Meeting the Challenges of N-way Videoconferencing

By Anil Kumar, Lead, Voice and Video over IP Team
Sriram Sethuraman, Technologist, Distinguished Member Technical Staff
Ittiam Systems

Transitioning videophones and videoconferencing solutions from expensive multi-chip solutions to a single System-on-a-chip (SoC) is essential to fueling their mass-market adoption. Compared to other applications such as media player/recorder, the number of simultaneous components that need to co-exist in a latency critical n-way audio-visual communication application presents several challenges. Figure 1 illustrates the number of concurrent tasks that need to happen in a 3-way conferencing system. In a true multi-point conferencing unit, there will be additional requirements to perform the composition/mixing of multiple decoded video and speech streams prior to encoding the composite version. Multiple versions of the mixed speech may have to be encoded as well.

The TI DM6446 ™ is a very highly integrated SoC that features a RISC CPU (ARM9) and a powerful DSP (C64x+) in addition to a rich set of on-chip peripherals such as video capture, video display, and Ethernet MAC. This architecture allows partitioning of the software into UI, call control and networking tasks running on the ARM and all speech and video processing to happen on the DSP. The DSP sub-system with its co-processors provides the necessary processing power to consider even high-end n-way conferencing. The Enhanced DMA controller (EDMA) and the set of carefully designed on-chip data busses ensure a high throughput from external memory to the ARM, DSP, co-processors and peripherals.

Harnessing the power of such a device to realize the n-way conferencing capability, and to ride on its future versions, poses the following major challenges:

- Optimizing the components to leverage platform features
- End-to-end Latency reduction and handling network jitter
- Keeping the design modular and flexible to facilitate easy migration to future versions and be scalable in terms of adding more instances or features

Optimizing the components

The approach to optimizing the components for a specific platform consists of:

1. Balancing the memory access time and processing load on the DSP/co-processors
2. Leveraging DMA capabilities to hide the memory access time against processing
3. Re-designing code to be compiler friendly and introducing pragmas to improve compiler performance.
4. Tuning the code and data placement to minimize cache thrashing due to multiple modules within a component and increasing the amount of internal memory available to each component through scratch memory overlays.
5. Determining and minimizing the impact of cache thrashes due to pre-emption by other components which are unavoidable in latency critical applications.
6. Minimizing the development time by leveraging tools such as simulators, linear assembler, etc. and testing the components pro-actively as in-system debugging will be very time consuming.

Addressing end-to-end latency and network jitter

End-to-end Latency is very important for the quality of the conference and this latency can be divided into a controllable component and an uncontrollable component. Controllable components include all processing related latencies introduced at the sending and receiving ends. Uncontrollable component is typically the network latency. End-to-end latency can be reduced by going to a finer granularity of pipelining of the components (for e.g., slice level capture, encode, decoding and display for video). However, this conflicts with achieving higher processing performance. Some instances of this are, increased number of task preemptions (for example, to send a packet of data as soon as it is produced), increased ARM/DSP interactions to send or receive data on a finer granularity, and reduced pipelining efficiency within components. Hence, a careful balance between task scheduling and performance impact needs to be struck. Compositing video from multiple end-points also goes against achieving low latency as the compositor needs to wait for corresponding portions of the frame from each end point for scaling (if necessary) and feeding the resultant output to the encoder.

Though network latency cannot be directly helped, monitoring the network jitter gives an indication of the network congestion and by adapting the output bit-rate, congestion can be eased. Allowing speech to continue with minimal losses or delays provides a higher perceived quality of the session. The rate adaptation is made complex with the multiple end-points, as reacting to the worst link affects the quality seen by all the end-points. An important part of managing latency is using an adaptive jitter buffer (AJB). The AJB delays video display and/or voice playout in order to synchronize the speech and video, as well as to provide a small backlog in case later packets are delayed. AJB design can also be used to combat clock skew between the two ends. Along with good concealment algorithms, and time scale modification, the overall perceived quality in case of packet loss and jitter changes can be drastically improved.

A good and robust Acoustic Echo Canceller (AEC) is essential for good quality conversation. It is very important to have a full-duplex AEC to give a natural conversational experience for hands-free telephony, especially in a jittery and latency prone network. With additional conferencing feature supported, the AEC should be designed to be more sensitive for feedback paths from multiple peers.

