White Paper

G.PAK-C6472: A Rapid Way to Build High Density Voice Solutions

Adaptive Digital Technologies, Inc.
September 30, 2009

Issue 1

Revision History

Issue	Revision History	Date	Initials
1	Initial Creation	9/30/2009	SDK

Portions of this document contain advance information.

1. Introduction

In the past decade, Voice over IP (VoIP) has gone from the “Can this work well enough?” and “Will it be cost effective?” to “How quickly can we get our next VoIP-based product to market?”

VoIP equipment comes in many forms, sizes and capabilities. There are IP phones and intercoms IP PBXs and gateways, ATAs. There are wired, cordless, and wireless devices. There are high, medium, and low density devices. And there are narrowband and wideband systems. And within each type of product and system, the designer must choose from many different possible features – vocoders, tone handling, fax relay, voice quality enhancement.

Programmable DSPs are ideally suited to handle such a wide variety of needs. And no company offers a wider range of available programmable DSPs than Texas Instruments. But the DSP chip is only part of the solution. Getting the right software is arguably the more difficult problem.

In the early days of VoIP, software designers were faced with building blocks such as vocoders, tone detectors, tone generators, echo cancellers, fax modems, etc. The software designers needed to complex, efficient, real-time applications to glue together the right building blocks into a DSP, write device drivers. It is a daunting and time consuming task even for those familiar with the details of VoIP.

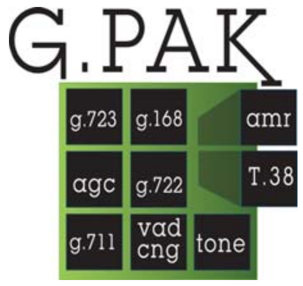
The software technology has caught up with the DSP chip technology in that it is now possible to obtain a complete DSP software solution to run on a DSP without the need to do any programming on the DSP. But when using such a highly integrated solution, one tends to lose the flexibility.

G.PAK, Adaptive Digital’s configurable VoIP DSP software, provides the needed flexibility while maintaining the high degree of integration. G.PAK can be used in equipment ranging from ATAs to high density gateways using a broad range of TI DSPs. G.PAK can be configured to include only the necessary building blocks. And, for those who need to tinker beyond the capabilities of standard G.PAK, **application source code is available**.

G.PAK can run in low-end applications on the TMS320C642X chips to the high-end, high-density applications using the TMS320C6472 device. With six C64X+ the C6472 provides the necessary CPU power for high density applications. Furthermore, the C6472 shared memory architecture is ideal for running the same DSP software application across multiple cores. And the C6472 provides the right set of peripherals for voice infrastructure equipment including three TSIPs (Telecom Serial Interface Port) and two Ethernet MACs.

2. G.PAK Overview

Figure 1 is a block diagram of a system that includes G.PAK. There are two components included in G.PAK – the DSP software and the Host API. The DSP software is a VoIP application that handles the complete data flow between the TDM and Packet interfaces. The second G.PAK component is the G.PAK API, which is the second component of G.PAK. The G.PAK API, provided in “C” source code format, enables the host application to download and configure the DSP, set up and tear down channels, and read and write packets. The G.PAK API is designed to be flexible enough to run under any operating system.



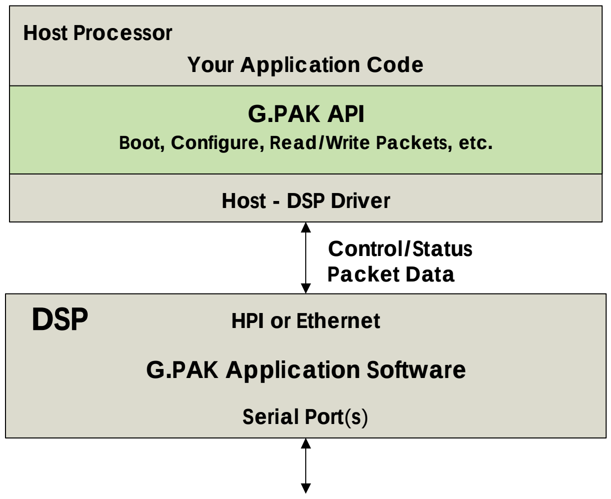


Figure 1 is a block diagram of a system that includes G.PAK

Sample APIs include:

Configuring a Channel:

```
gpakConfigureChannel( . . )     /* Configure a DSP's Channel. */
```

When a New Network Packet is Received:

```
gpakSendPayloadToDsp( . . )  /* Send a Payload to a DSP's Port. */
```

To Retrieve an Outbound Payload:

```
gpakGetPayloadFromDsp( . . ) /* Read a Payload from a DSP. */
```

Tearing Down A Channel

```
gpakTearDownChannel( . . )    /* Tear Down a DSP's Channel. */
```

Retrieving Status:

```
gpakGetSystemStatus( . . )     /* Read a DSP's System Status. */
gpakGetChannelStatus( . . )    /* Read a DSP's Channel Status. */
```

2.1 G.PAK Channel Types

Each G.PAK DSP supports a fixed number of channels. The number of channels available in a DSP is dependent on the DSP type and the capabilities selected at build time.

There are several types of channels: TDM to Packet, PCM to TDM, Packet to Packet, TDM to Conference, Packet to Conference, and Conference Composite. Each channel in a DSP can be dynamically setup as any type at run time. Frame sizes, audio codec types, and tone detection types are selected when a channel is setup from capabilities configured at build time.

All channels are designed to operate as full duplex channels. A full duplex channel may be configured to operate as a half duplex by setting the end points of one-half of the full duplex channel to NULL end-points

2.1.1 TDM to Packet Channel Type

TDM to packet channel types provide a linkage between analog telephones (TDM data) and digital telephones (packet data). They are full duplex channels used to convert TDM data to/from packet data. This type of channel is typically used in a voice-over-packet application. Channels of this type have two threads of processing as shown in Figure 1.

