The D/A diaries: A personal memoir of engineering heartache and triumph

By Hitoshi Kondoh, Burr Brown Products Group of Texas Instruments; English translation by Rusty Allred, Distinguished Member of the Technical Staff, Texas Instruments*

One Night at 10

The Developer Awaits that Moment of Ultimate Happiness...

So-called "engineering samples" are created when wafers come from the fab and the die are attached to lead frames at the assembly plant. Next we fire up the device to check performance. This is the most nerve-racking, heart-throbbing time for a designer.

When the chip has been soldered to the previously-prepared test board, we confront our feelings and, in a prayerful mood, flip the switch.

The confidence that the device will definitely work, which is amassed through extensive simulations and tests prior to tape out gives way to uncertainty as the mind has fleeting thoughts of worst-case scenarios. Even if the device does function, we don't believe our eyes. We flip the switch and continually check the power supply and traces on the board because it could be functioning improperly, which means we can't really say that it is functioning at all. But after verifying that it really is functioning, we feel like we have gone to heaven. This is the moment of ultimate happiness for a designer.

However, even if a single portion of the device is not functioning (the usual case!), the designer sinks to the depths of despair.

The Non-functioning Chip

The PCM2702 didn't work at all for 2 days after receiving the engineering samples. Furthermore, one of us was scheduled to take the board to the US a couple of days after the samples arrived. Saturday passed and it was now Sunday at 10 PM and still no response from the chip. Three of the responsible parties arrived early and worked all day, even on the weekend. At some point, we were staring at a single point (one of the engineering samples) and saying, "This is weird. The gate-level simulations were perfect. We did ample verification using the large-scale PLD (Programmable Logic Device). And we even took a board with the PLD installed to the US and passed the USB compliance test..."

Part 1: The USB Compliance Test

The Development of a D-to-A Converter for Digital Speakers

We (Burr-Brown) were building a USB controller with a D-to-A converter (DAC) inside. The application for such a device is clear - digital speakers.

USB has been standardized as a digital serial interface and, in the PC world, is replacing RS-232-C, which had been the industry standard. Furthermore, it appears that USB will also be used for various applications outside the PC. Applications that were unimagined at the

beginning of USB development, such as MD/CD radio cassette systems with USB audio adapters and USB ports, seem likely to appear.

USB can connect not only input devices such as a keyboard or a mouse, but also output devices such as a printer or speakers. We focused on these USB speakers. Sound information is an extremely important medium for human beings. If the digital packetized information streaming across USB can be converted to analog with high fidelity, high quality sound can be delivered. We thought that there would be many applications for a one-chip USB interface and DAC if it could be provided at a low cost.

But if a USB chip is to be developed, it is necessary to have a certification. So let's begin with the story of the USB compliance test.

The Plan for the USB Compliance Test

The USB compliance test is sponsored by the USB Implementers Forum (USB-IF) and is held approximately four times per year at irregular intervals. Only those chips that have passed this test are allowed to use the USB logo, and product ID's of passing devices are included in the Implementers List that is published by the USB-IF.

As mentioned earlier, the engineering samples we created to formally undergo this test didn't work. I say formally since we actually had a dry run test using a large-scale PLD board implementation.

There were two reasons for undergoing the dry run test:(1) To assure that there was nothing we had missed. In other words, this test acted as our test bench to assure that we had done sufficient debugging prior to taking the design to silicon. (2) Since this was our first, we wished to see just what is involved in this compliance test.

From there our plan was to take the design to silicon and be ready for the next compliance test.

This test has not recently been held in Japan, it is only held about once every three months on irregular intervals. Therefore, thinking that you can't win if you don't play, we put together an aggressive plan: hoping for a first-pass success, we would put the engineering samples on the evaluation board as soon as they had arrived from assembly, and go to the US for the test. After making careful preparations, from getting the test board ready to getting hotel and flight reservations, we waited eagerly for the samples to arrive.

The Silent Samples

But the engineering samples didn't work.

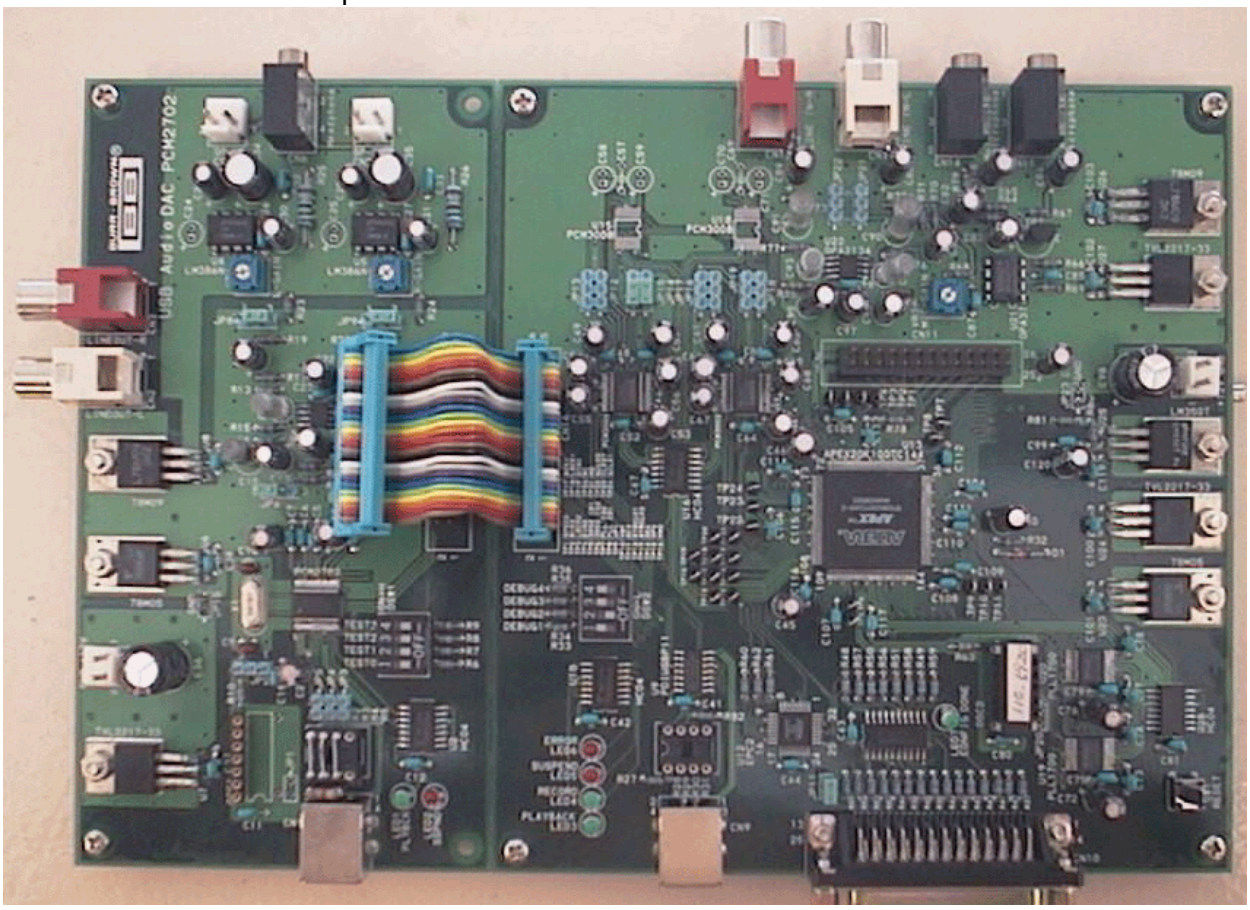
It was a shock to us designers because we had confidence (although not 100%) that it should work. To back that up we had done ample testing on the PLD implementation. Of course, since we were aware that the circuits were not exactly the same in the PLD as in actual device, we knew there was some chance that the samples would not work. If they didn't, nothing could be done immediately (it takes time to make changes to the silicon), so we just planned to wait for the next test (give up with a smile) if necessary.

But even though we had planned that way, what was to become of all the overtime effort and expectation? Looking for anything that might help, we checked the circuit board traces until they were nearly worn out, and used an oscilloscope to verify the operation of the external clocks... Finally we decided to probe the chip's internal circuits directly.

An Engineering Sample on the Dock

After using chemicals to remove part of the plastic package we put the chip under an electron microscope and use a remote control apparatus to apply an oscilloscope probe to specific signals. This job is one that requires great skill - it is realm of a great craftsman who can use a manually controlled lever to maneuver at the micron level.

Photo 1. Large Scale PLD Board: The right side is the PLD evaluation portion; the left side is the structure for the actual chip



Even as we marveled at this probing process, we designers were half sobbing because a bug at this level requires a change at the wafer level. In this situation, we might as well be doing all of this after taking some time off; but not a single person went home.

The process continued to the point of finally checking the master clock, which, it was discovered, was operating at the wrong frequency. At this frequency there was no possibility that the USB bus could communicate. Prepared for the worst, we double-checked the source file.

And found the bug!

An Engineering Sample Gets Outpatient Surgery

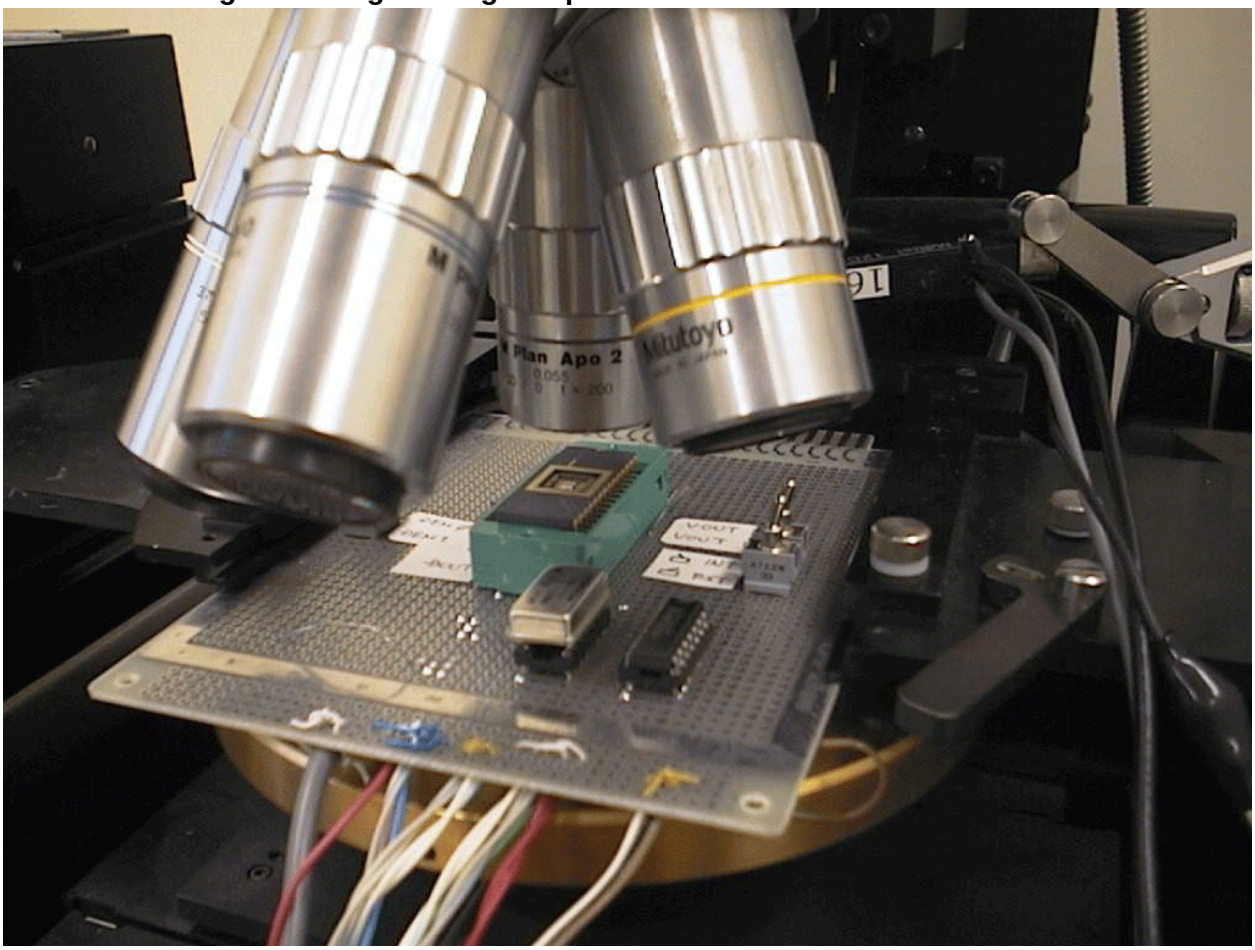
Could we call it happiness amid unhappiness? It wasn't a problem with the USB core or the DAC core, but with the control circuit for the integrated phase locked loop (PLL). And it was determined that it could be repaired by using a laser to sever a particular connection.

An electron microscope was used to locate the spot and a laser scalpel was applied.

Now the chip with the hole in it was again put on the board.

And the device was once again prayerfully fired up. This time it worked!!

Photo 2. Probing of the engineering sample



Personal Sidebar A:

The USB-DAC Development Strategy

I am employed at an American company (Burr-Brown) with a long history in the business of providing high-performance operational amplifiers for industrial and audio applications. I am in the Japanese branch, which has responsibilities for development of a variety of chips. It may not

be well-known, but a large number of devices are actually developed in Japan. In particular, our emphasis is on DACs and PLLs. In Japan devices have been developed for use in high-performance audio products, of course, as well as for CD-ROMs for PCs and well-known electronic games. In addition, we also produce extremely high-performance DACs that require individual trimming and cost several thousand yen each (1000 yen is approximately $10).

I myself have not so much changed jobs, but changed occupations, and have not been with Burr-Brown for long enough to know the company very well; but I have found it to be very strict about the quality of analog products. One of the first things I heard on arriving is "We must maintain this performance level!"

