

# Cross Fade of Digital Audio Streams

Fitzgerald J. Archibald

A digital audio stream cross fader requires two streams to be faded and mixed. This paper describes the dual core System-on-Chip (SoC) architecture and capabilities required for optimally implementing a cross fader in audio playback equipment. In addition, the paper provides in-depth analysis, architecture and design of systems software for utilizing an audio cross fader in low power audio playback and recording apparatus.

Index terms: Audio Mixer, cross fader, cross fade system for digital audio.

## Contents

1	Introduction .....	1
2	Cross Fade Algorithm Overview .....	3
3	System Design for Cross Fade .....	4
4	Conclusion .....	11
5	Acknowledgment .....	11
6	References .....	11

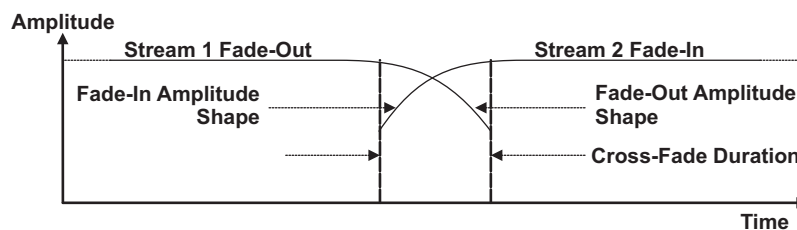
## List of Figures

1	Cross Fade of Two Streams .....	1
2	Simple Cross Fade System .....	2
3	Block Diagram for the Cross Fade Algorithm .....	3
4	Hardware Architecture for Optimal Cross Fade Implementation .....	4
5	Software Block Diagram for Cross Fade .....	5
6	Cross Fade System State Diagram .....	6
7	Sequence Diagram of Cross Fade System .....	8
8	Flowchart of Timing Control Firmware .....	9

## 1 Introduction

Sound producing and reproducing equipment needs audio effects like cross fade to enhance listener experience and mood [1]. Cross fade is a common feature in sound mixing consoles found in recording studios and disk jockey (DJ) consoles. The cross fade of audio streams refers to the mixing of the end of the present stream and the end of the next stream to produce seamless audio as illustrated in [Figure 1](#).

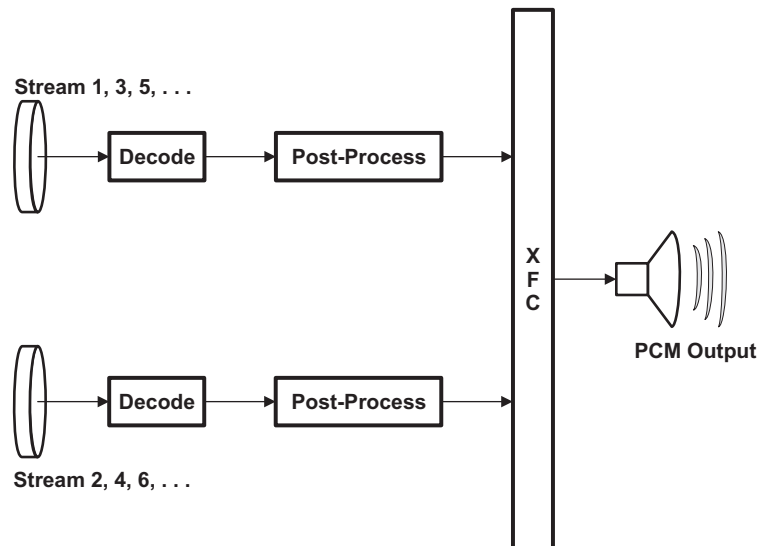
**Figure 1. Cross Fade of Two Streams**



Before mixing part of the stream, a suitable gain change is applied on incoming and outgoing streams. The gain change on the outgoing stream has the causes the fade-out of the stream. That is, the stream gradually reduces in volume. The gain change on the incoming stream is called a fade-in, referring to a gradual increase in volume. The duration for which the incoming and outgoing streams overlap in time is referred to as fade duration. The fade-out duration (time taken for fade-out of the outgoing stream) does not need to be the same as the fade-in duration. Also, the gain curve during fade-in need not match with fade-out. Though various gain curves are possible for fade-in and fade-out, the most effective for cross fade is linear gain change (with respect to amplitude voltage). In practice, sound engineers and technicians apply varying degrees of fade-in, fade-out duration, and gain change during cross-fading (gain curve shape).

**Figure 2** illustrates the conventional method for reproducing audio with cross fade effect. As seen before, the cross fade algorithm needs two streams simultaneously for the cross fade effect. The conventional cross fade system requires higher memory requirement for codec, and algorithm data memory and streaming buffers, because of there are two instances of decoder and post-processing modules. The codec program memory also would increase if the streams to be cross faded are encoded in different audio formats. This would require the audio playback or reproducing devices to be capable of running simultaneously two decoders and post-processing algorithms. The common post-processing algorithms for audio are equalizer and/or bass/treble.

