DTP2_PROJ_3

School of Engineering

Codec Digital Audio Interface

I2S Protocol

Block Aufteilung für Audio Interface Driver

- I2S_Master Block
- Audio_Control Block

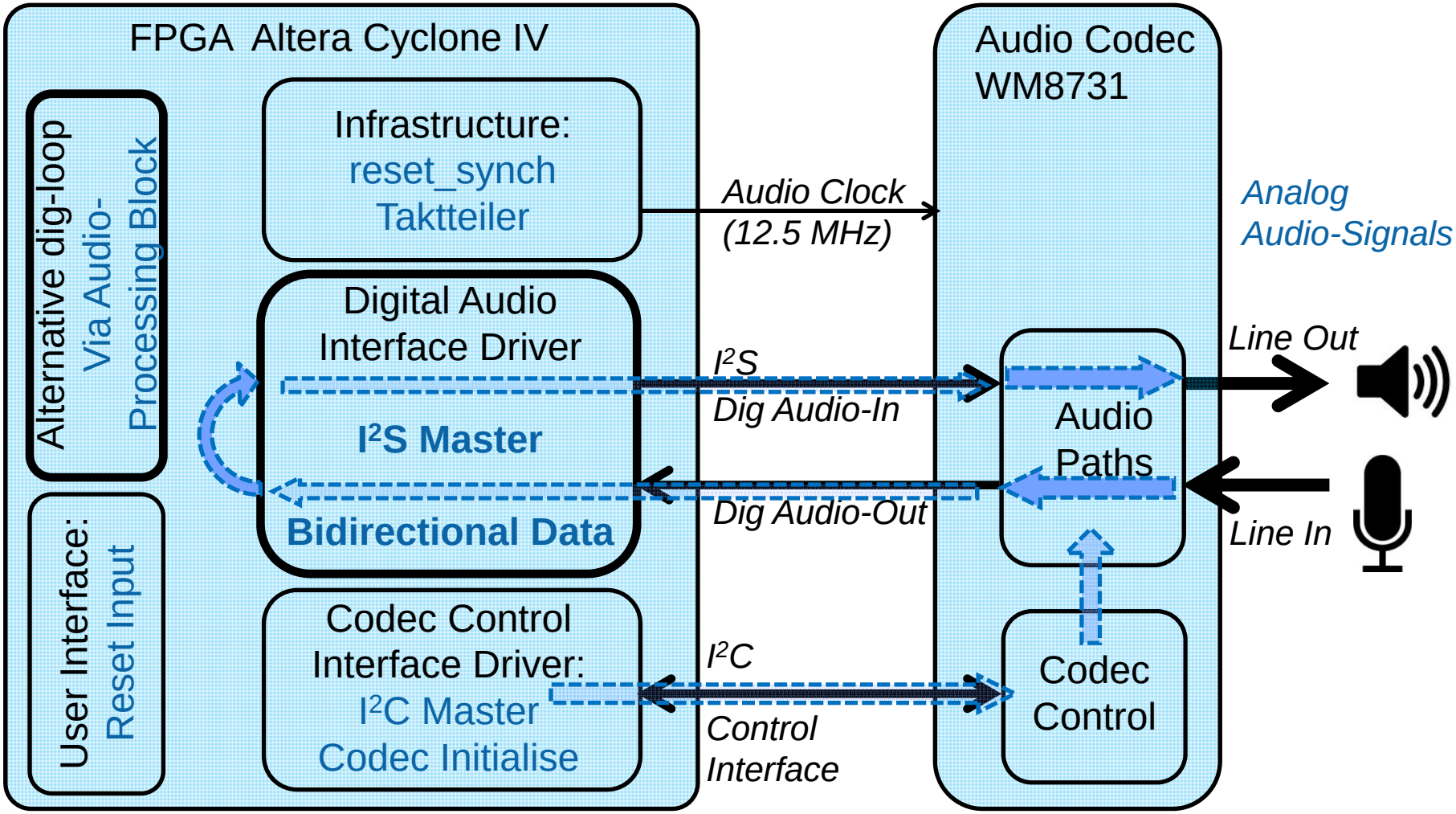
I2S Master Design

- Subblocks und Zeitverlaufsdiagramm

Phase-2 *Milestone-2*

Digital Audio-Loop Test

School of Engineering



Objective: test for digital audio interface driver *(I²S Block Design)*

WM8731 Digital Audio Interface Protocol

School of Engineering

Format: I^2S
Data Width: 16 bits
Operation: slave mode (BCLK generated in FPGA)

