I2S TRANSMITTER

JUNE 2020

Doc. Name: Design Document - IIS TRANSMITTER

Contents

List of Figures

List of Tables

1

Theory, Principle of Working and Calculations

1.1 I2S protocol

Inter-IC sound (I2S) bus is a serial link especially for digital audio.The bus handles audio data only, while the other signals, such as sub-coding and control, are transferred separately. To minimize the number of pins required and to keep wiring simple, a 3/4-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line. In a system I2S master is the device which is generating the clock. The I2S transmitter mentioned here is an I2S master capable of generating the clock to an external audio DAC.

1.2 The I2S bus

The bus has three lines:

- master clock(MCLK)
- continuous serial clock (SCK)
- serial data (SD)
- word select (WS)

and the device generating SCK and WS is the master.

1.2.1 Master clock (MCLK)

The master clock is always an free running clock that is fed to the transmitter to generate the SCK and also to the receiver as the oversampling clock. Several I2S receivers are able to generate the MCLK internally thus supporting a 3 wire I2S interface. The I2S master clock is absolutely necessary for jitter free operations. The MCLK is avoided on the ground that it is a fractional clock based on the sampling frequency of the audio data. eg : for 48khz (fs) audio sample the MCLK can be 192fs or 384fs which is 9.216 Mhz or 18.432 Mhz. A clock value of 36.864 Mhz can support most of the sampling frequencies and oversampling requirements.

1.2.2 Serial clock (SCK)

The serial clock is also called the bit clock BCLK in the industry. It is the clock on which the audio data samples are shifted out. The data is shifted on the trailing edge of this clock.

1.2.3 Serial Data (SD)

The transmitter shifts out the audio data bit by bit on to the SD line on the trailing edge of the SCK. MSB is always transmitted first. The transmitter always sends the MSB of the next word one clock period after the WS changes. Such a format is called I2S justified format. Left channel data is shifted during the low period of WS and right channel data is shifted during the high period. Serial data is always latched on to the receiver on the leading edge of the serial clock signal, so the transmitter registers the data on SD and WS on the trailing edge of the SCK.

1.2.4 Word select (WS)

The word select signal is also called LRCLK in the industry since this is the signal to identify the left/right audio data on the bus. A low signal indicated left channel is being shifted out and a high signal indicates a right channel audio data on the bus. There is a delay of one SCK period bewteen the serial data and word select signal.

1.3 Advanced Peripheral bus protocol

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) protocol family. It defines a low-cost inter-

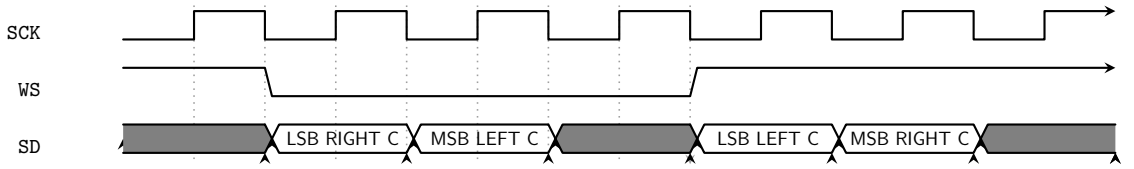

Figure 1.1: 3 wire I2S

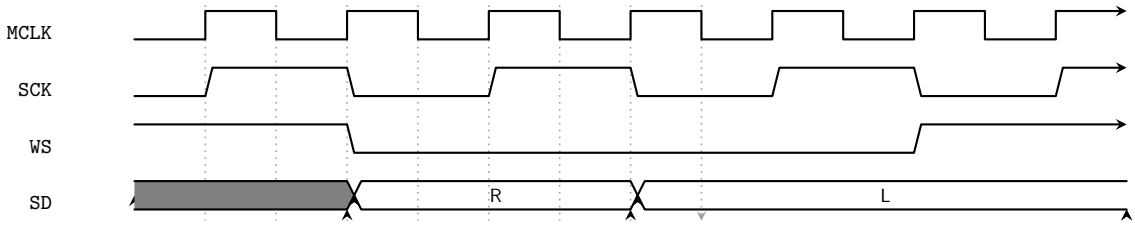

Figure 1.2: 4 wire I2S

face that is optimized for minimal power consumption and reduced interface complexity. The APB protocol is not pipelined. It is used connect to low-bandwidth peripherals that do not require the high performance of the AXI protocol. The APB protocol relates a signal transition to the rising edge of the clock, to simplify the integration of APB peripherals into any design flow. Every transfer takes at least two cycles.