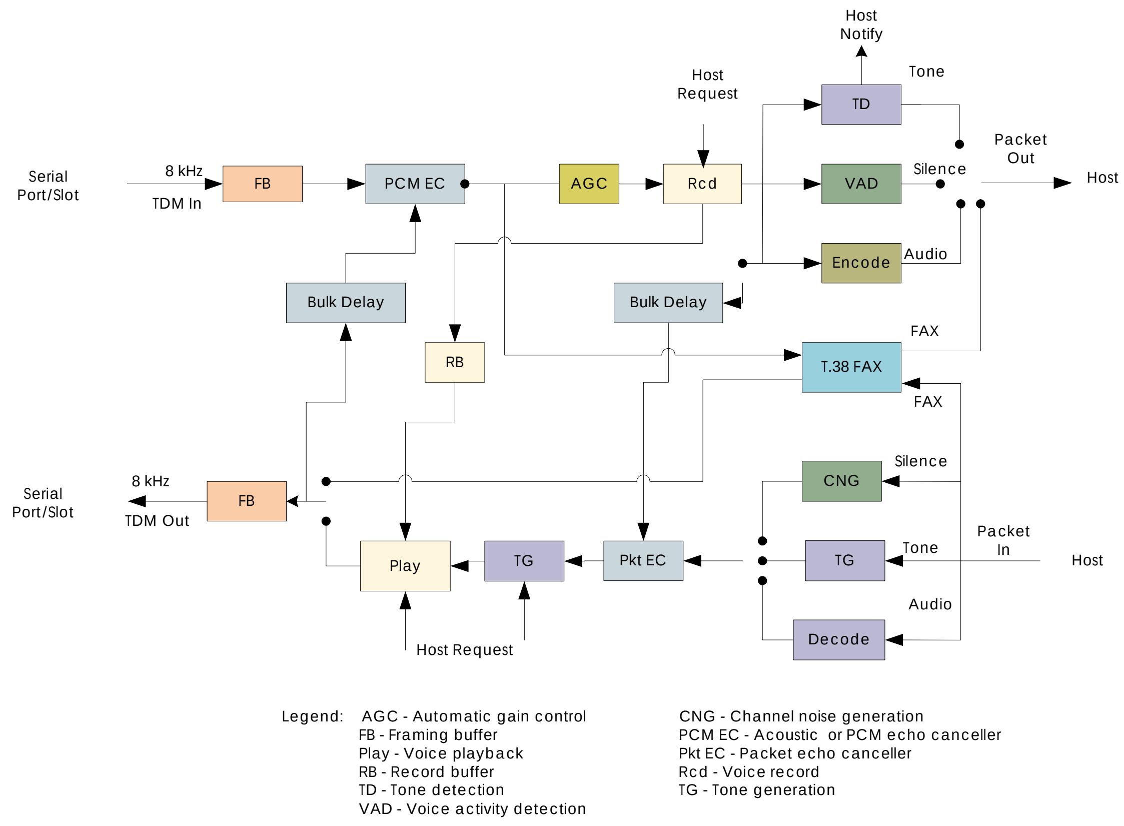


Figure 1 – TDM to Packet Channel

TDM to packet processing. The TDM to packet processing thread inputs 8 kHz PCM samples from a TDM time slot and buffers as many samples as are needed to build an output frame. When a frame's worth of samples are buffered, the TDM input data is optionally passed through the PCM echo canceller or the acoustic echo canceller, automatic gain control (AGC), the voice recorder, the tone detecter (TD), T.38 FAX relay, and the voice activity detector (VAD).

Tone detection is optionally performed on the PCM data after the G.168 echo cancellation [1] but prior to the non-linear post-processing of the echo canceller. G.PAK notifies the host at the start and end of each tone. When tone relay is enabled, tone packets are generated for each frame until the tone ends; an end of tone packet is generated when the tone stops.

Voice activity detection is optionally performed if no tone is detected. A silence packet is generated when VAD does not detect activity,

Voice encoding is performed whenever tone, T.38 FAX, or silence packets are NOT generated. The channel's encoder type identifies which vocoder will be used. The vocoder output is formatted and packed into the vocoder's packet payload and sent to the host for transmission.

Packet to TDM processing. The packet to TDM processing thread reads a packet from the host control processor and generates PCM samples according to the received packet type. If a silence packet is received, comfort noise is generated. If a tone packet is received, the specified tone is generated. If a T.38 packet is received, the corresponding FAX tones are generated. If an audio packet is received, it is decoded according to the input packet codec type.

The generated PCM samples are then optionally passed through the packet echo canceller and buffered. Buffered samples may be overridden by either host requested generated tones or host requested playback of pre-recorded messages. The buffered samples are then output to the channels TDM time slot.

The PCM echo canceller, acoustic packet echo canceller, AGC, VAD, tone detection, T.38 Fax relay, and tone relay algorithms are independent options selectable at channel setup.

2.1.2 TDM to TDM Channel Type

TDM to TDM channel types provide a linkage between two analog phones. They are full duplex channels used to transfer PCM data between TDM slols. These channels are typically used for PCM echo cancellation or time slot interchanging. They are implemented as two half duplex channels (A-to-B and B-A) as shown in Figure 2.

TDM to TDM Processing. Each half duplex channel inputs 8 kHz PCM samples from a serial port time slot and buffers as many samples as specified by the framing size. When enough samples are buffered, the PCM input data is optionally passed through the echo canceller (acoustic or PCM), tone detectors (TD), automatic gain control (AGC), noise suppression, voice recording, tone generation (TG), and voice playback before being buffered for output. The buffered output samples are then delivered to a TDM slot.

The echo canceller, tone detection, automatic gain control, noise suppression, and tone generation algorithms are independent options that are configurable at channel setup.   The voice record, tone generation, and voice playback functions are initiated by host commands at run time.

[1] Performing tone detection on the pre-NLP echo cancellation data provides more reliable tone detection.

