임베디드 하드웨어 설계 – Final project

Audio Player & Recorder

2019. 12. 20. 2014231183 최 찬 호.

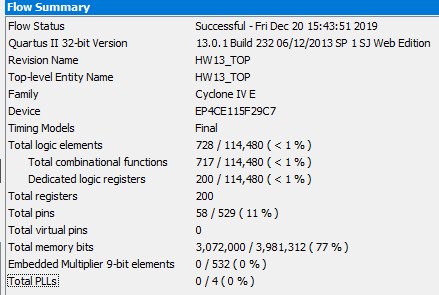
2019\_fall 임베디드 하드웨어 설계 과목 기말 과제로 Altera-DE115 board를 이용한 Audio Player 및 Recorder를 제작하였다. Quartus 2 13.0sp1 프로그램과 Verilog HDL 코드를 이용하여 구현되었다.

오디오 입출력은 Altera-DE115 board에 내장된 WM8731 Audio Codec과 통신하여 구현된다. WM8731과의 과의 통신을 위해서 [ I2C\_SDK, I2C\_SCLK, AUD\_ADCDAT, AUD\_ADCLRCK, AUD\_BCLK, AUD\_DACDAT, AUD\_DACLRCK, AUD\_XCK ] PIN을 사용하였다. WM8731은 Slave-Mode로 동작하게 설계하였으며, 통신에 필요한 신호들을 DE2-115보드에서 직접 생성하여 제공하였다. 모든 수신호 생성/확인 및 데이터 입출력은 Modulo-N Counter를 이용한 State Machine을 통해서 제어된다. 모든 Audio Player 관련 Module은 CLOCK\_50에 동기화된 동작을 실행한다.

실제 동작 시에는 Audio Player Interface Module을 이용하여 오디오 입출력을 제어할 수 있다. 총 8가지의 Control/Manipulate 로 구분된 동작을 실행한다. Control 동작은 오디오 입출력 장치 초기화 또는 초기값 설정 등의 기능을 수행한다. Manipulate 동작은 오디오 재생 도중 Stop/Play, Restart, Volume Up/Down 과 같은 재생 관련 기능을 수행한다.

최종적으로 24초 분량의 8khz sampling 16bit mono PCM 형식의 오디오파일을 저장 및 재생하는 장치를 구현하였다. 8,000 \* 16 \* 24 = 3,072,000 (bit)의 메모리 저장공간을 확보하기 위하여 Altera-DE115의 메모리 소자를 이용하였다. (3,981,312 bit의 메모리 소자 중 약 77%)

오디오 장치 확인을 위해서 각각 14초, 8초 분량의 음악파일을 준비하여 초기화하였다. 각각의 음악파일이 저장된 공간은 구분되어 있으며, Audio Player Interface에 의해서 선택된다. Recorder 기능을 사용한다면 음원을 1번 또는 2번 음악파일이 저장된 메모리 공간에 덮어쓴다.

 [그림 1] 빌드 성공 화면

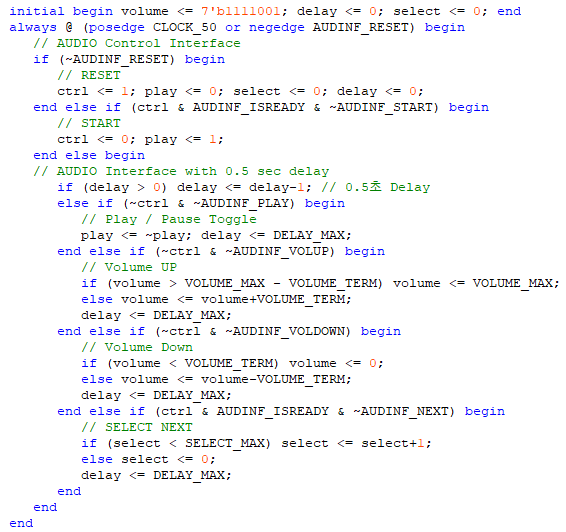
Contents

1. **Audio Player Interface**
   1. Audio Player Interface 기본 동작
      1. Control Mode
      2. Manipulate Mode
      3. Output
   2. Sub Module
      1. AUD\_init
      2. AUD\_inout
      3. Mem (MegaWizard Plug-in: RAM 1port)
2. **Communication with WM8731**
   1. I2C(Inter Integrated Circuit) 통신
      1. I2C Interface
      2. AUD\_init Module
   2. I2S(Integrated Interchip Sound) 통신
      1. AUD\_inout Module
3. **Demo Audio Source**
4. **Top Module With CPU**
5. **Reference**
6. **Audio Player Interface**
   1. Audio Player Interface 기본 동작

Audio Player Interface는 AUD\_main Module에 구현되어 있다. 오디오 입출력 동작을 구현하는 모든 Module을 Sub Module로 가지고 제어한다.

총 8가지의 Control/Manipulate 로 구분된 동작을 실행한다. ctrl값은 현재 상태가 Control 동작을 실행하는지, Manipulate 동작을 실행하는지에 대한 정보를 가진다. (Control Mode = 0, Manipulate Mode = 1)만약 현재 상태에 맞지 않는 동작 신호가 주어질 경우, 해당 동작 신호를 무시한다.