1.3.1 AMBA APB signals

- PCLK - The rising edge of PCLK times all transfers on the APB.
- PRESETn -The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal.
- PADDR - This is the APB address bus. It can be up to 32 bits wide and is driven by the peripheral bus bridge unit.
- PPROT - This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
- PSELx - The APB bridge unit generates this signal to each peripheral bus slave. It indicates that the slave device is selected and that a data transfer is required. There is a PSELx signal for each slave.
- PENABLE (APB bridge Enable) This signal indicates the second and subsequent cycles of an APB transfer.

- PWRITE (APB bridge Direction) - This signal indicates an APB write access when HIGH and an APB read access when LOW.
- PWDATA (APB bridge Write data) - This bus is driven by the peripheral bus bridge unit during write cycles when PWRITE is HIGH. This bus can be up to 32 bits wide.
- PSTRB (APB bridge Write strobes) - This signal indicates which byte lanes to update during a write transfer. There is one write strobe for each eight bits of the write data bus.
- PREADY - Slave interface Ready. The slave uses this signal to extend an APB transfer.
- PRDATA - Slave interface Read Data. The selected slave drives this bus during read cycles.
- PSLVERR - Slave interface error, This signal indicates a transfer failure.

1.3.2 Write transfers

There are two types of write transfers:

- Write transfer without wait state
- Write transfer with wait states

A write transfer starts with address PADDR, write data PWDATA, write signal PWRITE, and select signal PSEL, being registered at the rising edge of PCLK. This is called the Setup phase of the write transfer. At the next clock cycle enable signal PENABLE, and ready signal PREADY, are registered at the rising edge of PCLK. When asserted, PENABLE indicates the start of the Access phase of the transfer. When asserted, PREADY indicates that the slave can complete the transfer at the next rising edge of PCLK. The address PADDR, write data PWDATA, and control signals all remain valid until the transfer completes. The enable signal PENABLE, is de-asserted at the end of the transfer. The select signal PSEL, is also de-asserted unless the transfer is to be followed immediately by another transfer to the same peripheral

The PREADY signal can be extended low if the slave device is not able to accept the write transfer, in such a case the PENABLE is extended. Such and transfer is called write transfer with wait states.