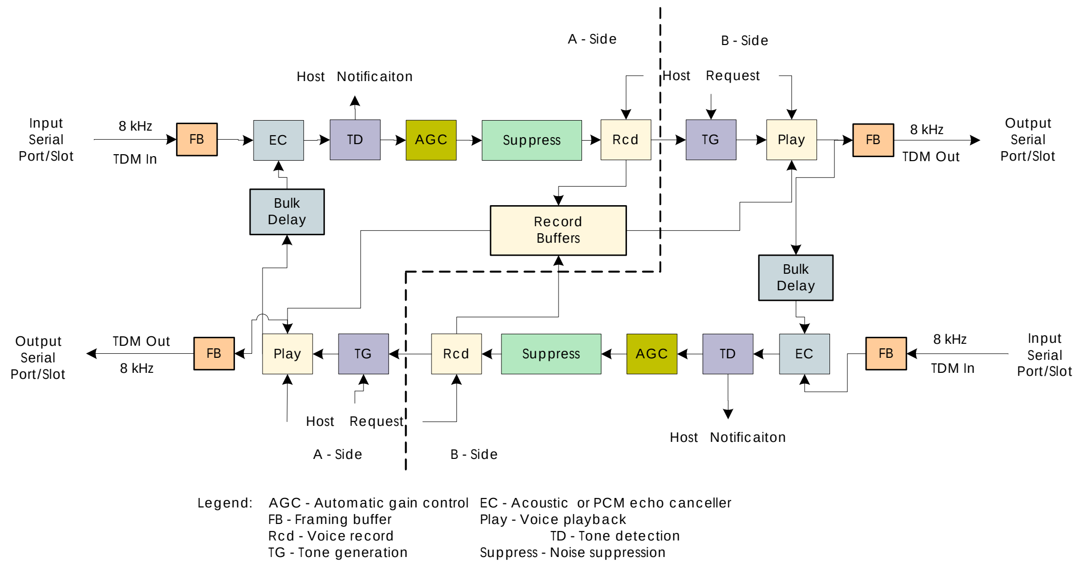


Figure 2– TDM to TDM Channel

2.1.3 Packet to Packet Channel Type

Packet to packet channels provide a linkage between two digital telephones. These channels are typically used in gateway applications to perform frame size conversions, codec type conversions, and/or echo cancellation of packet data. This type of channel can also be used to convert silence and tone packets to audio packets or to convert audio packets to silence and tone packets. Channels of this type have two streams of processing as shown in Figure 3. Each G.PAK packet to packet channel is half duplex; however, these channels are always allocated in pairs (channels A and B) to allow full duplex operation.

Packet decode processing. The packet decoding thread reads a channel 's packet from the host control processor and generates and buffers 8 KHz samples according to the received packet type. If a silence packet is received, comfort noise is generated. If a tone packet is received, the specified tone is generated. If an audio packet is received, it is decoded according to the input packet codec type. The generated PCM samples are buffered for encoding by the paired channel.

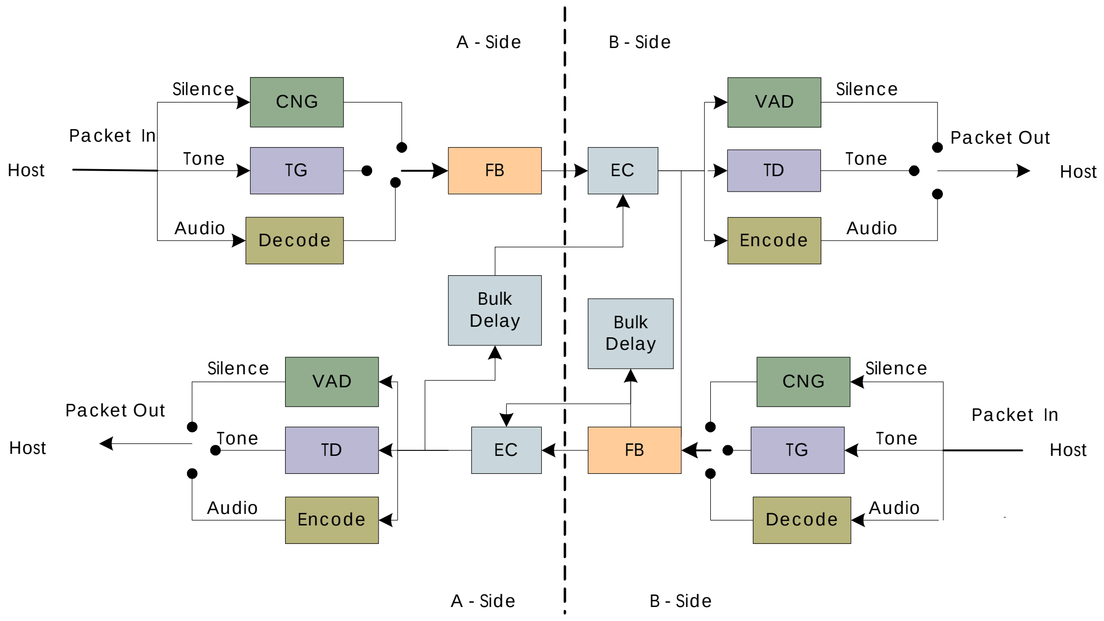
Packet encode processing. When enough of the paired channel's PCM samples are buffered to build an output frame, this PCM data is optionally passed through the packet echo canceller, the tone detecter (TD) and the voice activity detector (VAD).

Tone detection is optionally performed on the PCM data after the G.168 echo cancellation [2] but prior to the non-linear post-processing of the echo canceller. G.PAK notifies the host at the start and end of each tone. If tone relay is enabled, tone packets are generated for each frame until the tone ends; an end of tone packet is generated when the tone stops.

[2] Performing tone detection on the pre-NLP echo cancellation data provides more reliable tone detection.

Voice activity detection is optionally performed when no tone is detected. A silence packet is generated when VAD does not detect activity,

Voice encoding is performed whenever tone or silence packets are NOT generated. The channel's encoder type identifies which vocoder will be used. The vocoder output is formatted and packed into the vocoder's packet payload and sent to the host for transmission to the paired device.