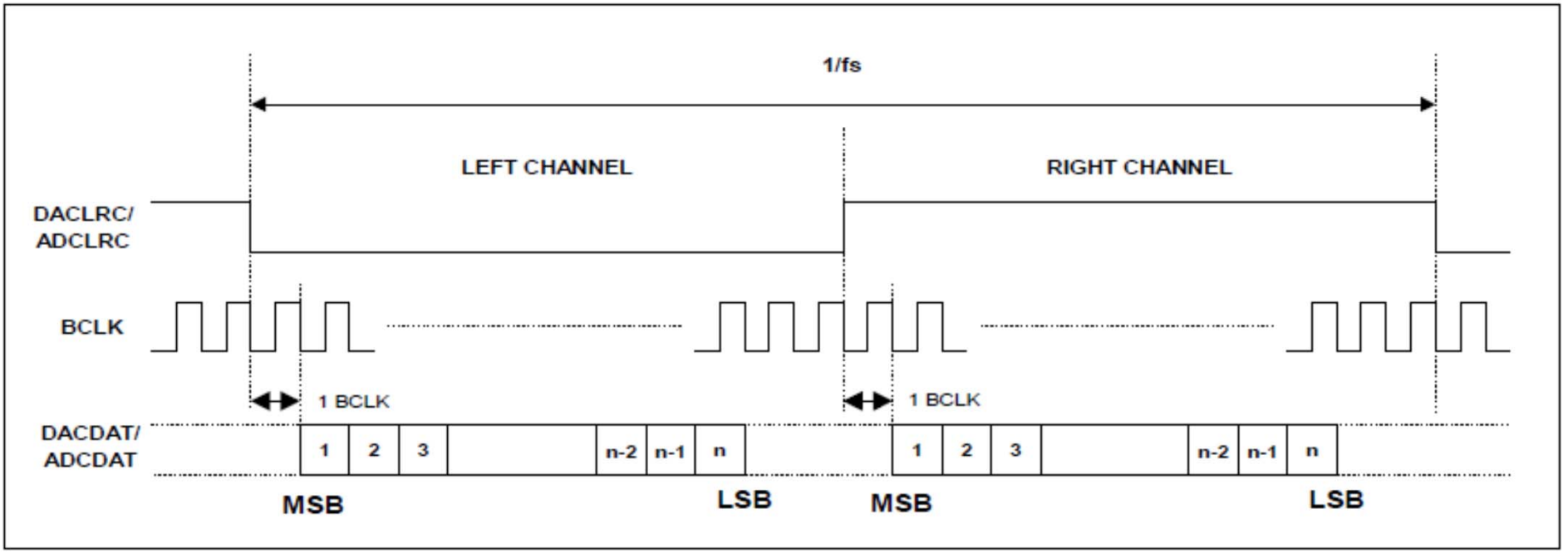