각 동작을 실행시키는 input port가 0을 의미하는 low value를 가질 때 해당 동작을 실행시킨다. 비동기 Reset을 제외한 모든 동작은 50Mhz Clock에 동기화되어 있다. Play/Stop, Volume Up/Down, Select와 같은 동작은 0.5초의 공유된 동작 후 대기시간을 가진다. (Altera DE115 board의 key 1회 입력을 쉽게 하기위한 용도이다.)



[코드 1] AUD\_main 동작 구현

* + 1. Control Mode
* Initialization

Initialization 동작은 always 문을 사용하지 않고, 직접 AUD\_init Module에 입력을 전하는 방식으로 구현된다. Control Mode 일 때, AUDINF\_INIT 값이 0이 된다면, AUD\_init Module에 Start (Enable) 신호가 전달된다. 곧, AUD\_init Module에 의해서 WM8731 설정을 초기화 시킨다. 구체적인 초기화 내용은 2.1.2에 기술되어 있다.

……

[코드 1-1] Initialization 동작

* Start

Start 동작은 Audio 재생 및 녹음을 시작하면서, 현재 상태를 Manipulate Mode로 전환한다. Start 동작 이후에는 Manipulate Mode이므로 Reset 동작 이전에는 Control 동작을 실행시키지 못한다. (Start 동작 또한 Control 동작이므로 실행 불가능하다.)

Start 동작이 올바르게 주어진 경우에도, 현재 초기화 작업이 끝나지 않은 상태라면 해당 동작을 실행하지 않는다. 곧, Initialization 동작 이후에 AUD\_init Module이 Done 신호를 전송하기 전까지 Start 동작은 무시된다.

* Reset

negative edge에서 비동기 Reset을 실행한다. AUD\_main Module뿐 아니라, 다른 모든 Sub Module 또한 Reset 신호를 입력하여 초기화 시킨다.

* Select

메모리 공간 또는 R/W (음원 재생 또는 녹음)을 설정하는 동작이다. 동작을 실행시킬 때 마다 Select 상태 값을 1 더한다. Select 상태 값은 2bit (0 ~ 3) 정보를 가진다. LSB는 메모리 공간 선택을 의미하며, MSB는 R/W 선택을 의미한다. R/W 동작 변환 후에는 Initialization동작을 통해서 다시 설정을 초기화해야 오디오 입출력이 정상적으로 동작한다.

(LSB = 0: Address1, LSB = 1: Address2. MSB = 0: Read, MSB = 1: Write)



[코드 1-2] Select 메모리 주소 설정

* + 1. Manipulate Mode
* Play / Stop

Play/Stop 동작은 현재 재생중인 음악을 정지 또는 재생시킨다. Input port의 값이 low value일 경우, always문에서 play값을 Toggle 시킨다. (play = 1: 재생, play = 0: 정지) play 값은 AUD\_inout Module의 Start(Enable) 신호로 직접 연결되어 해당 동작을 구현한다.



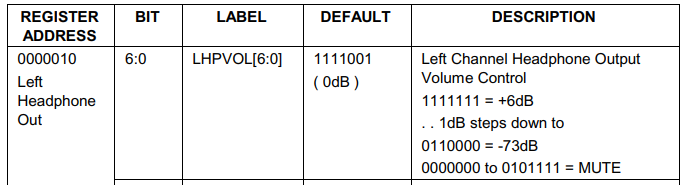
[코드 1-3] Play/Stop 동작 / Restart 동작

* Restart

Restart 동작은 현재 재생중인 음악을 재시작 한다. AUD\_inout에 Reset 신호를 입력하여 다시 Address를 Load 하도록 하는 방식으로 동작을 구현한다. ([코드 1-3] 참조)

* Volume Up / Down

Volume Up/Down 동작은 volume 7bit 값을 30만큼 가산하거나 감산한다. volume = 127일 때 최대 음량이며, volume값은 0의 최소값을 가진다. WM8731 Data Sheet에 의하면 최대 6dB, 최소 -73dB의 음량 범위를 가진다. volume값 변경 이후에는 AUD\_init Module을 통해서 WM8731을 해당 음량 설정으로 다시 초기화하는 방법으로 구현한다.



[그림 2] WM8731 Data Sheet\_ Volume 정보

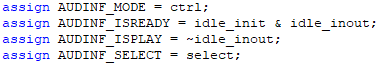


[코드 1-4] AUD\_init Volume 설정

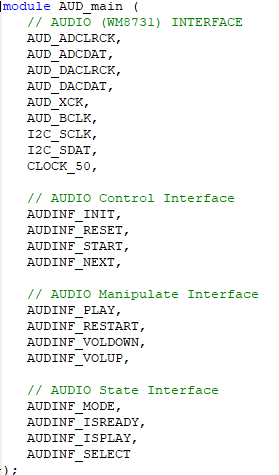
* + 1. Output

|  |  |
| --- | --- |
| AUDINF\_MODE | (=ctrl) 현재 동작 상태를 반환한다. |
| AUDINF\_ISREADY | AUD\_init Module의 현재 동작 여부를 반환한다.  Aud\_init Module이 Idle 상태일 때 1(high value)값을 가진다. |
| AUDINF\_ISPLAY | (=play) 현재 Play/Stop 상태를 반환한다. |
| AUDINF\_SELECT | (=select) 현재 선택된 메모리 공간 및 R/W정보를 반환한다. |