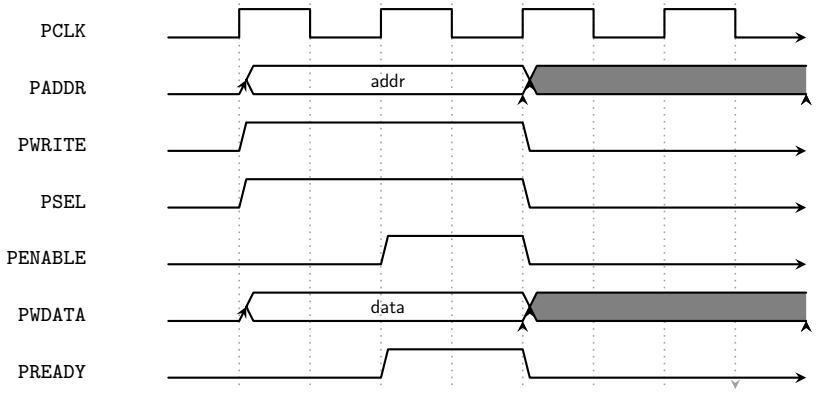

Figure 1.3: Write transfer with no wait states

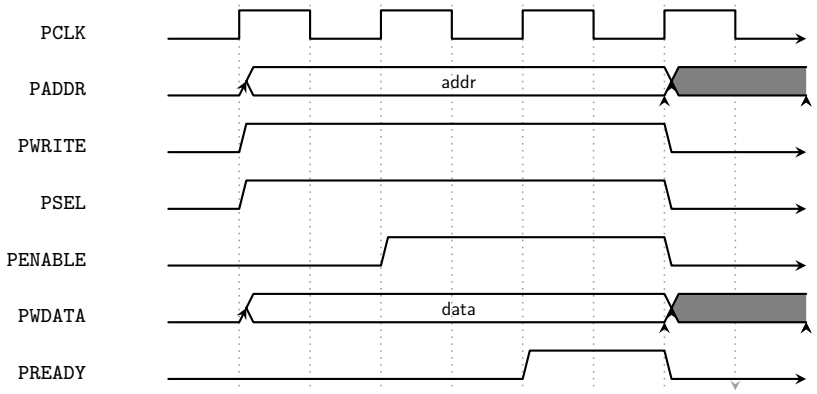

Figure 1.4: Write transfer with wait states

1.3.3 Read transfers

Two types of read transfer are described in this section:

- With no wait states
- With wait states.

The signals timings for read transfer with and without waits states is shown in the figure.

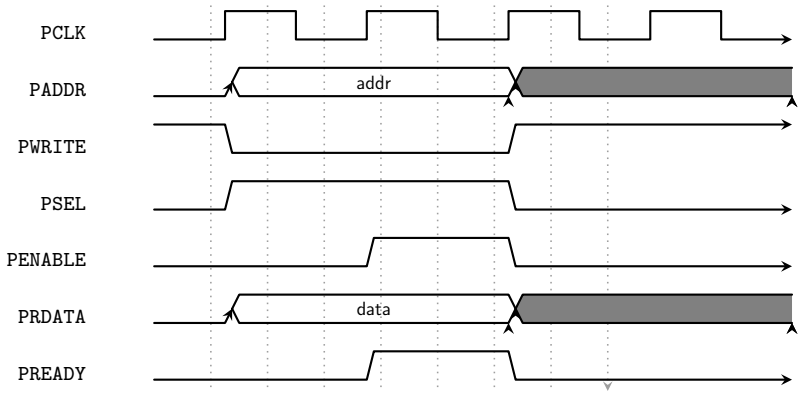

Figure 1.5: Read transfer with no wait states

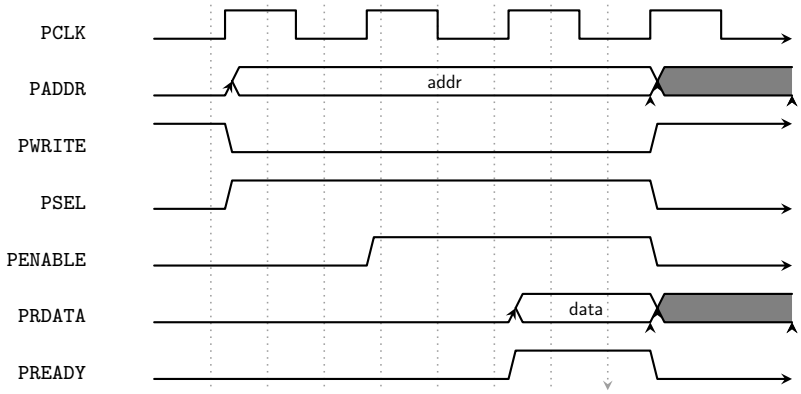

Figure 1.6: Read transfer with wait states

1.4 Calculations

1.4.1 Master clock (MCLK)

The master clock is also called the oversampling clock and is fixed at 36.864 Mhz in the design. So

$$\#\ MCLK = 36.864Mhz$$

1.4.2 Serial/Bit clock (SCK/BCLK)

The serial/bit clock is generated internally using a divider circuit on the MCLK. The I2S transmitter is limited to the input frequency of the MCLK. For a MCLK of 36.864 Mhz the serial/bit clock possible frequncy is dependent on the sampling frequency of the audio sample. This clock is designed

Audio sample Bit Resolution	8Khz	16Khz	48Khz	96Khz	192Khz	384Khz
16	0.256	0.512	1.536	3.072	6.144	12.288
24	0.384	0.768	2.304	4.608	9.216	18.432
32	0.512	1.024	3.072	6.144	12.288	24.576

Table 1.1: SCK values (Mhz) for some common audio bit rate and resolution

to be programmed based on the audio specification required. Some of the configurations are :

The exact values of SCK for various audio sampling frequencies is given in the table 1.1.