**Figure 2. Simple Cross Fade System**

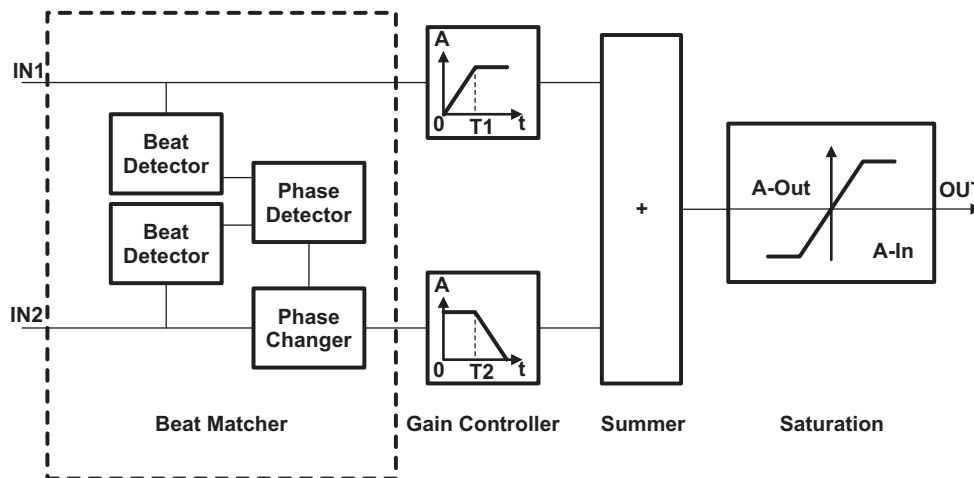


Streaming buffers are data buffers carrying data between various elements within the modules such as a decoder, a sample rate converter (SRC), and a cross fade controller (XFC). The hardware for portable audio players consists of one or more processor(s), limited on-chip memory, and peripherals. The processor could be a digital signal processor (DSP) or a general purpose processor (GPP) with DSP extensions. In systems where on-chip memory is low, the placement of two simultaneous audio playback systems, required for cross fade, gets challenging in terms of memory. The solution for memory constraints is to overlay the on-chip memory region with code and data that are mutually exclusive in terms of program execution [1], [2]. The modules that need to be executed are loaded into on-chip memory from external memory before execution commences. Once the execution is completed, the state of the present module is saved in external memory and/or on-chip memory that is not used for overlay. Then, the next module that needs to be executed is overlaid on top of the current module in on-chip memory. The reproducing system with excessive overlay is less efficient in terms of power efficiency due to the overlay execution cycle overhead, and to frequent external memory access.

## 2 Cross Fade Algorithm Overview

As illustrated in [Figure 3](#), the cross fade algorithm performs summation on the two input streams. To be able to sum two audio streams, both streams need to have the same sample rate. Reproducing systems could be expected to support various compressed audio formats like Motion Picture Expert Group (MPEG) 1 Layer 3 (MP3), Advanced Audio Coding (AAC), and Windows Media Audio (WMA) (reproducing systems also might be expected to support compressed lossless streams like WMA lossless) along-with uncompressed Pulse Code Modulation (PCM) samples (WAV format). Thus, the reproducing systems should be capable of playing all supported sample rates. Typical sample rates for audio are 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, and 96 KHz. This would require the presence of sample SRC as one of the post-processing modules in [Figure 2](#). The sound reproducing system could have speech decoders as well.

**Figure 3. Block Diagram for the Cross Fade Algorithm**



The cross fade algorithm (cross fade controller, XFC), as shown in [Figure 3](#), therefore takes two input streams, matches the beats of the fade-in and fade-out streams, applies fade-in and fade-out gain, and performs saturated sum of the faded-in and faded-out streams [4]. The beat matcher is optional and if used, the processing unit should have enough cycles to perform beat matching [3]. Beat detection and phase changer need to be enabled only during the end of the fade-out stream (maybe a few seconds) before the fade-out operation starts, so that the fade-out stream beats can be phase aligned with the fade-in stream.

The phase change needs to be applied gradually to reduce audible skips or pitch change (depending upon method of implementation of phase changer). Dynamic clock change is suitable to handle this increased load during brief period [2], [5], [6]. If dynamic clock change is not possible, the device needs to be clocked higher as compared to normal to accommodate the beat matcher near to the fade-out time. The saturation is to avoid overflow in finite precision arithmetic module. For a positive 45 degree gain curve slope for the fade-in gain curve and a 45 degree negative slope for the fade-out gain curve, the clipping or saturation is not essential. The gain curves for fade-in and fade-out also could be set in such a way as to create headroom, which prevents clipping/saturation of the result of the mixer.