[표 1] Audio Player Interface Output



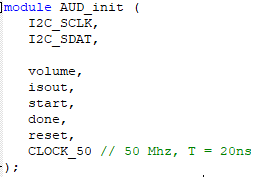
[코드 1-5] Audio Player Interface Output 연속할당문

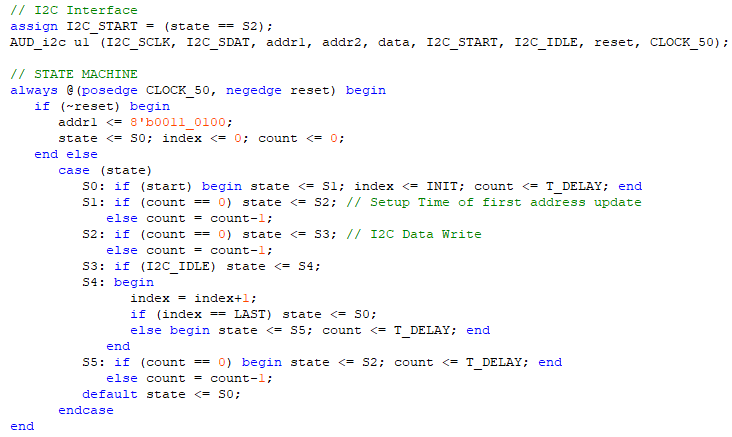
[코드 2] Audio Player Interface Module

* 1. Sub Module

mem Module을 제외한 모든 Audio Player Interface의 하위 Module은 Start(Enable), Done(Idle), Reset 신호를 가진다. Sub Module에 Start 신호가 입력되면, clock에 동기화되어 동작을 순차적으로 실행한다. 모든 동작이 완료된 후에는 Done 신호를 상위 Module에 반환한다. 비동기 Reset을 제외한 모든 동작은 50Mhz Clock에 동기화 되어있다.

* + 1. AUD\_Init

[코드 3] AUD\_init Module

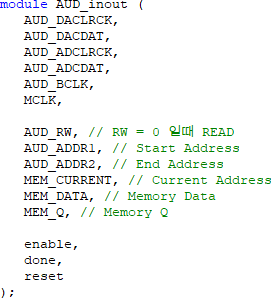


[코드 3-1] AUD\_init State Machine

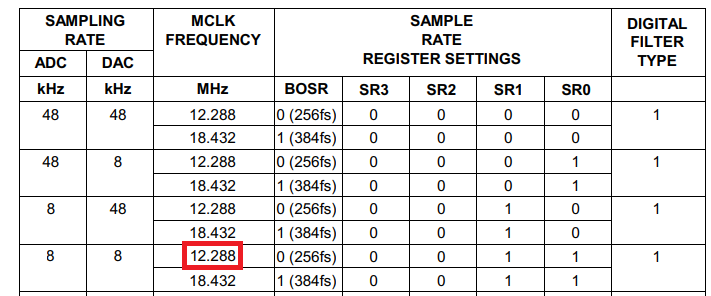
AUD\_init Module은 WM8731과의 I2C통신을 통해서 WM8731 초기 상태를 설정한다. Start 신호가 입력으로 주어지면, Command Table의 설정 값을 하나씩 적용한다. (Command Table은 index 값을 통해서 참조되며, 자세한 초기 설정은 2.1.2 참조)

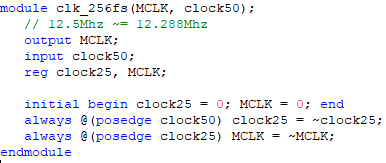
실질적인 I2C 통신은 AUD\_i2c Sub Module을 통해서 이루어진다. S2 상태에서 1 Clock 동안 AUD\_i2c Module에 Start 신호를 공급한 후, S3 상태에서 AUD\_i2c Module의 Done 신호를 대기한다. 모든 Command Table값을 참조한 후에는 Done 신호를 반환한다.

* + 1. AUD\_inout

[코드 4] AUD\_inout Module

AUD\_inout Module은 WM8731과의 I2S 통신을 통해서 오디오 재생 또는 녹음을 실행한다. RW = 0 일 때 Ready 및 오디오 재생 동작을 실행하며, RW = 1 일 때 Write 및 오디오 녹음 동작을 실행한다. WM8731 Data Sheet는 8kHz sampling Normal Mode 설정 시 12.288Mhz의 MCLK 공급을 요구한다. 50Mhz Clock과 동기화된 MCLK 공급을 위하여, 50Mhz Clock을 두 번 분주한 12.5Mhz (~= 12.288Mhz) MCLK를 공급한다. (I2S 통신은 2.2 참조)

[그림 3] WM8731 Data Sheet\_ MCLK 정보

[코드 4-1] MCLK 공급 Module

* + 1. mem Module

mem Module은 Quartus2 Mega Wizard Plug-in을 이용하여 구현된 1-Port RAM Module이다. 18bit Address를 이용하여 384KB (3,072,000 bit)의 저장공간 입출력을 제공한다. (주소당 저장 공간은 16 bit이다.) 50Mhz Clock에 동기화되어 동작한다. select의 MSB를 이용하여 R/W 동작을 확인한다.



