Freescale Semiconductor
White Paper

Document Number: IMX31MULTIWP
Rev. 0, 02/2006

Multimedia Support in the i.MX31 and i.MX31L Applications Processors

by: Oskar Pelc

1 Executive Briefing

In the endeavor for differentiation in wireless handheld mobile entertainment devices, multimedia capabilities tops the features list. Dynamic, rich and fast-paced video with 3D graphics and stereo sound can make the heart race, but it creates a bevy of challenges that can make an embedded device engineer's heart palpitate.

This paper presents how Freescale design engineers address the issues of performance and low power in handheld devices while delivering an incredible multimedia experience in the i.MX31 multimedia applications processors.

The i.MX31 and i.MX31L multimedia applications processors provide outstanding capabilities for multimedia applications. The following features demonstrate the multimedia capabilities:

- High performance:
  - Still image capture:
    - up to 16 Mpixel resolution
    - up to 40 Mpixels/sec, for example, 2 Mpixel image at up to 20 fps

Contents

- — Motion video capture (including real-time encode):
  VGA resolution, 30 fps, 4 Mbits/sec
- — Video playback (including real-time decode):
  VGA resolution, 30 fps, 2 Mbits/sec
- — Video call:
  CIF resolution, 30 fps, 384 Kbps

- Versatile connectivity:
  - — 2 displays (smart/memory-less) and a TV encoder
  - — 2 image sensors (smart/RAW)
  - — Storage devices: MMC card, Flash card, SDIO, HDD
  - — Synchronization with a PC: USB HS—up to 480 Mbits/sec

These performances are achieved while preserving battery life. Table 1 summarizes the system power-consumption and battery life during some important multimedia use cases.

Table 1. Multimedia - Battery Life[1, 2]

Use Case	With Smart Display	With Memory-less Display
Audio Playback (MP3, 128 Kbps)	67 hours (67 mW; with display off)	
Camera Preview (QVGA @ 30 fps)	34 hours (85 mW)	29 hours (100 mW)
Video Clip (Audio and Video) Capture - MPEG-4 SP		
CIF @ 30 fps, 384 Kbps	13 hours (230 mW)	12 hours (250 mW)
VGA @ 30 fps, 1-4 Mbps	8 hours (350 mW)	7 hours (400 mW)
Movie (Audio and Video) Playback - MPEG-4 SP		
CIF @ 30 fps, 384 Kbps	12 hours (150 mW)	11 hours (170 mW)
VGA @ 30 fps, 1 Mbps	6 hours (500 mW)	5 hours (550 mW)

1 The power results are quoted for a full system, including all components (camera, display, speakers, etc.) and the estimate is based on pre-silicon simulation.

2 The battery used in the calculation is rated at 3.6V, 800 mAh

To achieve the performance described in Table 1, the i.MX31 and i.MX31L architectures have a highly-optimized implementation of the video data path providing the most demanding aspect of multimedia applications.

Figure 1 on page 3 shows the imaging and video system.