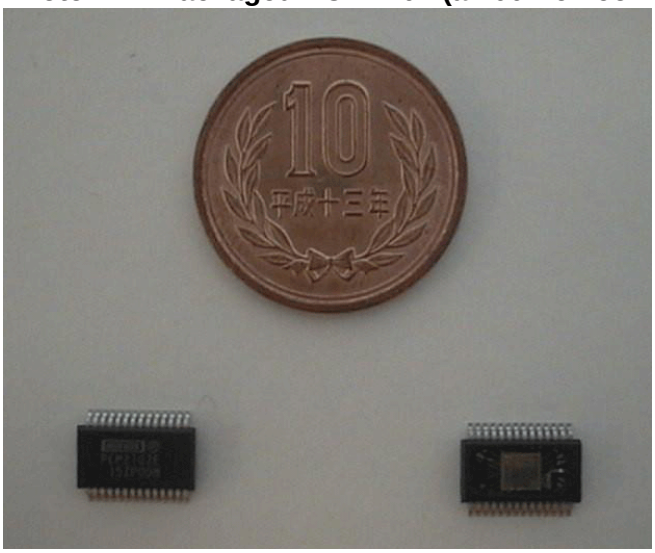
However, Burr-Brown also has great interest in the massive PC market since we expect that IT business will become more popular, and that the distribution of music over networks will catch on. Both Windows 98 and the Mac OS support USB, so to provide a single chip USB interface plus DAC solution, with the kind of DAC performance that we have provided in the past, was the first step in our plan of attack.

By only connecting the USB connector, audio line out level is available directly from the chip. This chip is an interface between a USB terminal and an analog audio amplifier. Of course it is important that this be simple, but, on the other hand, since the data coming from the USB terminal is distortion-free digital audio, if the DAC performance is like what we have offered previously, the story is that even the PC can become a superior audio device.

The IDs that can be returned by the PCM2702 are shown in Table A.

USB Revision	USB Revision 1.0
Device Class	0x00 (device defined interface level)
Device Sub-Class	0x00 (not specified)
Device Protocol	0x00 (not specified)
Max Packet Size for Endpoint 0	8 byte
Vendor ID	0x08BB
Device ID	0x2702
Release	1.0

Photo A. A Packaged PCM2702 (a 100 Yen coin is near the size of a US quarter)



The Sample Makes its Flight

We three who had just been delivered from the jaws of death used the electron microscope and the laser cutter to prepare a spare device. By the time we finished it had become Sunday.

The device with a hole in its package (somewhat embarrassing) caught its flight, went to the US, and passed the second compliance test without incident.

With the feeling of having hit the game-winning home run after having been down in the count with two outs in the bottom of the ninth, today's beer was the best ever! In cases like this we can make the necessary design change in the best of moods. From there, after living through the trials of six months of rigorous reliability and manufacturability tests, and having completed the design change, a new device was completed - the USB-DAC, PCM2702. (See Sidebar A, earlier.)

Part 2. The Pitfalls of USB Isochronous Audio Data Transfer

The Audio Capability of the PC

Although it makes a nice story to say that as long as a good DAC is used, great audio can come from the PC, there is the problem that the PC is a haven for digital noise. Not only that, but there are bothersome problems with real-time digital transmission that are not even limited to USB.

The former can be alleviated through use of the USB cable, which allows the power supply and analog circuitry to be separated into a different box. But, without a lot of forethought about the latter, it can become impossible to reach the desired audio quality level.

There are four USB transmission modes (please see Table 1). The two of those that are used for sending large quantities of data are (1) Isochronous Mode - A fixed number of packets is guaranteed to be sent and received. This mode is used with multimedia data such as images and audio. (2) Bulk Mode - A fixed quantity of data is sent at one time. If for some reason some of the data, is lost it is resent.

For data storage or printer applications the bulk mode is best because speed is of utmost importance and, through retransmission, data errors will be eliminated. But for audio data, real-time transfer is even more important than occasional missing data. (Noise is more tolerable than interruptions in the data. Of course pops and clicks are intolerable, but even more unpleasant is an intermittence of the data.) In this case, the isochronous mode is used. In other words, a real-time transmission scheme, with no re-sending of packets, is used for audio data, which streams from the PC in an RS-232-C-like manner.

Photo 3a. USB Connectors (Series A)

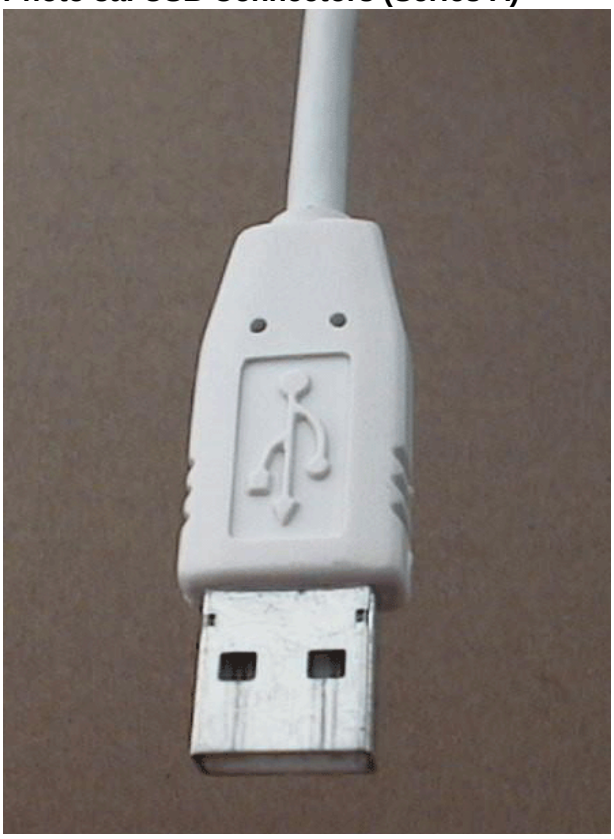
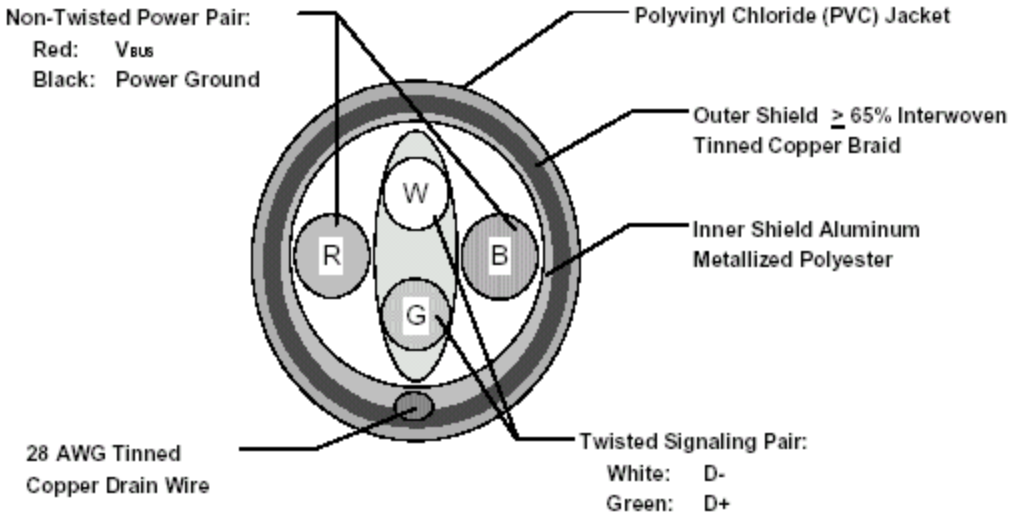


Photo 3b. USB Connectors (Series B)

USB is Clockless, Differential, Serial Transmission

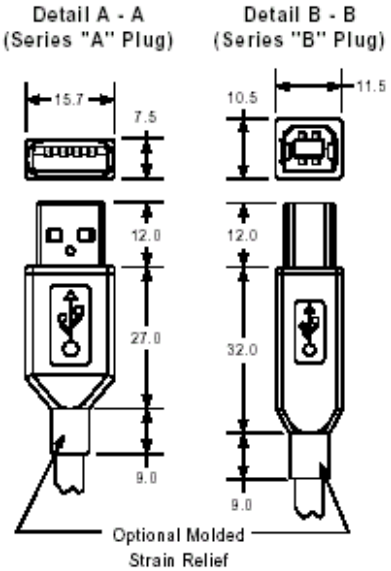
It is not our intent to fully discuss the USB specification here. (Please refer to sister publication, Interface, March 2000.) USB 1.1 is a 12 Mbps bi-directional, serial bus, which is connected with 4 conductors as shown in Figure 1.

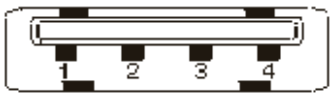


Two of those conductors are power (VBUS) and ground (GND). Information is transferred through the other two: 1) D+ 2) D- Since there are only two data lines, only four states can be used for data transmission.

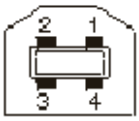
To prevent noise and corruption due to the cable and connectors from affecting the receiver's detection, which is accomplished by thresholding, a basic differential signaling scheme is used: When D+='1', D-='0' When D+='0', D-='1' This signaling scheme uses these two states to send data (please refer to Figures 2 and 3). In addition, a special meaning is ascribed to the state where D+ and D- are both zeros (SE0: Single Ended Zero). The state where both D+ and D- are both '1' is not used.

Figure 2. Arrangement of USB Connector Pins





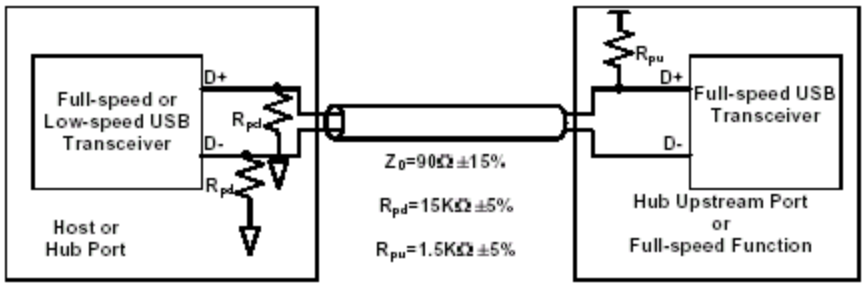
Type A



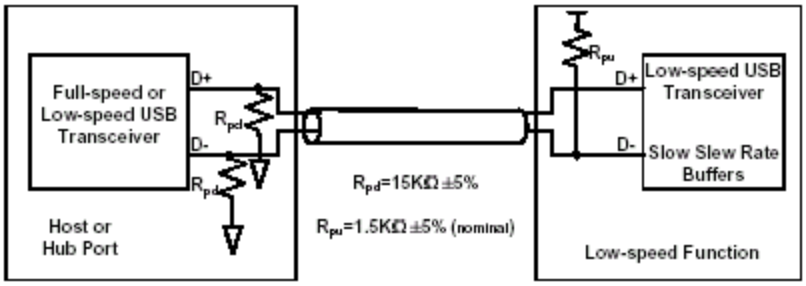
Type B

1	Red	5 Volt
2	White	Data-
3	Green	Data+
4	Black	Ground

Figure 3. USB Pull-ups and Pull-downs



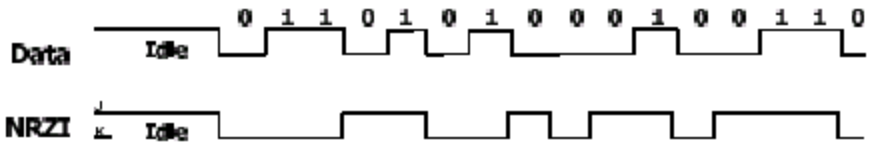
Full-speed Device Cable and Resistor Connections



Low-speed Device Cable and Resistor Connections

The NRZI Method

USB data transfer is basically a two-conductor signaling method wherein, in the case where the data is '1', the signal does not change, and when the data is '0', the signal does change. This is the so-called Non Return to Zero Inverted (NRZI) method (please refer to Figure 4).



It follows that there is no explicit clock on the USB cable (this compounds the problem). Rather, the signal is restored based upon the intervals between edges of the data. In this type of digital communication, if the sender uses a perfect clock to create the signal, and the receiver uses a perfect clock to interpret the data, the original data can be reconstructed. Since NRZI reconstruction is possible if there is a clock that is four times the bit rate, it can be accomplished if both the sender and receiver both have 48 MHz clocks (the transmission rate is 4 times 12 Mbps).

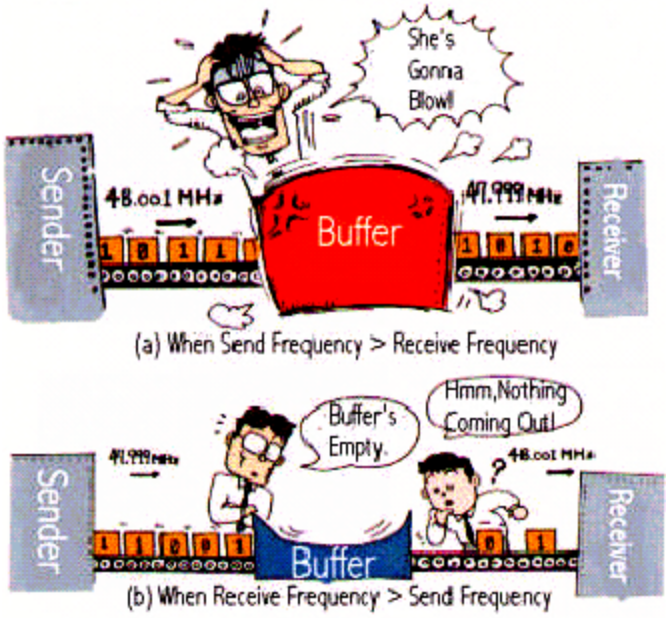
However, when viewing this from the standpoint of an audio device, the very fact that the sender and receiver both have local clocks becomes a stumbling block.

The Evil of Clocklessness

The fact that there is no clock line within the USB cable leads to a thinner cable which is an advantage. But, no matter how good the crystal oscillators are at the send and receive ends, there will always be some difference between the two. For example, if the sender is sending

audio data at a rate of 48.001 MHz and the receiver is receiving at 47.999 MHz, the receiver is reconstructing data slightly slower than the transmission rate. When a large quantity of audio data is sent under these conditions, the buffer will soon overflow, resulting in lost data (please see Figure 5).

