# Implementing AC-link With ESAI

by

Ilan Naslavsky and
Leonid Smolyansky

# CONTENTS


**1 Introduction**

1.1 Scope . . . . . 1-1
1.2 AC '97 Digital Controller . . . . . 1-1
1.3 The AC '97 Digital Serial Interface (AC-link) . . . . . 1-2

**2 Physical Connection**

2.1 AC-link Lines . . . . . 2-1
2.2 ESAI Pin Connections . . . . . 2-2

**3 Data Flow**

3.1 Output Data Streams . . . . . 3-1
3.2 Input Data Streams . . . . . 3-1
3.3 Data Flow Model . . . . . 3-1
3.4 FIFOS . . . . . 3-2
3.5 Workflow . . . . . 3-4

**4 System Implementation**

4.1 Resources . . . . . 4-1
4.2 AC-link Application . . . . . 4-1
4.3 ESAI Usage . . . . . 4-3
4.4 DMA Usage . . . . . 4-3
4.4.1 Transmission . . . . . 4-3
4.4.2 Reception . . . . . 4-4
4.5 Timer Pulse Modulation . . . . . 4-4
4.5.1 Cold Reset . . . . . 4-4
4.5.2 Warm Reset . . . . . 4-5
4.5.3 ATE Test Mode . . . . . 4-5
4.5.4 Vendor-Specific Test Mode . . . . . 4-6
4.6 Interrupts . . . . . 4-6
4.6.1 Receiver Last Slot Interrupt . . . . . 4-6
4.6.2 Timer Compare Interrupt . . . . . 4-7
4.6.3 DMA Channel 0 Interrupt . . . . . 4-7
4.7 Assumptions And Recommendations . . . . . 4-8

# FIGURES


Figure 2-1 ESAI - AC '97 Connection: The AC_Link . . . . . 2-2

Figure 3-1 Audio Application-CODEC Interaction Model . . . . . 3-3

Figure 3-1 AC-link Application Data Flow . . . . . 3-4

Figure 4-1 AC-Link Implementation . . . . . 4-2

Figure 4-1 AC '97 Cold Reset. . . . . 4-5

Figure 4-1 AC '97 Warm Reset . . . . . 4-5

Figure 4-1 AC '97 ATE Test Mode. . . . . 4-6

Figure 4-1 Buffer Swapping . . . . . 4-7

Figure 5-1 Transmit Clock Control Register (TCCR). . . . . 5-1

Figure 5-2 Transmit Control Register (TCR) . . . . . 5-2

Figure 5-3 Receive Clock Control Register (RCCR) . . . . . 5-3

Figure 5-4 Receive Control Register (RCR) . . . . . 5-4

Figure 5-5 Common Control Register (SAICR) . . . . . 5-5

# TABLES


| | | |
|---|---|---|
| Table 2-1 | AC-link Signals | 2-1 |
| Table 3-1 | AC-link Application Workflow | 3-5 |
| Table 5-1 | PCR, PRR and PDR Values | 5-6 |
| Table A-1 | Data X-Memory Usage | A-1 |
| Table A-2 | Data Y-Memory Usage. | A-2 |
| Table A-3 | FIFO Table, Initial Status | A-2 |

# 1 Introduction

The Enhanced Serial Audio Interface (ESAI) of the DSP56300 Family provides full capabilities for interfacing with a general AC ‘97 CODEC through an AC ‘97 Digital Serial Interface, the AC-link.

The DSP56362 is the first DSP56300 Family derivative that presents the ESAI and is readily enabled to run this application. **Appendix B** lists reference materials pertaining to the DSP56300 Family, the DSP56362, and the *Audio CODEC ‘97 Component Specification*.

## 1.1 Scope

This application report describes how to implement an AUDIO CODEC ‘97 Digital Serial Interface (AC-link) using the ESAI. It is recommended for developers who have previous knowledge of Motorola’s DSP56300 Family of Digital Signal Processors and of the AC ‘97 Component Specification.