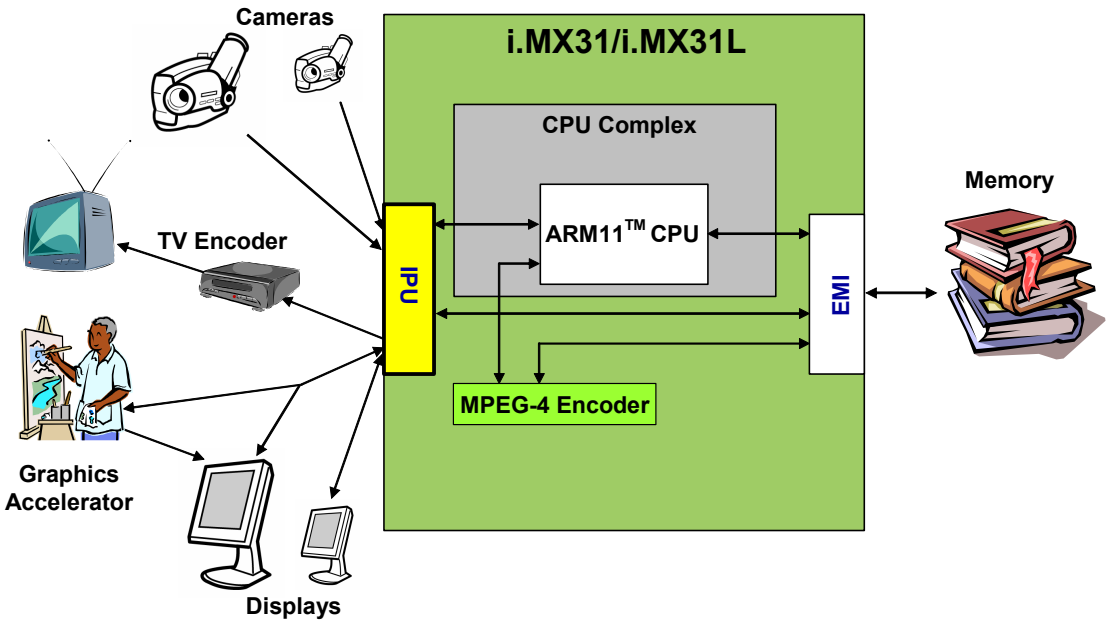

Figure 1. The Imaging and Video System

The major components of the i.MX31 and i.MX31L applications processors involved in image and video processing are the following:

- Image Processing Unit (IPU) hardware accelerator
  - Image processing (filtering, conversions)—up to VGA resolution, 30 fps.
    To perform the same processing on ARM11™, it would require 1.2 MHz.
  - Power efficient autonomous functionality (requiring no involvement of the ARM® CPU), such as screen refresh/update, camera preview
  - Versatile interface to displays and cameras
- MPEG-4 Video encoder
  - Formats: MPEG-4 SP (simple profile) and H.263 baseline
  - Rate: up to VGA resolution, 30 fps, 4 Mbps
  - To perform the same processing on ARM11, it would required 1.4 MHz.
- ARM11 CPU
  - MPEG-4 SP video decode: up to VGA resolution, 30 fps, 2 Mbps
  - Video decode in advanced formats (H.264, WMV, RV...)
    At least HVGA resolution, 30 fps, 1 Mbps
  - MPEG-2 Main Profile decoding: D1 resolution up to 9.6 Mbps

2 Image Processing Unit (IPU)

A schematic block diagram of the IPU is presented in Figure 2 on page 4.

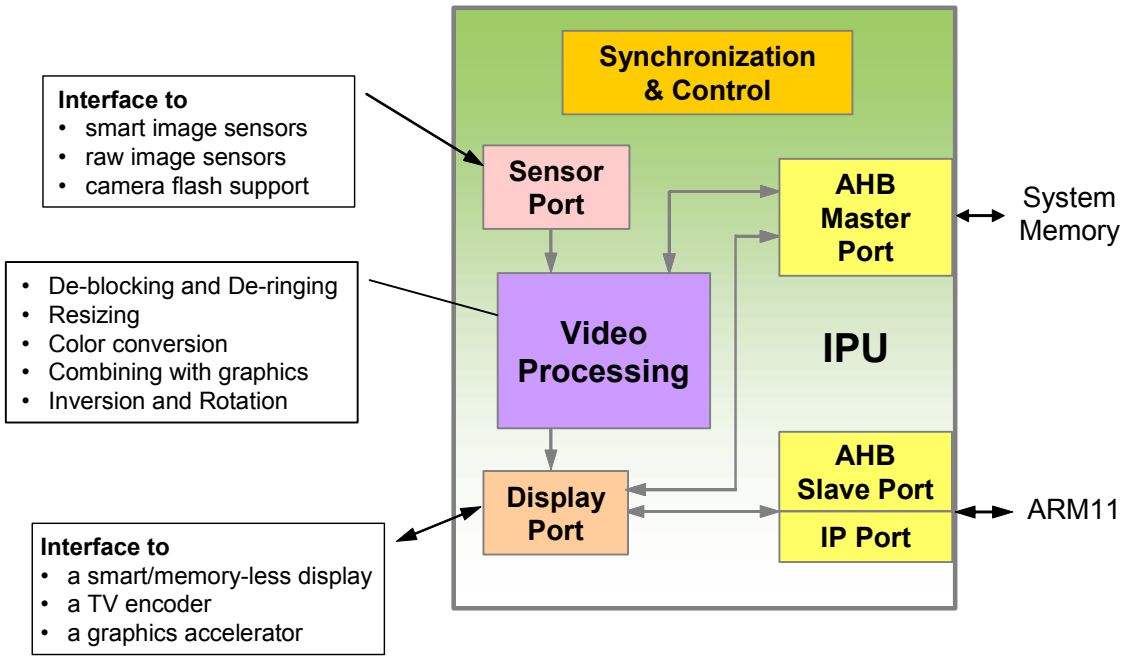

Figure 2. Image Processing Unit (IPU)