For any other non standard audio bit rate and sampling frequency, The value of SCK is calculated as follows :

$$\#\ SCKfrequency = (2 * Fs * B)$$

where the multiplier 2 is for stereo , Fs is sampling frequency and B is the audio sample bit resolution.
For example for 48Khz 16 Bit audio SCK is 2*48*16 = 1.536 Mhz. **The multiplier is always 2 in I2S even for mono audio.**

1.4.3 Word select/Left-Right clock (WS/LRCLK)

The word select or the left-rigth clock is always the **sampling frequency** of the audio signal. For example for 8Khz audio data, WS/LRCLK is 8Khz. It is an non programmable auto generated signal based on the SCK clock.

2

System Architecture

The system architecture is shown below. The deisgn consists of the following submodules.

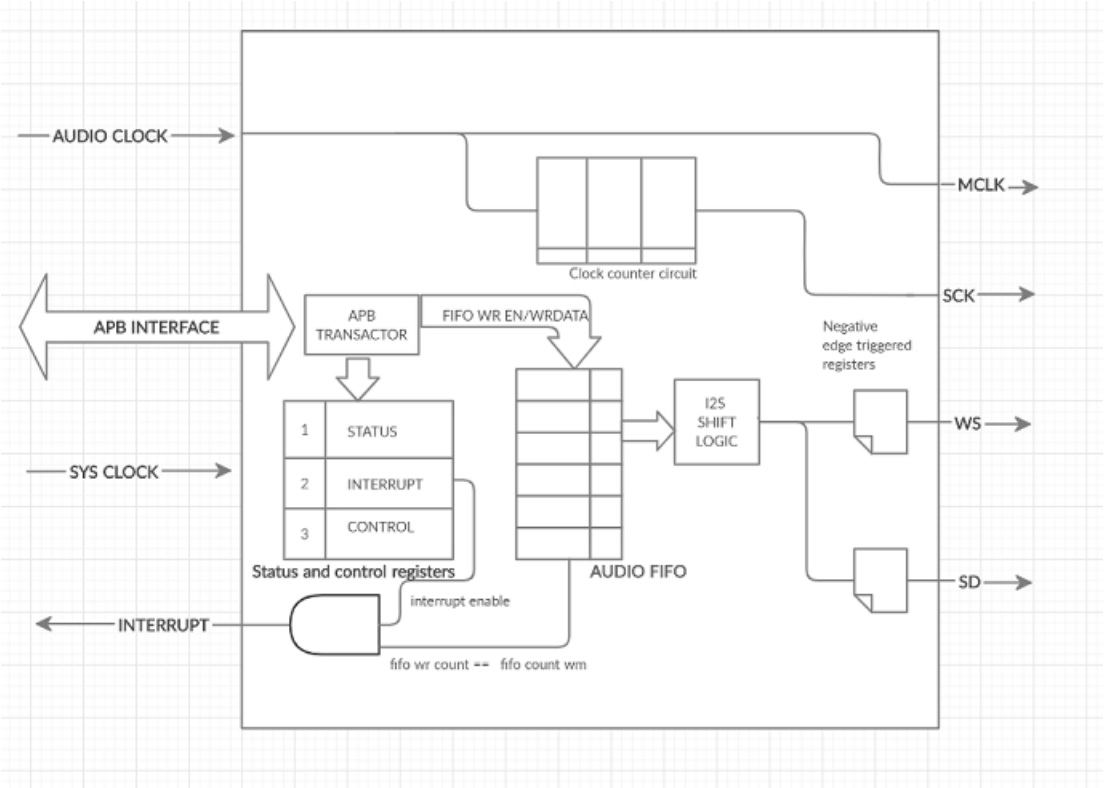

Figure 2.1: Block Diagram

- APB TRANSACTOR - The APB transactor is a state machine capable of handling the APB signals on the interface port.The APB transac-