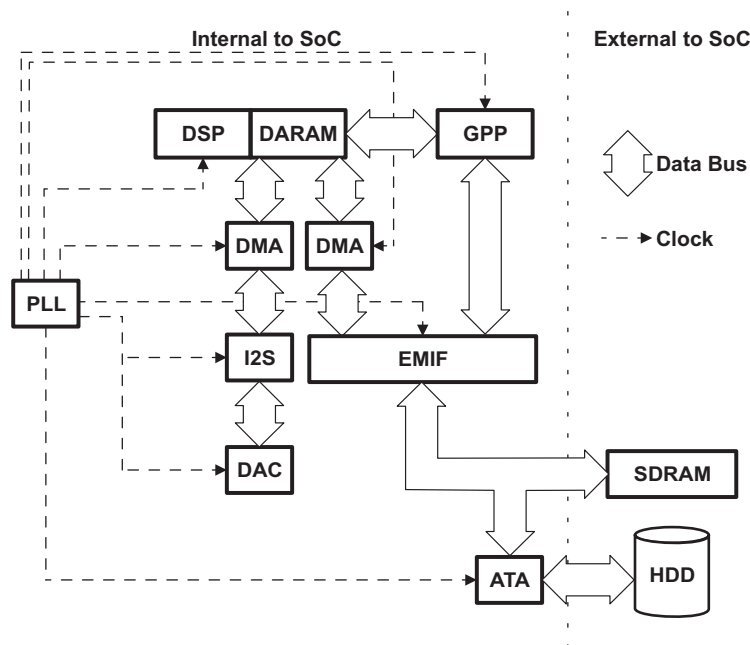
The side effect is loss of dynamic range. In the case of continuous playback of audio streams, the same stream needs to be faded-in at the start and faded-out at the end. This would require the beat matcher and gain controller to alternate between the two inputs to the summer. In simple terms, the same stream is switched between the inputs to cross fader inputs IN1 and IN2. That is, the stream starts on fade-in input (IN1) and as soon as fade-in is completed, the stream is switched to fade-out input (IN2). However, the fade-out operation or gain controller is applied only for the fade-out duration at the end of stream.

### 3 System Design for Cross Fade

#### 3.1 Hardware Architecture

Figure 2 shows the hardware components of the SoC to perform cross fade of audio streams [2]. The storage media contains one or both streams to be cross faded. The storage media is capable of providing data to external memory and/or on-chip memory via interfaces like AT Attachment (ATA). The stream for cross fade (compressed or uncompressed) in on-chip memory is processed by a DSP. The GPP could contain a DSP extension and on-chip memory in the case of a single core SoC. The compressed/uncompressed stream is stored in synchronous dynamic rAM (SDRAM) or retrieved from SDRAM by the DSP/GPP by using direct memory access (DMA). The external memory interface (EMIF) provides the interface to external memory (SDRAM) for direct access by the GPP/DSP or DMA. The PCM samples are delivered to the digital to analog converter (DAC) by the DMA by way of the Inter-IC sound bus (I2S) interface. The DAC can be external to the SoC. An external crystal can be used to deliver a clock to the various peripherals, interfaces, and processor core in SoC. Phase locked loop (PLL) can be used to derive different clock frequencies from crystal by the use of multipliers and dividers. The PLL can be programmed to a different set of frequencies for the processor core, depending upon the computational load. The programming of the frequency would need voltage control of core as well (via the general purpose IO (GPIO) or power management unit (PMU) controlled through the I2C interface. A dynamic clock change for the processor during audio playback can be achieved if the processor clock is decoupled with the DAC/DMA clock. The clock domain grouping (number of sets of clock frequencies) and isolation is not necessarily required as shown in Figure 2.

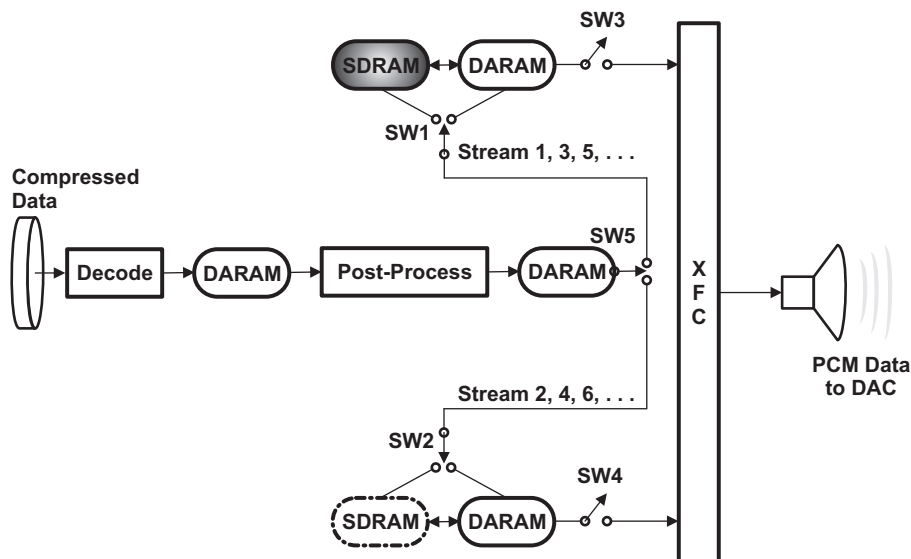
**Figure 4. Hardware Architecture for Optimal Cross Fade Implementation**



### 3.2 Firmware Design

Figure 5 shows the software components required to realize the cross fade of audio streams.