Legend: CNG - Comfort noise generation; EC - Packet echo canceller; FB - Framing buffer; TD - Tone detection; TG - Tone generation; VAD- Voice activity detection

Figure 3– Packet to Packet Channel

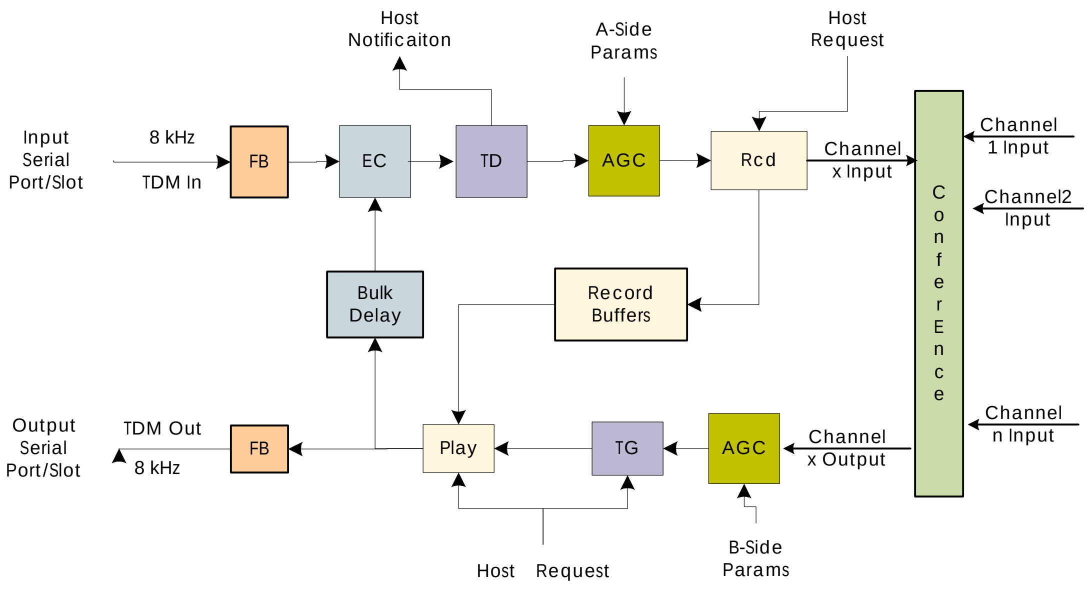
Echo cancellation. Channel A's echo canceller cancels the echo of channel B's input from channel A's output. Similarly, channel B's echo canceller cancels the echo of channel A's input from channel B's output.

Each of the voice enchancement algorithms: packet echo canceller, VAD, and tone processing (detection and relay); are independent options selectable at channel setup.

2.1.4 TDM to Conference Channel Type

TDM to conference channel types are used to allow analog telephones to join a conferencing call. These channels invoke are processed as two half duplex channels as shown in Figure 4.

Conference input. The conference input function inputs 8 kHz PCM samples from a TDM slot and buffers as many samples as needed for a frame used by the conference. When enough samples are buffered, the PCM data is optionally passed through PCM echo cancellation, tone detection and automatic gain control (AGC) alogrihms before being passed to the Conferencing algorithm for use as one of the conference inputs. The PCM data can be recorded at the host's request for future playback.



Legend: AGC - Automatic gain control
FB - Framing buffer
Rcd - Voice record
TG - Tone generation
EC - Acoustic or PCM echo canceller
Play - Voice playback
TD - Tone detection

Figure 4 – Conference TDM Channel

Conference output. The conference output function buffers the conference's output data for the channel. Each channel has its own conference output data that removes a speaker's voice from his own output. The buffered samples are optionally passed through AGC and then output to a TDM slot. The host may override the TDM output with either a previously recorded message or a generated tone.

The pcm canceller and automatic gain control algorithms are independent options selectable at channel setup.

2.1.5 Packet to Conference Channel Type

Packet to conference channels allow digital telephones to join a conferencing call. These channels are processed as two half-duplex channels as shown in

Figure 5.

Conference input. The conference input function reads a packet from the host control processor and generates PCM samples according to the received packet type. If a silence packet is received, comfort noise is generated. If a tone packet is received, the specified tone is generated. If an audio packet is received, it is decoded according to the input packet codec type.
The generated PCM samples are then optionally passed through the packet echo canceller and automatic gain control algorithms before being passed to the Conferencing algorithm for use as one of the conference inputs.

Conference output. The conference output function buffers the normalized sum of all conference member's input data minus the channel's own input data. The samples are buffered until there are enough samples to build an output frame. When enough samples are buffered, the samples are optionally passed through the tone detecter (TD) and the voice activity detector (VAD).

