant Bit Rate, GOP IPPPPPPP.

H.264 Video Decode Performance: 720p 30fps

Figure 6. 720p multi-channel decoding results.
Main Profile, Constant Bit Rate, GOP IPPPPPPP.
Video decoding, processing, and content analysis in NVR applications requires significant memory resources and frequent memory access. Because of these requirements, more than 8 GB of dual-channel DDR3 memory is recommended.

**Software Selection**

The 64-bit operating system is recommended to take optimal advantage of the processor and large memory. See the System Architecture Optimization section of this paper for more details.

Graphics hardware acceleration is heavily constrained by the driver architecture model of the operating system and the latest graphics standards. Microsoft introduced a new Windows Display Driver Model (WDDM) driver architecture starting in the Vista operating system, and Microsoft fully supports DirectX 11 in Windows 8. In order to obtain improved hardware acceleration capabilities in NVR products, Intel recommends that customers use the mainstream 64-bit Windows 7 or Windows 8 operating systems. Because Windows 8 is relatively recent, this paper will use 64-bit Windows 7 as an example for discussion.

**Analysis of NVR Functional Modules**

**Decoding and Video Post-Processing Module**

In the field of IP video digital security surveillance, third-party decoders are often required to decode proprietary encapsulation or video streams used by specific manufacturers. In addition, many of these third-party decoders are designed to meet the demand for standard media decoding. For standard decoding tasks, developers have several hardware acceleration options.

**Option 1:** Comply with Microsoft’s DirectX standard. Use a variety of APIs in the DirectX Video Acceleration (DXVA) components in DirectX for programming.

**Advantages:**

- **Good hardware compatibility.** Because DirectX Video Acceleration is based on Microsoft’s standards, the shielding of underlying hardware is good. It can be compatible with GPUs equipped with hardware decode acceleration functionality.

- **Excellent software interactivity.** Because DXVA belongs to Microsoft’s DirectX architecture, the DXVA-enabled decoding function can be smoothly connected to other DirectX functional modules. For example, the Direct3D* component is used for complex video post-processing and DirectDraw* (DXGI) component is used for display.

**Disadvantages:**

- **Complex programming requirements.** DirectX comprises a large collection of APIs with many functional components. In addition there are numerous DirectX versions, ranging from DX9 (popular from Windows XP), to DX11.1, supported by Windows 8.1. Microsoft has introduced many modifications, including the removal of outdated functional components, addition of new functional components, new display driver models, and new programming concepts. These changes can increase complexity for programmers.

- **Failure to open some hardware-specific acceleration functions.** DirectX is a standard for Windows-based systems. While it provides excellent versatility for hardware from independent hardware vendors (IHVs), it does not support some IHV-specific functions. For example, hardware encoding acceleration and special video post-processing developed by Intel cannot be implemented directly in the DXVA API library.

**Option 2:** Load known third-party decoder plug-ins with hardware acceleration functions.

**Advantages:**

- **Simple programming for rapid development.** Developers with no video coding experience can easily integrate mature third-party decoders by using the Microsoft .NET* framework, which enables developers to focus on user interfaces and other functionality. In this way, Microsoft .NET can shorten the software development cycle.

- **Relatively stable.** Developers will typically encounter fewer problems with mature third-party decoders.

**Disadvantages:**

- **Cost.** Copyright fees may apply.

- **Lack of flexibility.** Problems may be difficult to resolve.

- **Lack of proprietary core value.** This may not be conducive to long-term product development.

**Option 3:** Adopt the Intel® Media SDK.

The Intel Media SDK is a multimedia development kit based on Microsoft’s DXVA architecture. It enables developers to optimize the media acceleration capabilities of software and hardware on an Intel® architecture-based hardware platform, while implementing a variety of specific hardware features.

Advantages:

- **Hardware flexibility.**
  As shown in Figure 7, the Intel Media SDK can help optimize the characteristics of multiple platforms, including software implementations enabled by Intel SSE and Intel AVC instruction sets and hardware implementations based on the Intel HD Graphics GPU. Note that hardware encoding cannot be achieved through DXVA common programming.

- **Backward compatibility.**
  The Intel Media SDK can collect the underlying hardware information and select hardware acceleration libraries depending on the hardware platform needed to accomplish specific tasks. This enables the same program to run smoothly on multiple hardware platforms, and the performance of existing software products can be maintained or improved following hardware updates.
Modular programming.
As shown in Figure 8, the Intel Media SDK enables implementation of multiple functional modules including decoding, conversion, and encoding. Another benefit is that developers can use modules enabled by the SDK together with existing modules. For example, developers can use their DXVA decoding module and an SDK-enabled encoding module to implement complete transcoding hardware acceleration, or achieve certain special post-processing effects by using an OPENCL-based video processing module after SDK decoding.