Figure 5. Software Block Diagram for Cross Fade



The compressed/uncompressed stream is transferred in small chunks from external memory to on-chip memory. The decoder program decodes the compressed data contained in the streaming buffers, located in on-chip memory. In the case of an uncompressed stream as in linear PCM samples in WAV format, the decoding operation would be simple streaming or passing through the PCM samples. The input stream buffer is refilled for consumed data for the next frame decode. The decode data is processed by post-processing module. In the simplest configuration, the post-processing module might be an SRC. The SRC is essential to ensure that the fade-in and fade-out streams are converted to the same sample rate before passing to XFC.

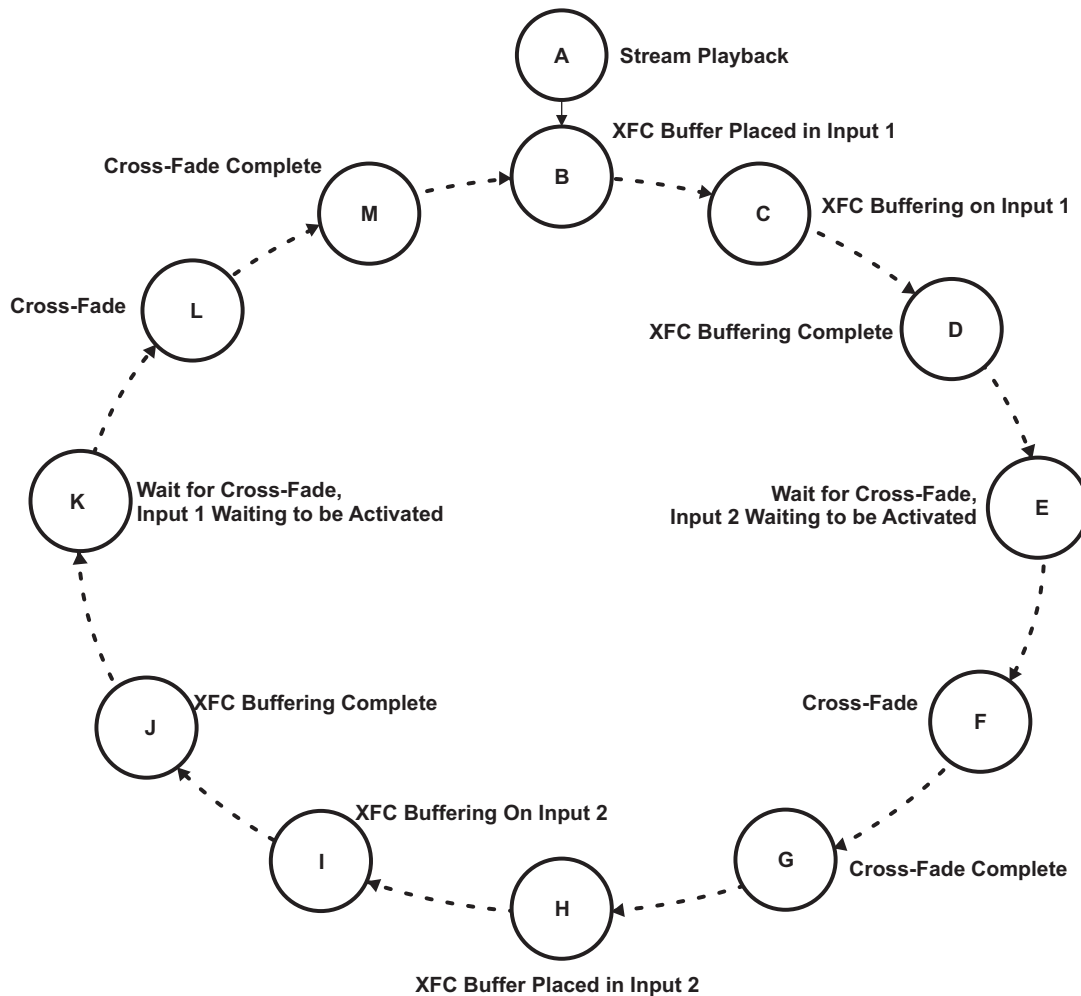
The post-processed data is available for streaming in buffers in on-chip memory. The switch sw5 controls the streaming data to either switch sw1 or sw2. The odd numbered streams are passed onto sw1 and even numbered streams are passed onto sw2. The first stream is odd and every subsequent stream alternates between even and odd. The stream here refers to the streams for cross fading. Initially, switch sw1 routes data to the streaming buffer. The switch sw3 is in closed position delivering data to the cross fade input. The stream passes through XFC without any alteration to the buffered samples for playback by DAC. The external memory buffering between the DAC driver and the cross fade module is needed only if a program swap or memory overlay is needed to accommodate the reproducing device in on-chip memory. The samples reservoir, in SDRAM, is activated once the buffering time is met, by changing switch sw1 state. The real time playback of audio continues by streaming PCM samples through buffers from the samples reservoir via the path containing switch sw3. On reaching the end of the stream for fade-out, switch sw5 changes position for streaming any even numbered stream to sw2. The switch sw4 will be held in open position to prevent streaming of fade-in stream. On reaching fade start time, the switch sw4 is closed to allow streaming of data to XFC input.