Figure 5. When there is Clock Frequency Error



On the other hand, if the receiver is running faster, an underflow will occur resulting in a discontinuity in the audio data. In a CD player, angular control can be used to control the motor such that it will synchronize with the playback data rate. But the USB receiver cannot control the sender. The resulting missing data can be digitally compensated (using a smoothing filter, please see Figure 6), but our company's development philosophy does not allow for such deception! (As an aside, there is no problem at all if the data is reconstructed with the receiver's clock after it has all been sent.)

*A version of this article entitled, "Story of the Development of USB D-A Converter," appeared in the Japanese publication, Design Wave Magazine, June, 2000, pp. 28 - 47.

Do we use an existing USB core? How large should the FIFO be? The D/A Diaries, Part 2

3. Dealing with the PC (software) is a Bigger Problem than the Specification

We began the development with the USB interface. The USB specification itself is not particularly complicated, but we came to understand that the real problem is the software that comes loaded on the PC. Even if a USB-DAC meets the USB specification, it is useless if it will not operate under Windows or Mac OS.

There actually exist USB speakers that will not work (don't make a nice sound) with certain PCs. Even though these USB speakers fulfill the USB specification; that alone is not enough.

Obviously, certain know-how is necessary in dealing with interfaces. In LSI development we recognize that the use of previously-developed cores is the key to effectiveness. But we recognize that it is not always a simple matter to use an existing core even when dealing with a standard interface. In that sense, the decision of whether to use existing cores is very important. Certainly there are cases where an existing core could be used but the cost is too high to make business sense.

With this in mind, we tested that actual operation of a PC using a large-scale PLD.

Using a Large-scale PLD We Decided to Develop a USB Core

First we mocked-up a system containing a PLL and a DAC and tested the operation on various PCs. Using as cores a PLL1700, a PLL chip with a high Carrier/Noise (C/N) performance specification, and a PCM1716, a well-known 96 kHz - 24 bit DAC, we were able to use a PLD to efficiently test the USB operation. Use of the PLD allowed us to observe internal signals and test conditions that do not actually occur. This is particularly useful when communicating with PC software, as in our present case.

A FIFO is Used to Deal with Packets that are not in Order

USB sends audio data packets on 1 ms intervals. Since, as mentioned previously, pauses in the audio cannot be tolerated, audio playback begins when the first packet arrives, and the next packet must arrive before all of the data in the previous packet has been played. Although we are discussing audio packets in particular, it is possible for the order of packets to be disrupted by other USB packets. In other words, a FIFO large enough to hold at least two packets is required to deal with the possible change of order.

In the case of dealing with 48 kHz, 16-bit stereo data, the buffer capacity must be at least 48 x 16 x 2 x 2 = 3,072 bytes. However, since we know that FIFOs require significant die area, we want to make them as small as possible to save cost.

USB Clock Error Uses up the FIFO!

On the other hand, the USB specification allows for clock frequency error of 500 ppm. This is an easy-to-accomplish specification for a crystal oscillator and makes the design of the USB

circuitry rather easy. However, this is an allowance for an error between the send and receive clocks, and poses a problem for audio.

In this case, the read and write clocks for the FIFO are different. As the 500 ppm error accumulates, the 1 packet buffer margin will be completely used up in 2,000 packets. Since 1 packet is 1 ms, 2,000 packets works out to 2 seconds. If one packet is lost and the device jumps to the next, a popping sound will be heard.

A Clock Tracking PLL Circuit is Essential

It would never do to create a DAC that makes noises every few seconds, or even every few dozen seconds, depending upon the clock precision. In order to avoid this, it is necessary to have the receive clock track the send clock.

For this reason we determined that it is necessary to have an excellent C/N performance PLL on the chip. Although this might mean a cost increase, it must be done. At this point any hope of using a standard USB core was completely eliminated.

4. Test Chip Development

The Capabilities of the First Test Chip

We decided to develop a single chip containing a USB core, a DAC, and a PLL. It was determined that we would use our existing DAC and PLL cores.

By integrating the PLL, we were able to develop a USB-DAC that did not make popping noises. At such a time it is human nature to want various people to see (hear) the result, so we demonstrated it to all of those purported to be 'Golden Ears.' The audio signal came through the PCM1716, a DAC with an industry-wide reputation, and the PLL as the PLL1700, which has excellent C/N performance.

Since Windows 98 has a 48 kHz, 16-bit limit for audio data, at first glance it would appear that the 96 kHz - 24-bit PCM1716 was over specified for this task (please see Personal Sidebar B). But inside this chip the bit rate is oversampled by a factor of 8, and the precision increases to 24 bits. I thought the sound performance was sufficient.

Sidebar B
Sampling Frequencies
There are various sampling frequencies used in digital audio. Probably the most common is 44.1 kHz which is used by CD players.

However, in some systems, such as DAT and the PCM audio of BS broadcasts, 48 kHz or 32 kHz is used.

It seems very strange when considering how all of these different rates came to be. In many PC environments all three of these sampling rates are available.

Table B: Consumer Digital Audio Sampling Frequencies

	CD	LD	DAT	BS PCM	MD	DVD
Sample Rate	44.1 kHz	44.1 kHz	32/48 kHz	32/48 kHz	44.1 kHz	48/96 kHz
Bit Width	16	16	16	16	16	16/20/24

The Distortion is an Order of Magnitude too High you say?! Why....?

When the guys in charge listened to the prototype I saw dubious faces and was asked a variety of questions such as "Is the source coming from the PC corrupted?" In the end I was told to measure the audio performance. When I announced the results in a subsequent meeting I was told the distortion was an order of magnitude too high; the THD+N was 0.03%.

I wondered what was wrong with 0.03%, but was told that "We could never sell a device with this performance as one of our own."

For a 16 bit, 48 kHz system, I would have to achieve at least 0.003%!

I was faced with (attacked by) a problem. Some asked, "Is the digital data getting corrupted somewhere?" But rigorous VHDL simulations did not locate such a bug. For the first time I had the feeling that analog is awful...

I went into this thinking "Since we are processing digital signals, we can expect good sound as a matter of course, and from here on we are dealing with digital!" So this experience was a real shock.

"Is noise getting in to the signal somewhere? Maybe it's crosstalk through the signal lines. Or is there a timing problem with the data transmission?"

Troublesome thoughts, day after day.

I'm Responsible for Tracking Circuitry? Let's First do an FFT Analysis.

As the person in charge of tracking circuitry, I was praying (?) that there would be a bug in the digital system somewhere. This is because previous experience had taught me that it can be difficult to tune a tracking circuit. And there are stability issues as well. I was afraid the development effort would drag on. Because of this I first did an experiment to prove that there was no problem with the tracking circuitry. Since we have distortion, we should be able to see something on an FFT.

So I did an FFT analysis using the Audio Precision. From the USB port I output a full scale, 1 kHz signal with a sampling rate of 44.1 kHz. Since there could be some strange components, I looked at the spectrum out to about 70 kHz.

This spectrum is shown in Figure 7. The 1 kHz fundamental is fine, but there are also noise or distortion products. The frequency characteristic that results from the Delta-Sigma DAC's noise shaper can be seen as the noise floor rises after about 40 kHz, but there are no strange components in the 40 kHz to 70 kHz region. However, this is certainly a shape that one does not often see. The noise floor is pulled up to about 105 dB and there are broad skirts around the signal.

Figure 7. Full-scale Signal FFT of First Prototype (Noise floor has risen to around 105 dB)

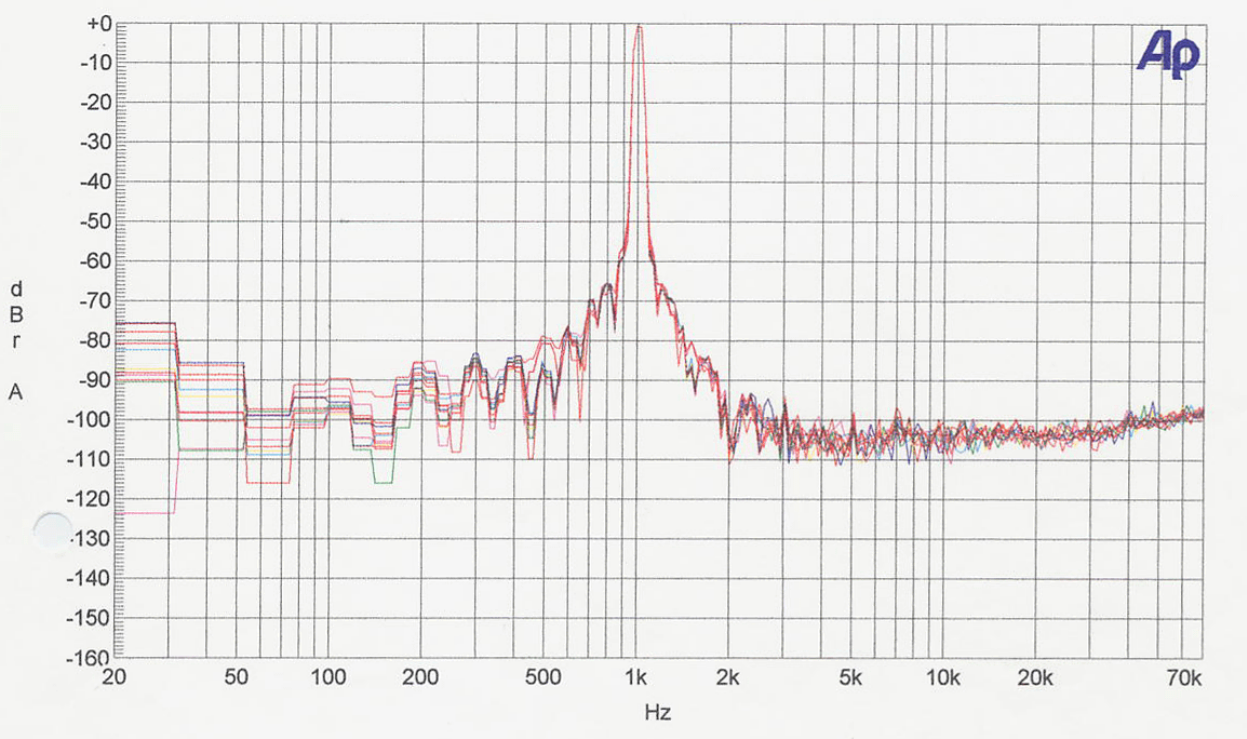
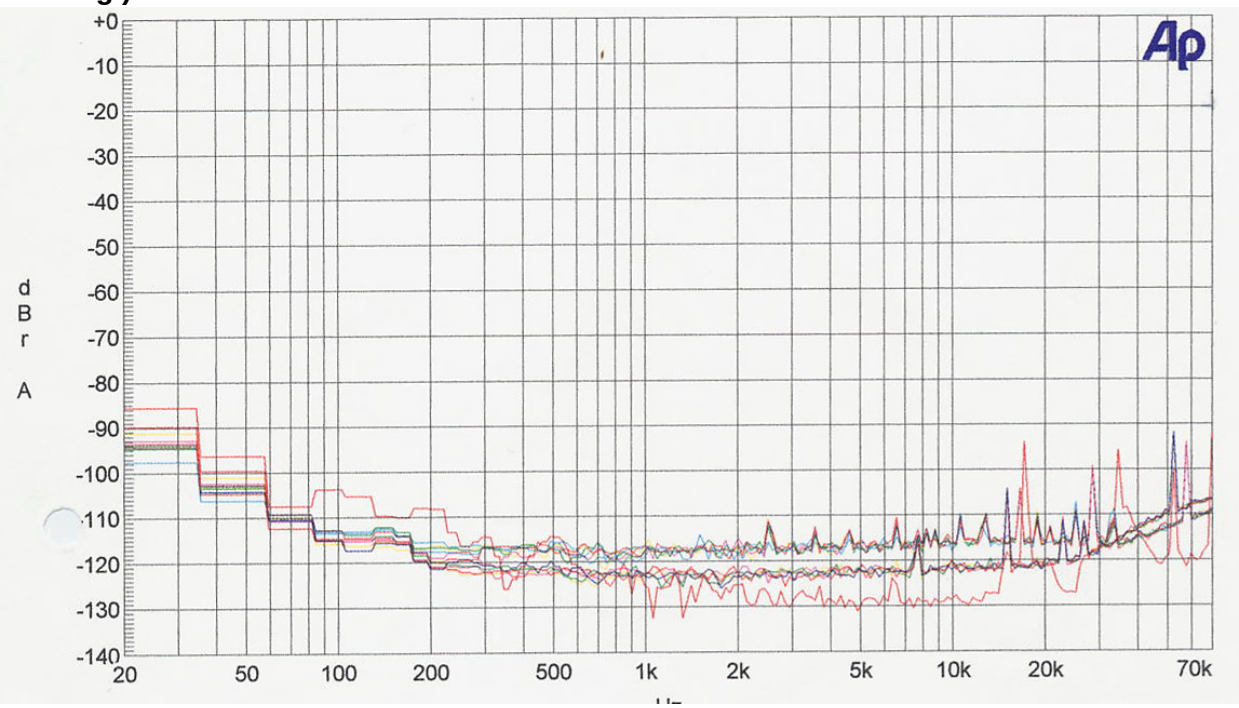


Figure 8. FFT Analysis with Tracking On/Off (Noise floor rises -123dB to -118dB due to tracking.)