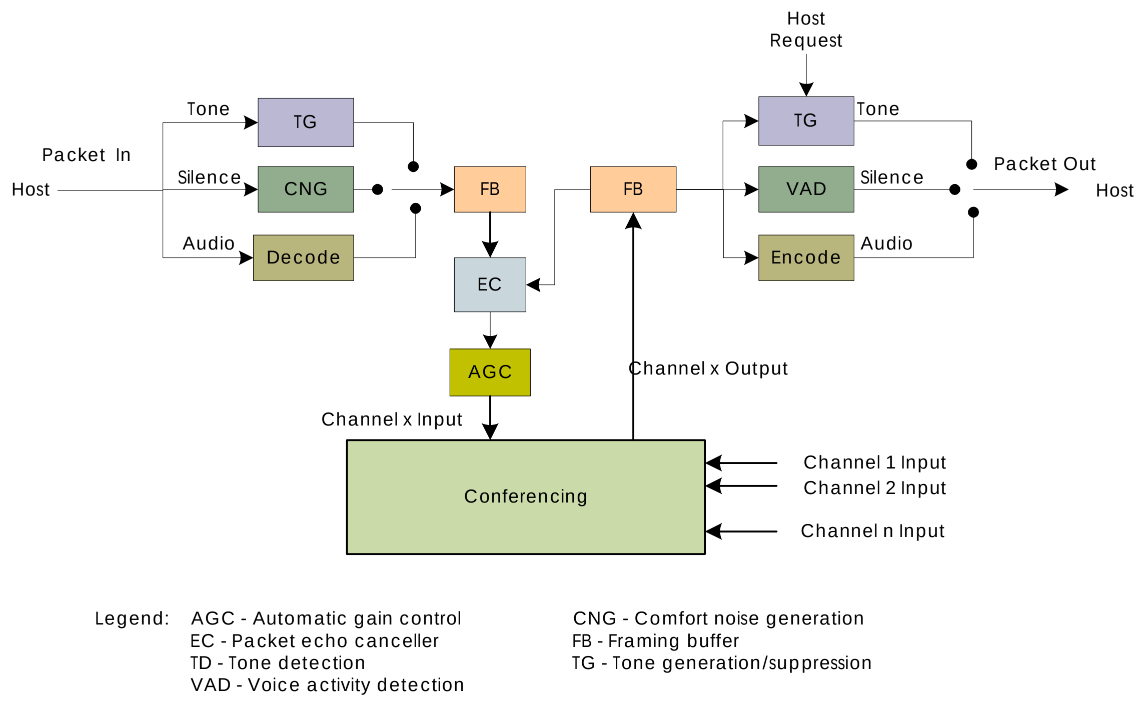


Figure 5 – Conference Packet Channel

G.PAK notifies the host at the start and end of each tone. If tone relay is enabled, tone packets are generated for each frame until the tone ends; an end of tone packet is generated when the tone stops.

Voice activity detection is optionally performed when no tone is detected. A silence packet is generated when VAD does not detect voice activity,

Voice encoding is performed whenever tone or silence packets are NOT generated. The channel's encoder type identifies which vocoder will be used. The vocoder output is formatted and packed into the vocoder's packet payload and sent to the host for transmission.

The packet echo canceller, tone detection, tone relay, voice activity detection, and automatic gain control algorithms are independent options selectable at channel setup. Input packets are read and output packets are generated with the same frequency as the conference's frame size.

2.1.6 Conference Composite Channel Type

Conference Composite channel types are used to output the composite output from a conference as PCM data and/or packet data as shown in Figure 6.

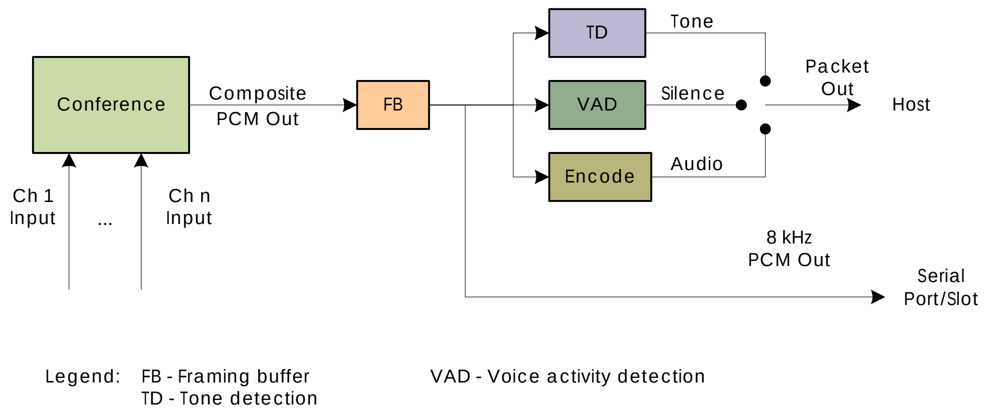


Figure 6 – Conference Composite Channel

The conference composite function buffers the normalized sum of the dominant conference member's input data into a frame buffer. The samples are buffered until there are enough samples to build an output frame. When enough samples are buffered, the samples are optionally passed through the tone detecter (TD) for tone relay and the voice activity detector (VAD).

If tone relay is enabled, tone packets are generated for each frame until the tone ends; an end of tone packet is generated when the tone stops.

Voice activity detection is optionally performed when no tone is detected. A silence packet is generated when VAD does not detect voice activity,

Voice encoding is optionally performed whenever tone or silence packets are NOT generated. The channel's encoder type identifies which vocoder will be used. The vocoder output is formatted and packed into the vocoder's packet payload and sent to the host for transmission.

PCM data is optionally transferred to a TDM slot.

The tone detection, tone relay, and voice activity detection algorithms are independent options selectable at channel setup. Output packets and tone event packets are generated with the same frequency as the conference's frame size.

For more detailed information about G.PAK, please refer to the G.PAK Users Guide.

3. DSP Selection and G.PAK Build Process

The G.PAK Build Process refers the process that a designer goes through to obtain a G.PAK solution that meets his or her needs. This step-by-step process is outlined below.

1. Identify the required building blocks (vocoders, echo cancellers, tone handling, fax relay, RTP/SRTP, TCP/UDP/IP, etc.)
2. Determine the channel density
3. Narrow down the TI DSP selecting using Adaptive Digital's DSP Resource Wizard based upon the above requirements. The DSP Resource Wizard is a web-based applet that can be accessed at http://wizard.adaptivedigital.com.
4. Identify the required DSP interfaces. (McBSP, TSIP, HPI, PCI, Ethernet, etc.)
5. Select the appropriate TI DSP based upon the results obtained from the DSP resource wizard, further narrowing down the selection based upon the DSPs that include the required interfaces.
6. Identify the required channel types (TDM to TDM, TDM to Packet, etc.)
7. Using Adaptive Digital's G.PAK Build utility, configure G.PAK with the all the above requirements and build the DSP image. The G.PAK build utility interoperates with TI's Code Composer Studio and the associated compile tools.

When you are done this process, you will have a downloadable DSP image for the selected DSP!

