Laboratory 3
All-digital FM stereo modulator

Diogo Correia e Pedro Augusto

January 10, 2019

Abstract

This report details the work developed to implement the FM stereo modulator studied in the lab classes of the course PSDI, Digital Systems Design of the Integrated Master in Electrical and Computer Engineering of University of Porto.

1 Architecture

The FM modulator datapath provided in the Guides for the report was implemented not as one unique top module but in the organization in the picture below:

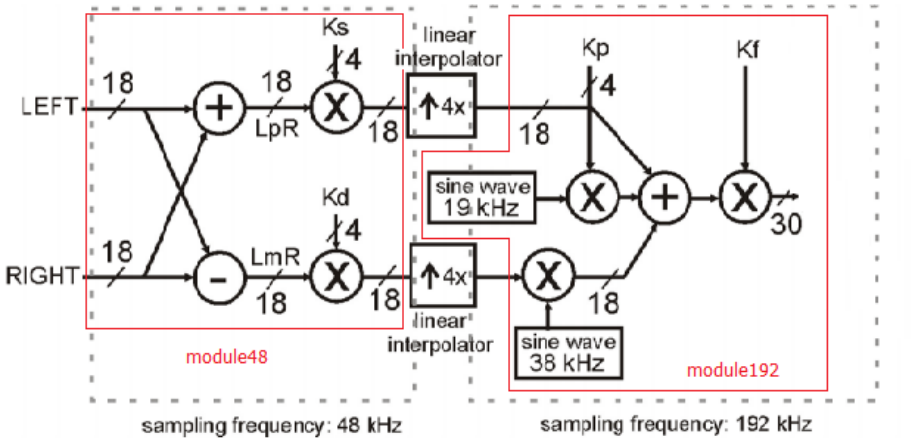


Figure 1: FM modulator.

For modularity purposes and ease of implementation, the system was separated in two sections, module48 and module192, with the two interpolators bridging the two. The dds modules included in module192 are instances of a dds parametrizable module detailed in the next section. The top level module was named stereo_fm_mx and instantiates module48 as module48_1, module192 as module192_1, the interpolators as interpolL and interpolR.

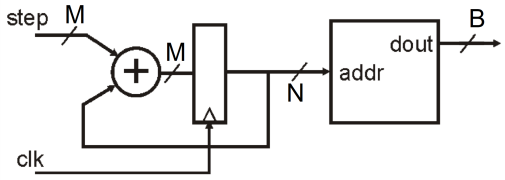
2 DDS parametrizable module implementation

From a lookup table holding 2^N samples of a single sinusoidal wave period a value is selected, based on the input's N more significant bits, to be the output. These samples were generated by a MATLAB's script and stored on a separated file, this file's path is one of the parameters of our module. The values stored on this file are B bit sized, also a module's parameter.

```
module dds (clock, reset, enableclk, phaseinc, outsine);

    parameter M = 13;
    parameter N = 7;
    parameter B = 32;
    parameter FILE = "../simdata/DDSLUT.hex";
    parameter NSamplesLUT = 64;
```

(a) in Verilog



(b) on Diagram

Figure 2: DDS Parameters

The input's bit size, M, the LUT address bit size, N, and the number of samples in the LUT, $NSamplesLUT$, are the other parameters.

3 Control Path Mechanism

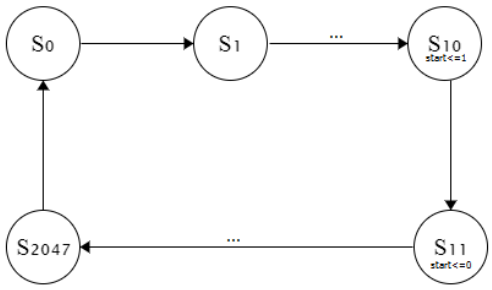
The Control Mechanism implemented was based in a finite state machine approach.

Two finite state machines were developed, one for the control of the sequential multipliers in module48 and other for the control of the sequential multipliers in module192.