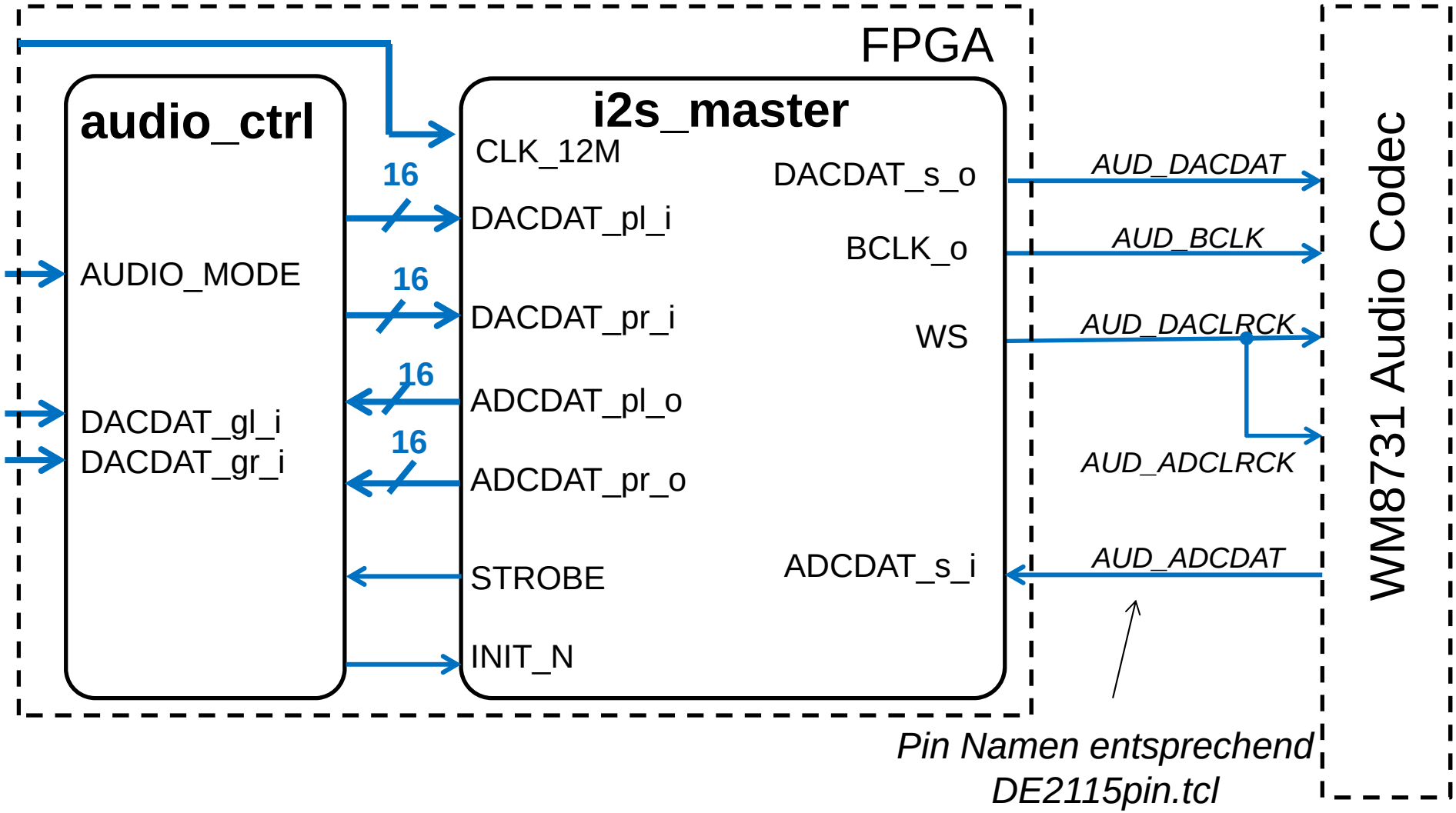