The input from the samples reservoir in input sw3 and the decoded samples from input sw4 are faded-out and faded-in, respectively, by XFC. The faded streams are mixed by XFC before outputting to DAC. Since the output is driven by DAC periodically, the fade-in stream will be available in real time (both inputs and output are present) for XFC. XFC feeds current PCM samples in real time to DAC, after the past samples are played out. In the event of a large PCM sample buffer following XFC (might be needed for overlay application to control overlay frequency), the cross fade operation is allowed only if audio samples are available at both inputs of XFC. Otherwise, the samples reservoir would be emptied by the buffer following the cross fader. In the case of a real time miss reported by DAC or a bit stream error reported by the decoder, the presence of both inputs condition is relaxed to allow the fade-out stream to continue playback until recovery from error. Once the samples reservoir is empty, the reservoir buffer is transferred to the even stream path (path having switch sw2). The samples reservoir and cross fade operation starts from the even stream path as the timing conditions are met.

### 3.2.1 Cross Fade System State Transition

Figure 6 illustrates the state transitions in a sound reproducing system.

**Figure 6. Cross Fade System State Diagram**



#### State A

In the case of a reproducing system without the presence of cross fade, XFC is set up for passing through the PCM samples. Smooth fading-in/fading-out are required in features of the reproducing system such as mute, unmute, volume change, pause, resume, stop, and start. XFC can be used for achieving smooth fade-in and fade-out by configuring gain curves and disabling mixing (summer and saturator).

#### State B

When the cross fade feature is enabled in the reproducing system, the samples reservoir buffer memory is to be provided by the system. The reservoir has to be large enough to buffer PCM samples whose playback time is greater than or equal to the fade-out or fade-in duration, whichever is larger. The samples reservoir can be in external memory or on-chip memory. Until buffering time/reservoir filling time is reached, the data streams to XFC bypass the reservoir. Though there are no streams to cross fade initially, the input stream is faded-in.

### **State C**

When the buffering start time is reached, the samples reservoir is enabled. The buffering start time is the time before the cross fade start time at which the samples reservoir would hold PCM samples to play for the duration of cross fade. When the samples reservoir is enabled, the decoded PCM samples are directed to the samples reservoir. SRC can be after the samples reservoir, which would reduce samples streamed in and out of the reservoir, if SRC converts a lower sample rate stream to a higher sample rate. DAC consumes PCM samples in chunks. As soon as DAC consumes a PCM chunk, the next PCM chunk is made ready by streaming samples from the samples reservoir.

### **State D**

On reception of the end of stream (EOS), the samples reservoir would contain samples equivalent to or exceeding fade-out duration. Once the decoder propagates EOS, the decoder can be replaced with the decoder for next audio stream. For example, MP3 decoder may be replaced by AAC decoder.

### **State E**

Decoding of the fade-in stream is started but streaming to XFC is suspended till the fade-in trigger time is reached, by opening the streaming connection to XFC. The alternative method is to signal XFC to stop consuming data from input.

### **State F**

Once the samples reservoir holds samples equal to or less than the fade duration, the fade-in stream is fed to XFC input, thus starting up streaming of the fade-in stream. The presence of both fade-in and fade-out streams triggers the cross fade module to perform cross fading of the two streams with pre-configured gain curve. Alternatively,

- If the fade-in time is not equal to and less than the fade-out time, the samples reservoir module signals XFC to start fading-out once the samples reservoir level reaches less than or equal to the fade-out duration. Once the samples reservoir level reaches below or equal to the fade-in duration, the fade-in operation is triggered, by connecting SRC output to XFC input.
- If fade-in time is not-equal to and greater than the fade-out time and when the sample level reaches fade-in duration, fade-in is initiated by connecting SRC output to XFC input. Fade-out is started by samples reservoir signaling the XFC to start fade-out.