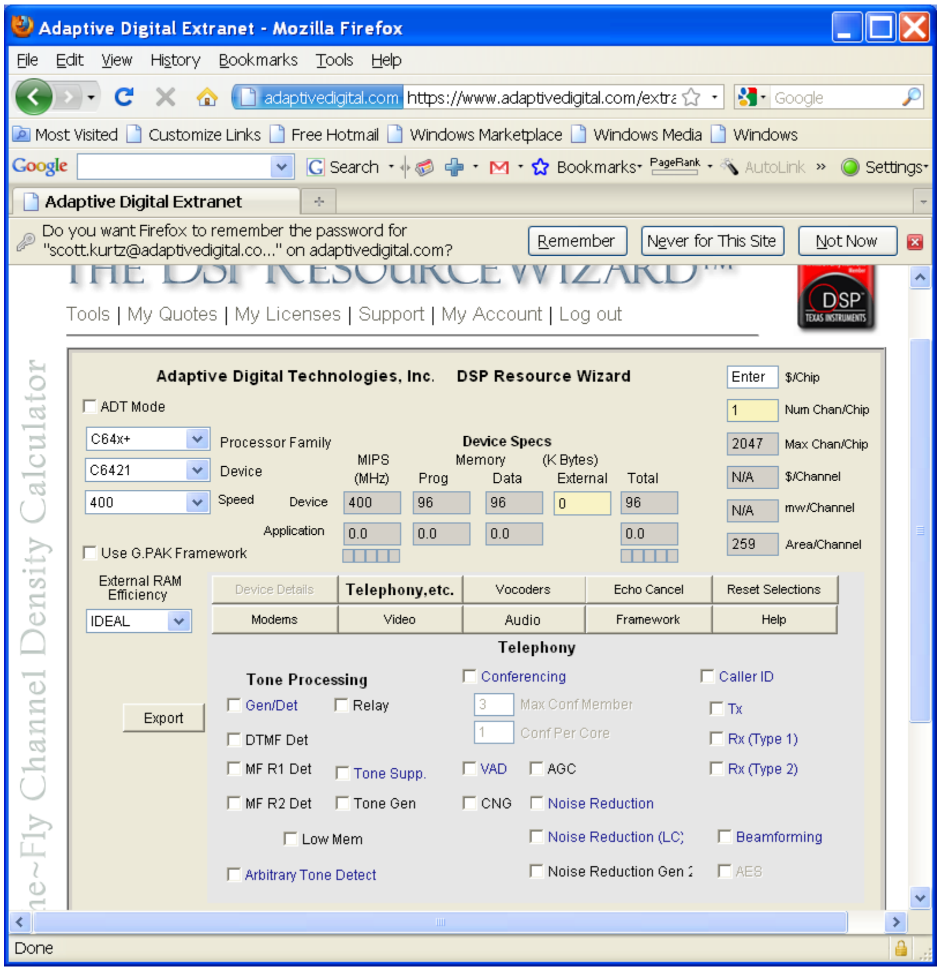
Let's go through a high-density gateway example.

Requirements:

- Channel Density: 256 channels per chip
- Vocoders: G.711 G.729AB
- Echo Cancellation: G.168 @ 128 msec
- DTMF relay
- Channel Type: TDM to Packet
- Protocols: RTP
- Packet Interface: HPI
- TDM Interface: McBSP

First, we browse to the DSP Resource Wizard at http://wizard.adaptivedigital.com

We first select the processor family to be C64X+ since this is a high-density application. Figure 8 is a screen shot of what you will see.


Adaptive Digital Extranet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
adaptivedigital.com https://www.adaptivedigital.com/extra
Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Adaptive Digital Extranet
Remember
Never for This Site
Not Now
Tools | My Quotes | My Licenses | Support | My Account | Log out
Device Details
Telephony,etc.
Vocoders
Echo Cancel
Reset Selections
Modems
Video
Audio
Framework
Help
Export
Done

Do you want Firefox to remember the password for "scott.kurtz@adaptivedigital.co..." on adaptivedigital.com?

Adaptive Digital Technologies, Inc. DSP Resource Wizard

ADT Mode

Processor Family	C64x+
Device	C6421
Speed	400
External RAM Efficiency	IDEAL

Use G.PAK Framework

Device Specs

	MIPS (MHz)	Memory (K Bytes) Prog	Data	External	Total
Device	400	96	96	0	96
Application	0.0	0.0	0.0		0.0

Enter	$/Chip
1	Num Chan/Chip
2047	Max Chan/Chip
N/A	$/Channel
N/A	mw/Channel
259	Area/Channel

Telephony

Tone Processing: Gen/Det, Relay, DTMF Det, MF R1 Det, Tone Supp., MF R2 Det, Tone Gen, Low Mem, Arbitrary Tone Detect

Conferencing, Max Conf Member 3, Conf Per Core 1, VAD, AGC, CNG, Noise Reduction, Noise Reduction (LC), Noise Reduction Gen 2

Caller ID: Tx, Rx (Type 1), Rx (Type 2), Beamforming, AES

Figure 8 shows the DSP Resource Wizard

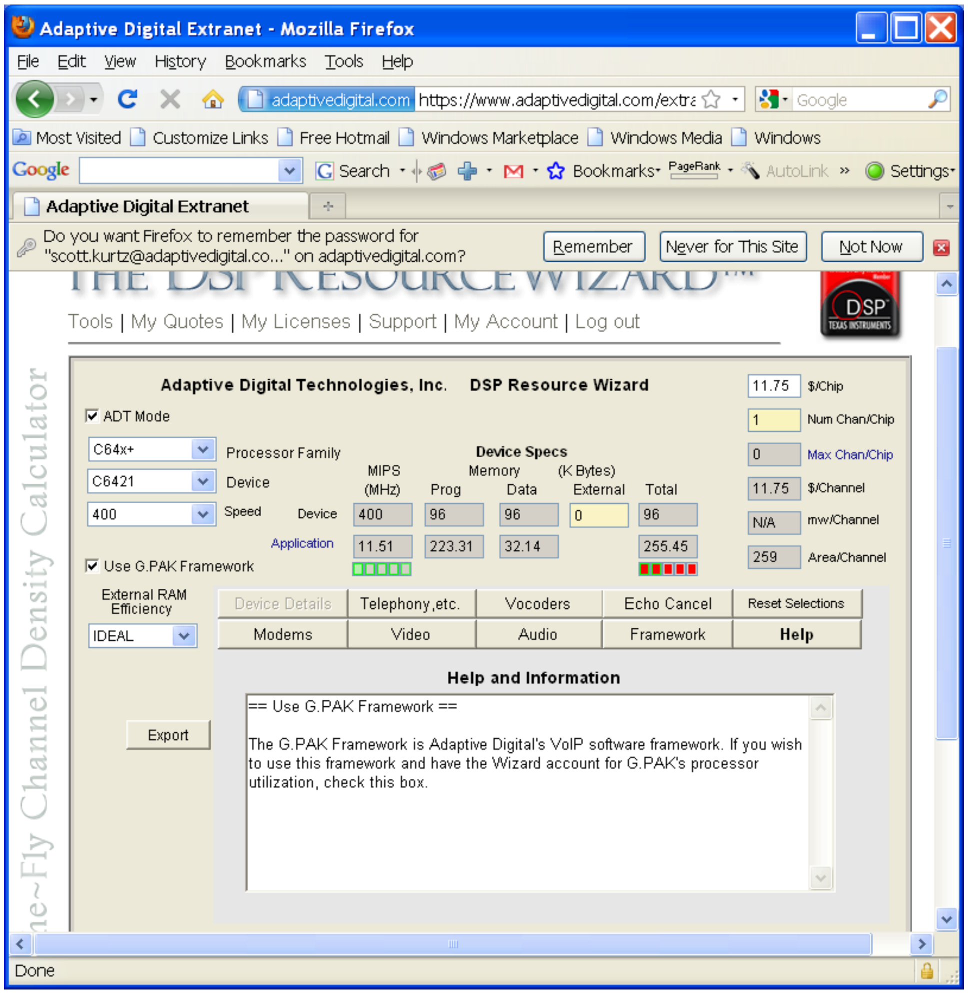
The first thing we want to do is let the wizard know that we wish to use the G.PAK framework by clicking on the “Use G.PAK Framework” checkbox.