Figure 27 I^2S Mode *Quelle: WM8731L Datasheet Rev4.3*

I^2S mode is where the MSB is available on the 2nd rising edge of BCLK following a DACLRC or ADCLRC transition.

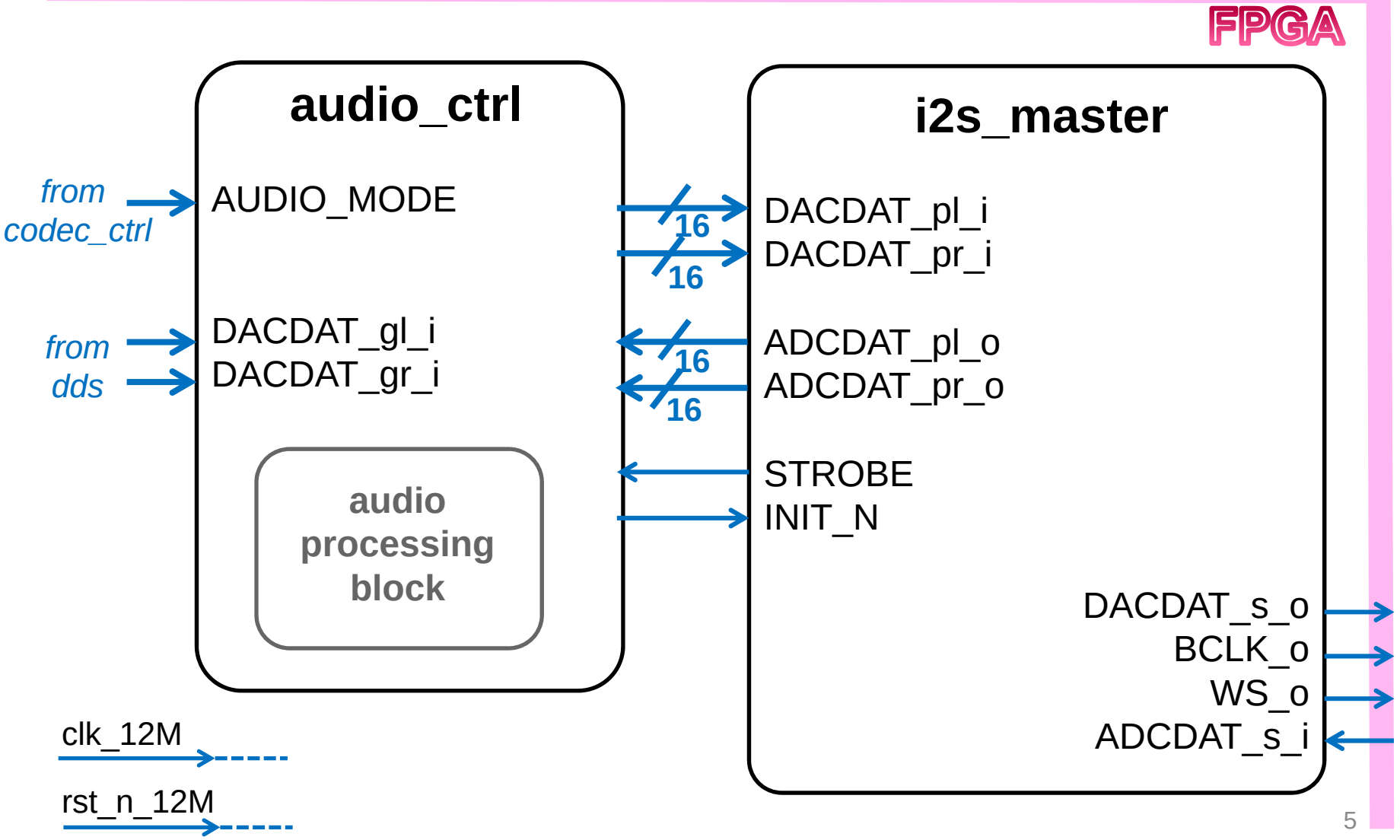
WM8731 Digital Audio Interface Pinning

School of Engineering



Digital Audio Interface Driver Blocks