2.1 Integrative Approach

The IPU includes all the functionality required for image processing and display management. This integrative approach leads to several significant advantages, as described below.

2.1.1 Automation

The IPU is equipped with powerful control and synchronization capabilities to perform its tasks with minimal involvement of the ARM CPU. These include the following devices and capabilities:

- An integrated DMA controller (with two AHB master ports), allowing autonomous access to system memory
- An integrated display controller, performing screen refresh of a memory-less display
- A page-flip double buffering mechanism, synchronizing read and write accesses to the system memory to avoid tearing
- Internal synchronization

As a result, in most cases, the CPU is involved only when it also performs part of the processing (e.g. video decoding). In particular, for some situations which extend for long periods—such as, screen refresh/update and camera preview—the ARM complex is idle and can be powered down, reducing considerably the power consumption and extending the battery life. This is described further below.

2.1.2 Optimal Data Path

The integration combined with internal synchronization, allows avoiding unnecessary access to system memory, reducing the load on the memory bus and reducing further the power consumption. In particular:

- Input from a smart sensor (in YUV or RGB pixel formats) can be processed on-the-fly, before being stored in system memory.
- Output to a smart display can be processed on-the-fly, while read from system memory.
- In some cases, input from a sensor can be sent directly to display, without passing through system memory

2.1.3 Resource Sharing

The integrative approach enabled an efficient hardware design, in which the hardware is reused whenever possible for different applications. For example:

- The DMA controller is used for video capture, image processing, and data transfer to display
- The image conversion hardware is used both for captured video (from camera) and for video playback (from memory).

2.2 Flexible Connectivity

i.MX31 supports connectivity to a wide range of external devices—cameras, displays, graphics accelerators, TV encoders and decoders etc. This support was achieved by sophisticated configurability. The resulting flexibility is important also because the relevant interfaces are still significantly evolving, so expected changes need to be anticipated and accounted for.

2.2.1 Display Port

To support the wide variety of relevant display devices, two types of interfaces are provided:

- Synchronous interface: for transfer of display data in synchronization with the screen refresh cycle. This is the interface used for memory-less displays, and TV encoders. It is also used to transfer video to smart displays that have a video port.
- Asynchronous interface—serial and parallel: for random access (read and write) to the memory and registers of smart displays and graphics accelerators

The data bus is 18-bit wide (or less) and it can transfer pixels of up to 24-bit color depth. Each pixel can be transferred during 1, 2 or 3 bus cycles and the mapping of the pixel data to the data bus is fully configurable: the user can choose for each cycle an offset (on the bus) of each color component and which bits are transferred during this cycle.

For output to a TV encoder, a YUV 4:2:2 format is supported. Additional formats can be supported by considering them “generic data”—that is, they are transferred byte-by-byte, without modification, from the system memory to the display.

The display port does not include an address bus and the asynchronous interface uses “indirect addressing”—that is, the address and related commands are embedded in the data stream. This method was

adapted by display vendors to reduce the number of pins and wires between the display and the host processor. The problem with this method is that typical software running on the ARM CPU does not support it, since it assumes direct addressing (with a dedicated address bus). To allow such software a direct access to the memory of the display device (meaning, to map this memory to the system memory space), an automatic translation mechanism—from direct addressing to indirect addressing—is required. Implementing such a mechanism is particularly challenging, since at the present stage, indirect addressing is not standardized, and each vendor implements it differently. To address this problem, an innovative—patent pending—mechanism was developed. When the IPU is configured, the IPU driver stores in its internal memory a "template" specifying the access protocol to the display device. The IPU then uses this template to access the device without any further CPU intervention. In particular, SW running on the CPU can request an access to the display device, the IPU captures the request (through the AHB slave port) and performs the appropriate access procedure.