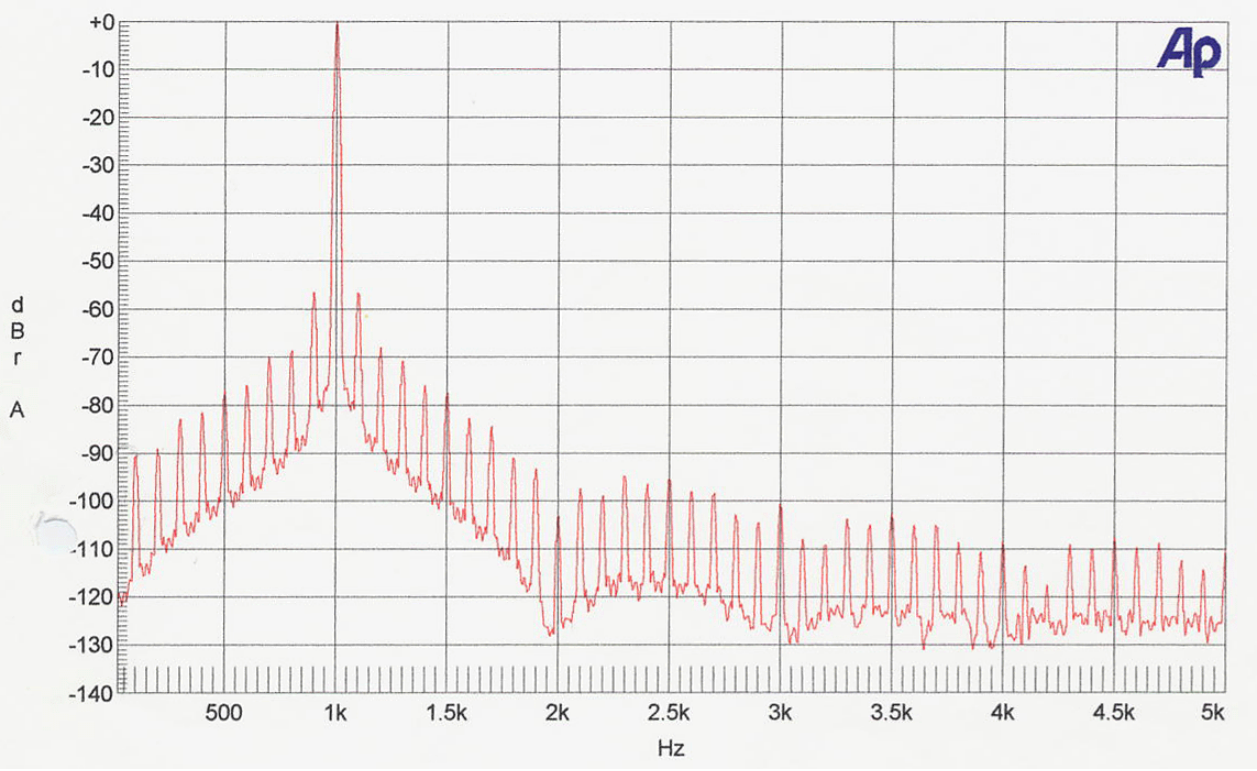


Next I investigated whether there was any difference with the PLL tracking turned on versus when it was turned off. Since the internal circuitry follows the clock, even for a zero signal input, we should see noise as a result of the fluctuations in the bipolar zero level if the clock is fluctuating. Figure 8 shows the FFT result (noise floor) for a zero input in the cases where the tracking circuit is On versus when it is Off.

Of course, an experiment like this can be done without having to solder or go through other manufacturing steps; the PLD can make very surprising contributions to the development. It's hard to imagine doing development without using a large-scale PLD. Since it was hard to see at a glance, I plotted the On and Off cases five times each. In the 200 Hz to 30 kHz region, the noise floor rose from -123 dB to -118 dB. There still remains the question as to whether tracking alone can cause this much change, but, in any case, this did confirm the fact that some degree of change was present.

Upon Raising the FFT Resolution . . . A 100 Hz Monster Appeared!

Next, in order to investigate the skirt around the fundamental, I decided to increase the FFT resolution to a higher setting than I usually use. Naturally it took longer to make the measurement. After a wait time that would best be measured in a fractions of an hour, I was amazed at the FFT analyzer's output graph. The measured FFT is shown in Figure 9.



It was completely opaque from the lower-resolution FFT, but at this higher resolution, the figure looked like a chestnut that had sprouted sea urchin-like spikes. It was an impulse train with spacing of exactly 100 Hz! Since the noise floor under conditions of no input signal is flat, this appears as signal distortion. If the frequency response shows an impulse train, the time waveform will also contain a 100-Hz impulse train.

The True Character of the 100 Hz Impulse Train...

Even for a sample rate of 44.1 kHz, the USB isochronous mode packets have a period of 1 ms (1 kHz). In order to distribute 44.1 kHz across 1 ms intervals, one 45-sample packet is sent for every nine 44-sample packets. The tracking pulse (as we will call it here) for every 45 sample packet occurs once every 10 packets, or with a frequency of 100 Hz. Since the PLL loop filter, a so-called low pass filter, has its corner in the tens of kHz range, this 100 Hz tracking pulse goes right on through and shows up on the PLL's VCO control voltage. It appears as frequency jitter.

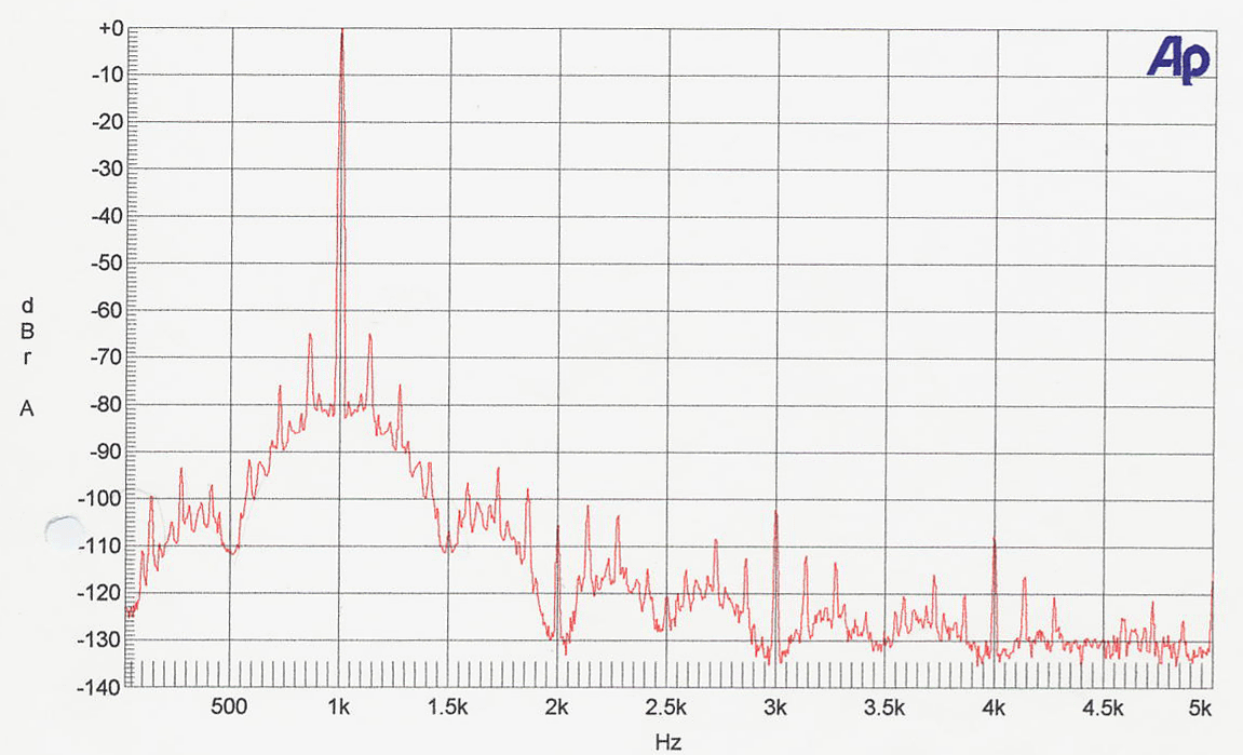
From the graph it is seen that the PLL frequency fluctuates impulsively right at 10 ms intervals. As a test I changed the sampling frequency to 48 kHz and measured the same 1 kHz signal.

Upon Changing the Sampling Rate to 48 kHz...

This time, since each packet always contains 48 samples, there should be no 100 Hz tracking pulse.

Figure 10 is the FFT for this case. The 100 Hz impulses have certainly disappeared, but some weak 142 Hz pulses have appeared in their place. And there is a succession of lobes. This is probably the result of phenomenally excessive ringing that goes along with the tracking operation!

Figure 10. FFT for 48 kHz Sample Rate



The impulse-type artifacts are a result of the open loop gain being high (due to the cost priority, a minimum-sized buffer is used and the gain of the tracking function is set high). It seems that we have a problem for a control theorist (this was my college major). However, increasing the buffer size and decreasing the gain only caused a low frequency disturbance on the signal and did not turn out to be a viable solution.

The Terrors of the Isochronous Mode

We still have a problem. It is a problem with a USB mode: in the adaptive isochronous audio transmission mode, the receiver has to determine the bit rate. This means that the bit rate is unknown prior to the time the data arrives.

The bit rate cannot be known prior to actually observing the packet.

Another terror of USB is that, according to the specification, it would not be unusual for the bit rate to change when the operating system is busy. Since the packets arrive on 1 kHz intervals, the PLL must lock within 1ms. In most PLLs, if we say that 1 kHz fluctuations are clearly audible and decrease the gain, we cannot track! Terror of terrors, we have just bumped into a brick wall. Upon doing some investigation, we were actually able to observe fluctuations in the audio frequency characteristics of one company's USB-DAC. Upon listening this could be detected as a disruption in the rhythm of the music. In reality, fluctuations in the time domain will probably result in an unpleasant listening experience. This is probably because they are delaying the lock-up time in order to reduce the jitter distortion.

Also, for isochronous USB data, a buffer is necessary for the time between the beginning of the packet until PLL lock, so the PLL lock-up time is reflected directly in the chip cost. The more audio quality is pursued, the longer the necessary buffer and the longer the time lag when playback begins. On the other hand, if the time constant of the loop filter is increased, a large RC is necessary and the chip area increases (recent progress in semiconductor technology has brought about minimization of digital devices, but analog devices have not changed).

If an external filter is needed, not only does the part count increase, but the board area also changes making the total cost increase significantly...

5. Delta-Sigma DACs and Jitter Control

The Advantages of Delta-Sigma DACs

Recently Delta-Sigma DACs have become the most popular type. At my company this is true at least for the lower-cost products. It is not my intention to give a primer on Delta-Sigma DACs, but to state it simply, this is a method of re-creating analog data using a one- or multi-bit Pulse Density Modulation (PDM).

Earlier DACs re-created analog signals using resistor ladders and the like. But, as the bit precision became higher, the errors in resistive ladders, as well as the current leakage, made it difficult to minimize semiconductor devices using these technologies. In the PDM used by Delta-Sigma DACs, the voltage and current are not controlled by controlling the combinations of resistors. Instead, integrators are used and the analog signal is re-constructed by controlling their charge times. It is basically the same as in the 1-bit Pulse Width Modulation (PWM) based DACs used in motor control. However, since the error is integrated, the precision is greater even for the same time resolution. The Burr-Brown PCM1716 is a charging-type, enhanced multilevel Delta-Sigma DAC that uses 8 times oversampling and 8 analog levels to achieve higher precision than 1-bit DACs.

The Disadvantages of Delta-Sigma DACs - Jitter Control

When considering ladder resistance error and current leaks, Delta-Sigma DACs certainly have a time control advantage given the exceptional precision of crystal oscillators. However, it must be remembered that they also require low jitter.

In usual circumstances this is not a big problem. However, as in the present instance, system clock tracking pulses, if they are needed, appear directly as signal distortions. This is not limited to USB, but will definitely be a problem for all digital transmission devices.

For example, since the receiver clock has to track the sender clock when a digital cable, in other words an SPDIF optical cable, is used to connect a CD player to the amplifier, the tracking pulses on the receiver clock will be a source of distortion. Of course this problem can be masked using digital filters and the like, but, since it decreases performance, it is not ideal for audio applications.

In other instances too, for example in products where tracking is used to synchronize image and sound data, signal distortion occurs as a result of the PLLs used. Care must be taken since this distortion will not be found characterized in data sheets. It appears that there are some systems in which the image frame rate is adjusted instead of the audio, to make the adjustment less noticeable (human eyes are not as sensitive as human ears, so if 2 or 3 frames of video information is dropped, it will not be noticed).

Back to the Beginning

We took a detour but now the USB-DAC development has come full circle. If we are to adopt an expensive, multi-level DAC, we will have to go back and re-investigate the marketing situation.

Feedback Control Doesn't Cut It

Since it is useless to be continually bewildered, I decided to reopen the tracking experiments. I adjusted the gain with an eye on the jitter analyzer. But I did not find a solution that satisfied the jitter and lock-up time requirements for an audio device. The biggest problem is that the 1 kHz feedback frequency is smack dab it the middle of the audio range. If the loop filter characteristic is shifted toward the low end the lock-up time become too long. If the PLL loop filter does not receive a reference signal for several clock cycles, it does not lock.

For several days I debated this within my own head: "If I don't use feedback the sound skips. If I do, distortion arises..."

I Try Feed-forward Control

And then, finally, I thought, "Wait! If I use feed-forward I'll bet they'll be no distortion!" At a time like this a PLD is certainly handy. No matter what the algorithm, as soon as it is written in VHDL it can be tested. In the days where silicon was fabricated based only upon simulations, the cycle time was so slow it was probably difficult to even entertain the notion of doing experiments!

Simulation is also a useful technique, but the test bench is, of course, written by humans. Missed hypotheses are never forgotten, and since the answer is often close to that desired by the experimenter, most folks probably find unexpected peace of mind. I think that people who

feel relief because they simulated something are probably over confident or a little slow. At any rate, I can't feel good about a design until I have actually built and tested the circuit.

I Achieve Low Distortion, but...

But, for feed-forward control, the one side must be able to specify the frequency of the other. However, since this is an experiment, I tried a two-pass approach. The FFT became unbelievably clean. The THD+N was 0.003%. I arrived abruptly at the goal.

However, lock-up takes 5 seconds!

"When I evaluate a 1 kHz signal the frequency counter digits continue to change... When I listen on a speaker the sound continues to change. There's nothing to do now but laugh!"

And in the instant after thinking that, I began beating my head with a hammer for the second time.