The telephony algorithms show in the lower half of the screen. Since tone relay is needed, click on Tone Relay, DTMF Detect, and Tone Gen.

Next click on the Vocoders tab. Select G.711 and G.729AB.

Next, click on the Echo Cancellation tab. Select G.168 128 msec.

The results are shown in Figure 9.


Adaptive Digital Extranet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
adaptivedigital.com https://www.adaptivedigital.com/extra
Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Search Bookmarks PageRank AutoLink Settings
Adaptive Digital Extranet
Remember Never for This Site Not Now
THE DSP RESOURCE WIZARD™
Tools | My Quotes | My Licenses | Support | My Account | Log out
On-the-Fly Channel Density Calculator
Device Details Telephony,etc. Vocoders Echo Cancel Reset Selections
Modems Video Audio Framework Help
Export
Done

Do you want Firefox to remember the password for "scott.kurtz@adaptivedigital.co..." on adaptivedigital.com?

Adaptive Digital Technologies, Inc. DSP Resource Wizard

☑ ADT Mode

C64x+	Processor Family		Device Specs				
		MIPS (MHz)	Prog	Memory Data	(K Bytes) External	Total	
C6421	Device						
400	Speed	Device	400	96	96	0	96
		Application	11.51	223.31	32.14		255.45

☑ Use G.PAK Framework

External RAM Efficiency: IDEAL

11.75	$/Chip
1	Num Chan/Chip
0	Max Chan/Chip
11.75	$/Channel
N/A	mw/Channel
259	Area/Channel

Help and Information

== Use G.PAK Framework ==

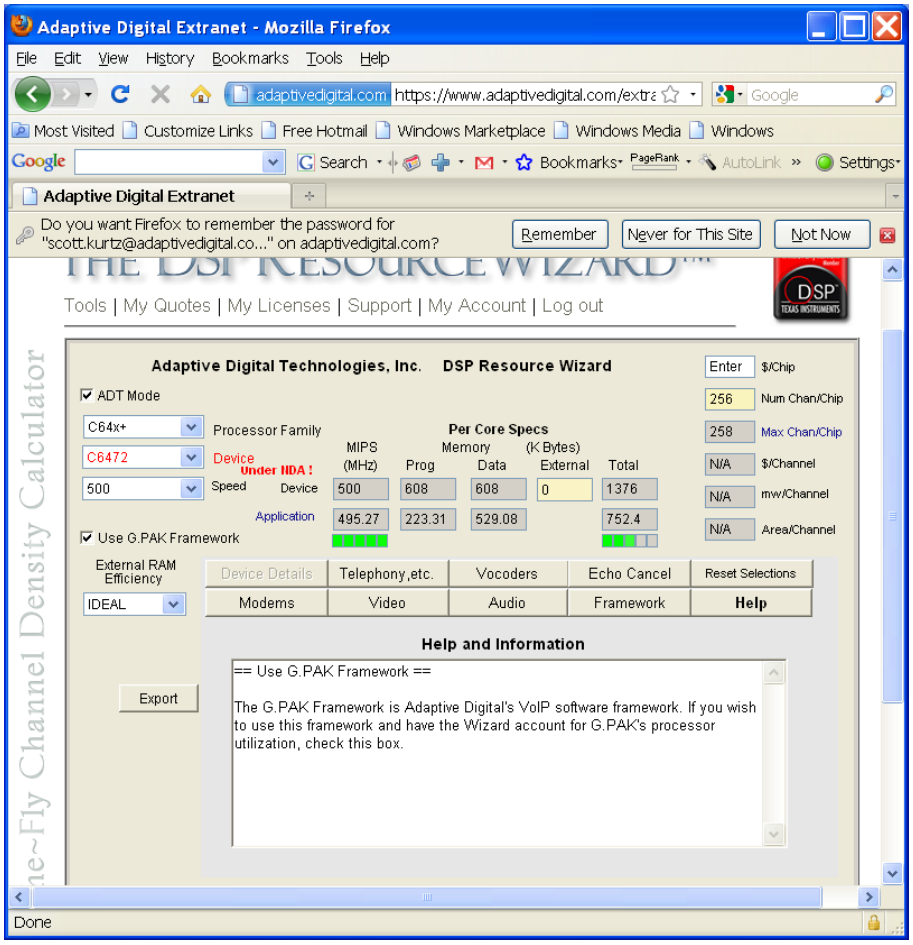
The G.PAK Framework is Adaptive Digital's VoIP software framework. If you wish to use this framework and have the Wizard account for G.PAK's processor utilization, check this box.

Figure 9 shows the DSP Resource Wizard after algorithm selections have been made.