[코드 4-2] mem Module

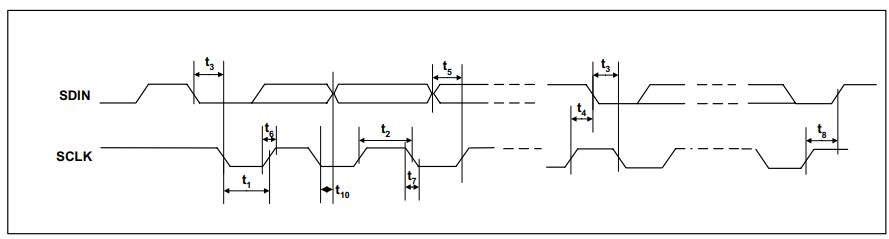
1. **Communication With WM8731**

Audio Player/Recorder 구현을 위한 WM8731과의 통신은 다음과 같은 두 가지로 구분된다.

1. I2C 통신: WM8731 기기의 입출력 설정을 지정한다.
2. I2S 통신: WM8731의 DAC또는 ADC를 이용하여 음원 재생/녹음을 실행한다.

I2C 통신은 AUD\_init Module, I2S 통신은 AUD\_inout Module에서 구현된다. 각각의 Module은 Modulo-N Counter로 제어되는 State Machine을 이용하여 신호를 전송 및 확인하거나 데이터 입출력을 제어한다.

* 1. I2C 통신



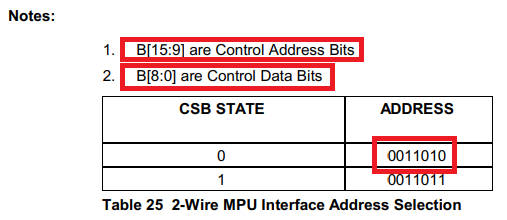
[그림 4] 2-wire Serial I2C 통신 (WM8731 Data Sheet 자료)

I2C 통신은 SDIN, SCLK를 이용하여 1bit 단위의 신호를 전송 및 수신한다. Altera DE-115 board는 WM8731과 수신하기위한 I2C\_SDK, I2C\_SCLK Pin을 제공한다.

1. SDIN = 1, SCLK = 1 일 때 IDLE 상태이다.
2. SCLK = 1, SDIN = 0 일 때 데이터 전송 신호가 시작한다.
3. SCLK의 low에서, AUD\_i2c Module은 SDIN에 데이터를 전송한다.
4. SCLK의 high에서, WM8731은 Data를 수신한다.
5. 8bit 전송 후, 수신호가 정상적이라면 WM8731은 ACK 신호를 반환한다.
6. SCLK = 1, SDIN = 1일 때 Stop 신호가 생성된다.

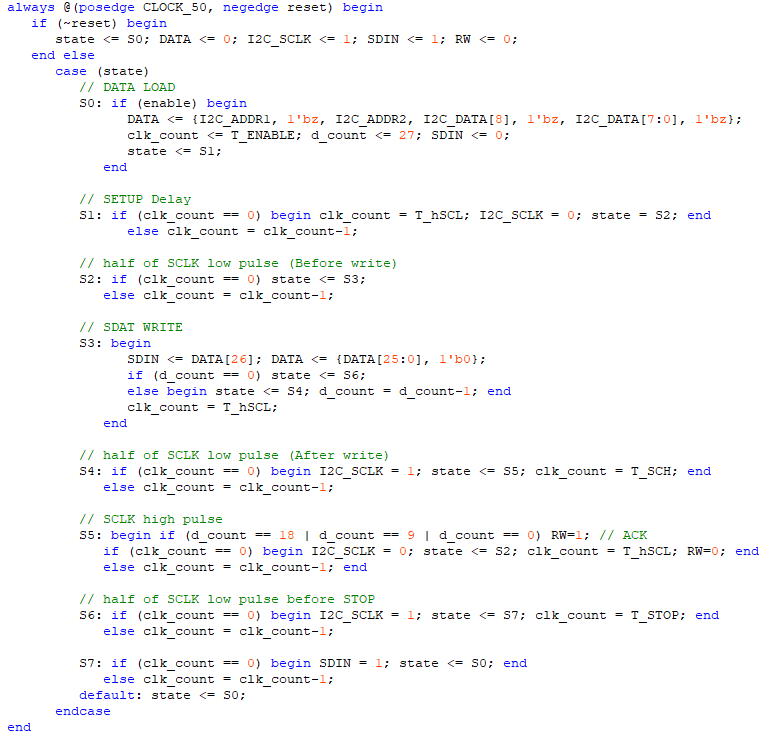
WM8731과의 I2C 통신은 7bit Slave Address + 0 (Write), 7bit Register Address, 9bit Data. 총 24bit Data 전송을 통해서 1개 WM8731 내부 Register를 설정할 수 있다.

* Slave Address는 0011010을 사용한다. (WM8731 Data Sheet 참고)

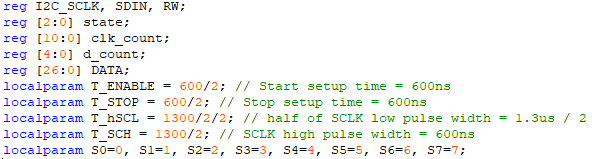
[그림 4-1] WM8731 Data Sheet\_ I2C 통신

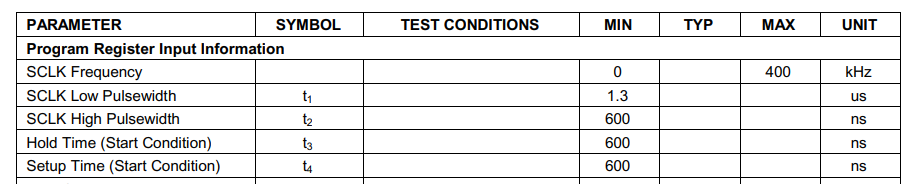
* + 1. I2C Interface (AUD\_i2c Module)

AUD\_i2c Module은 Slave Address (7bit), Register Address (7bit), Data (9bit)를 입력으로 받는다. Idle 상태에서 Start 신호를 입력 받으면, 위의 정보들을 Load 하는 동작을 실행한다. Start 신호 이후에는 순차적으로 I2C 통신을 실행한다. 통신 종료 후에는 Done 신호를 전달한다.