Figure 8. Modular design of the Intel® Media SDK provides the flexibility to use multiple modules.
• Simple programming. Based on DXVA, the Intel Media SDK shields a variety of complex DirectX APIs, which simplifies programming complexity. The SDK development package also contains application samples and provides comprehensive source code and project files. Developers can master development features in a short time by studying the samples.

• No cost. The Intel Media SDK is available free of copyright issues.

Disadvantages:

• Limited support. The Intel Media SDK supports platforms based on Intel architecture only.

• Relatively simple functions. The Intel Media SDK provides video features including decoding, video processing, and encoding. It does not provide comprehensive support for video, 3D, display, and general-purpose computing. These functions can be implemented by developers through modular programming.

Intel recommends that customers use Intel Media SDK to implement decoding and video post-processing on NVR products. The remainder of this paper provides information pertaining to the use of the Intel Media SDK. For information regarding other methodologies, refer to relevant Microsoft documentation.

The Intel Media SDK: issues and limitations
The Intel Media SDK can be used to implement decoding and post-processing modules in NVR applications, but some limitations apply.

1) Support for encoding and decoding formats
The Intel Media SDK supports encoding and decoding of a variety of popular video formats:

• Video encoding: H.264, MPEG-2, MJPEG and MVC codec supporting 3D videos. MJPEG is implemented in software only.

• Video decoding: H.264, MPEG-2, MJPEG/JPEG, VC-1, and MVC.

Because hardware acceleration relies on the support of underlying the hardware and driver, the new features supported by the latest Intel Media SDK 2013 version can be fully implemented only on the 4th generation Intel Core platform. For further information on the acceleration features supported by previous platforms, please contact your Intel technical representative or refer to the product manuals for the respective platforms.

2) Video processing functions
The Intel Media SDK 2013 supports the following video processing functions: TV signal interlace/progressive conversion, image scaling, color space conversion, noise reduction, frame-rate conversion, brightness, contrast and color saturation, sharpening, and graphics stability. Note that input and output of each feature cannot be arbitrarily determined. For example, color space conversion supported by MSDK only supports the following input and output formats listed in Table 2. For information on detailed specifications and parameters supported by each feature of the SDK, please refer to the Intel Media SDK product manuals.

<table>
<thead>
<tr>
<th>Input Color</th>
<th>NV12</th>
<th>RGB32</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB4 (RGB32)</td>
<td>X limited</td>
<td>X limited</td>
</tr>
<tr>
<td>NV12</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>YV12</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>YUY2</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2. Color conversion formats supported by the Intel Media SDK.

3) Intel Media SDK operating environment check
To provide the basis for software development, the operating environment must be checked after a hardware platform is selected and the SDK is installed.

Intel provides the Mediasdk_Sys_analyzer tool to help customers obtain accurate information about their particular platform. This tool is included in the Tools Directory under the installation path of the Intel Media SDK. Typical operation of the tool is shown in Figure 9.
The operating results of mediasdk_sys_analyzer show the specific features supported by the platform. Figure 9 shows that the 2nd generation Intel® Core™ platform supports hardware decoding and encoding of version 1.4 and below.

Figure 10 shows that the GPU version of Fourth Generation Intel® Core™ Platform is GT2, but all the hardware encoding and decoding functions are not supported. This result is due to the fact that the monitor is not connected to the processor graphics. Although the Graphics Device category lists Intel® HD Graphics P4600/P4700, the State is 08. This error return value indicates that the GPU is currently in offline status. The correct configuration of the SDK operating environment can avoid the failure to open hardware acceleration during development.

4) Selection of the appropriate Intel Media SDK API version

After both hardware and software environments are selected, users need to select the correct version of API library based on to their particular application scenario. The version number of the API library increases, corresponding to Intel processors. Each new version of the API is used to implement new features enabled by each successive platform, while maintaining support for previous functions. Customers should carefully analyze their functional requirements and select the minimum version that meets these requirements.

Video Analysis Module

Similar to the video codec, the video analysis module can be implemented through either software or hardware.
The software implementation is more mature. Developers can implement video analysis algorithms through software development tools including the Intel IPP library and the Open Source Computer Vision (OpenCV) library, or through the direct use of Intel processor instruction sets including Intel SSE and Intel AVX. The advantage of software implementation is that flexible programming can be used to create algorithms of multiple types and complexities. In addition, using processor performance debugging tools such as the Intel® VTune™ Performance Analyzer, developers can easily discover any performance bottlenecks in their algorithms to achieve more accurate and efficient implementation.