**Section 2** describes the physical connection between ESAI and an AC ‘97 CODEC. **Section 3** describes the Data-Flow model. **Section 4** delineates the system concept implemented in the AC-link application. **Section 5** details the configuration of DSP56300 Family resources used in this application. **Appendix A** shows the assembly code and equates of the application. **Appendix B** lists relevant reference information.

## 1.2 AC ‘97 Digital Controller

The AC ‘97 Digital Controller runs one or more audio applications and exchanges data with its analog counterpart, the AC ‘97 CODEC, through the AC-link. The DSP56300 Family may implement the AC ‘97 Digital Controller functionality.

# 2 Physical Connection

This section describes the physical connection between the Enhanced Serial Audio Interface (ESAI) and a generic AC ‘97-compatible CODEC. The AC ‘97 DSI or AC-link implementation suggested in this application report uses seven ESAI pins, providing the whole AC-link functionality. This section describes this glueless connection.

## 2.1 AC-link Lines

Five lines make up the AC-link. On the DSP side, all of the lines are connected to ESAI pins. In some cases, these lines are programmed as GPIO pins in order to accomplish a particular signal’s characteristic. **Table 2-1** briefly describes their function and direction related to the ESAI.

**Table 2-1** AC-link Signals

| Signal | Description | Direction |
|---|---|---|
| BIT_CLK | Serial bit-clock @12.288MHz | input |
| SYNC | Frame Sync signal @48kHz; high for 16 BIT_CLK periods, low for 240 | output |
| SDATA_IN | Serial Data Stream | input |
| SDATA_OUT | Serial Data Stream | output |
| $\overline{\text{RESET}}$ | Asynchronous Hardware Reset | output |

The AC-link lines should be connected to the CODEC according to specification.

The Frame Sync signal (SYNC) is generated by the transmitter side of ESAI and is output through its P4 pin, configured as Frame Sync for Transmitter (FST). In order to comply with the timing constraints of the AC '97 and DSP56300 Family, the ESAI transmitter and receiver are programmed to work asynchronously. Therefore, Frame Sync must be fed for the receiver side as well. Thus, a feedback connection between FST, output configured, and the ESAI P1 pin (Frame Sync for Receiver - FSR), input configured, is needed. This pin can be reconfigured temporarily as GPIO, permitting the DSP to signal the AC '97 CODEC to enter the Vendor-Specific Test mode.

Similarly, the serial data clock provided by the AC '97 CODEC, (BIT_CLK), must be fed for both the transmitter and receiver portions of ESAI. The required connection links the BIT_CLK line through the ESAI P0 and P3 pins—Receiver Serial Clock (SCKR) and Transmitter Serial Clock (SCKT), respectively—are both configured as input.

The Serial Data Input line, SDATA_IN, is supplied by the AC '97 CODEC, driving the ESAI P6 Serial Transmit/Receive Data pin, SDO5/SDI0, and configured as the ESAI #0 receiver input.

# 3 Data Flow

This section describes the AC-link application data-flow model, which is an implementation with a five-slot-output and a six-slot-input. This modular concept enables a smooth upgrade for a larger number of data streams.

Two Time-Division-Multiplexed (TDM) one-wire data channels constitute the AC-link data path, one channel for each direction. Each channel can transfer up to 12 data streams, plus a leading validation stream.

## 3.1 Output Data Streams

The AC-link implementation defines three output data streams that demand five slots of the TDM output channel:

- **PCM Playback**: two-channel composite PCM stream, 2 slots
- **Control**: control register write port, 2 slots
- **Modem Line CODEC Output**: modem line CODEC DAC input stream, 1 slot

## 3.2 Input Data Streams

In the input direction, the AC-link implementation defines four data streams that demand six slots of the TDM input channel:

- **PCM Record**: two-channel composite PCM stream, 2 slots
- **Status:** control register read port, 2 slots
- **Modem Line CODEC Input**: modem line CODEC ADC output stream, 1 slot
- **Dedicated Microphone Input**: dedicated microphone input stream, 1 slot

## 3.3 Data Flow Model

The AC ‘97 Digital Controller sees the AC-link data path as two sets of software implemented as First-In-First-Out queues (FIFOs). Each set reflects one direction of the link, being composed by one FIFO for every data stream, summing up three output FIFOs and four input FIFOs in this particular implementation.