6. An Honored Professor and a Time-optimal PLL

Seeking Advice by E-mail

The Internet is essential to current research and development activities. This is because it allows discussion even without meeting face-to-face. Sometimes very beneficial advice from others can be found through e-mail. This is because different points of view or conceptualizations can be an opportunity to break a deadlock.

I sent e-mail to my professor from the university and to a fellow student from the research lab. The former is Professor Fuminori Kobayashi of the Kyushu Institute of Technology. He is jointly researching PLLs and motor rotation distortions so I expected some good advice. The latter, Hidekazu Machida, teaches at the Maizuru National College of Technology. When I told them, "I am looking for a fast lock-up time PLL," Professor Kobayashi said, "How about some papers I wrote 20 years ago on 'Time-optimal PLLs'?" and sent me several papers. Mr. Machida, who was copied on my message, pointed out that "the lock-up time is short but the jitter performance is bad." But since this is the first step, I put this method into the PLD and began the experiment.

The Time-optimal PLL Passes the Lock-up time Test but...

The conclusion regarding the Time-optimal PLL is that its lock-up time performance is extraordinary. Due to the structure of a PLL, it cannot measure the frequency in less than two reference clock cycles. But the PLL of Professor Kobayashi's paper is locked perfectly by the third clock cycle.

However, the jitter performance, as suggested by Mr. Machida, did not meet the specification. The result of tuning up the circuit, using the PCM1716, was a THD+N of 0.01%. This is a factor of 3 lower than the original THD+N of 0.03%, but still a long way from the desired 0.003%.

But I had a feeling that it might just be possible to make this work. If this algorithm is used, the buffer can be the minimum size. Now just to start with this research as the basis and try to improve the jitter performance. In other words, the desire was to use the Time-optimal PLL and gets its jitter performance into the range of crystal oscillators.

Actually, even as I read the PLL papers I had the intuition that the feed-forward concept could still be put to use. That intuition turned to conviction as I tested the PLL algorithm. In order to lock to the connected device in minimum time, the connected device's frequency must have been estimated. This was the birth of SpAct (s-pact).

We will discuss those results after a discussion of the principles behind the Time-optimal PLL.

Principle of Operation of the Time-optimal PLL

Figure 11 is a top-level timing chart to explain the principle of the Time-optimal PLL. This figure depicts the following: (a) The PLL's reference frequency. (b) The VCO's waveform (oscillator frequency/phase) when the PLL responds in minimum time. (c) The output waveform of the phase comparator (phase error) given the above two reference signals.

Figure 11. Principle of Operation of the Time-optimal PLL

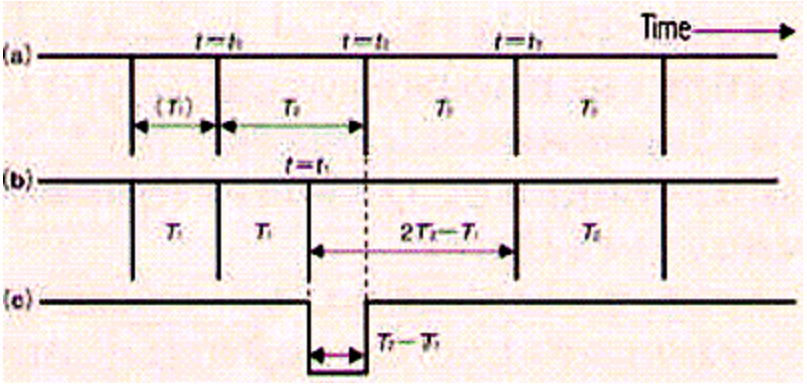
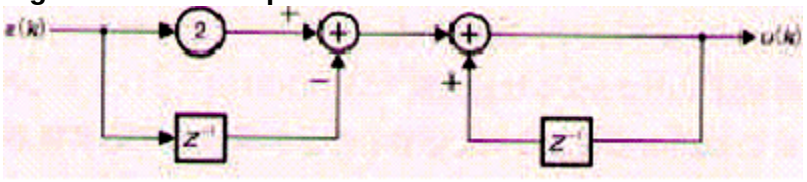


Figure 12. PLL Loop Filter Structure



Now to explain the operation. Figure 11 shows the response when the reference clock period changes from T1 to T2 at time t0: (1) Until time t0 the PLL is locked and the phase error is zero. (2) At time t0, immediately after the period of the signal changed from T1 to T2, since the phase comparator is not yet cognizant of the sender's period, the VCO's control voltage stays the same as before. (3) Next, the VCO output acknowledges the comparator at t1. (4) Then the evaluation of the phase comparator's error begins and is completed by time t2. We can see from the figure that the phase error measured at this point is T2 - T1.

The Time-optimal PLL exploits this single error to lock, as shown in the response in (b), at the next sample point, t3.

The Transfer Function is Derived from the Phase Error Next we shall derive the transfer function between the phase error in Figure 11 (c) and the response of Figure 11 (b). We can represent the signal of Figure 11 (c) using Sequence 1: ..., 0, 0, T2 - T1, 0, 0, ... (Sequence 1)

The Time-optimal PLL's response of Figure 11 (b) can be represented as a time sequence by Sequence 2: ..., T1, T1, 2T2 - T1, T2, T2, ... (Sequence 2)

Sequence 3 is derived by taking the difference of neighboring elements in Sequence 2: ..., 0, 0, 2(T2 - T1), -(T2 - T1), 0, ...(Sequence 3)

Here, by studying Sequences 3 and 1 the following can be discerned: Sequence 3 can be created from Sequence 1, the phase error signal, alone. Use a doubled Sequence 1 (Sequence 4) and a one-sample delayed Sequence 1 (Sequence 5). Now, invert the signs and add. In other words, this can be implemented using a multiply by 2 and a difference.

..., 0, 0, 2(T2 - T1), 0, 0, ... (Sequence 4) ..., 0, 0, 0, T2 - T1, 0, ... (Sequence 5)

Sequence 3 integrated [1/(1-Z^-1)] is Sequence 2, the Time-optimal PLL's response we were trying to derive. The transfer function that takes (c) to (b) is:

$$\frac{2 - z^{-1}}{1 - z^{-1}}$$

The block diagram that implements this is found in Figure 12. Here, e(k) is the phase error output from the phase comparator, and u(k) is the VCO set voltage (frequency/phase).

The left side of Figure 12 shows the multiply that doubles the error; Z^-1 is the unit delay element. Also, the right side of the figure shows the integrator that returns Sequence 2 when Sequence 3 is applied.

The Time-optimal PLL's loop filter can be realized with this simple circuit. (Actually, as for the basic structure, it is necessary for the phase comparator through the VCO gain to be strictly identified in a physical sense.)

7. Completion of the Control System - The Extended Time-optimal PLL

The Problem is the PLL Loop Filter

First let's make it clear where the problem lies. The problem is with the PLL loop filter. This is basically a low pass filter and its cut-off frequency determines the PLL lock-up time. Since the base of a PLL is typically considered to be a crystal oscillator, when jitter performance is questioned, the problems typically occur at high frequencies. Because of this, a low-pass filter that sufficiently attenuates high frequencies is designed. Since the tracking performance or lock-up time is determined by the lower frequencies, the gain at these frequencies is set moderately high.

The problem is in cases where the receiver tracking frequency enters the lock-up frequency band. In such cases, the loop filter allows those frequency fluctuations to pass unimpeded to the VCO.

The Countermeasure

For that reason the following countermeasure was used. The loop filter's frequency response was changed using adaptive control techniques. When not locked it has the response of the Time-optimal PLL; when locked the cut-off frequency moves lower according to the degree of lock.

When this is done, the cutoff frequency is only moved to the left and the Time-optimal PLL's transfer function is preserved. We named this PLL the Sampling Period Adaptive Controlled Tracking System: SpAct (s-pact).

The Meaning of Feed-forward Control

As I have mentioned previously, the greatest benefit of this system comes from feed-forward control.

Sidebar C
History of the Time-optimal PLL - An Interview with Professor Fuminori Kobayashi

Q: How did this PLL come about?
A: I was trying to make a faster PLL and recognized that ordinary PLLs, which are based upon approximation, are limited. If that was the case, I decided I'd look at this a new way a build a new PLL structure from the ground up.

Q: What are the basic elements of high-speed operation?
A: First, 1) I linearized the circuit completely, and then 2) I implemented the circuit completely in discrete time.

Q: Could you explain further?
A: A digital phase comparator is actually a time comparator. Since this is being used to view phase, it is non-linear. If this is completely considered in terms of time (therefore, rather than a VCO, we use a variation of a VPC: voltage-to-period converter), it is linear.

In addition, if an ordinary continuous time loop filter is used, in spite of the fact that the phase comparator operates in discrete time, the two interfere with each other, disallowing further speed improvement. Therefore, I used a discrete time filter.

Q: In what application was this high-speed technique used?
A: It was used in the synchronization circuit for an analog Fourier Analyzer.

Q: How did you make these results known, and how was it received?
A: I published a paper in the *Journal of the Society of Instrument and Control Engineers*. I don't know about the reception, perhaps this was not well understood. Control theorists are not generally familiar with PLLs. I am a control theorist, but my own professor only went so far as to say, "Well…it seems like interesting work…"

Feedback control is a useful and powerful technique since, even if the connected device's (in this case the host's) frequency is not known, the system's frequency can easily be set by calculating the difference. However, the biggest weakness of feedback control is that the only way a frequency fluctuation in the connected device can be known is through the error; dealing with the problem is always relegated until later. In particular, the loop filter's transfer function causes delay, so it is easy for tracking pulses to arise, and it takes great intuition and experience to improve the jitter performance.

On the other hand, in feed-forward control, the receiver must know the sender's frequency. Of course the receiver's own control system must also be identified correctly, so the designer has to be alert. However, if the sender's frequency can be estimated one time, the frequency will be stable even without the feedback loop that causes delay.

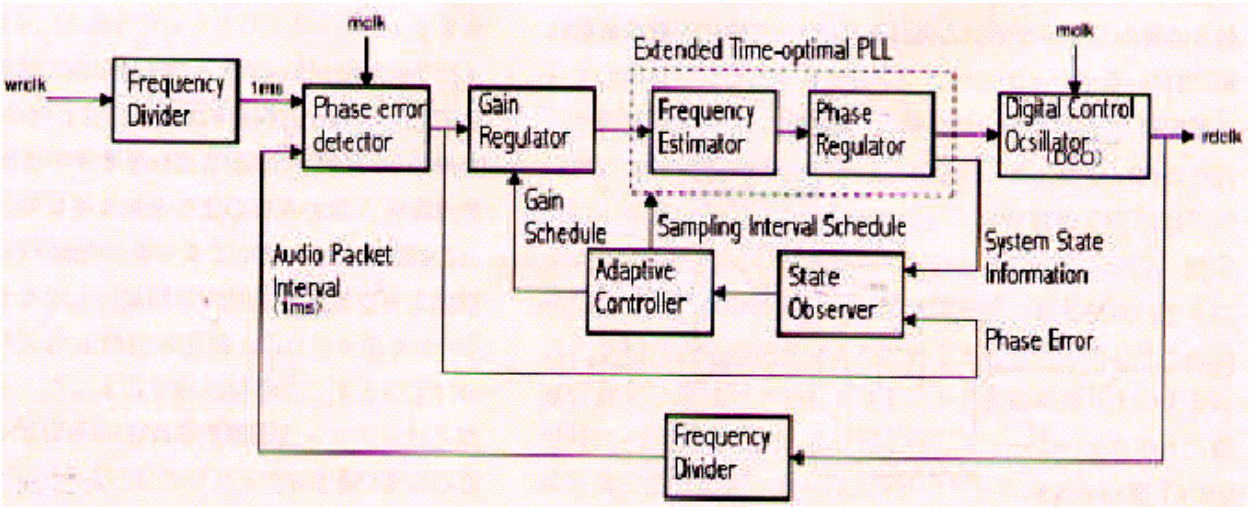
In other words, SpAct deals with this using a two-stage structure: (1) The Time-optimal PLL concept is used and the sender's frequency is estimated. (2) After estimating the frequency, stabilization is accomplished using feed-forward control techniques, and crystal oscillator-like performance is preserved.

Figure 13 shows a block diagram of SpAct.

The signal wrclk is the FIFO write signal that accompanies the arrival of a USB packet. SpAct uses a frequency divider to transform this into a 1 ms signal. The Digital Control Oscillator at the right side of the figure creates the FIFO read signal and the DAC system clock. This rdclk is divided to create a 1 ms signal. The Time-optimal PLL's input signal is created by the Phase Error Detector operating on this signal and the 1 ms signal derived from the wrclk.

The section set off by a dashed line near the center of the figure is the "Extended Time-optimal PLL Circuit." The Sender Frequency Estimator and the Phase Regulator are found here. Beneath that are the State Observer and the Adaptive Controller. The Adaptive Controller uses the Phase Error Detector and the State Observer to adjust the Time-optimal PLL's feedback gain and time constant.

Figure 13. SpAct Block Diagram



SpAct's Principle of Operation

Figure 14 is an overview of SpAct's operation.

The horizontal axis is the USB packet arrival times (the reference clock) and is in discrete time. The vertical axis shows phase error. Let's follow along the horizontal (time) axis.

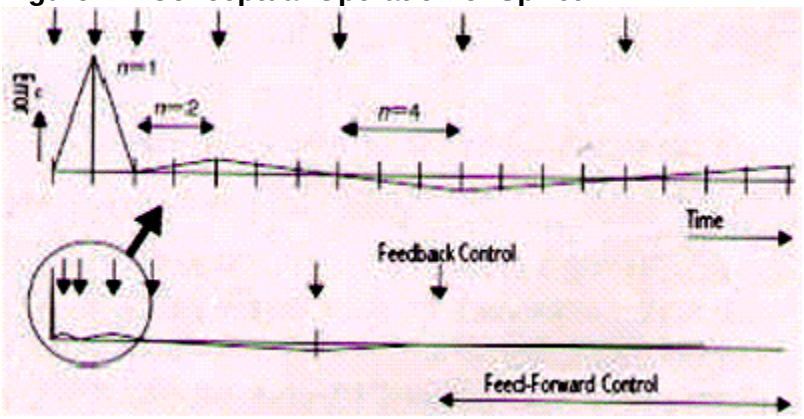
At first, SpAct uses all of the reference clock cycles for control. In this mode, SpAct is a Time-optimal PLL. In the beginning, the sender's frequency has not been identified, so the error is quite large; the USB specification allows for 500 ppm. Then, when the second packet arrives, the first error is measured.