2.2.2 Camera Port

The interface with cameras (and TV decoders) is much more systematic (than for displays) and much less flexibility is required.

The interface receives one data sample per bus cycle, with 8-16 bits/sample. There is one exception—a nibble mode—in which 8-bit samples are received through a 4-bit bus, each during two cycles.

Synchronization signals (Vsync, Hsync) are either embedded in the data stream (following the BT.656 protocol) or transferred through dedicated pins.

The main pixel formats are YUV (4:4:4 or 4:2:2) and RGB. Any other format (e.g. Bayer or JPEG) can be received as "generic data", which is transferred, without modification, to the system memory.

2.2.3 Memory Port (AHB)

Although it is an internal port, considerable flexibility is required in the pixel formats, to comply with the various SW conventions. The main supported formats are:

- YUV: interleaved and non-interleaved, 4:4:4, 4:2:2 and 4:2:0, 8 bits/sample
- RGB: 8, 16, 24, 32 bits/pixel (possibly including some non-used bits), with fully configurable size and location for each color component. An additional component for transparency is also supported.

2.3 Versatile Image Processing

Last but not least, the IPU performs some very processing-intensive image manipulations, adding considerable processing power to the system: ~1.2 MHz of equivalent ARM11 performance.

- Filtering
  - Post filtering for MPEG-4: deblocking and deringing, high-quality proprietary algorithms
  - In-loop deblocking for H.264, as specified in the standard
- Resizing:
  Fully flexible resizing ratio, essentially between any two resolutions

- Color space conversion:

  Fully flexible conversion coefficients; can be combined with color adjustments
- Video and graphics combining:

  Transparency specified by a key color, global alpha value or per-pixel alpha values interleaved with the pixel components
- Rotation (90°, 180°, 270°) and inversion (horizontal/vertical)

Filtering and rotation are performed by reading/writing 2D blocks from/to system memory. The other tasks are performed row-by-row and, therefore, can be performed on the way from the (smart) sensor and/or to the (smart) display.

2.4 Autonomous Activities

The following activities are performed by the IPU completely autonomously:

- Screen refresh of a memory-less display
- Periodic update of the display buffer in a smart display
- Display of a camera preview (view-finder window)

The ARM platform—being idle—can be powered down. This is important, since these activities typically last for extended periods of time, so reducing the power consumption during these activities extends significantly the battery life.

2.4.1 Screen Refresh/Update

In many devices, most of the time the system is in idle state, the only activity being screen refresh. When a memory-less display is used, the screen refresh is performed by the IPU. When a smart display is used, the periodic screen refresh is performed internally, without i.MX31 intervention, so i.MX31 must only update the frame buffer, when this is needed. This activity is also performed by the IPU, which provides for it some specialized capabilities as described below.

2.4.1.1 Optimized Display Update

In some situations, in addition to a frame buffer in the display, there may be a “shadow” buffer in system memory, which the SW (including the operating system) must access. When this buffer is modified, these changes must be transferred to the display. To minimize power consumption, such data transfers should be avoided when there is no change in the data. In i.MX31, optimized updates are performed fully by HW. The EMI (External Memory Interface) includes a “snooping” mechanism, which monitors and records write accesses to the shadow buffer and upon detection, it sets a flag in the IPU. The IPU periodically checks the flag and if it is set, it performs one of the following

- Performs an update of the full frame, without any SW intervention.
- Interrupts the ARM CPU, to check (by monitoring flags in the EMI) what parts of the frame were modified and program the IPU to perform a more refined update.

2.4.1.2 Automatic Display of a Changing/Moving Image

When the system is idle, the user may want to display on the screen a changing image, such as an animation or a running message. In i.MX31, this can be performed automatically. The CPU stores in system memory all the data to be displayed, and the IPU performs the periodic display update without further CPU intervention. For an animation, there would be a sequence of distinct frames, and for a running message, there would be a single large frame, from which the IPU would read a "running" window. During this display update, the CPU can be powered down. When the IPU reaches the last programmed frame, it can perform one of the following:

- Return to the first frame (displaying a cyclic animation/message); in this case, the CPU can stay powered down.
- Interrupt the CPU to generate the next frames.