### **State G**

Fade-out completes when the samples reservoir gets empty or the fade-out gain curve reaches inaudible gain. The fade-in completes on reaching target gain or the end of stream, whichever is the earlier. Completion of fade-in and fade-out marks the completion of the cross fade operation.

### **State H**

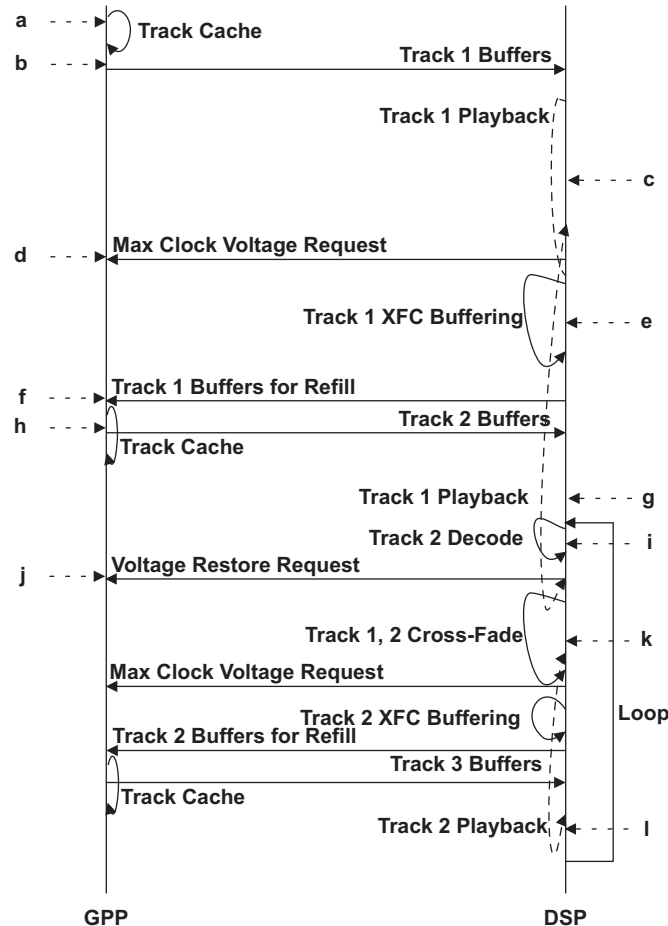
The samples reservoir is shared or transferred to the faded-in stream, thus preparing the faded-in stream for the fade-out operation. In the case of the abrupt ending of the faded-in stream (before fade-out), the decoder for the next stream format is loaded by the system and the stream starts fading-in. This state is identical to state B with XFC input pin roles swapped.

### **States I-M**

The states I, J, K, L, and M are identical to state C, D, E, F, and G with XFC input pin roles swapped, thus completing a cycle.

## **3.2.2 Cross Fade Signaling System**

[Figure 7](#) is the sequence diagram of the system for reproducing audio with cross fade illustrating a dynamic clock change. The sequence diagram illustrates the cross fade system with the presence of a GPP and a DSP, though a dual processor core is not mandatory for grouping control and signal processing.

**Figure 7. Sequence Diagram of Cross Fade System**


The description flows from the top to bottom of the diagram corresponding to the timeline.

1. The stream for reproducing is buffered into on-chip or external memory (track cache).
2. The buffers are provided to playback system.
3. The playback system converts the input stream (compressed or uncompressed) to PCM samples for analog conversion and amplification, using DAC, to drive speakers.
4. On reaching buffering start time, the clock frequency change request is raised. Clock frequency change might require operating voltage change prior to increase in clock frequency.
5. The samples reservoir is enabled for buffering, once clock setting is completed. In devices where dynamic clock change is not permissible while audio playback is ongoing, the device could operate at an increased clock over the entire duration of the playback thus eliminating the need for clock change. In such case, the samples reservoir is filled from start of stream decode till end, gradually increasing level of samples in reservoir and reaching the level needed for fade duration when the stream reaches end.
6. On completion of decoding track buffers, the track buffers are available for refill.
7. The stream playback continues from samples reservoir in the mean time.
8. The next track buffers which are already buffered, before present track buffers are returned for refill, is sent for fade-in and playback. The fade-in track format, if different, requests the system to install corresponding decoder in place of the previous decoder. In systems where 2 or more tracks can not be buffered due to limited memory, the fade-in stream is buffered on receiving buffers for refill. In this case, the samples reservoir has to have adequate number of samples (in addition to fade-out duration) to handle the delay arising from track buffer refill operation.