Modularity, Flexibility, Programmability

Given the rapidly changing landscape of the video telephony industry, modularity, flexibility, and programmability are critical shorten the time to market with new SoCs. Some of the considerations in this front are:

1. Supporting a wide base of codecs to ensure interoperability with other similar devices

2. Adopting API standards such as TI's eXpressDSP Multimedia Interface Algorithm Standard (XDMI) or Khronos OpenMAX IL standards to facilitate plug in or plug out of various components to a class (such as video encoders, speech decoders, etc.) while shielding the application from the specifics of the codec.

3. Developing an ARM/DSP communication framework that eases ARM side application development by automating the resource management and scheduling of the components and providing a seamless interface whether a component executes on ARM or DSP side, while providing sufficient control to achieve the latency and performance goals. For instance, it may be beneficial

from the cache performance point of view in a n-way session to group the calls to video or speech decoder across multiple instances.

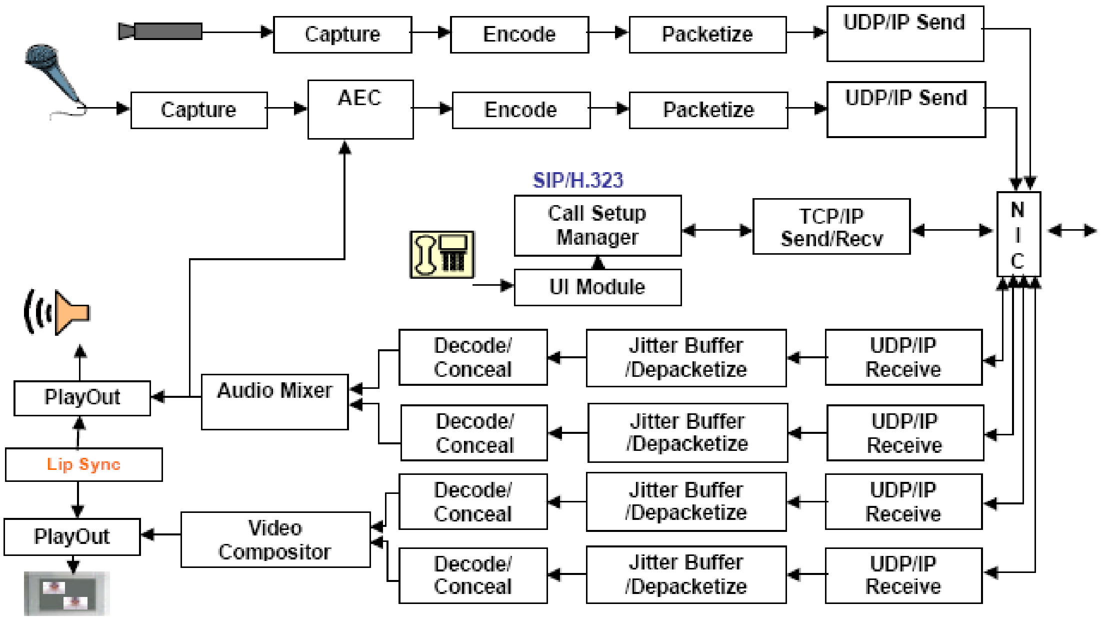
4. Developing a software architecture that cleanly abstracts the call control, media processing and user interface features to shield each of them from changes in the other. The software architecture also needs to provide appropriate OS and driver abstraction.
5. Ensuring co-existence with other applications (such as running the videophone application along with a video on-demand application in an IP-STB).

Ittiam product offering

Ittiam Systems offers an n-way Video Conferencing reference design based on TI's DM6446 and DM64x platforms. The reference design supports multiple video codecs, (H.264, H263 and MPEG-4) and a whole suite of speech codecs. Ittiam's reference design is a complete hardware and software development platform for the rapid development of client video phone designs.

Figure 1:

Caption: Block Diagram of an N-way voice and video conferencing system


Capture
Encode
Packetize
UDP/IP Send
Capture
AEC
Encode
Packetize
UDP/IP Send
SIP/H.323
Call Setup Manager
TCP/IP Send/Recv
N I C
UI Module
PlayOut
Audio Mixer
Decode/ Conceal
Jitter Buffer /Depacketize
UDP/IP Receive
Decode/ Conceal
Jitter Buffer /Depacketize
UDP/IP Receive
Lip Sync
Decode/ Conceal
Jitter Buffer /Depacketize
UDP/IP Receive
PlayOut
Video Compositor
Decode/ Conceal
Jitter Buffer /Depacketize
UDP/IP Receive