

# VLSI Implementation of the Farrow Structure

Marek Wróblewski\*

Artur Wróblewski\*

Josef A. Nossek\*

**Abstract** — This paper discusses measures that need to be taken in order to attempt a VLSI implementation of the Farrow structure. An extension of the interpolator is presented that enables a robust operation independently of the ratio of sampling rates. Quantization effects are studied and recommendations for choosing bit-widths are given. Based on a synthesized design and transistor-level simulations estimates of signal quality, area and power consumption are included.

## 1 INTRODUCTION

Sample Rate Conversion (SRC) by non-integral factors by means of up-sampling and subsequent down-sampling is costly in terms of area and power consumption, and sometimes simply infeasible due to high clock rates of the interpolator [2]. Structure presented by Farrow [3] provides an elegant solution to this problem. Originally conceived for changing the sampling instant while maintaining the sample rate, the circuit can be utilized as a universal sample rate converter by rational factors.

While the circuit has been studied extensively by the filter community [4], [1], aspects related to actual implementation of the structure on a chip have hardly been considered.

This paper investigates issues involved in VLSI realization of the Farrow structure, particularly those stemming from the fact that the circuit is clocked using two independent oscillators and no provisions are made for synchronization of input and output data.

We limit our considerations to the case of an interpolator, i. e.  $T_{\text{src clk}} > T_{\text{tgt clk}}$ , where  $T_{\text{src clk}}$  and  $T_{\text{tgt clk}}$  denote the sampling periods of incoming (source) and outgoing (target) signals, respectively. Furthermore, in order to avoid case differentiation, we use flip-flops triggered by rising edge of the clock signal.

## 2 THE FARROW STRUCTURE

Figure 1 shows a simplified block diagram of the Farrow structure. As a detailed description of the circuit can be found in [3] we limit the following overview to its aspects pertaining to the matters discussed in this paper.

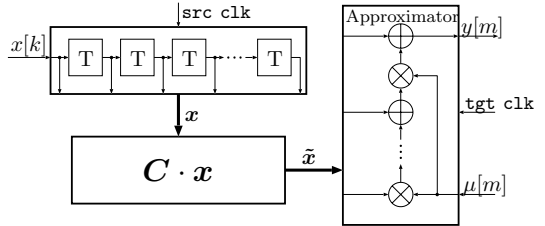


Figure 1: The Farrow structure.

Two distinct blocks constitute the structure: a block of FIR-Filters with the state vector  $\mathbf{x}$  and each filter's coefficients forming a row of matrix  $\mathbf{C}$ , and an approximator computing a polynomial in  $\mu$  whose coefficients are updated every  $T_{\text{src clk}}$  by the filter block. The *inter-sample position* (ISP)  $\mu$  states at what instant, expressed in relation to the clock period of  $\text{src clk}$ , a new output sample should be computed (cf. Fig. 2).

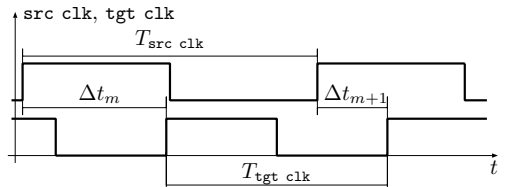


Figure 2: The inter-sample position.

This position changes every  $T_{\text{tgt clk}}$  and can be determined from

$$\mu_m = \frac{\Delta t_m}{T_{\text{src clk}}} \quad (1)$$

Unfortunately time measurements can only be obtained with the resolution of a clock signal. Even assuming that a clock signal with period  $T \ll T_{\text{tgt clk}}$  were available in the circuit, the precision of the measurements would still have to be much higher, in the range of setup times of flip-flops. To justify this claim let us consider the situation depicted in Fig. 3.