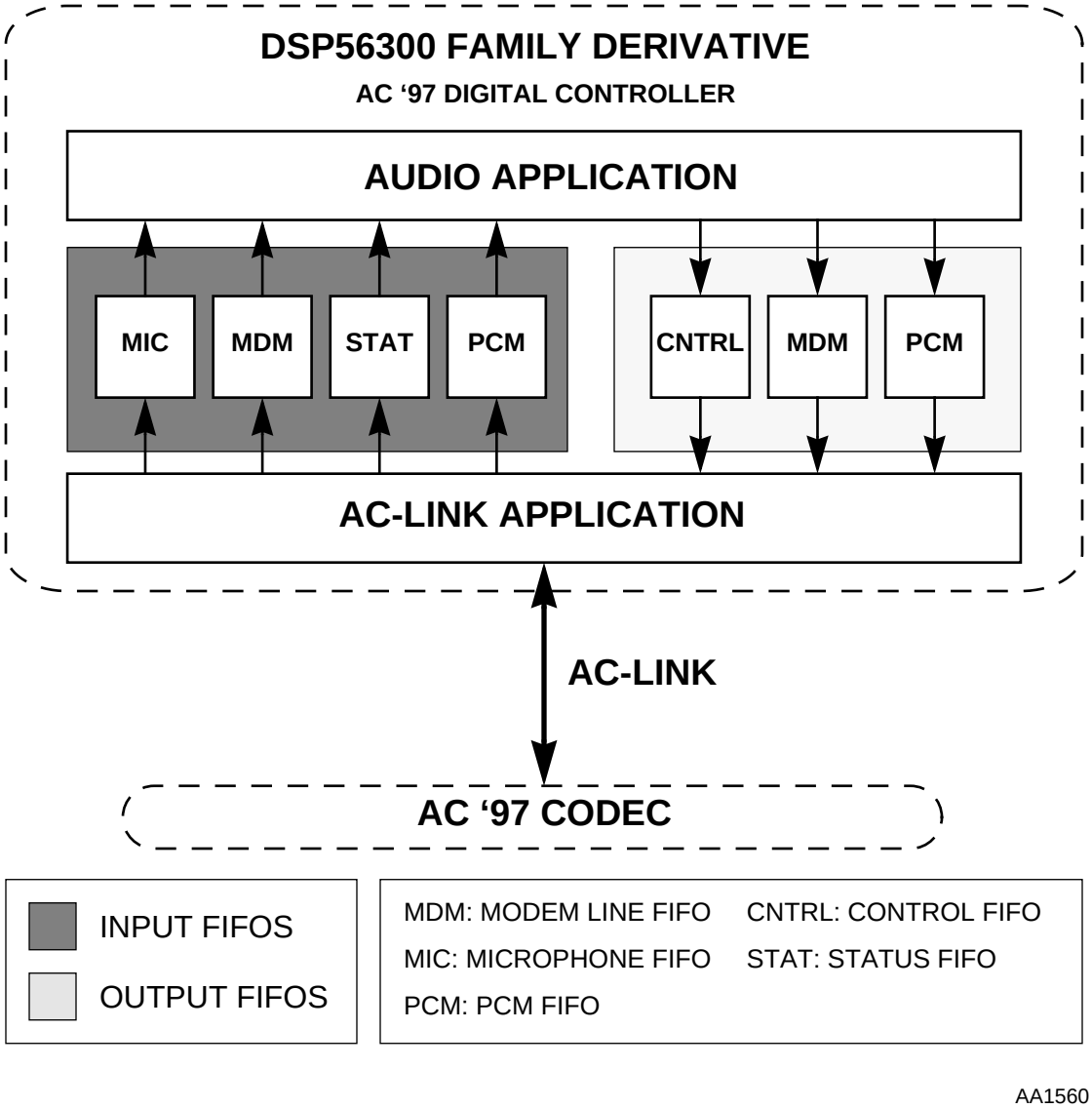


**Figure 3-1** Audio Application-CODEC Interaction Model

As **Figure 3-1** shows, during every AC '97 Audio Frame one of the output buffers is *active*, meaning its data is currently being transmitted. At the same time, the second buffer is not-active, and is fed by the AC-link application with the output FIFOs data. Slot validation—that is, slot 0 determination (Audio Frame TAG)—is accomplished concurrently. In the course of the subsequent audio frame, the buffers roles switch.

A symmetrical scheme is adopted in the reception section. The *active* buffer is filled with AC-link incoming data, while the data of the not-active buffer is transferred to input FIFOs. Reception buffers alternate active and not-active states in the same way as the transmitter buffers.

The first and second transmitted frames data after the AC-link reset ($F_0$, $F_1$) are entirely invalidated in order to enable the AC-link Application to enter the steady state. **Table 3-1** outlines the buffer swapping procedure.

**Table 3-1** AC-link Application Workflow

| | **Transmission Section** | | **Reception** | **Section** |
|---|---|---|---|---|
| **Frame** | **Active Buffer (Buffer -> ESAI)** | **Not (FIFO -> Buffer)** | **Active Buffer (ESAI -> Buffer)** | **Not Active Buffer (Buffer -> FIFO)** |
| $F_0$ | B1(-1) (whole buffer invalidated) | - (no RLS int. occurred) | B1(0) | - (no RLS int. occurred) |