tor provides exact APB timings for read and write transfers and no wait states are involved in the design. The read data is transferred on the same clock cycle as the PENABLE signal.The APB PPROT, PSLVERR, PSTRB are not used in the current architecture. All data transfers are 32 bit long. This transactor works at the system clock.

- AUDIO FIFO - 32-bit wide and 16 entry deep asynchronous fifo bewteen the two clock domains. The audio data is such that first left channel audio is to be written then right channel audio.
- I2S Shift logic - The I2S shift logic read the audio data from the fifo and generates the WS and SD signals.
- Registers - The system consistes of a set of registers to control the data transfer on the I2S bus. The detailed description of these registers is provided.
- Clock counter circuit - A simple clock divider unit is used to change the frequency of the SCK signal depending on the audio sample frequency. Legal values must be provided into the clock divider circuit.
- Interrupt Control - Generate an active high interrupt when enabled. The interrupt is activated by writing into the interrupt enable register. This is a programmable interrrupt based on the fifo occupancy. A value can be programmed in the fifo water mark register and a interrupt is generated whenever the fifo occupancy falls below the water mark value. It is cleared by reading the status register or by just disabling the interrupt enable bit.

2.1 Register Descriptions

2.1.1 ENABLE CORE REGISTER (ECR) [Read/Write]

Address - 0x00

Writing any non-zero value will enable the I2S core and the core will start SCK , dequeue audio data from AUDIO fifo and shift it out on the SD line. All other registers if needed must be configured before enabling the core for normal operation. Reading returns the core enable status (HIGH - core active).

2.1.2 AUDIO FIFO (AF) [Write Only]

Address - 0x04

Write audio data to this address. Maximum audio sample resolution supported is 32bit. Read as zero.

The order of data is left right left with left data written always first.

For mono mode this order is meaningless

Stereo data can be made mono by writting zero on the corresponding channel

Some audio DAC support mono out only. For such cases the DAC will handle the audio data and configure the transmitter as for normal stereo mode

2.1.3 SCK COUNTER REGISTER (SCR) [Read/Write]

Address - 0x08

Clock divider circuit configure value. The value written is such that :

$$\#\ SCR = (\frac{MCLK\_frequency}{2 * SCK\_frequency})$$

Since MCLK is always 36.864 Mhz

$$\#\ SCR = (\frac{36.864 * 10^6}{2 * SCK\_frequency})$$

where SCK frequency for some of the standard audio bit rate and resolution is depicted in table 1.1

The exact values for SCR for various audio bit rate and bit resolution is :

Audio sample Bit Resolution	8Khz	16Khz	48Khz	96Khz	192Khz	384Khz
16	72	36	12	6	3	1.5*
24	48	24	8	4	2	1
32	36	18	6	3	1.5*	0.75*

Table 2.1: SCR values for some common audio bit rate and resolution
*not supported

2.1.4 AUDIO SAMPLE BIT REGISTER (ASBR) [Read/Write]

Address - 0x0C

Write audio sample resolution into this register.Although any values can be written, standard values are 8, 16, 24, 32.

2.1.5 AUDIO FIFO RESET (AFR) [Write Only]

Address - 0x10

Writing any non zero value will reset the audio fifo. New data must be written before enabling the core once the reset has been issued. Read as zero.

