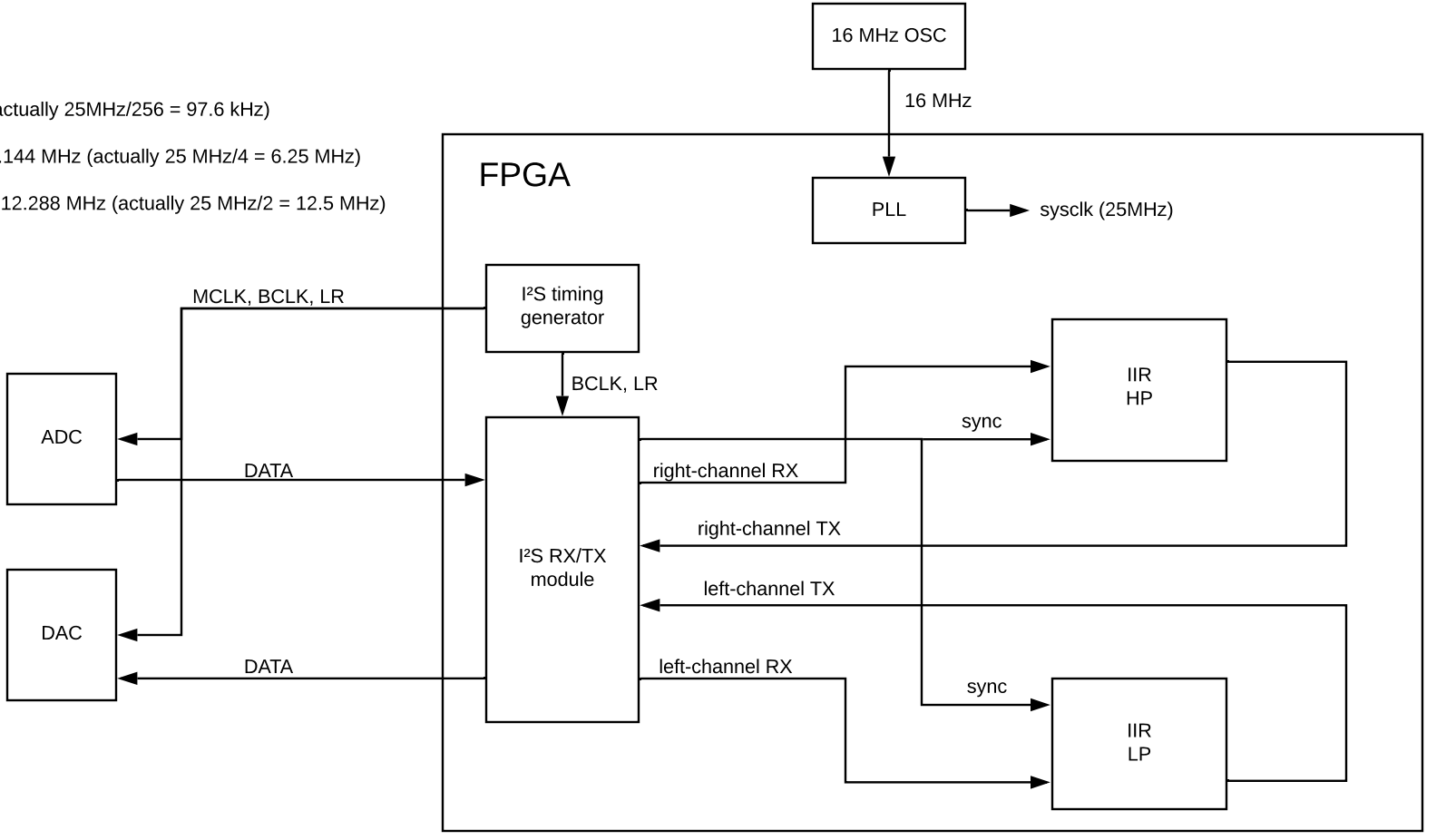
LR (fs) = 96 kHz (actually 25MHz/256 = 97.6 kHz)

BCLK = 64x fs = 6.144 MHz (actually 25 MHz/4 = 6.25 MHz)

MCLK = 128x fs = 12.288 MHz (actually 25 MHz/2 = 12.5 MHz)



Floating-point and fixed-point

> Audio-Samples are represented in 32-bit integers coming from I²S module
> Filter coefficients for IIR / FIR filters typically represented in fractional numbers

Example for DSPs / Microcontrollers (with floating-point support)

Sample: 18745
Coefficient: 0.007938475

```
int result = (int) ((float) sample * 0.007938475f)
//result would be = 149
```

FPGAs

> No floating-point support by default. However, a dedicated floating-point unit can be implemented (but not efficient in terms of FPGA usage)
> By default support for signed and unsigned integers for multiplications, summings and substractions
> Bit-shifting very easy to implement

Sample: 18745
Coefficient: 0.007938475

Step 1 (preparation)

What fixed-point format is needed?
> IIR parameters can vary between typically -2.0 and + 2.0
> Using Q2.30 format (for 32-bit resolution)
>can represent numbers between -2.0 and +1.9999999...
> Multiply all coefficients with 2^30 -> e.g. 0.007938475*(2^30) = 8523873

Step 2 (FPGA implementation)

Multiply incoming sample and fixed-point coefficient
> 18745 * 8523873 = 159779999385
> Caution: a 32bit by 32bit multiplication will result in a 64 bit output vector on a FPGA

> Apply shift_right by 30 bits on the output result. This is effectively a division by 2^30.
159779999385 >> 30 = 149

> Reduce output result from 64-bit again to 32-bit