| $F_1$ | B0 (whole buffer invalidated) | B1(1) | B0(1) | B1(0) |
| $F_2$ | B1(1) (first valid buffer transmitted) | B0(2) | B1(2) | B0(1) |
| ... | ... | ... | ... | ... |
| $F_{t-2}$ | B1(t-3) | B0(t-2) | B1(t-2) | B0(t-3) |
| $F_{t-1}$ | B0(t-2) | B1(t-1) | B0(t-1) | B1(t-2) |
| $F_t$ | B1(t-1) | B0(t) | B1(t) | B0(t-1) |
| $F_{t+1}$ | B0(t) | B1(t+1) | B0(t+1) | B1(t) |
| $F_{t+2}$ | B1(t+1) | B0(t+2) | B1(t+2) | B0(t+1) |
| ... | ... | ... | ... | ... |

# 4 System Implementation

This section describes the system concept implemented in this application. The AC-link application is implemented with a set of DSP56300 derivative routines that supports ESAI in its AC ‘97 mode, handling several on-chip resources.

## 4.1 Resources

The implemented application uses the following resources of the DSP56300 derivative:

- 1 ESAI transmitter
- 1 ESAI receiver
- 2 Direct Memory Access (DMA) channels
- 1 timer module
- Variable quantity of internal data memory for buffers and FIFOs

## 4.2 AC-link Application

The AC-link application implements an AC ‘97 Digital Interface using ESAI on its AC ‘97 mode and is supported by two DMA channels and a timer module. **Figure 4-1** presents the application block diagram.

The ESAI performs AC-link Audio Frame multiplexing and demultiplexing, as well as timing and control functions. DMA channels transfer data between the ESAI and the audio application running on the DSP56300 derivative. A timer module is necessary in AC-link special signaling.