[코드 5] AUD\_i2c State Machine

[코드 5-1] AUD\_i2c Delay (10배)



[그림 4-2] WM8731 Data Sheet\_ Timing ([그림 4]의 Timing)

* 실제 동작 시 문제가 없도록 Data Sheet의 10배 시간을 적용하였다.
* SCLK의 duty cycle이 50%가 되도록 1.3us의 High Pulse With (여유를 위해 13us)를 가지게 하였다.

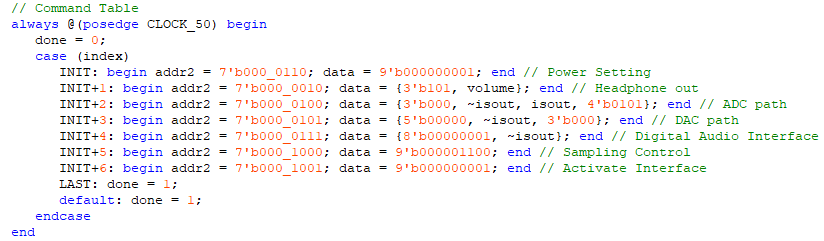
1. S0: Enable 신호가 전달되면 Load 작업을 실행한다. 총 24bit의 Data를 전달하지만, 8bit마다 1번씩 ACK 신호를 받으므로, 3bit의 high-impedance 값을 포함한 27bit Data를 Load한다. SDIN = 0으로 설정하여 I2C Start 신호를 생성한다.
2. S1: T\_Enable값 만큼 Setup Delay를 기다린다. Delay 이후에는 SCLK = 0으로 설정하여 데이터 입력을 준비한다.
3. S3~S5: SCLK low의 50%지점에서 DATA 입력이 실행된다. DATA 입력 후에는 Left Shift를 통해서 다음 DATA를 설정한다. d\_count 값을 이용하여 남은 데이터의 개수를 확인한다. 모든 데이터를 전송한 이후에는 S6 State로 탈출한다.
4. S5: ACK 신호를 받는다. RW를 이용한 tri-state buffer로 I2C ACK 신호를 받을 수 있다. 수신한 I2C ACK 신호를 별도로 처리하지는 않는다. ([코드 5-2] 참조)
5. S6: 남은 SCLK low의 절반을 진행한다. S3 State로 돌아가지 않으며 끝나기 직전의 상태라는 점에서 S4 State와 다르다.
6. S7: SCLK = 1일 때 SDIN = 1으로, I2C Stop 신호를 생성한다. S0 State로 돌아간다면 Done(Idle) 신호를 상위 Module로 전송한다. Idle 상태 이후, 곧바로 Start 신호를 받을 수 없도록 연속 할당문에 지연 시간을 적용하였다. ([코드 5-2] 참조)



[코드 5-2] Idle 지연 시간

* + 1. AUD\_init Module

AUD\_init Module은 AUD\_i2c Module을 이용하여 WM8731 Register를 하나씩 초기화 시킨다. 초기화 시키는 Register Address와 Data는 always문을 이용한 Command Table에 저장되어 있다. Index 값을 통해서 Command Table을 참조하며, 모든 데이터 참조 이후에는 Done 신호를 반환한다.



[코드 6] AUD\_init Command Table

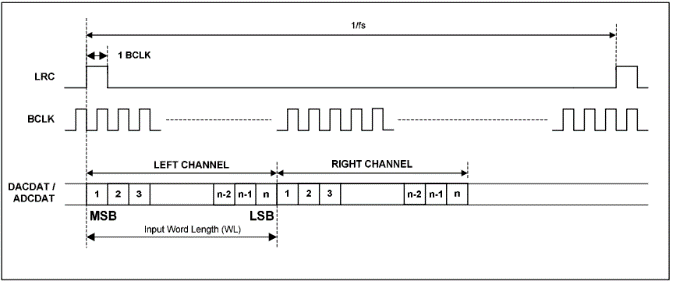
Register Address 및 Data 동작은 WM8731 Data Sheet를 참조하였다. 세부 설정은 다음과 같다.