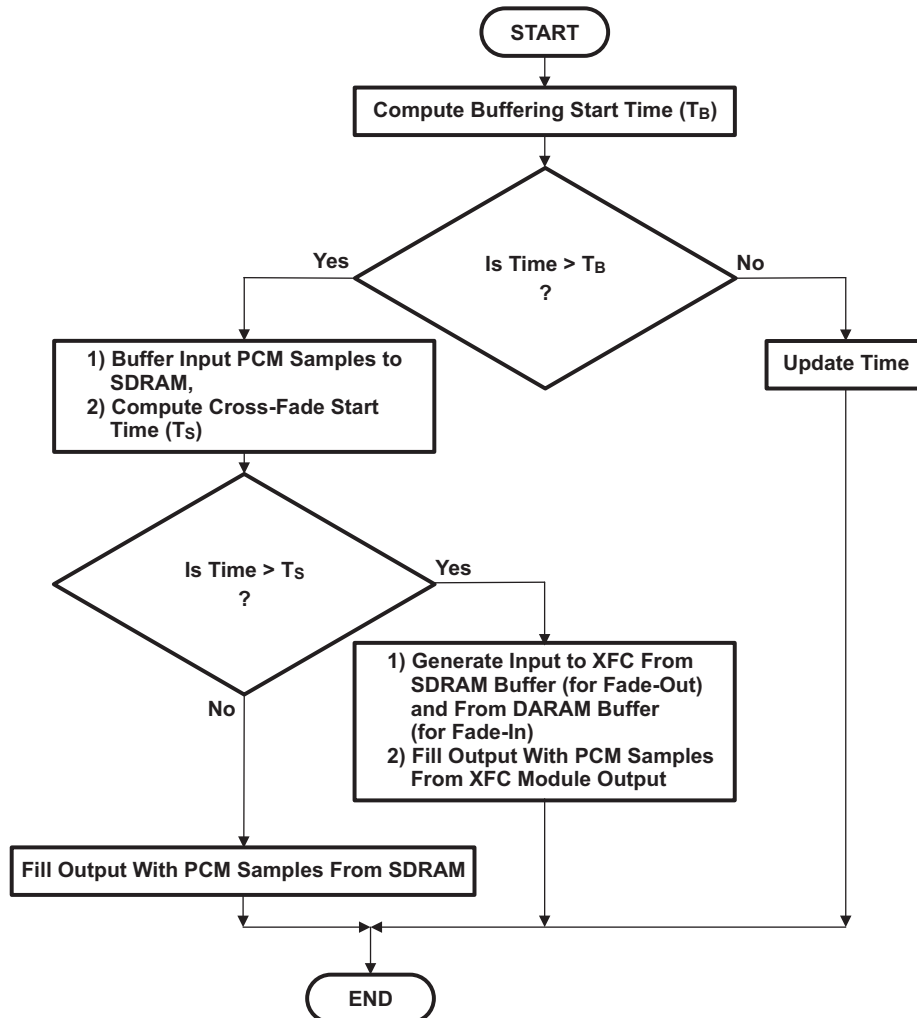


9. The fade-in track starts in the mean time, while fade-out stream playback is in progress. The fade-in track progresses up to few frames for which the on-chip memory has space for holding the PCM samples. Then the decoder idles until fade-in operation starts.
10. The clock is restored to the minimum required by the cross fade system for playback, if the clock was increased in step d. The clock once restored to minimum required, the core voltage can be lowered to the operating voltage needed for the selected frequency. The voltage and frequency tables could be pre-computed and saved in table format.
11. Once the samples remaining in samples reservoir correspond to fade-out duration in play time, cross fade operation is started. This is applicable when fade-out time is greater than or equal to fade-in time. Otherwise, fade-in starts first followed by fade-out so as to meet different fade-in and fade-out time.
12. On completion of fade-in operation, the stream plays back at constant gain, as mentioned in step 3. Further processing continues by repeating steps d to i till end of playback.

### 3.2.3 Timing Computation

Figure 8 illustrates the timing conditions required for the cross fade operation by means of a flow-chart representation. Determine the buffering start time. Update the playback time if the buffering time is not met. If the current playback time is greater than or equal to the buffering start time, stream data to the samples reservoir and update the playback time. Compute the cross fade start time. Check if the playback time has exceeded the cross fade time. If the cross fade time is not yet reached, stream data to DAC from the samples reservoir.

Figure 8. Flowchart of Timing Control Firmware



If the cross fade time is reached, the following occur:

1. Faded-out the stream from samples reservoir
2. Fade-in the stream from decoder as the decoder would have started decoding fade-in stream
3. Mix the faded-in and faded-out streams
4. Stream the cross fade stream to DAC.