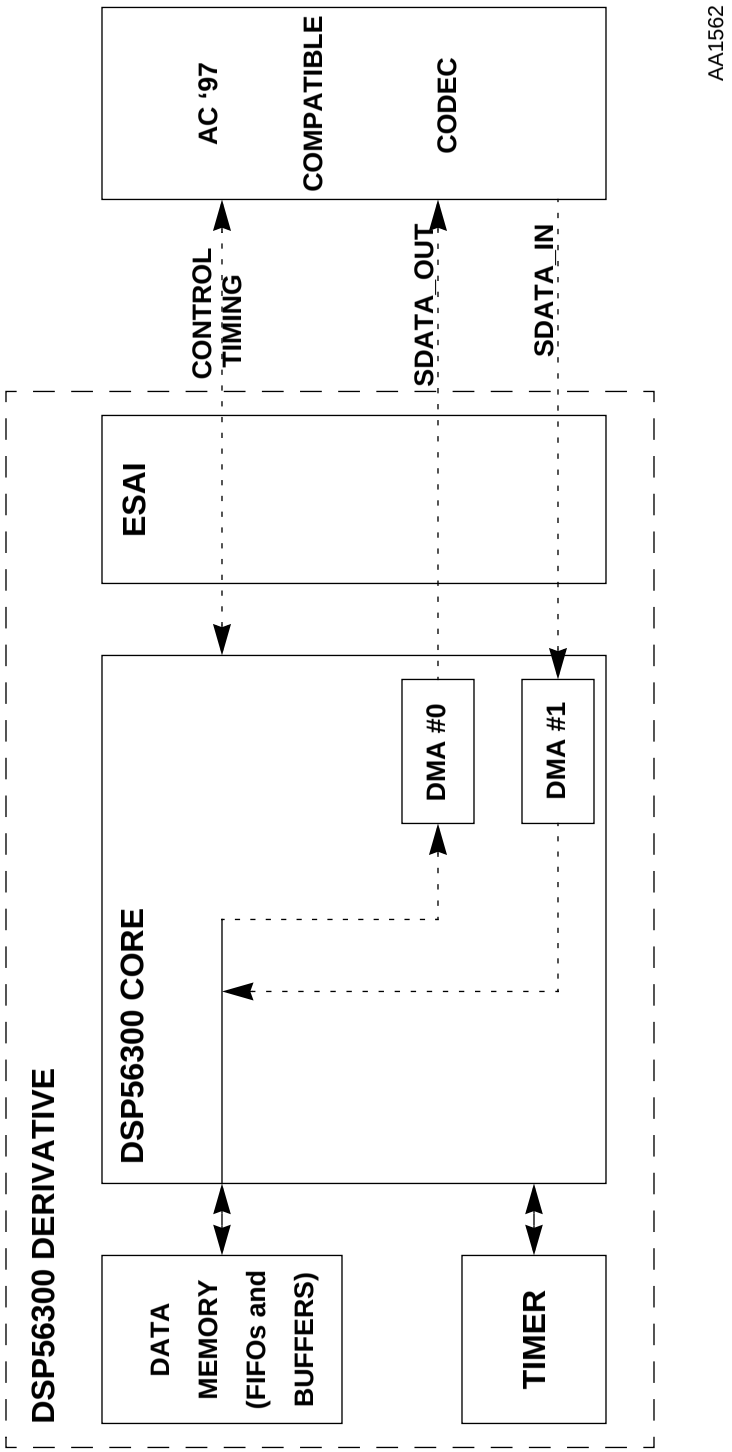


**Figure 4-1** AC-link Implementation

## 4.3 ESAI Usage

The ESAI is programmed to transmit with TX #0 and receive with RX #0 in asynchronous mode:

- with an external clock
- under AC '97 mode
- with 12 × 20-bit slots per frame plus a leading 16-bit slot (slot 0, Tag Phase).

The external clock for both the receiver and transmitter is the BIT_CLK signal generated by the AC '97 CODEC. The transmitter section generates the frame synchronization signal (FST/SYNC) supplied to the CODEC and to the receiver section. FST is driven with the falling edge of BIT_CLK, one bit before the beginning of the first slot in the frame.