Audio data begins to playback as soon as the first packet arrives. As mentioned before, since SpAct is a Time-optimal PLL, when the next packet arrives SpAct locks completely, including phase. Since Figure 14 is conceptual, the error scaling is overemphasized. But, other than the hump at the beginning, the error is extremely small.

However, even with a Time-optimal PLL, there is no getting around the frequency estimation error. There is measurement error, and system identification error. It follows that if we only had a Time-optimal PLL, it would be oversensitive to this error and begin pulsating (the loop filter has a gain high enough to lock in minimum time, after all). To resolve this, SpAct expands the sampling time.

In Figure 14, every other reference clock cycle is being skipped. The interval is chosen using "adaptive control techniques."

Figure 14. Conceptual Operation of SpAct



The Time-optimal PLL is a sample value type control system, and the change in sampling time changes the system response. But SpAct uses this characteristic in a positive sense, and preserves the Time-optimal PLL's characteristics, except that the sampling time is lengthened. An increase in the sampling time shifts the loop filter toward lower frequencies, which improves the jitter performance.

In this way, since it is able to change the sampling time according to the state of the sender frequency estimate, SpAct is a no-feedback control system. In this case, the gain of the loop filter has become zero. This can correctly be called a feed-forward control system.

Improving the Transfer Function - Scaling the Input Signal

It is necessary to use a scheme to preserve the Time-optimal PLL's performance even though the sample time changes. For that, we modify the transfer function (2 -- z^-1)/(1 -- z^-1). To make a long story short, since the object of control operates in continuous time within some boundaries, changing the sampling time within the sample value system changes the system response.

Figure 15. Changing the Sampling Time

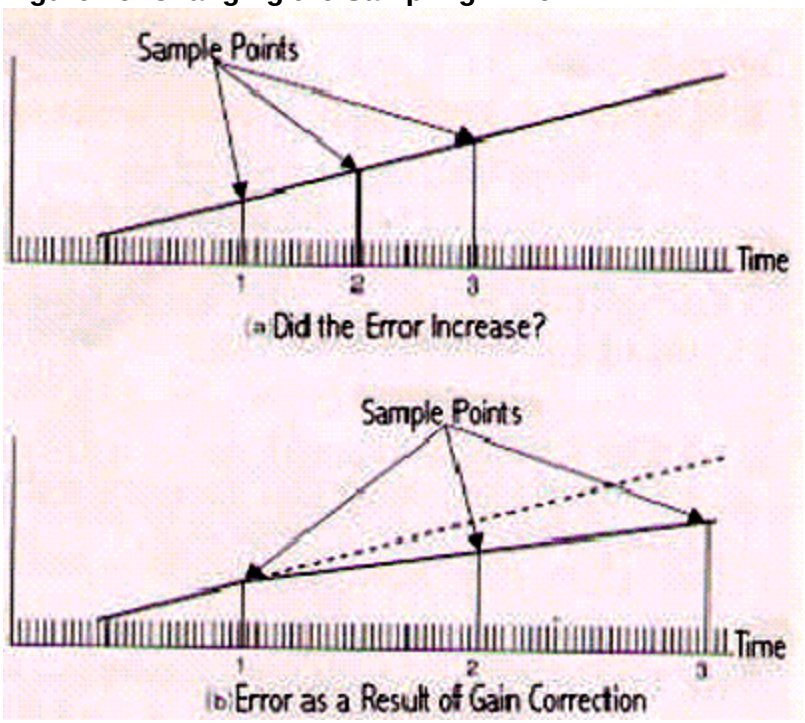
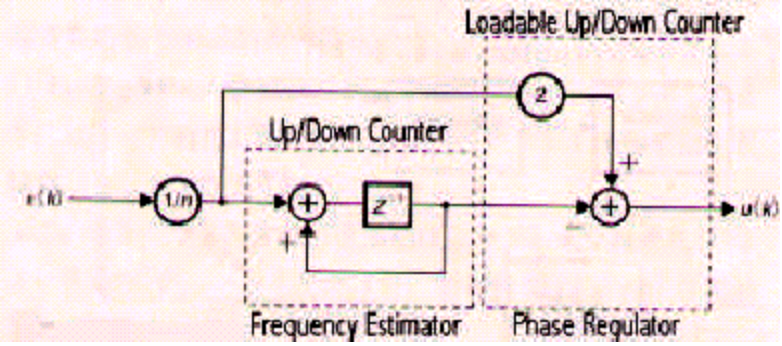


Figure 15 is an example of what happens when changing from one sampling time to another. When the sample time is changed without also changing each sample's excursion (or time constant), the sample value system makes an error as shown by the dotted line in the figure. Here the Time-optimal PLL would mistakenly think a large error had been made. In order to correct for that, we place a scalar on the Time-optimal PLL's input. It can be thought of as a gain correction that allows the error that occurs during n clock cycles to be cancelled within n clock cycles.

The block diagram of the Time-optimal PLL, including the scale factor 1/n, is shown in Figure 16. Notice that we have changed the order of the summers for the delay elements and that the multiply by two now couples in at the end. Also, the two delay elements have been combined into one, which is in the main signal path. Physically, the multiply by 2 performs phase regulation, and on the next sample the delay element does frequency correction. For this reason, we call the left part of Figure 16 the 'Frequency Estimator' and the right part the 'Phase Regulator'.

Figure 16. The Correction to Figure 12



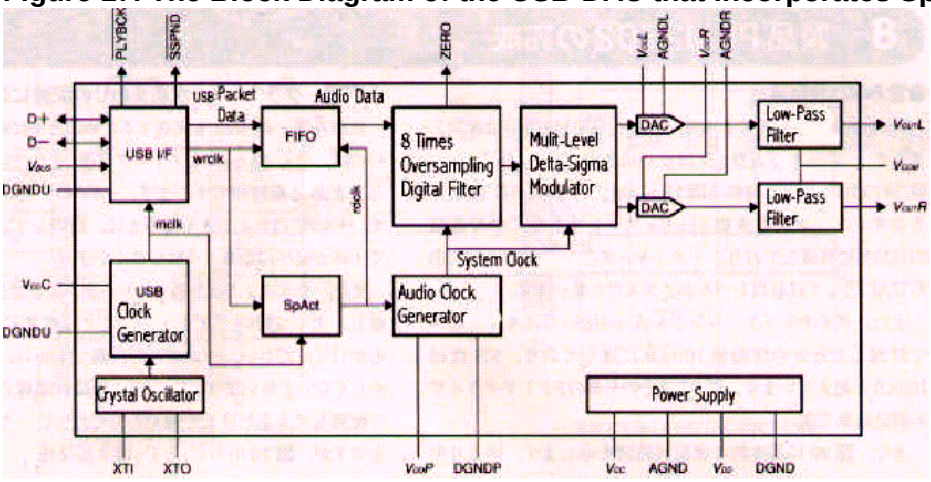
Considering the fact that this is a sample value system and that its operation will not be affected by anything that happens between samples, the digital implementation of this circuit can actually be realized using only a couple of counters (adders). The portion within the left dotted box is an integrator and can be realized by counting, with the clock, up or down based upon the error (the phase error output time). The counter within the dotted box on the right is taking the difference between the data of one sample time ago and twice the error. This can be realized using a loadable counter where the data from one sample time prior is reloaded into the counter every sample time, and the counter counts up or down according to twice the error. In addition, by making the sampling time lengthening factor, n, a power of two, the multiplier gate count can be greatly reduced (only a shift is needed). In this way, the heart of SpAct can be realized using only two counters (adders) and a shifter.

The Structure of the USB-DAC that uses SpAct

The block diagram for the USB-DAC that uses SpAct is shown in Figure 17. SpAct is depicted beneath and to the left of the figure's center.

The clock that is reconstructed by SpAct goes into the Audio Clock Generator which generates the system clock used in the DACs shown at the top of the diagram. In addition, the FIFO read timing is in perfect synchronization with the DAC system clock. The reference clock input into SpAct is the FIFO write clock from the USB interface. The read clock (rdclk) tracks the frequency of that reference.

Figure 17. The Block Diagram of the USB-DAC that Incorporates SpAct



Optimization of the Feed-forward Period

Since SpAct is fundamentally a Time-optimal PLL, it locks by the time of playback of the second packet and from there the feed-forward period is modified according to the state of the sender frequency estimator. Of course, during this time the phase error is continually measured, and if something unusual occurs, the operation returns quickly to that of the Time-optimal PLL.

When the adaptive condition is favorable, the feedback loop gain is zeroed for some periods. These are called "Feed-forward Control Periods." The minute fluctuations of the sender frequency are completely unobserved during these periods. In addition, during the Feed-forward Control Periods, the phase error is detected and shifted to the right by a factor of n. As a direct benefit, the number of effective bits of accuracy of the measurement increases and, with it, the accuracy of the frequency estimate increases as well. Then, the error measurement if fed back, through the Time-optimal PLL, which makes the Feed-forward Period 1, to the integrator and the error is updated in minimum time.

Meeting the Distortion Goals. The D/A Diaries, Part 3 of 3

8. Completion of the PCM2702 Product

The development of SpAct brought about success in the product (PCM2702) development as well. The THD+N is less than 0.002%. The PCM2702 distortion characteristics are shown in Figure 18. This is very close to the theoretical minimum of 0.0015% for a 16-bit, 48 kHz, stereo signal, and must be the top level for a USB DAC that tracks at 1 kHz.