From measurement we obtain  $\mu_m = 0.95$ , which means that the rising edge of  $\text{tgt clk}$  was observed at the instant  $t_m$ , i. e. from (1)  $\Delta t_m = 0.95 \cdot T_{\text{src clk}}$  after the rising edge of  $\text{src clk}$ . Because of the

\*Munich University of Technology, Institute for Circuit Theory and Signal Processing, D-80290 Germany, e-mail: Marek.Wroblewski@nws.ei.tum.de

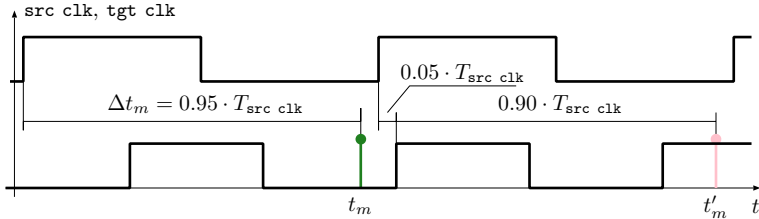


Figure 3: Small inaccuracies during measurement of  $\Delta t_m$  may lead to significant errors.

limited precision of the circuitry used this value is wrong and in reality the rising edge of **tgt clk** appeared briefly after a new rising edge of **src clk**, so that the precise value is  $\mu_m = 0.05$ . Because new input data has arrived at the rising edge of **src clk**, by using the wrong value of  $\mu_m$  the approximator computes a value that is valid for an instant  $t'_m$  that lies  $\Delta t_m = 0.95 \cdot T_{\text{src clk}}$  after the new rising edge of **src clk**. Thus the error is much greater than the inaccuracy of the measurement.

Given  $\mu_{\text{step}} = \frac{T_{\text{tgt clk}}}{T_{\text{src clk}}}$  and  $\tau = \frac{\Delta t_0}{T_{\text{src clk}}}$  the sequence  $\mu_m$  can alternatively be computed for  $m \in \mathbb{Z}^+$  as follows:

$$\mu_m = \begin{cases} \mu_{m-1} + \mu_{\text{step}} & \text{if } \mu_{m-1} + \mu_{\text{step}} \leq 1 \\ \mu_{m-1} + \mu_{\text{step}} - 1 & \text{if } \mu_{m-1} + \mu_{\text{step}} > 1 \end{cases} \quad (2)$$

with  $\mu_0 = \tau$ .

The ratio of clock periods  $\mu_{\text{step}}$  is either known at design time and can thus be read from non-volatile memory when needed, or can be determined at runtime. This can be achieved by means of two counters, one of which is clocked by **src clk**, while the other is clocked by **tgt clk**. Assuming that each of counters is  $b$  bits wide, after  $2^b$  **tgt clk**-periods the counter clocked by **src clk** contains  $\mu_{\text{step}}$ .

The estimation of  $\tau$  is subject to exactly the same problems as explained above for  $\Delta t_m$ . Therefore the goal is to find a way of computing  $\mu_m$  where  $\tau$  can be chosen arbitrarily. This condition can be satisfied if the value of  $\mu_m$  used by the approximator is chosen independently from the actual intersample position as determined by the timing relation of the signals **src clk** and **tgt clk**. This presumes that incoming data used by the approximator together with  $\mu_m$  remain available as long as it is needed, independently of **src clk**-cycles. As will be shown in Sec. 3.4 the storage required here can also be used to solve timing related problems.

### 3 EXTENSION OF THE FARROW STRUCTURE

We extend the Farrow structure as shown in Fig. 4.

The output of every filter in the block FIR is connected to a ring buffer, each containing 3 registers<sup>1</sup>. Write access to all the ring buffers simultaneously is controlled by the block IRC, while reading is supervised by the block ORC. We explain the working principle of the extension as we take a closer look at both blocks.

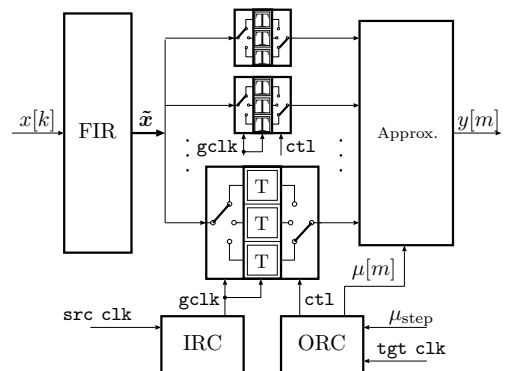


Figure 4: The modified Farrow structure.

#### 3.1 Writing (IRC)

Incoming data is latched at the rising edge of **src clk**. IRC contains a counter which is incremented every time new data arrives. It overflows above 2 so that its output can be used as address of the register to be written to and used to generate the gated clock signal **gclk**.