You will notice that each time you click on an algorithm, MIPS and Memory utilization change. At this point, you'll notice that we have exceeded the available memory in the selected DSP, which happens to be the C6421. This is highlighted by the red bar that appears below the total memory usage column.

In the Devices pull-down menu, select the C6472 device. This will give us plenty of horsepower for the application. The red line should go away.

You'll also notice that the wizard computes the maximum number of channels that the C6472 chip can handle given the specifications that we have provided to it. We can tell the wizard that we want 256 channels by entering that number in the “Num Chan/Chip” box. When we do that, the MIPS and Memory usage are updated. Figure 10 shows the results.


Adaptive Digital Extranet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
adaptivedigital.com https://www.adaptivedigital.com/extra
Google
Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Google Search Bookmarks PageRank AutoLink Settings
Adaptive Digital Extranet
Remember Never for This Site Not Now
THE DSP RESOURCE WIZARD™
DSP TEXAS INSTRUMENTS
Tools | My Quotes | My Licenses | Support | My Account | Log out
Line~Fly Channel Density Calculator
Device Details Telephony,etc. Vocoders Echo Cancel Reset Selections
Modems Video Audio Framework Help
Export
Done

Do you want Firefox to remember the password for "scott.kurtz@adaptivedigital.co..." on adaptivedigital.com?

Adaptive Digital Technologies, Inc. DSP Resource Wizard

☑ ADT Mode

C64x+	Processor Family		Per Core Specs				
		MIPS		Memory	(K Bytes)		
C6472	Device Under NDA !	(MHz)	Prog	Data	External	Total	
500	Speed	Device	500	608	608	0	1376
		Application	495.27	223.31	529.08		752.4

Enter	$/Chip
256	Num Chan/Chip
258	Max Chan/Chip
N/A	$/Channel
N/A	mw/Channel
N/A	Area/Channel

☑ Use G.PAK Framework

External RAM Efficiency: IDEAL

Help and Information

== Use G.PAK Framework ==

The G.PAK Framework is Adaptive Digital's VoIP software framework. If you wish to use this framework and have the Wizard account for G.PAK's processor utilization, check this box.

Figure 10 shows that the C6472 can handle the application's requirements.

If you review the MIPS and memory specs, you'll see that the MIPS are close to the limit. We're using 495 out of the available 500 MIPS. The memory is not so tight. It is advisable to use the faster 625 MHz device. This can be selected using the "Speed" pull-down menu.

Now that we have found that the C6472 can handle this application, we can use the TI selection guide to verify that the right peripherals are available on the C6472. Had this been a lower-density application, we may have found multiple DSPs that could handle the job. In that case, we would have needed to narrow down the search based upon the available peripherals as well as price, package, etc.

A video demonstration of the use of the DSP Resource Wizard can be found on Adaptive Digital's web site at http://www.adaptivedigital.com/video_training/wizard3/wizard3.htm.

Now that we have selected a DSP, we need to build the G.PAK image. We do this by starting Code Composer Studio and opening the G.PAK project. When we build the project, a Windows program starts as shown in figure 11.

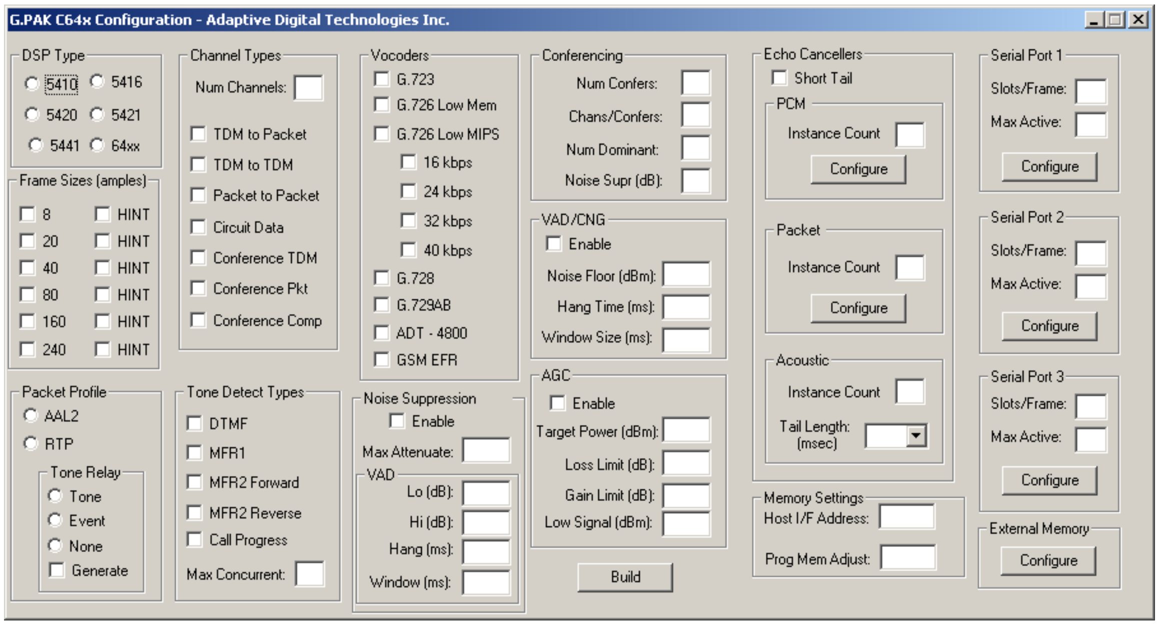


Figure 11 shows the G.PAK configuration builder.

Once you make the appropriate selections and click the "Build" button, source code files will be updated and Code Composer will build a DSP-downloadable image.

A video demonstration of the G.PAK build process can be found on Adaptive Digital's web site at http://www.adaptivedigital.com/video_training/GPAK/GPAK.htm.

4. Summary

Adaptive Digital's G.PAK VoIP DSP Application software can be configured to meet a wide variety of VoIP system requirements, including high density. Using Adaptive Digital's DSP Resource Wizard and G.PAK Configuration Builder, a designer can obtain the right DSP software solution on the right DSP quickly and efficiently.

When high density is needed, the Texas Instruments TMS320C6472 is an ideal DSP choice.