School of Engineering



I2S Master Subblocks

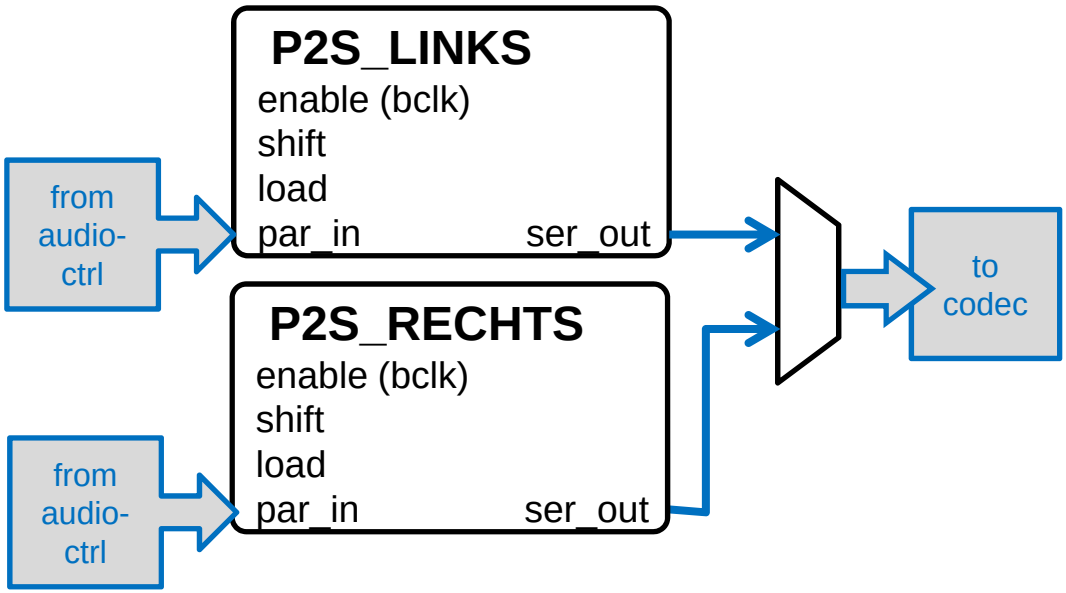
ALL SUBLOCKS HAVE
- clk_12M
- rst_n_12M

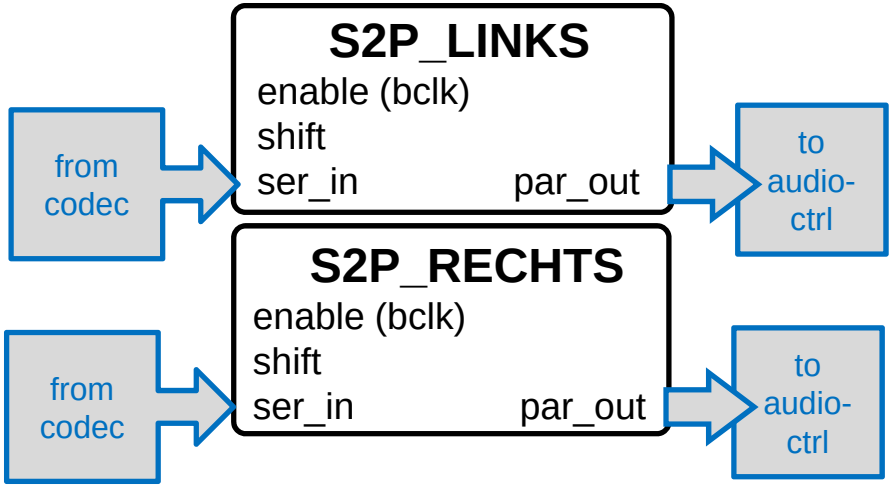
School of Engineering

BCLK_GEN
: 2
bclk

BIT_CNTER
0…127
bit_cnt
enable

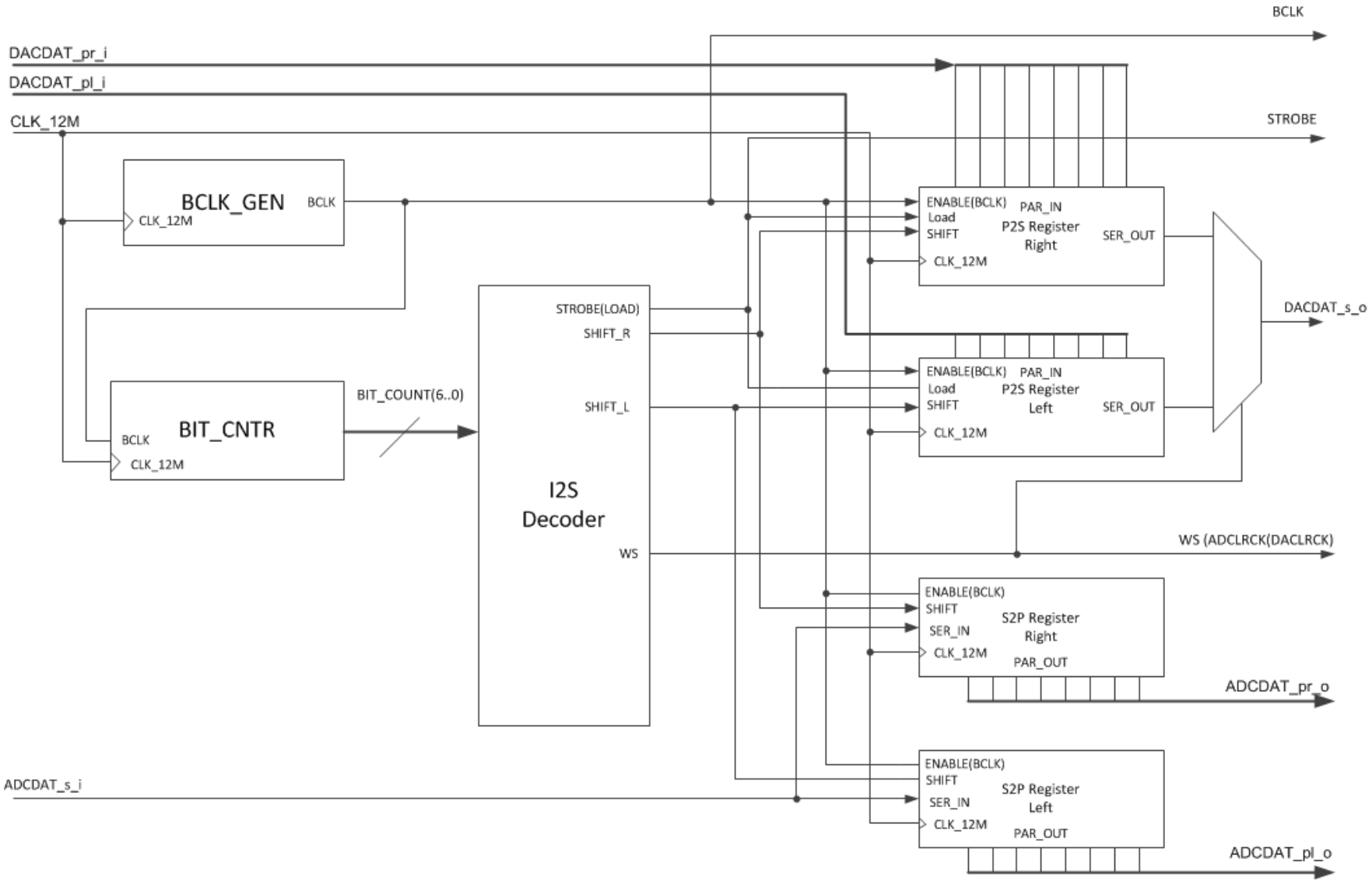
I2S_Decoder
bit_cnt
enable
shift_L
shift_R
strobe(load)
WS





I2S Master _ Blockdiagram

School of Engineering



Digital Audio Interface
Setup with fs = 48kHz

School of Engineering

Gegeben

Master Clock clk_12M : 12,288MHz **(nominal value)**

Vorschlag

Bit Clock BCLK : (clk12_M/2) = 6,144MHz

Word Select WS : 48kHz = BCLK/128

(also called LRCK)