Figure 18a. Distortion Performance for the PCM2702/2703

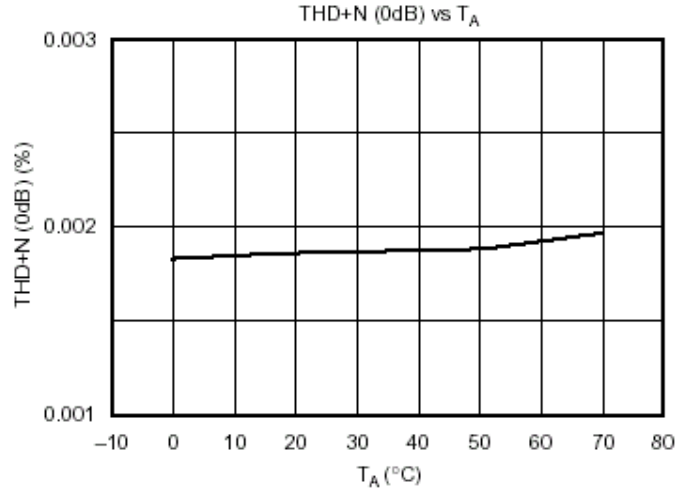


Figure 18b. Distortion Performance for the PCM2702/2703

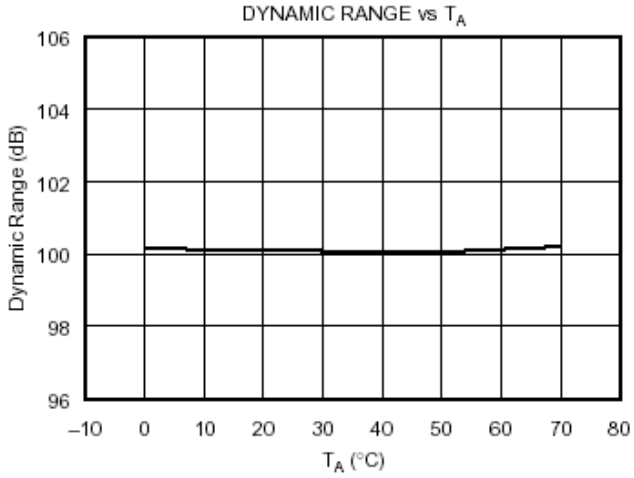


Figure 18c. Distortion Performance for the PCM2702/2703

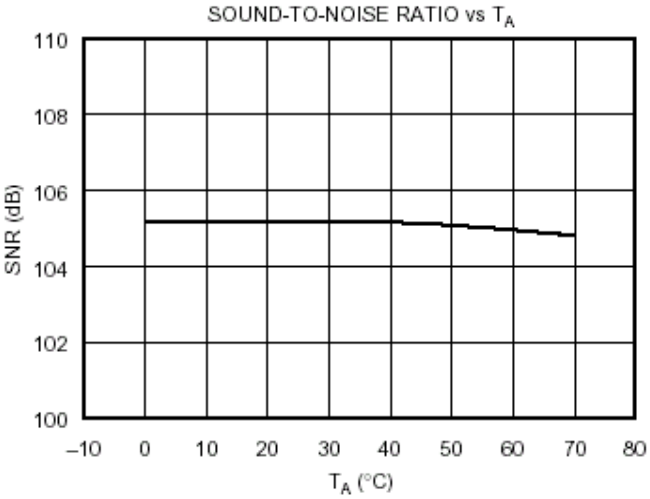


Figure 18d. Distortion Performance for the PCM2702/2703

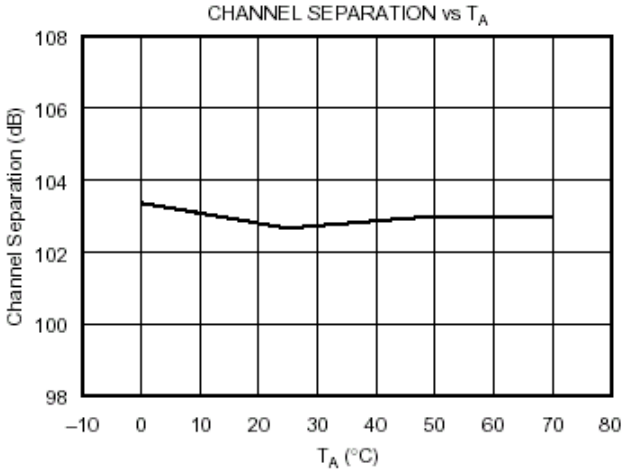


Figure 18e. Distortion Performance for the PCM2702/2703

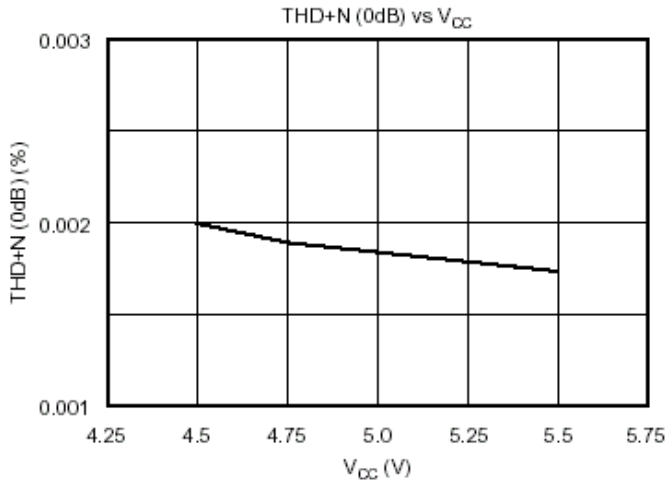


Figure 18f. Distortion Performance for the PCM2702/2703

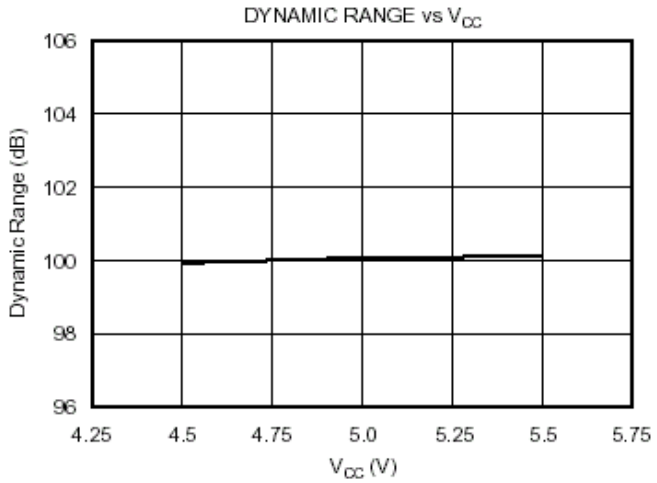


Figure 18g. Distortion Performance for the PCM2702/2703

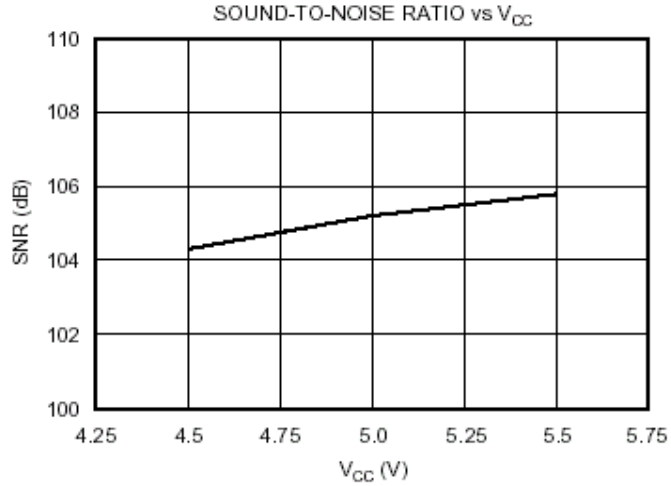
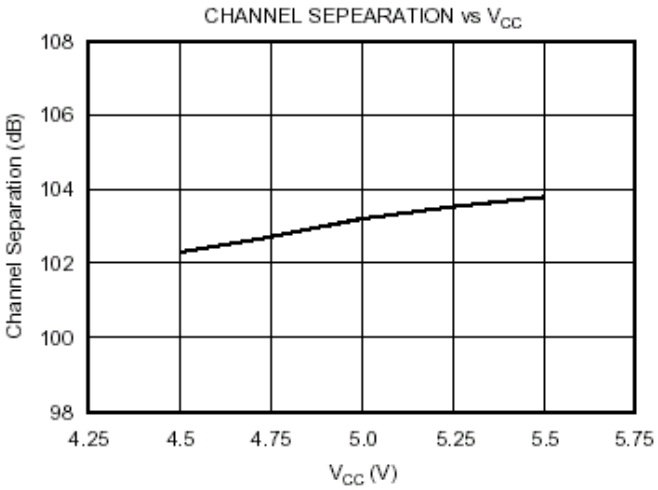


Figure 18h. Distortion Performance for the PCM2702/2703



The A-weighted dynamic range reaches the theoretical value of 100 dB, and the signal-to-noise ratio is over 105 dB. The appropriate FFT is shown in Figure 19.

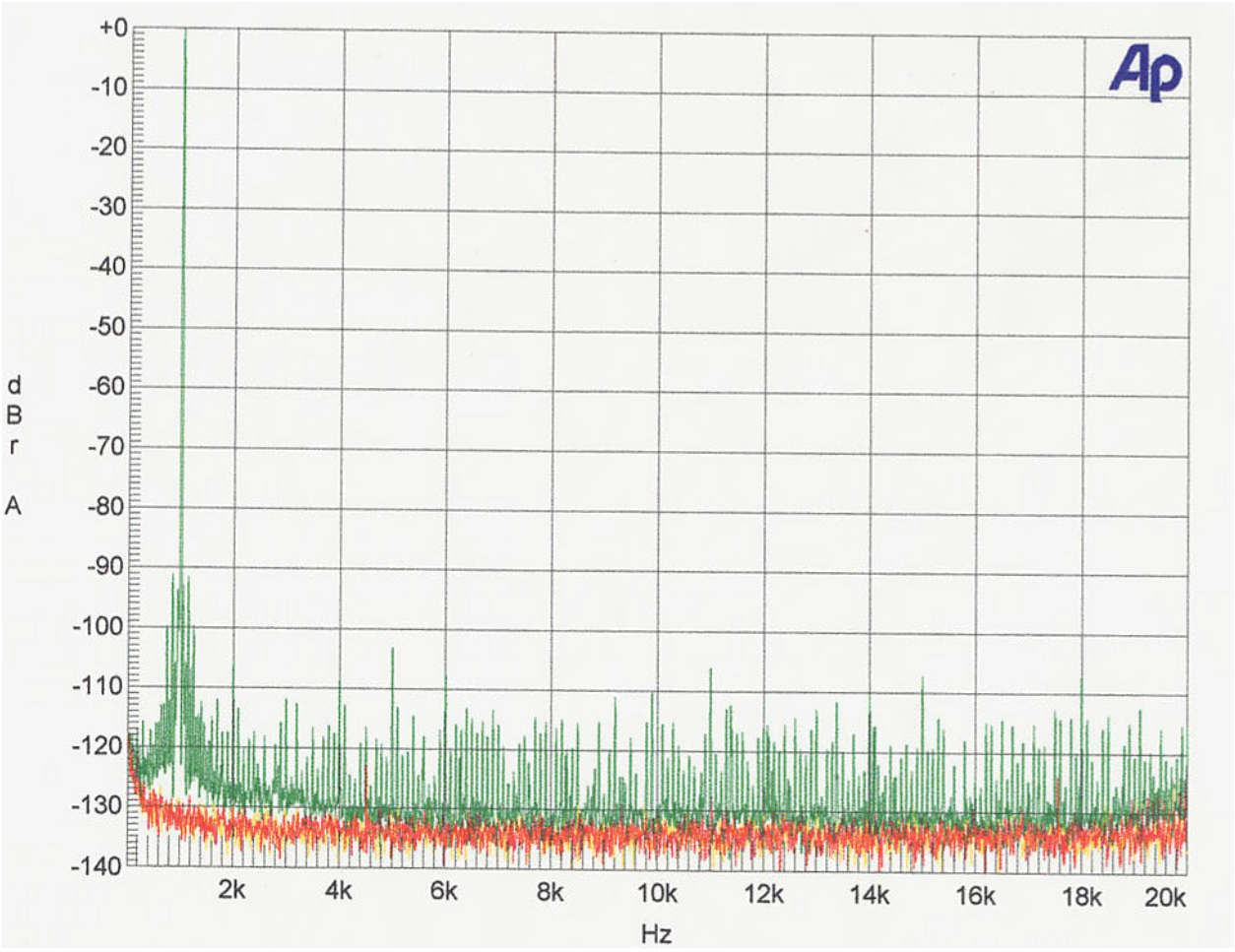
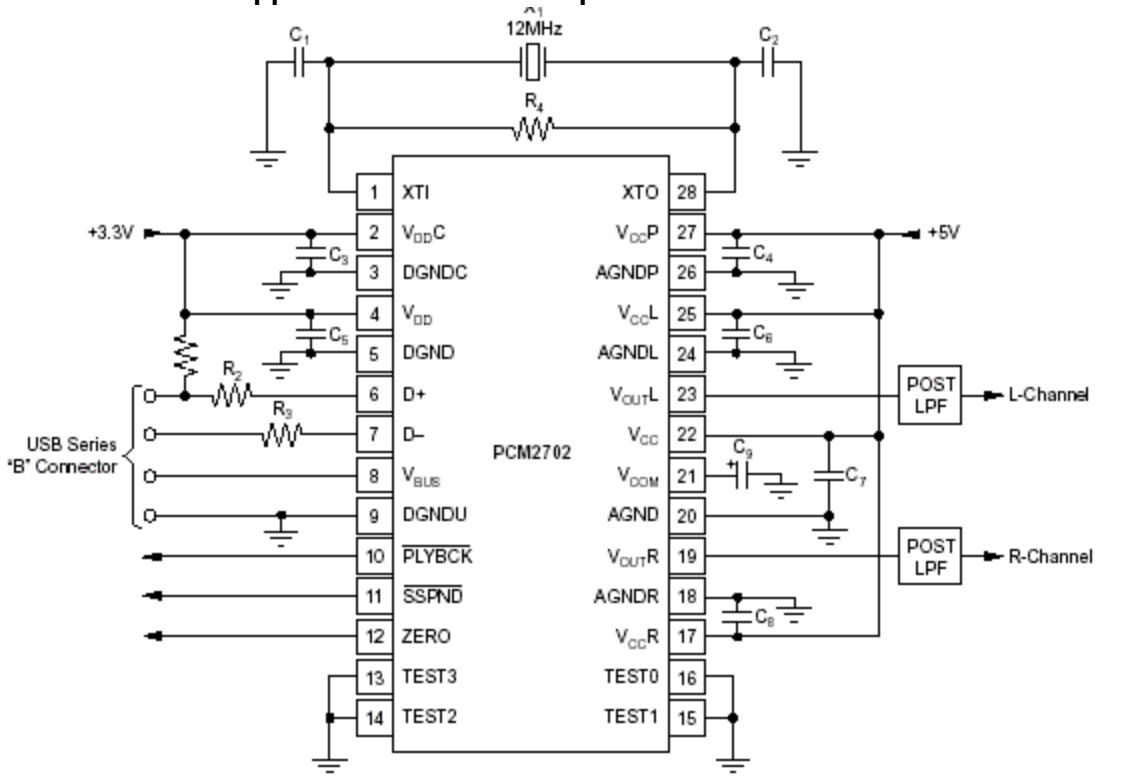


Figure 20. PCM2702 Application Circuit Example



NOTE: C_1, C_2: 10pF to 33pF (depending on Crystal Resonator); C_3, C_4: 0.1µF || 1-100µF; C_5 to C_8: 0.1µF Ceramic each and two 1µF to 100µF for 5V and 3.3V; C_9: 10µF; R_1: 1.5kΩ; R_2, R_3: 22Ω; R_4: 1MΩ; X_1: Crystal Resonator (fundamental mode, parallel resonant).

Also, Figure 20 shows the basic application circuit. As seen from the figure, the PCM2702 requires only a 12 MHz crystal and a few capacitors and resistors as external circuitry.

Take Care in Laying Out the Power, Ground, and Crystal

In order to achieve a low-distortion, high SNR design, please take extra care in laying out the analog power and ground, and the crystal oscillator circuit. First, a +5V analog power supply is needed. By using a separate regulator from that used in the +3.3 digital power supply, the interference can be minimized.

Take care to avoid a common impedance on the ground. Use short, fat traces for ground. It's a simple circuit so shouldn't be too hard. As for the area around the crystal oscillator, it is best to use short fat traces here too. It is well known that interference tuning of C1 and C2 in Figure 20 will be required, depending upon the crystal oscillator.

While this may be redundant, the gain of the oscillator depends upon the ratio of C2/C1. It is easy to design a high gain, but if it goes to far, 3rd, 5th, and higher harmonics are created, and this can be a cause of unusual oscillations. So try to make C1 a little larger than C2.

Damping resistors R2 and R3 (22ohm) are needed to prevent ringing between the device and the USB connector. Also, on D+ a pull-up resistor, R1 (1.5 kohm) is needed; according to the USB specification, it must be pulled up to 3.3V. This is not VBUS (the power line available at the USB connector), so please exercise caution.

PCM2702 Operation 1 - The Interface Protocol

Figure 21 shows the interface protocol.