The first fsm has 2048 states to comply with the timing constraints of the signal enableclk48. At the tenth state the sequential multipliers' start input is set to high and in the eleventh state is set to low. From this point on, the 2 sequential multipliers in module 48 start the shift-add iterative process at the same time. The choice of state 10 was semi-arbitrary because as the multiplication is over in $5 + 2 = 7 << 2048$ clock cycles, the state number could vary from 2 to 2046. Just to be safe, it was chosen 10.

```
always @(posedge clock)
if (reset)
    state <= 11'd0;
else
begin
    case (state)
        11'd0: if ( enableclk )
            state <= 11'd1;
        11'd10: begin
                start<=1;
                state <= state + 1;
                end
        11'd11: begin
                start<=0;
                state <= state + 1;
                end
        default: if ( state == 11'd2047 )
                state <= 11'd0;
                else
            state <= state + 1;
endcase
end
```

(a) in Verilog



(b) on Diagram

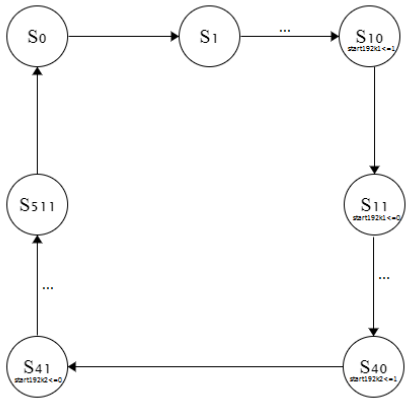
Figure 3: Module48 finite state machine.

The second fsm has 512 states and defines the timings of the start inputs of the three multipliers present in this module. This time around, there are necessary two start signals, start192k1 and start192k2.

The start input of multipliers Kp x sinewave19kHz and interpolRout x sinewave38kHz, start192k1, is set to high at state 10 and set to low at state 11. However, the start input of multiplier Kf x sum, start192k2, is only set to high at state 40 and set to low at state 41 in order to guarantee that this final multiplier has not a desynchronized input.

```
always @(posedge clock)
    if (reset)
    state <= 9'd0;
    else
    begin
        case (state)
            9'd0: if ( enableclk )
                    state <= 9'd1;
            9'd10:  begin
                    start_192k1<=1;
                    state <= state + 1;
                    end
            9'd11:  begin
                    start_192k1<=0;
                    state <= state + 1;
                    end
            9'd40:  begin
                    start_192k2<=1;
                    state <= state + 1;
                    end
            9'd41:  begin
                    start_192k2<=0;
                    state <= state + 1;
                    end
            default: if ( state == 9'd511 )
                    state <= 9'd0;
                    else
                    state <= state + 1;
    endcase
    end
```

(a) in Verilog



(b) on Diagram

Figure 4: Module192 finite state machine.

4 Optimizations

In terms of optimizations, the ready outputs of the sequential multipliers weren't used.

Due to the use of the same *start* signal for all the sequential multipliers in module48, the use of a *ready* output signal in its multipliers is needless. In fact, as the number of clock cycles between each posedge of the clock enable signal (2048 for clkenable48) is far superior to the number of clock cycles necessary for each multiplication (Nmax+2=7), the multiplications results are available well before the next rising edge of the clkenable48. The same logic applies to module192 so the output ready of all the multipliers in this module was not used.

As a result of this optimization, during the synthesis proccess, all the seqMultNM's ready signal generators are discarded from the final circuit saving area for only the crucial parts of the system.

5 FPGA resource occupancy

As specified, our circuit was synthesized with Area optimization in mind. The table bellow shows how much of the FPGA's total "area" was used when selecting 'High Area Optimization' on Xilinx's Synthesis Options.

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1,695	54,576	3%
Number used as Flip Flops	1,695		
Number of Slice LUTs	1,34	27,288	4%
Number used as logic	1,208	27,288	4%
Number used as Memory	0	6,408	0%
Number used exclusively as route-thrus	132		
Number of occupied Slices	520	6,822	7%
Number of MUXCYs used	712	13,644	5%
Number of LUT Flip Flop pairs used	1,787		
Number with an unused Flip Flop	416	1,787	23%
Number with an unused LUT	447	1,787	25%
Number of fully used LUT-FF pairs	924	1,787	51%
Number of unique control sets	52		
Number of slice register sites lost to control set restrictions	185	54,576	1%

6 Additional Developments

No relevant additional developments were made. It was attempted to use different gains in order to improve our circuits performance, but without any substantial results.