1. 전원 설정: LINE Input Power Down
2. Headphone 설정: LPHPBOTH = 1 (Headphone 두 출력이 동일하도록 설정)
3. ADC 장치 설정: Microphone Input Level Boost 사용, Microphone Input 사용
4. DAC 장치 설정: 오디오 녹음 시 DAC Mute
5. Digital Audio Interface Format 설정: I2S Format (MSB-First left-1 justified),   
   16bit Input Audio Data Bit Length
6. Sampling Control 설정: Normal Mode, Base Over-Sampling Rate = 256fs,   
   8kHz DAC/ADC sampling rate

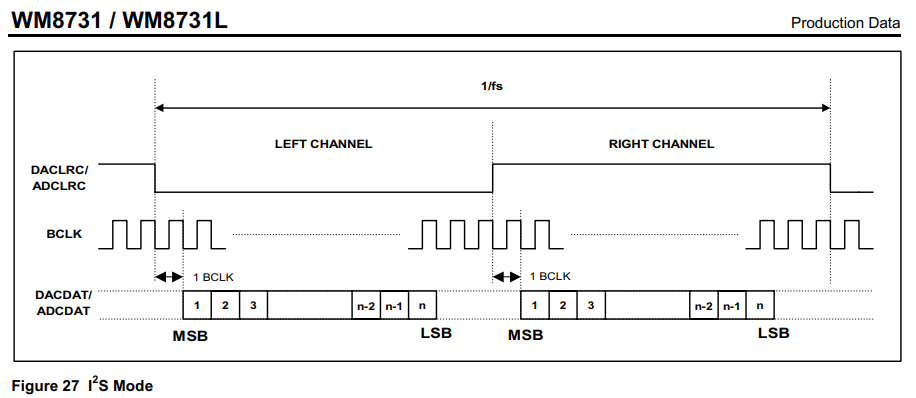
* Digital Audio Interface Format 설정 – 알 수 없는 성능 향상…

Digital Audio Interface Format Register의 [1:0] Data는 Audio Data Format을 설정한다.   
Data = 11일 때에는 I2S Format, Data = 10 일 때에는 DSP-PCM Format이다. 본 기기는 I2S Format 통신을 구현하였으므로, 오디오 출력의 경우 DSP-PCM Format 설정 시 정상적으로 통신 되지 않는다.

하지만 오디오 입력의 경우, 알 수 없는 현상이 발견되었다. I2S Format을 통한 Audio 녹음이 구현되었으나, 잡음이 발생하는 현상이 있었다. 잡음 발생의 원인이 WM8731 설정 값 문제라고 판단하여, 여러 설정 값을 시도하던 도중, 우연히 I2S Format을 DSP-PCM Format으로 바꾸었다. 결과로는 I2S Format에서는 발생하던 잡음이 사라진 정상적인 상태로 오디오 입력 (녹음)이 적용되었다. 곧, 이해할 수 없는 성능 향상을 이루었다…

[그림 5] DCP-PCM Mode…

* 1. I2S 통신



[그림 6] WM8731 Data Sheet\_ I2S 통신

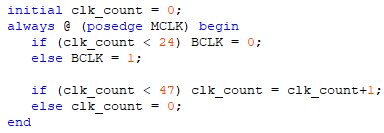
I2S 통신은 입력(녹음) 시 ADCLRC, ADCDAT를, 출력(재생) 시 DACLRC, DACDAT를 이용하여 1bit 단위의 신호를 전송 및 수신한다. Altera DE-115 board는 WM8731과 수신하기위한 Pin들을 제공한다. BCLK, ADCLRC, DACLRC와 같은 제어 신호는 WM8731이 Slave로 동작 할 때에는 WM8731의 Input이 된다. (Master 동작 시에는 WM8731이 생성하여 output으로 출력한다.) 본 기기는 WM8731가 Slave로 동작하므로, 직접 제어 신호들을 생성한다.

전송되는 bit의 개수는 Digital Audio Interface Format 설정을 통해서 정해진다.   
16/20/24/32 bit중 선택할 수 있으며, 본 기기는 16bit의 Audio Data bit를 선택하였다. DACLRC/ADCLRC가 Low일 때 신호가 시작하여, DACLRC/ADCLRC Low동안 Left Data를 전송한다. DACLRC/ADCLRC High 동안에는 Right Data를 전송한다. 본 기기는 mono 형식의 오디오 파일을 재생하므로, Left/Right로 전송되는 Data가 동일하다.

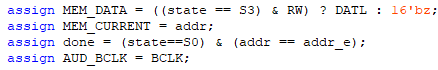
* + 1. AUD\_inout Module

AUD\_inout Module은 BCLK, DACLRC/ADCLRC와 같은 신호를 생성하고, BCLK 동작에 따라서 데이터 입출력을 제어한다. BCLK는 MCLK에 동기화되어 생성되는데, MCLK는 상위 Module의 50Mhz Clock을 두 번 분주한 12.5Mhz Clock으로 공급받는다.

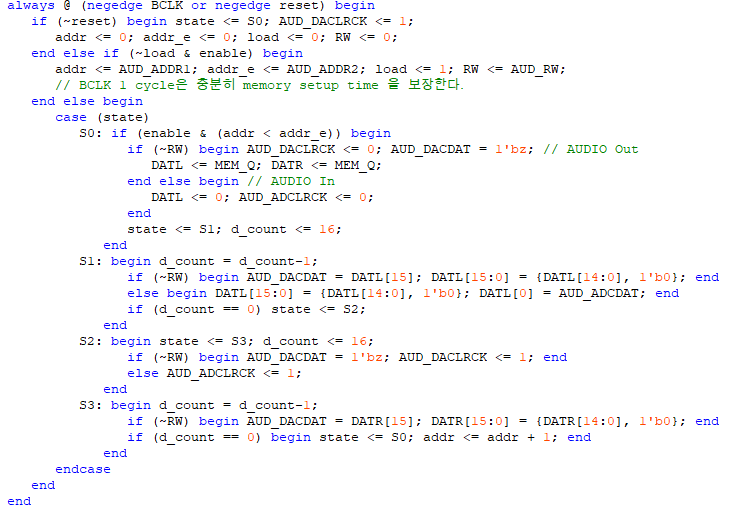
I2S 통신에 따라서 1회 데이터 전송동안 34 번의 BCLK cycle을 가진다. Sampling rate는 8Khz이므로, 1초 동안 8000 회 데이터가 전송된다.