The above mentioned process continues for the fade-in stream to cross fade with next stream for fade-in, on completion of present cross fade.

$$T2 = \left(1 + \frac{f}{f_{max} - f}\right) \times T1 \times \frac{BR}{8} \quad (1)$$

$$T = SZ - T2 \quad (2)$$

$$N1 = \frac{SZ}{BR/8} \times SR \quad (3)$$

$$N2 = T1 \times SR \quad (4)$$

$$N = N1 - N2 \quad (5)$$

In the equations, the following abbreviations are used:

- BR is the average bit rate (in bits per second) of the audio stream.
- SR is the sample rate (in samples per second).
- SZ is the file size (in bytes) excluding meta-data (such as id3 tag in mp3).
- T1 is the cross fade duration (in seconds).
- f is the clock frequency for normal playback (in MHz).
- fmax is the maximum clock frequency for cross fade buffering (in MHz).
- T is the samples reservoir buffering start point when decoder reaches the byte on the input audio stream.
- N is the PCM sample at which cross fade operation starts.
- N1 is the number of samples contained in the entire stream.
- N2 is the number of samples corresponding to cross fade duration.
- T2 is the number of bytes of compressed bitstream remaining to be decoded when the reservoir buffering is started.

Equations (1) to (5) provide the time computation for the start of the samples reservoir buffering and cross fade. Equation (1) computes the amount of input compressed/uncompressed data (T2) remaining to be decoded in the present stream before the start of samples reservoir buffering. Equation (2) computes the byte position T on the input stream to the decoder at which the samples reservoir needs to be enabled for buffering PCM samples. The file size SZ is the amount of input stream containing audio data (excluding meta-data, and file headers/trailers). In the case of a device where dynamic clock change is not possible, (1) and (2) can be used for computing the maximum clock frequency required (fmax) for continuous and gradual reservoir filling throughout the stream. T is set to the start of the audio stream and f is the worst-case clock required for a playback system without samples reservoir buffering. The computation can be done offline, and the clock frequency table generated. The clock frequency should account for the worst case clock requirement by any decoder if multiple format cross fades were to be supported. Otherwise, worst case clock frequency for a given decoder needs to be used in the system. Fade-out start time in terms of samples played by DAC can be determined by (5). In the case of bit stream errors being encountered by the decoder, the file size is adjusted and the time parameters recomputed. Also to handle variable bit rate streams, time computation needs to be performed periodically at a sufficient interval (about 1/10th of fade-out duration) to ensure fade-out duration accuracy. Also it is possible to detect fade-out time by detecting the samples reservoir level after the end of the input stream. The samples reservoir would hold as much or more PCM samples as required for real time playback during fade-out duration, when fade-out needs to be signaled or started.

## 4 Conclusion

The cross fade system is designed for optimal power and performance by using a dynamic clock frequency change of processor core during audio playback. The processor clock frequency is increased during buffering of PCM samples into the reservoir during fade-out preparation. The clock is restored back to the minimum required for real time playback on the start of the cross fade effect. The optimal timing used in triggering reservoir buffering and the cross fade effect minimizes external memory access, external memory size, and the duration of higher operating clock frequency during playback, as can be seen from (2). Thereby achieving low power (approximately 60 mW of total system power) design goals in SoC intended for handheld electronic devices.

## 5 Acknowledgment

The author acknowledges valuable inputs on SoC architecture from Mike Polley and Stephen Li of Texas Instruments.

## 6 References

1. *Audio System for Portable Market* by F. J. Archibald, AES Preprint 6906, Convention 121, Oct 2006.
2. *System-on-a-chip for Portable Audio Player* by F. J. Archibald, M. O. Polley, S. H. Li, J. D. Kridner, ICCE 2007, pp. 1-2, Jan. 2007
3. *Design of an Automatic Beat-Matching Algorithm for Portable Media Devices* by S. Fedigan, D. Jochelson, AES Preprint 6927, Convention 121, Oct 2006.
4. *Hang the DJ: Automatic sequencing and seam-less mixing of dance-music tracks* by D. Cliff. Available: <http://www.hpl.hp.com/techreports/2000/HPL-2000-104.pdf>
5. *Power Optimization and Management in Embedded Systems* by M. Pedram, Proceedings of the 2001 conference on Asia South Pacific design automation, pp. 239 – 244.
6. *Power optimization of variable-voltage core-based systems* by I. Hong, D. Kirovski, Q. Gang, M. Potkonjak, M. B. Srivastava, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, Volume: 18, Issue: 12 pp. 1702-1714, Dec 1999.



**Fitzgerald Archibald** earned a B.E (Electronics and Communication Engineering) from PSG College of Technology, Coimbatore, Tamil Nadu, India in 1996. He worked on control systems software development for geo-synchronous satellites from 1996 to 1999 in ISRO Satellite Centre, Bangalore, India. In 2001-2002, he worked on speech decoder, real-time kernel, and audio algorithms for DVD audio team in Sony Electronics, San Jose, USA. While in Philips Semiconductors (Bangalore, India, and Sunnyvale, USA) in 1999-2001 and 2002-2004, he worked on audio algorithms and systems for STB, DTV, and internet audio. He is part of the Personal Audio Video and Digital Imaging groups in Texas Instruments Inc, Bangalore, India from 2004-till date working on audio, video, and imaging systems and algorithm development. Interests include multimedia and control algorithms and systems.

Mr. Archibald is member of AES.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2008, Texas Instruments Incorporated