The principal reason for hardware implementation of video analysis is to maximize the capability of GPU parallel computing that permits operation of multiple analytical algorithms on the GPU. This approach generally results in better performance than a software-only implementation. Hardware implementation also provides a development path for video analysis. Developers can choose a GPU hardware platform with support for DirectX 11 and Open Computing Language (OpenCL) and use common software development tools such as Computer Shader* or the OpenCL SDK. Currently there are some limitations, including high demands on the hardware platform, reduced algorithm implementation, and lower efficiency of some algorithms.

Intel is also adding GPU functions into the Intel IPP library, and a preview version is available in the website: http://software.intel.com/en-us/intel-ipp/.

Both CPU and GPU implementations have their own respective advantages in the implementation of video analysis algorithms. Intel is committed to providing customers with the hardware and software environment for a seamless use of the CPU and GPU.
System Architecture Optimization

System architecture, as discussed in this section, refers to the system software architecture including evaluation of the implementation of functional modules and their connection. Poor system performance is usually not the result of underlying hardware performance, but is most often due to software design and implementation issues. These problems can be solved by optimizing software architecture and code implementation.

Memory Copy

As discussed earlier, decoding and post-processing tasks in NVR application scenarios can be performed by hardware acceleration enabled by the Intel Media SDK. However, most video analysis today relies on software implementation, and this means that the video data in memory requires two storage locations:

- In the video decoding and post-processing stages, the data is saved in video memory that the GPU can access rapidly and effectively.
- In the video analysis stage, the data is saved in the most efficient system memory that the CPU can access.

These dual storage requirements require a memory copy operation that can consume significant CPU resources and cause a pipeline bottleneck. Dealing with memory copy issues is the first problem to be solved during NVR system optimization. There are three principal scenarios.

1) The GPU acceleration section assigns data to video memory, and data for CPU analysis is allocated to system memory.

For optimal performance, data must be placed on the most efficient memory location accessible to the CPU for reads and writes. For this reason it is necessary to pay special attention to data allocation. For example, the hardware implementation of the Intel Media SDK supports the use of system memory for input and output of data. It should be noted that the display driver will copy input data from system memory to video memory internally in order to enable the GPU to work more effectively, and then copy output data from video memory back to system memory after GPU processing. Data copying is the main reason that Intel Media SDK hardware acceleration using system memory results in lower performance than video memory.

2) Select video formats that are natively supported by hardware, avoiding redundant data copies.

The Intel Media SDK natively supports the NV12 video format for encoding and decoding, which is recommended as a reference when color formats are being defined. Other color formats will be converted to NV12 before and after the GPU processing, which result in unnecessary data copying during color space conversion.

3) Replace memcpy with Fastcopy.

When a memory copy must be performed, a correct and optimized copy method will improve the overall performance of the program. Traditional memcpy is a linear copy method with very low efficiency. Intel SSE instruction sets provide a more efficient way to complete the copy process and significantly improve overall performance. Specific Fastcopy system requirements and implementation methods can be found in Intel white papers.

Synchronization and Asynchronization

Any operations that cause functional units to wait for each other can be regarded as synchronous operations, including synchronization between the CPU and GPU, hardware synchronization between hardware engines in the GPU, and synchronization (CPU execution logic) between multiple execution modules of software.

Typical hardware synchronization involves copying from video memory to system memory. In this case, the CPU must call LOCK to request access to the video memory. The LOCK call can return the correct address pointer and the CPU can start to copy only after the GPU ends the operation on relative position in video memory and converts the address space to system memory. When the copy begins, the GPU must hold all the read/write operations to corresponding video memory until the CPU finishes its tasks and calls the UNLOCK function to release the address translation resources. With respect to GPU internal synchronization and software synchronization, if the execution speeds of two hardware or software modules are different, the module running faster must wait for the end of the execution of the slower module.

In current NVR application scenarios, both hardware and software synchronization have problems relating to the connection phase of video post-processing module and video analysis. Functions within the processing pipeline, including video decoding and post-processing, do not involve synchronization.
Hardware synchronization

Because video memory copy cannot be avoided, developers can use the recommendations provided in the previous section to accelerate copy performance and shorten the waiting time for synchronization.

Software synchronization

Video analysis algorithms based on frame images are numerous, complex and often require long time, resulting in failure to handle a large number of decoding output data in time. The following methods can be considered:

- Select skipping analysis based on the content.
- Perform down-scale sampling during post-processing to reduce the volume of content requiring analysis.
- Optimize the internal algorithms and data transfer between modules to avoid memory conversion caused by different data formats.