[코드 7-1] BCLK 생성

* 48 (count) \* 32 (bit) \* 8000 (sampling rate) = 12,288,000 (~= 1 MCLK)
* 실제로는 48 \* 34 \* 8000 = 13,056,000 (~= 1.044 MCLK)   
  (Audio 입출력 동작에 문제는 없다.)

[코드 7-2] MEM 선택

* RW = 0일 때 Read (재생), RW = 1일 때 Write (녹음) 동작을 한다.
* tri-state buffer를 이용하여 메모리 입출력을 제어한다.
* ADDR은 RW 상태에 따라서 Read 또는 Write할 메모리 주소를 가리킨다.  
  ADDR output을 상위 Module에서 mem Module으로 전달하는 방식을 사용해 메모리 입출력을 구현한다.



[코드 7-3] AUD\_inout State Machine

1. State S0 ~ S3은 데이터 전송/수신 1 cycle을 의미한다. 해당 데이터 전송/수신은 초당 8000번 이루어진다.
2. 최초 Enable(Start)값이 주어지면 Load 동작을 통해, 메모리 시작 주소와 종료 주소를 적재한다. 데이터 전송/수신을 반복하고 메모리 종료 주소에 도달하여 모든 작업이 완료된 후에는 Done 신호를 반환한다.
3. Enable값이 주어지지 않으면, 데이터 전송/수신을 중단한다. 다시 Enable 값이 주어진 후에는 마지막 주소부터 다시 데이터 전송/수신을 재개한다.
4. mono 음원을 사용하므로, DATL = DATR이다. 녹음 시에는 DATL 만을 사용한다.
5. BCLK 1 cycle은 CLOCK\_50에 동기화되는 mem Module보다 충분히 느리므로, Setup 및 Hold time을 보장한다.
6. S0: Enable = 1 일 때, DATA를 초기화 (Load/Reset)하고   
   제어신호 (DACLRCK/ADCLRCK)를 0으로 설정한다.
7. S1~S2: 상태가 전환되는 1 BCLK cycle 이후, 데이터를 전송/수신한다. Shift Left 연산을 이용하여 데이터 위치를 갱신한다. (S1: Left Side, S2: Right Side)
8. S3: 메모리 주소를 추가하고 데이터 1회 전송/수신을 종료한다. (다음 상태 S0에서 메모리를 바로 사용하지만, BCLK 1 cycle은 Setup/Hold time을 보장한다.)
9. **Demo Audio Source**

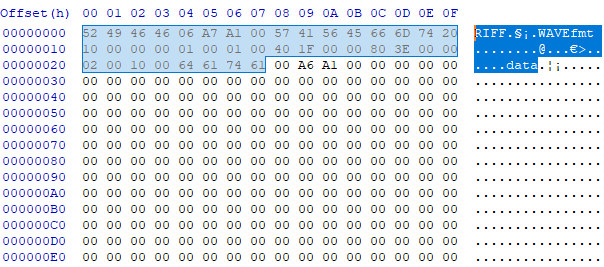
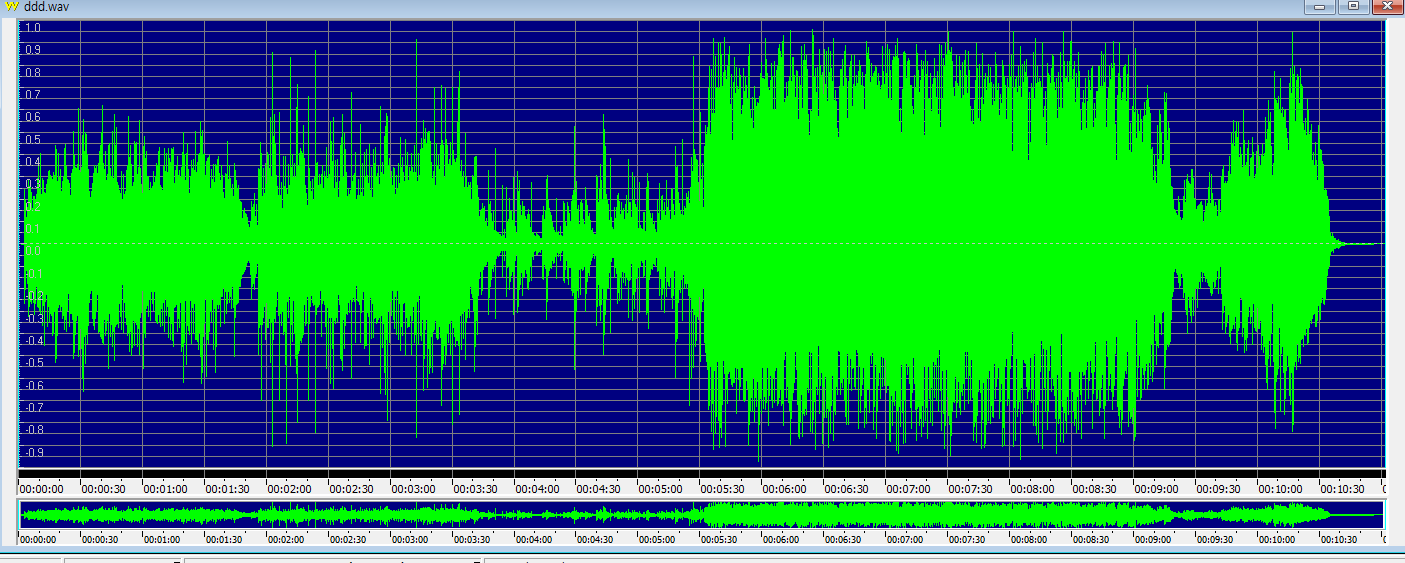
본 기기는 8Khz sampling rate 16bit mono PCM 형식의 Audio Source를 저장/재생한다. 음악 파일이 저장될 수 있는 공간은 384KB로, 해당 형식의 음악 파일을 24초 저장/재생 할 수 있다.