2.1.6 AUDIO FIFO STATUS REGISTER (AFSR) [Read Only]

Address - 0x14

Reserved	Aud fifo write count	Audio tx complete	Aud Fifo in Reset	Aud Fifo Full
31:7	6:3	2	1	0

- Audio Fifo Full - Audio fifo full bit. A high value indicates that fifo can no more accept any data. Any new data write during fifo full will result in audio data loss
- Audio Fifo in Reset - This bit indicates that the Audio fifo is in reset state and any write should be avoided during this time. It is asserted during power on and during software resetting the audio fifo. A logic high indicates fifo is in reset state.
- Audio transfer complete - A logic high indicates all the data written in audio fifo is shifted out on the bus. Can be used to ensure complete data transfer before shutting down the core.
- Audio fifo write count - Indicates the actual number of data entries in the audio fifo.**Total fifo depth is 16**.
- Reserved fields - Writing to these fields have no effect. Read as zero.

2.1.7 INTERRUPT PENDING REGISTER (IPR) [Read Only]

Address - 0x18

A value of '1' in this register indicates that the interrupt condition has met and the core has asserted the interrupt and is waiting for the external controller to reset this interrupt. This bit is cleared when AFSR register is read. This bit is only set when the core is enabled, interrupt is enabled and also the interrupt condition is met. This bit is connected externally to the pin also. This purpose of this register is to use polling mode to detect the interrupt condition in an interrupt-less system. Writing to this register have no effect.

2.1.8 INTERRUPT ENABLE REGISTER (IER) [Read/Write]

Address - 0x20

Writing any non zero value will enable the interrupt. This bit can be cleared by writing zero to this register. **The external interrupt can be disabled by writing zero to this register or by simply reading the AFSR register.** Reading will return the value written.

2.1.9 INTERRUPT WATER MARK REGISTER (IWMR) [Read/Write]

Address - 0x24

This register contains a 4 bit value for interrupt condition. The value in this register is compared with audio fifo data count and the interrupt is generated.

2.2 Interrupt Operation

The interrupt mode of operation is very simple. A active high interrupt is generated when the interrupt condition is met. There is only one interrupt condition. A value can be programmed in the IWMR register and interrupt is enabled using IER. For example , if IWMR is programmed with a value of 10 and the interrupt is enabled, an interrupt (logic high) is generated when ever the audio fifo count goes below or becomes equal to 10. Setting a value of zero can be a test for audio fifo empty condition. The interrupt can be disabled completely in 2 ways :

- Writing zero in IER.
- Reading the AFSR.

The interrupt can also be snoozed by writing more data into audio fifo so that the number of data in fifo exceeds the water mark value in IWMR register.

3

Software Environment

Operating the core is very simple.

1. Calculate counter value and write in SCR. See SCR register specification on how to calculate the SCR value.
2. Write in ASBR the audio sample bit resolution. for 16 bit audio write 16 in ASBR.
3. Reset the audio fifo by writing any non zero value into AFR.(not necessary if the device is resetted by power on)
4. Check if the audio fifo reset is complete by reading AFSR fifo in reset bit.
5. Enable the interrupt by writing into IER and IWMR.
6. Write some audio data into audio fifo. (A maximum of 16 write can be done if the fifo is empty).
7. Enable the core.
8. Poll the IPR to see if the interrupt condition is met. If interrupt is supported by the external system, then an logic HIGH will be generated on the irq pin and the external commanding unit can start its interrupt service.
9. Disable/ snooze the interrupt.
10. Write audio data into audio fifo.
11. Disable the core.
12. Reset the audio fifo.