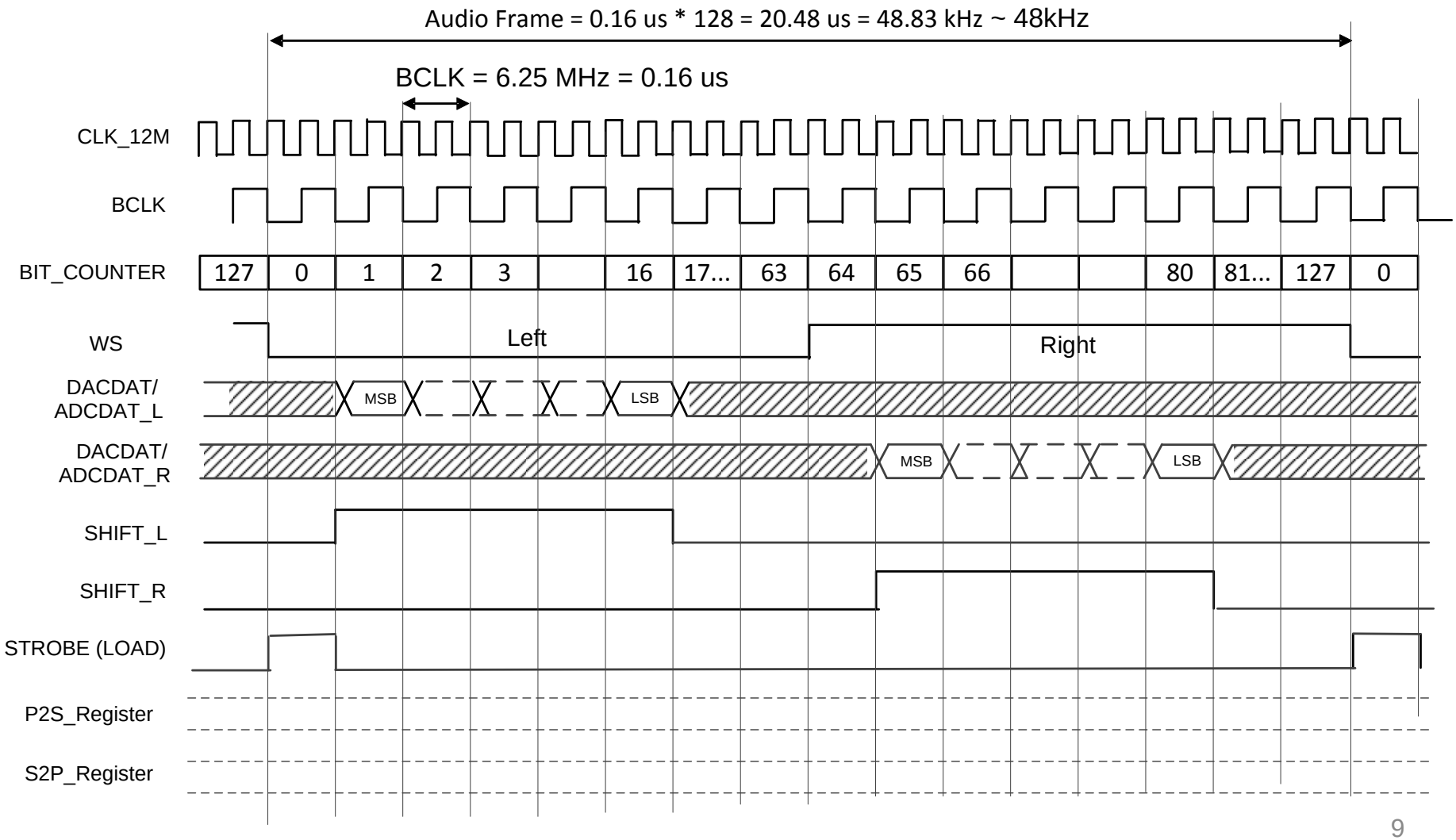
Hinweis VHDL Syntax

```
constant F_MCLK:          natural := 12_288_000;
constant F_BCLK:          natural := F_MCLK/2;
constant F_FS:            natural := 48_000;
constant COUNT_WS_TOGGLE: natural := F_BCLK/F_FS;   -- equals  128
```

Audio Frame

(strobe by counter=0; easier for reset value)

School of Engineering



I2S Master

School of Engineering

Counter	0	1--16	17--63	64	65--80	81--127
WS	0 (left)			1 (right)		
action	load	shift_L	hold_L		shift_R	hold_R
SIGNALS						
shift_L	0	1	0	0	0	0
shift_R	0	0	0	0	1	0
load	1	0	0	0	0	0

Observations:

- The input load has higher priority then shift for the P2S blocks
- Reset value is made so that you start by loading data right after reset;
- If you wish to have a «clean-transition» between modes, then introduce a init_n signal which can also take counter to reset value (this is just a nice to have, not a must)
- The I2S interface runs «continuously», only way to stop it is activating init_n (but again implementing init_n is a nice to have, not a must)