2.4.2 Camera Preview

To allow the user to prepare for capturing motion video (or a still image), a "view-finder" window is displayed on the screen. The data flow in this situation (with a smart sensor) is described in Figure 3. The IPU performs all the necessary processing. Moreover, it also performs all required synchronization and control.

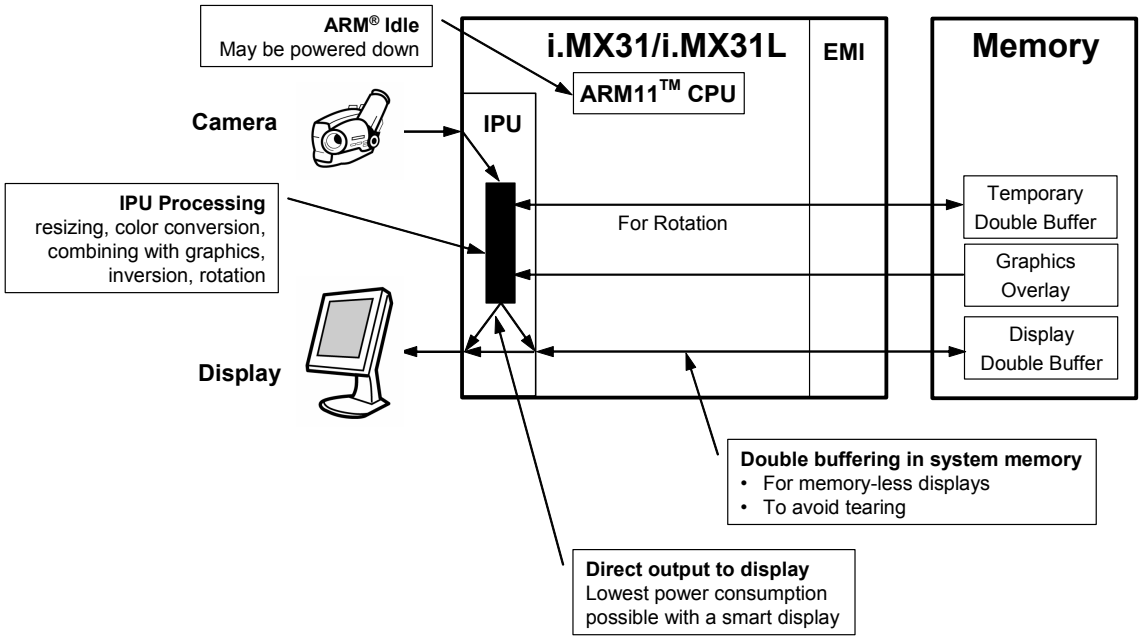

Figure 3. Camera Preview—Data Flow

Synchronization for a view-finder video stream can be quite a challenge. This is because both the sensor and the display have rigid operation cycles—sensor array scan and display refresh—and the periods of these cycles are different: at most 30 Hz for the sensor and at least 50 Hz for the display. Without proper synchronization, annoying "tearing" lines will appear on the display.

The IPU provides two complementary methods to prevent tearing. The robust method is called "page-flip" double buffering. The sensor output is written to one buffer, the data for display is read from another buffer

and at the appropriate time the IPU switches the roles of the buffers. This method can be used in all situations, and its only disadvantage is its need to access the system memory. In some cases, it is possible to completely avoid the access to system memory, reducing further the power consumption. This is possible when both the sensor and display are “smart,” the display can receive an external Vsync trigger and the display’s refresh rate can be exactly twice that of the sensor’s frame rate. In such a situation, the IPU can use the Vsync signals received from the sensor to generate Vsync signals for the display, allowing the display of a full-screen tearing-less view-finder.

3 Video Applications

The i.MX31 processor is optimized to support a variety of video applications. The video processing chain, and its implementation, is described in Figure 4.