64-bit vs. 32-bit Applications

Current computer system hardware and operating systems support 64-bit programming. In addition, the cost of memory and disk storage has declined substantially. Intel recommends that customers replace 32-bit applications with 64-bit applications. This optimization is particularly important in NVR applications used in IP video digital security and surveillance.

1) Using 64-bit systems and programming can maximize the advantages of address space addressing to help NVR applications access large amounts of data. Multi-channel video decoding and analysis in NVR applications requires the allocation of large memory resources. A single application in a 32-bit system only has 2GB of addressable space. If parallel execution of multiple tasks is conducted in a process, the fatal error of address space overflow is prone to occur. Using 64-bit programming can avoid the problem of address space shortage.

2) For the current mainstream hardware, supporting 64-bit addressing means that most of the hardware registers are 64-bit. Using 32-bit programming may require hardware to do additional completion operations on registers, and this can have negative effects on performance.

3) In addition, 64-bit operating systems perform additional conversion to run 32-bit applications. For such compute-intensive applications as NVR, avoiding the need for system conversion can improve the performance of the application.

Multi-process Programming vs. Multithreading

Today’s mainstream CPUs have multi-core architecture and hyper-threading parallel processing capabilities, both of which provide a good hardware foundation for multi-tasking processing.
Multi Thread Integer & Floating Point Performance

Figure 11. Relative performance comparison of multi-threading capabilities in selected processors.
Multi Thread Integer & Floating Point Performance

![Multi Thread Integer & Floating Point Performance graph]

Processor Integer Performance (SPECint2006 max rate user)
Processor Floating Pt. Performance (SPECFP2006 max rate user)

- Intel® Core™ i3-3120ME
- Intel® Core™ i5-3610ME
- Intel® Core™ i7-3612QE
With respect to software, applications with many parallel tasks typically can use single-process/multi-threaded, or multi-process solutions. Both of these approaches can benefit from the multi-core multi-threading capabilities of Intel processors, and each has certain advantages in NVR applications.

Single-process/multi-threaded software is currently the most widely employed software implementation. Its first advantage is simple programming. All threads occupy the same address space, reducing inter-thread communication overhead and providing more uniform access to data. This minimizes the overhead for mutual access between multi-channel data streams. Single-process multi-threaded data interaction simplifies applications that require graphical user interfaces (UIs). The UI thread can easily obtain data from other threads without significant overhead. On the other hand, because all threads run in the same process space, any fatal error occurring in one thread will trigger the collapse of the entire process, which significantly reduces the robustness of the system. This lack of robustness is generally not acceptable in NVRs used in IP video digital security and surveillance.

Multi-process programming exhibits far better robustness in a multi-task process, because a process fault is limited to that specific task. Other advantages of multi-process programming include:

- **Single-process logic.** Multi-process programming is a relatively simple and efficient implementation.
- **Flexible multi-core multi-process scheduling.** This enables more efficient access to CPU resources such as caching.
- **Flexible task allocation.** Multi-process system can easily assign some tasks to use hardware to conduct video acceleration, while others use software for acceleration. A multi-process system can also assign certain tasks to process high-definition video, and assign others to process standard-definition video.

One disadvantage is that inter-process data access efficiency across different process spaces is relatively more difficult than with the single-process/multi-threaded approach.

If there is no linkage among multiple-channel video in NVR applications, multi-process implementation can be considered. If synchronization processing or interactive access of video is needed, the single-process/multi-threaded approach may be the preferred option.

**Summary**

NVR products for IP video digital security and surveillance integrate three major functions including video decoding, processing, and analysis. This processing pipeline imposes stringent demands on the computing capacity of the processor. The selection of the right hardware platforms, operating systems, and development tools provides a solid foundation for optimizing NVR product performance. The 4th generation Intel Core processor product family helps developers meet this goal by providing enhanced computing performance, integrated GPU hardware acceleration, complemented by Intel® software, including the Intel Media SDK.

In order to coordinate with the Intel Media SDK, Intel provides customers with a comprehensive package of development software and tools designed to simplify the implementation of hardware and software solutions. Video decoding and processing are supported by the SDK, which greatly improves application encoding and decoding capabilities. For video analysis, Intel provides the Open CL SDK and Intel IPP library to support GPU acceleration. To optimize software performance, Intel’s debugging tools, Include Intel VTune Performance Analyzer and Intel® Graphics Performance Analyzer can be used improve program design to enhance the hardware execution efficiency. The industry-leading hardware products and development tools provided by Intel enable developers to analyze and design NVR products and applications that meet the growing demands of today’s IP video digital security and surveillance environment.

More information

To view additional white papers and videos related to what Intel is doing in DSS, please visit:

http://www.intel.com/info/dss