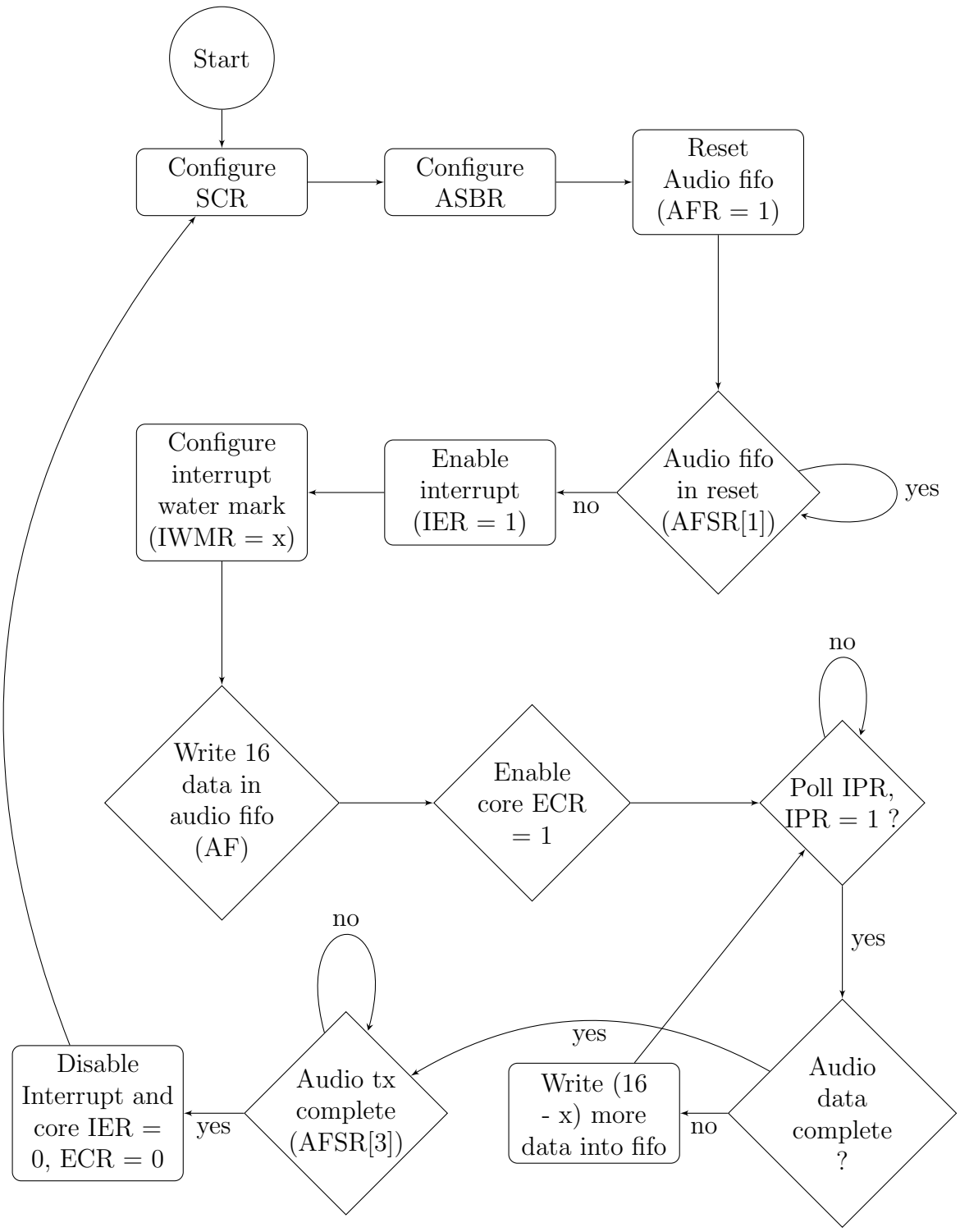

Figure 3.1: Polling mode of operation

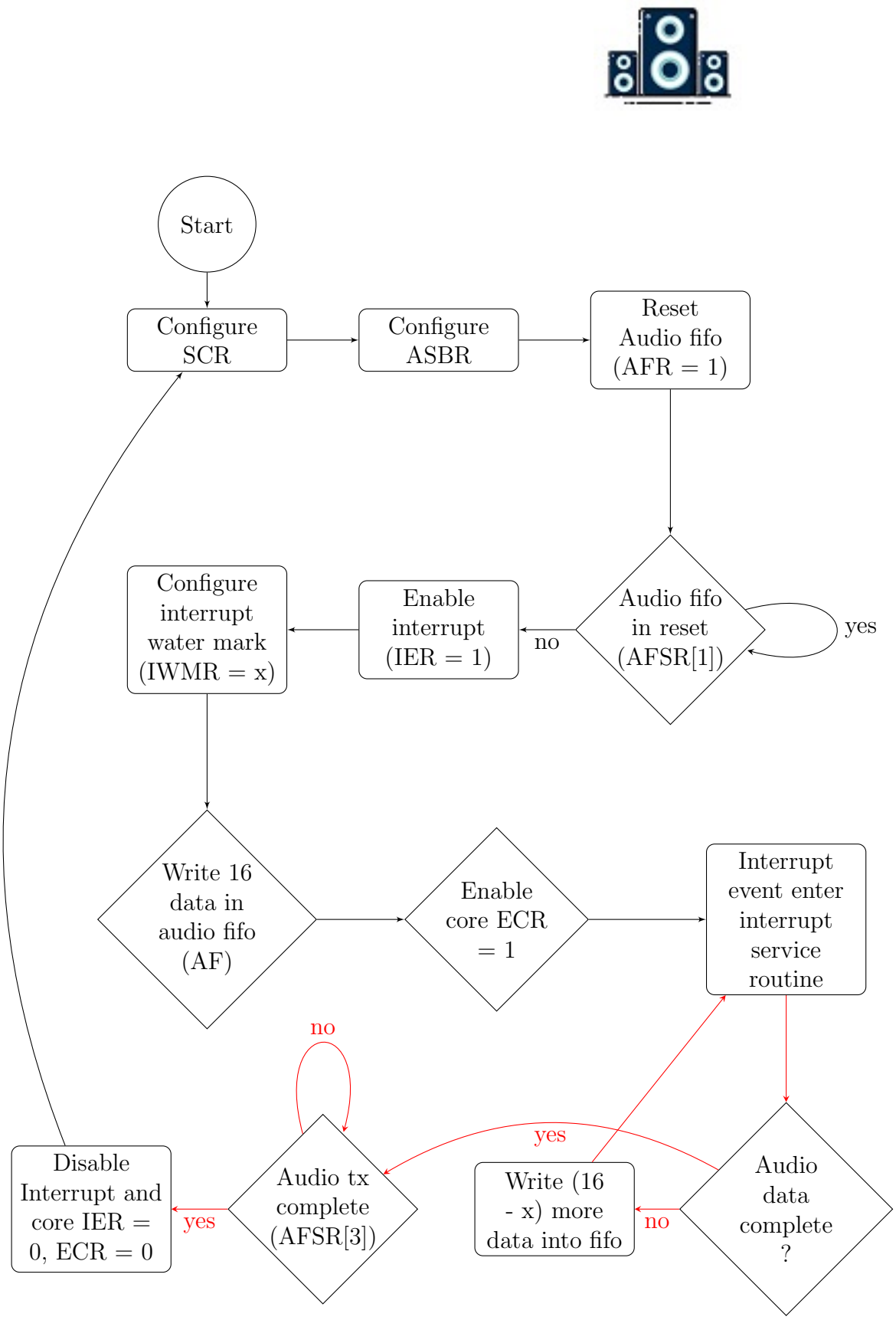

Figure 3.2: Inerrupt mode of operation : The paths in Red depicts the interrupt service procedure

4

Constraining the core

There are few timing sdc constraints required to time the design. The design has 2 negative edge triggered registers so there are few things to be constrained. The core is designed in such a way that all the logic is based on leading edge of the sys_clk and aud_clk, however as per protocol requirements the WS and SD signals are shifted to the external pins based on trailing edge of the clock. Internally WS and SD is generated on the leading edge of the clock and it is just registered on the trailing edge before assigning on the pins. So there will be 2 half cycle paths in the audio domain. The 2 half cycle paths will not have any combinational circuit. The timing sdc for the paths are :

- set_half_cycle_path -from */lrclk_aud_rp[CK] -to */lrclk_aud_rn[D]
- set_half_cycle_path -from */sdata_out_rp[CK] -to */sdata_out_rn[D]
- create_clock -name audio_clk -period 27.126 -waveform {0.0 13.56} -source [get_ports aud_mclk]
- create_generated_clock -name sck -source [get_ports aud_mclk] -master_clock audio_clk -divide_by 1 [get_ports sclk_out]