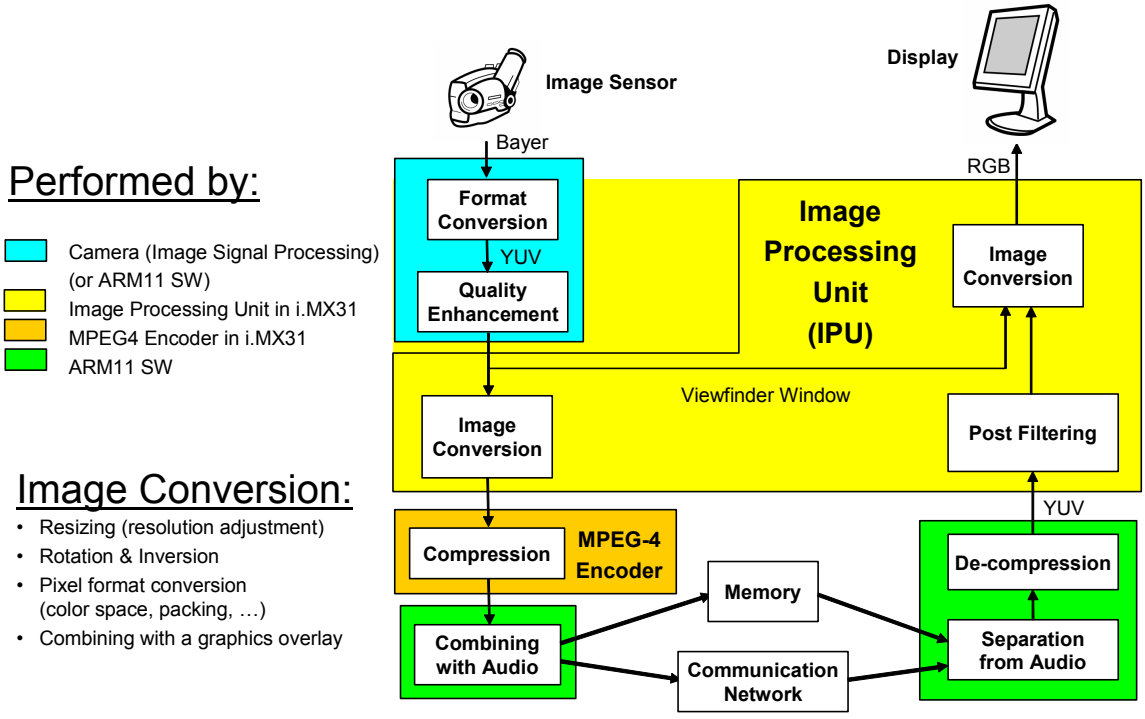

Figure 4. Video Processing Chain

This implementation is a result of a smart trade-off between performance, flexibility, and additional considerations as follows:

- Image processing required for a camera preview is performed fully in HW, to allow the CPU to be powered-down in this stage.
- Encoding in MPEG-4 SP and H.263-baseline formats is performed fully in hardware. This provides high performance and low power consumption, without compromising quality.

  These formats are sufficient for most purposes:

  — These are the formats used for video conferencing.

  — They are supported by most video players.

  Therefore, encoding in other formats is left to software.

- Decoding:

  For MPEG-4 SP, post-filtering is performed fully in HW, providing 75% acceleration

  For H.264—the most processing-intensive format—the deblocking filter is performed in HW, providing 30% acceleration

  The powerful ARM11 CPU (including its 2-level cache system) provides the flexibility to decode, at high rate, any other currently relevant format (at least HVGA @ 30 fps), as well as possible future extensions.

A system view of the data flow during video capture and playback, using the MPEG-4 SP format, is given in Figure 5 and Figure 6. The flow for other formats is very similar.