Figure 21a. Connecting to USB after power up

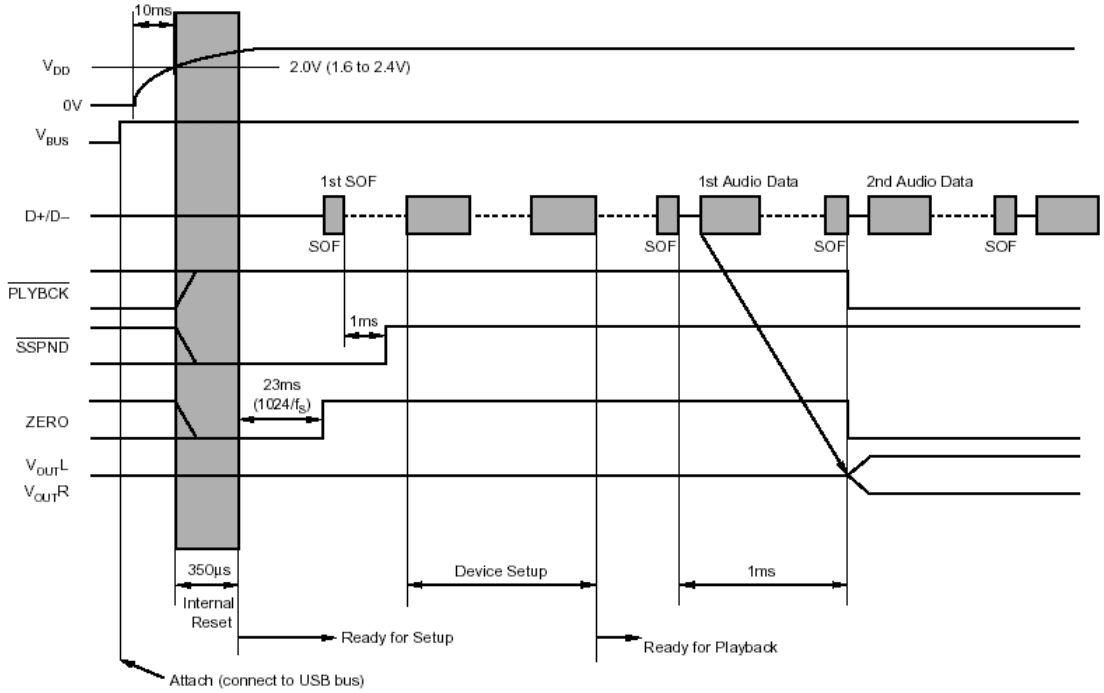


Figure 21b. When the power is turned on after USB attach

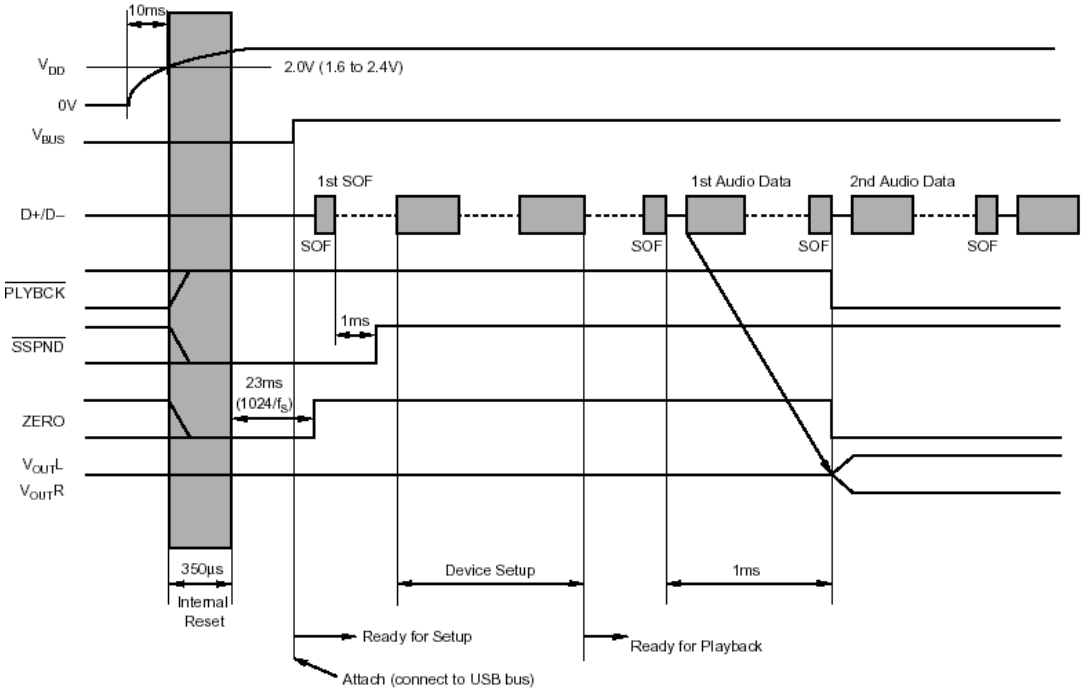
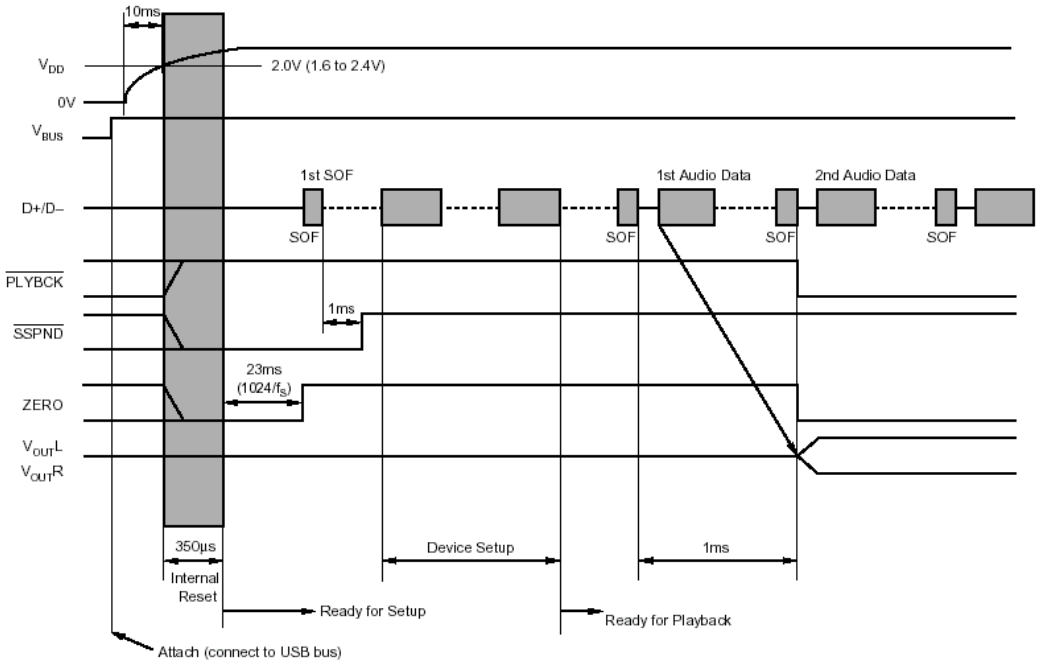


Figure 21c. End of playback, USB detach



(1) About 1,024fs (approximately 23 ms) after the power is connected, when Vdd gets above 2.0V, all device resets go low. At this time the analog output does a level shift to bipolar zero and the ZERO terminal changes from '0' to '1'. At this time the PLYBCK terminal goes low, and the SSPND terminal goes high (they are both low asserted).

(2) When the USB connector is connected to the PC, the VBUS terminal goes to 5V which is detected by the PCM2702. Then the PC (host) detects that the D+ terminal has the 1.5 kW pull-up resistor connected, and begins sending start of frame (SOF) packets at 1 ms intervals.

The operating system might send a reset (Single Ended Zero). We know when this happens because the SSPDN terminal blinks. Of course the PCM2702 will work fine whether or not the reset is sent.

(3) When the PCM2702 detects that the SOF is being sent, it sends SSPND low. After this, the PC issues the PCM2702 a control packet and recognizes PCM2702 as an audio device.

Here the OS comes to know that the connected chip is a Burr-Brown PCM2702.

(4) When the PCM2702 is first connected to the PC (when the first ID is made), the OS prepares the standard, built-in driver. The PCM2702 is designed to work with the standard, built-in drivers in Windows 98, Windows 2000, and Mac OS. There is no need to install a special driver. The user need only follow the instructions from the OS and click 'OK'.

By double clicking the 'Multi-Media' option under 'Control Panel' and selecting 'USB Audio,' the sound that had been coming from a Sound Blaster card or elsewhere, will now come through the PCM2702. When an audio data packet arrives, the PCM2702 awaits the next SOF packet and then begins playing back the audio. At this time the PLYBCK terminal is asserted. The PCM2702 is a high-performance DAC. The output terminal drive capabilities are low by design

in order to control noise. In the case where PLYBCK and SSPND need to drive LEDs, a 74HC04 is used as a buffer.

(5) When audio data packets stop arriving from the OS, the PCM2702 sends PLYBCK low. At this time the DAC output goes to BPZ (bipolar zero), and, in 1,024 fs (23 ms) the ZERO terminal is asserted. Also, if SOF packets stop arriving for 4 ms, because the PC itself has been put into suspend mode, for example, the PCM2702 goes into suspend mode and asserts SSPND.

Sidebar D
Summary of the USB-DAC "PCM2702"

The chip we developed is on the market under the name PCM2702. Here are some of the fundamental specifications that were not referenced in the paper.

Table C shows the pin functionality, and Figure C-1 shows the physical pin configuration. Figure C-2 shows the USB audio function topology, and the analog filter and digital filter responses are shown in figures C-3 and C-4 respectively.

Pin	Name	Type	Description
1	XTI	IN	Connection for a crystal oscillator or external clock[1]
2	$V_{DD}C$	-	Digital power supply terminal for reclocking circuit, +3.3V
3	DGNDC	-	Digital ground terminal for reclocking circuit
4	V_{DD}	-	Digital power supply terminal, +3.3V
5	DGND	-	Digital ground terminal
6	D+	IN/OUT	USB differential input/output terminal (+)
7	D-	IN/OUT	USB differential input/output terminal (-)
8	V_{BUS}	IN	USB Bus power terminal (this pin does not consume any USB bus power)[3]
9	DGNDU	-	Digital ground terminal for the USB transceiver
10	PLYBCK	OUT	Audio playback display pin, active low (L: playback, H: standby)
11	SSPND	OUT	Suspend display pin, active low (L: suspend, H: active)
12	ZERO	OUT	Zero detect pin (L: normal, H: zero detect)
13	TEST3	IN	Test pin 3, connect to digital ground[2]
14	TEST2	IN	Test pin 2, connect to digital ground[2]
15	TEST1	IN	Test pin 1, connect to digital ground[2]
16	TEST0	IN	Test pin 0, connect to digital ground[2]
17	VccR	-	Right channel analog power +5V
18	AGNDR	-	Right channel analog ground
19	$V_{OUT}R$	OUT	Right channel analog output
20	AGND	-	Analog ground
21	V_{COM}	-	Analog output common voltage
22	V_{CC}	-	Analog power +5V
23	$V_{OUT}L$	OUT	Left channel analog output
24	AGNDL	-	Left channel analog ground
25	VccL	-	Left channel analog voltage +5V
26	AGNDP	-	PLL analog ground
27	$V_{CC}P$	-	PLL analog power +5V
28	XTO	OUT	Crystal oscillator connection terminal

1: +3.3 V tolerant
2: Schmidt Trigger with pull down, + 5V tolerant

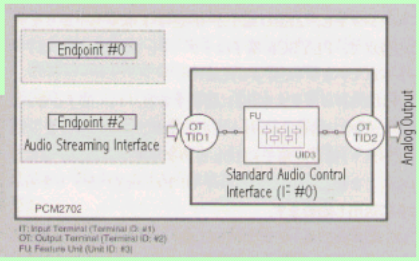


Figure D-1: PCM2702/2703 Pin Configuration **Figure D-2: USB Audio Function Topology**

And here it concludes

By integrating an Interface, a DAC, and a PLL on a single chip we were able to free the user from the trouble of designing a difficult clock circuit. It started as a chip design with a low-cost target, and concluded with the bonus that the user doesn't have to worry about the DAC system clock. By simply connecting the USB connector, a high-performance DAC can be realized.

Although the development can run into a variety of difficulties, semiconductor devices are tiny little things that are naturally expected to work. And, of course, there are few chances to emphasize these tiny devices.

In this case I just happened to get a chance to write. Furthermore, I made it into a story. It wanted readers to enjoy (?) hearing about the struggles of the development process. And I wanted students to know that there are people energetically developing semiconductor devices, even in Japan.

Finally, the Professor Kobayashi who made an appearance herein was my professor at the university. It's been twenty years since I met him. But we still enjoy frequent and friendly communication. Clamor about Industry-University Cooperation has been heard loudly and long. And large cooperative projects are fine. But lately I have come to think that these small but deep relationships with research labs might be even more important. And I feel that industry needs to have more concern for how students, who are Japan's future, will be trained.

References:

1) Fuminori Kobayashi et al., " Time-optimal PLL Using the High-Speed Null Method,)," Journal of the Society of Instrument and Control Engineers, Vol. 16, No. 4, 1980, pp. 573-578
2) Fuminori Kobayashi et al., "A Scheme of PLL with Finite Impulse Responses," IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, Vol. 43, No. 4, April 1996, pp. 340-343
3) Fuminori Kobayashi et al., "High-Speed PLL Frequency Synthesizer for Low Frequencies," IEEE Transactions on Circuits and Systems, Vol. CAS-31, No. 10, October 1984.
4) Fuminori Kobayashi et al., "Efficient Digital Techniques for Implementing a Class of Fast Phase-Locked Loops (PLL's)," IEEE Transactions on Industrial Electronics, Vol. 43, No. 6, December 1996.
5) Universal Serial Bus Specifications Revision 1.1, USB Implementers Forum.
6) Kuwano, Masahiro, "KORE DAKE WA SHITTEOKITAI USB KISOCHISHIKI (Just What You've Always Wanted to Know about USB Fundamentals)," Interface, CQ Publications, March, 2000.

Author's Biography

Hitoshi Kondoh became a ham operator in junior high, and makes an appearance on the airwaves night after night. His specialty is fast keying, and he holds a First Class Amateur Radio qualification. After becoming enchanted with Micon, he enrolled in Nagaoka University of Technology and met Professor Kobayashi (cited frequently in this article). Later he passed the First Class Information Technology Engineers examination. He went to graduate school to study image processing but became interested in parallel processing. He received the First Class Technical High School teaching certificate and returned to his Alma Mater, Tokuyama College of Technology, as a teacher. Then he transferred to the Tokyo Institute of Technology, took a PhD in multi-processing, and made the switch to private industry. After being involved in such pursuits as LSI development for image processing, he made a career change and accepted his present position at Burr Brown. E-mail: kondoh_hitoshi@ti.com