In AC '97 mode, frame sync is generated high for the first word, a 16-bit word corresponding to the AC '97 tag phase, and low for 12 words, each being one of the 20-bit slots of the AC '97 data phase.

Data is driven out (SDO0/SDATA_OUT) at the falling edge of BIT_CLK, being sampled by the CODEC on the *next* falling edge. Input data (SDO5/SDI0/ SDATA_IN) is driven by the CODEC with the rising edge of BIT_CLK and sampled by the receiver on the subsequent falling edge.

## 4.4 DMA Usage

The DMA performs data transfer between the Audio application and ESAI using one channel for each direction. DMA channel #0 transfers output data, and channel #1 transfers input data.

### 4.4.1 Transmission

The ESAI specification requires that the first transmitted word be written to the ESAI transmitter register by a DSP Core move *before* transmitter enabling. Accordingly, the first transmitted audio frame requires a DMA configuration in which only the data phase is transferred from the active output buffer to the ESAI. The tag phase (slot 0) is programmed as first data for the ESAI transmitter.

From the second frame on, DMA is re-programmed to work in a continuous mode, servicing the ESAI transmitter upon request and transferring both tag and data

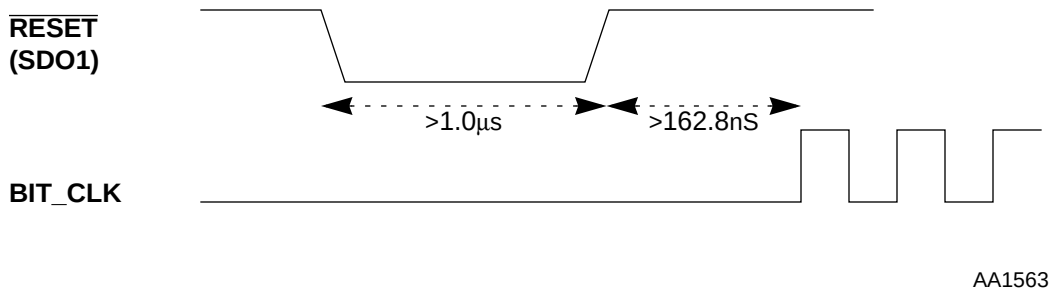


**Figure 4-1** AC '97 Cold Reset

### 4.5.2 Warm Reset

A similar pulse generation mechanism, supported by the timer module, is used for Warm Reset. This time the timer module is programmed to count to a typical 1.3 µs. Re-programming the ESAI P4 pin (FST/SYNC) as GPIO output allows a high pulse to be produced on this pin. The pin is driven high with timer enabling. Upon a timer Compare Interrupt that corresponds with the end of counting, the pin is brought back low.

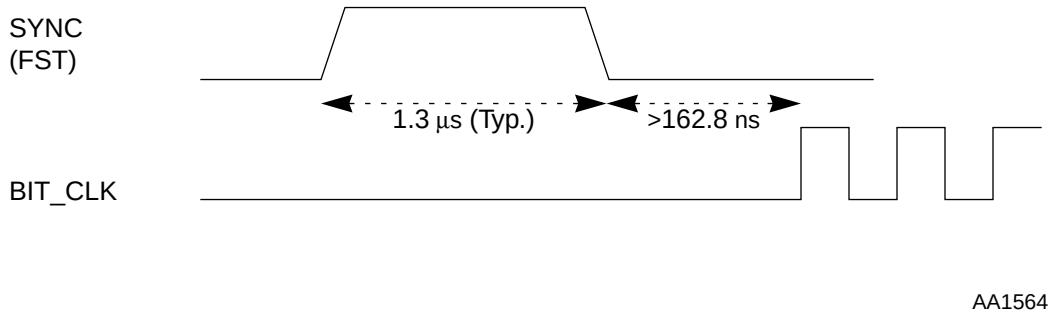


**Figure 4-1** AC '97 Warm Reset

### 4.5.3 ATE Test Mode

ATE Test Mode signaling uses the ESAI P11 pin (SDO0/SDATA_OUT). SDATA_OUT must be sampled high at the trailing edge of the $\overline{\text{RESET}}$ signal. This signaling is achieved by configuring the ESAI P11 pin as GPIO output and bringing it high with timer enabling for reset pulse counting. Upon a Timer Compare Interrupt and subsequent deassertion of $\overline{\text{RESET}}$, the pin is kept high.