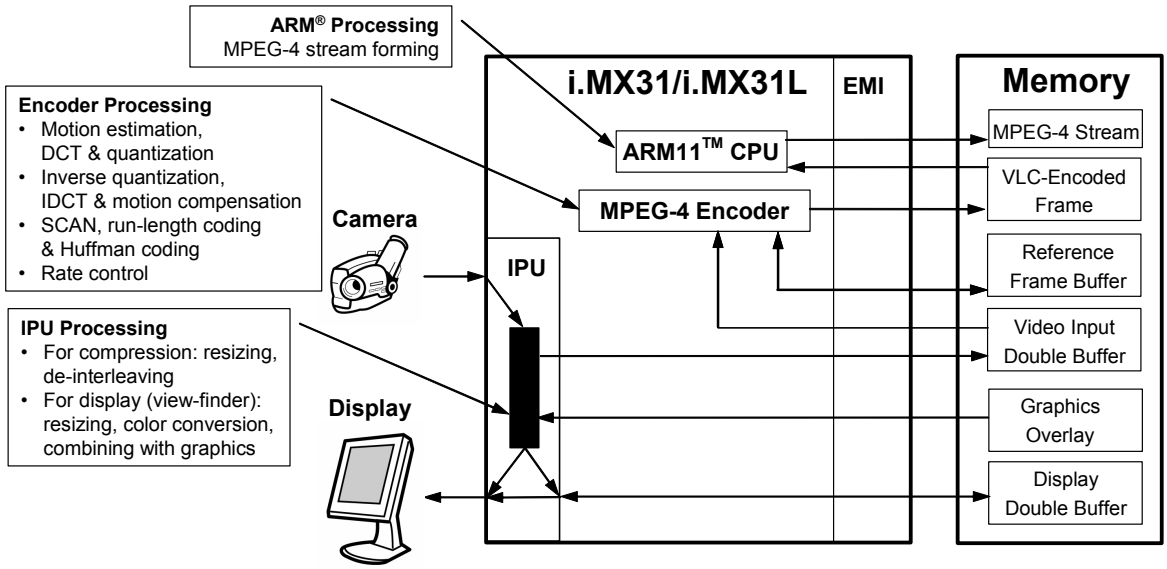

Figure 5. MPEG-4 SP Video Capturing—Data Flow

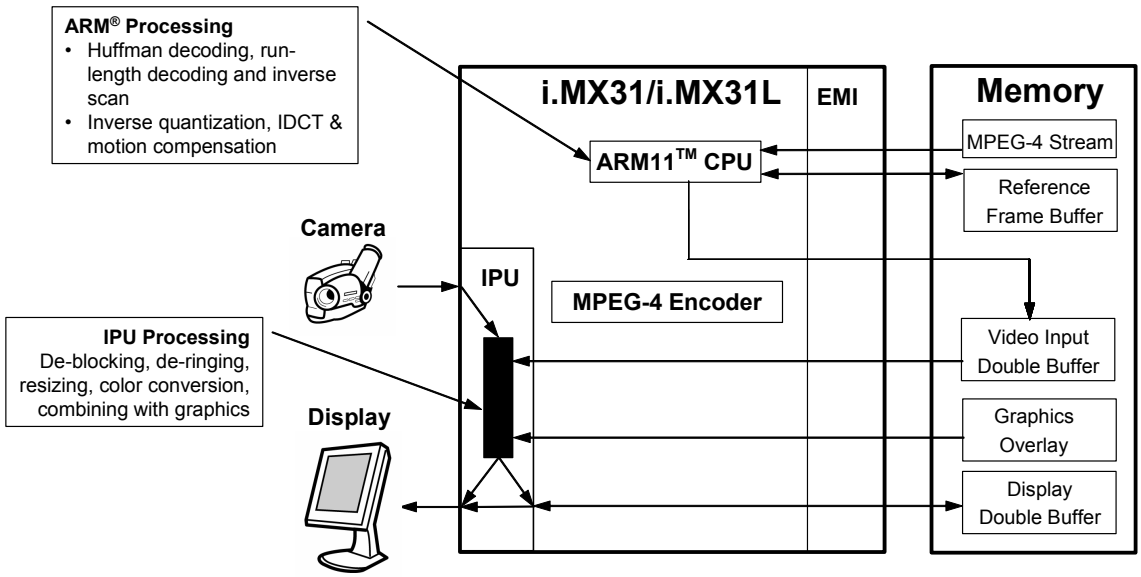

Figure 6. MPEG-4 SP Video Playback—Data Flow

The video processing performance is summarized in Table 3 (CPU load), Table 4 (memory bus load), and Table 5 (chip power consumption). These performance values are based on estimations and simulations, using the settings summarized in Table 2.