음악 파일이 저장되는 공간은 2개 공간으로 나누어져 있다. (18bit 주소, 주소당 16bit 저장)

1. 0h번지 ~ 1\_B580h 번지: 224KB, 14초 재생
2. 1\_B580h 번지 ~ 2\_EE00h 번지: 160KB, 8초 재생

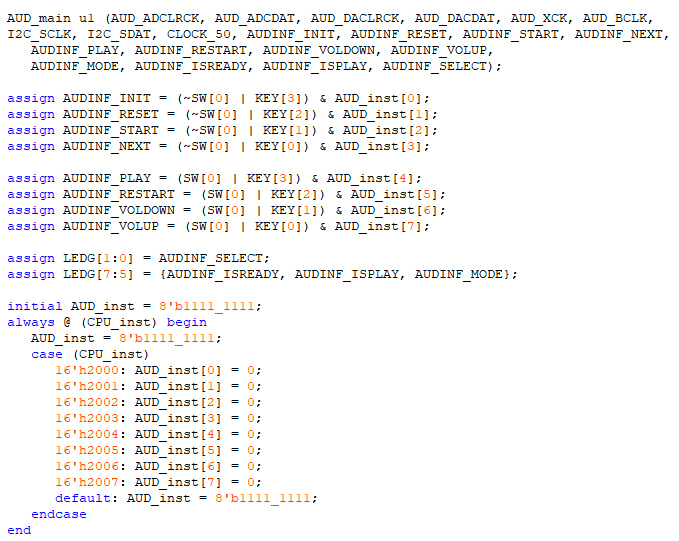
* Demo Audio Source 생성 과정

1. Mp3 또는 임의 형식의 음원 파일을 8Khz sampling rate 16bit mono PCM 형식의 wav 파일로 변환한다. (오디오 파일 편집 프로그램 사용 – GoldWave)
2. Hex Editor를 이용하여 Audio 파일 시작 및 종료 부분의 불필요한 데이터 제거
3. Matlab 프로그램을 이용하여 .mif 형식에 맞추어 저장. (외부 소스 변형하여 사용)  
   - uint16 (2바이트 부호없는 정수) 단위, little endian 형식으로 Audio 파일을 읽는다.
4. .mif 파일을 Mega Wizard Plug-In 1 port RAM Module 생성 시 초기값으로 입력.



[그림 7] 음원 편집 (Gold Wave-Hex Editor)

1. **Top Module With CPU**



[코드 8] Top Module (AUD\_main Part)

Audio Player/Recorder Interface는 Altera DE-115 board의 SW0, KEY0, 1, 2, 3 input과 4개의 AUD\_main output 정보를 표시하는 LEDG를 사용한다.

SW0 = 1일 때에는 Coontrol Mode 동작 신호를 생성하며, SW0 = 0일 때에는 Manipulate Mode 동작 신호를 생성한다.

Audio Player/Recorder Interface는 위의 Altera DE-115 board SW/KEY 입력 외에도, CPU의 명령어 호출로 동작 될 수 있다. CPU는 FPGA에 명령어에 해당하는 16bit 데이터를 전송할 수 있다. 전송된 명령어는 CPU\_inst값에 저장된다. CPU\_inst Event 발생 시, 명령어에 해당하는 CPU\_inst값을 해독하여 AUD\_inst에 저장한다. AUD\_inst는 연속할당문을 통해서 AUD\_main Input에 반영된다.

1. **Refrence**
2. <https://www.rockbox.org/wiki/pub/Main/DataSheets/WM8731_8731L.pdf>  
   (WM8731 Audio Codec Data Sheet)
3. <http://hamblen.ece.gatech.edu/DE2/>   
   <http://hamblen.ece.gatech.edu/DE2/DE2_demonstrations/> (DE2 I2C Demonstration)
4. <https://github.com/AntonZero/WM8731-Audio-codec-on-DE10Standard-FPGA-board/tree/master/project>  
   <https://www.youtube.com/watch?v=zzIi7ErWhAA>  
   Demo Audio Source 작성 방법 참조. Matlab Code: (MifConv)  
   VHDL기반 Audio Player 소스 참조 (USB-Mode 48ksps 오디오 파일 재생장치)
5. <http://csys.yonsei.ac.kr/lect/emhw/textlcd.pdf>  
   강의 자료: Text LCD Code  
   State Machine / Command Table 형식 참조