corresponding valid-bits of the OB0 TAG (slot 0). The new OB0 buffer data is transmitted next time OB0 is active, i.e. during $F_{t+1}$. Similarly, input FIFOs are written with data from valid slots received during $F_{t-1}$ and allocated at IB0.

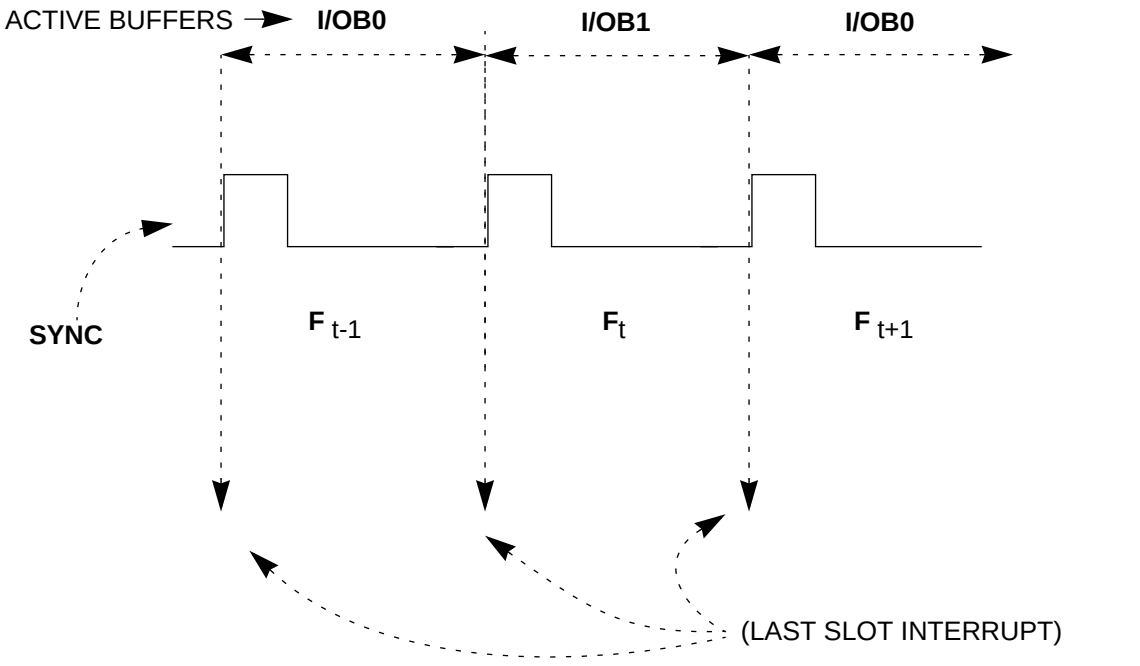


**Figure 4-1** Buffer Swapping

### 4.6.2 Timer Compare Interrupt

A Timer Compare interrupt occurs when the timer counter matches the compare register and permits the completion of the AC '97 reset and test signaling.

### 4.6.3 DMA Channel 0 Interrupt

Because of the special transmission at the first frame, the DMA Channel 0 interrupt is needed to enable DMA re-programming to work in a continuous mode. This re-programming includes disabling this interrupt.

Order by this number: APR37/D

### *How to Reach Us:*

**Home Page:**
www.freescale.com

**E-mail:**
support@freescale.com

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

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

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

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

***For Literature Requests Only:***
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com