Table 2. Video Performance - Simulations Settings

CPU Clock Rate	266/399/532 MHz (the minimum required)
Bus clock rate	133 MHz
External Memory	16-bit DDR SDRAM
Memory latency (out-of-page/in-page)	**Read: 9/3-1-1-1 Write: 6/0-1-1-1**
Display	Smart
Camera	Smart

The following assumptions are made:

- In decoding, the IPU is used to perform post-filtering (for MPEG-4/H.263) or in-loop deblocking (for H.264)
- Resizing/rotation/combining is not included (these would be performed fully by the IPU, affecting the bus load and power consumption, but not the ARM CPU load)
- Only video processing is included (e.g. no audio)

The CPU load provided in Table 3 is mostly described as the clock rate required to perform processing of 30 fps. Where this processing exceeds 500 MHz, the table presents instead, in brackets, the processing load for 15 fps.

Table 3. Video Processing - ARM11 CPU Load (clock rate at 30 fps or frame rate at 500 MHz)

Use Case	QCIF 128 Kbps	QVGA 256 Kbps	CIF 384 Kbps	HVGA 0.5–1.0 Mbps	VGA 1–2 Mbps
Decode (and Display)					
MPEG-4 SP, H.263 baseline	22 Mcps	65 Mcps	90 Mcps	140–190 Mcps	320–400 Mcps
H.264 baseline	40 Mcps	130 Mcps	180 Mcps	280–380 Mcps	[320–400 Mcps]
(Capture and) Encode					
MPEG-4/H.263—in HW	2 Mcps	6 Mcps	8 Mcps	13 Mcps	30 Mcps
MPEG-4/H.263—in SW	120 Mcps	380 Mcps	[270 Mcps]	[430 Mcps]	–
H.264 baseline	400 Mcps	–	–	–	–

Table 4 and Table 5 refer to the same use cases as Table 3, (meaning, as shown in brackets, frame rate is reduced to 15 fps).

Table 4. Video Processing - Memory Bus Load [Mbytes/s][1]

Use Case	QCIF 128 Kbps	QVGA 256 Kbps	CIF 384 Kbps	HVGA 0.5–1.0 Mbps	VGA 1–2 Mbps
Decode and Display					
MPEG-4 SP, H.263 baseline	8 (2%)	40 (10%)	60 (15%)	70 (18%)	160 (44%)
H.264 baseline	13 (3%)	60 (15%)	90 (23%)	130 (33%)	[130 (33%)]
Capture and Encode					
MPEG-4/H.263—in HW	20 (5%)	65 (16%)	90 (23%)	130 (33%)	260 (65%)
MPEG-4/H.263—in SW	30 (8%)	130 (33%)	[90 (23%)]	[130 (33%)]	–
H.264 baseline	45 (11%)	–	–	–	–

[1] Bus utilization is shown in parentheses.

Table 5. i.MX31 and i.MX31L Power Consumption [mW]

Use Case	QCIF 128 Kbps	QVGA 256 Kbps	CIF 384 Kbps	HVGA 0.5–1.0 Mbps	VGA 1–2 Mbps
Decode and Display					
MPEG-4 SP, H.263 baseline	17	50	70	110–130	200–250
H.264 baseline	30	85	130	190–240	[190–240]
Capture and Encode					
MPEG-4/H.263—in HW	20	60	80	120	240
MPEG-4/H.263—in SW	90	270	[180]	[270]	–
H.264 baseline	220	–	–	–	–

4 Conclusion

Big multimedia experiences can come in small wireless mobile entertainment devices. As shown in this paper, a highly integrative and modular silicon architecture is most desirable to achieve the optimal balance between multimedia performance and extended battery life. Performance is achieved via Freescale's Smart Speed™ Technology, a carefully balanced interweave of hardware acceleration and software flexibility that enables many functions to operate in parallel. In the i.MX31 processors, the most computationally-intensive parts of video processing are offloaded from the ARM CPU and accelerated in hardware via Freescale's Image Processing Unit. This keeps power demands very low, but does not come at the expense of design flexibility: engineers still have a choice in the decoders they choose to implement that best suits their requirements.

Battery life is further extended by optimized data path, superior control capabilities, and advanced power management techniques, such as dynamic voltage and frequency scaling and dynamic process temperature compensation (see Freescale White Paper: i.MX31 and i.MX31L Power Management).

Mobile entertainment devices based on the i.MX31 applications processors can drive hours and hours of multimedia performances that satiate even the most demanding consumers.

NOTES

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: IMX31MULTIWP
Rev. 0
02/2006

freescale™
semiconductor