#### 3.2 Reading (ORC)

Like IRC the block ORC contains a 2-bit counter used as address (**ct1**) generator. It is incremented synchronously with **tgt clk**, however only when new input data is to be supplied to the approximator. The decision whether it is the case is made based upon the values of  $\mu_{m-1}$  and  $\mu_m$ . Because

<sup>1</sup>We substantiate the choice of 3 registers in Sec. 3.3.

$\mu_{\text{step}} < 1$ , the condition  $\mu_m < \mu_{m-1}$  signifies the moment when the address needs to be incremented (corresponds to the second case of (2)). Thus the block ORC contains a hardware implementation of (2), which provides an `enable` signal for the clock-gating cell used to generate clock signal for the counter. We set  $\mu_0 = 0$ .

### 3.3 Input-output synchronization

Particular care needs to be taken in order to ensure proper synchronization of read and write addressing. It should hold that  $q \cdot T_o = \frac{T_o}{T_{\text{src clk}}}$  for a sufficiently long observation time  $T_o$ , where  $q$  denotes the number of address changes seen on the line `ctl` during that time. As these changes are generated based on the computed value of  $\mu_m$ , it is imperative that sufficient bit-width be provided for  $\mu_{\text{step}}$  and the circuitry performing the computation. It should be noted that the actual value of  $\mu_m$  used by the approximator need not to be supplied in that high a precision.

To justify the utilization of as many as three registers per data line we conduct the following argument. While it still holds  $T_{\text{src clk}} > T_{\text{tgt clk}}$ , we consider the case where  $T_{\text{src clk}} \approx T_{\text{tgt clk}}$ . If the rising edge of `src clk` appears briefly after the rising edge of `tgt clk` at time  $t_0$ , then the value stored at this time will not be used (i. e. the output of the register where it is stored will not be connected to the approximator's inputs) until approximately  $t_1 = t_0 + T_{\text{tgt clk}}$ . If additionally  $\mu_m < 1 - \mu_{\text{step}}$  at time  $t_1$ , then the value is going to be used for  $2 \cdot T_{\text{tgt clk}}$ , so that it will not be needed anymore only at time  $t_2 = t_1 + 2 \cdot T_{\text{tgt clk}} = t_0 + 3 \cdot T_{\text{tgt clk}}$ . Thus for  $T_d = T_{\text{src clk}} - T_{\text{tgt clk}}$  holds

$$\lim_{T_d \rightarrow 0} (t_2 - t_0) = 3 \cdot T_{\text{src clk}},$$

i. e. in order to guarantee robust operation a stored value needs to remain stable for at least 3 `src clk`-periods.

The above is also the worst case. While it is true that for  $\frac{T_{\text{src clk}}}{T_{\text{tgt clk}}} > 2$  the same value may be used for longer than  $2 \cdot T_{\text{tgt clk}}$ , the time is still in the range of  $2 \cdot T_{\text{src clk}}$ . Simultaneously  $t_1 - t_0 < \frac{T_{\text{src clk}}}{2}$ , so that  $t_2 - t_0 < 3 \cdot T_{\text{src clk}}$ .

It is important to realize that this situation takes place periodically, but, depending on the ratio of clock periods, only occasionally when considered in  $T_{\text{tgt clk}}$  time frame. The data stored in the next register is used for a time that is shorter than  $T_{\text{src clk}}$ . This is the case for several following data items, again depending on the ratio. As a result the

read address changes faster than the write address until again the second case of (2) occurs, which gives the write address chance to catch up.

### 3.4 Non-zero delay impact

The above considerations do not take into account signal delays introduced by cells used to implement the circuit, in particular setup and hold times of flip-flops.

In the original Farrow structure in Fig. 1 the FIR-filter block processes data synchronously with `src clk` while the approximator works with `tgt clk`. As both signals may be generated by two independent oscillators and time between rising edges of both clock signals changes constantly, there is no guarantee that no invalid data (as e. g. caused by glitching) will be used in the approximator. The extension in Fig. 4 eliminates glitch-related problems, as data is latched with `src clk` so that the inputs of ring buffer's flip-flops remain stable sufficiently long.

With proper synchronization of read and write address generators the circuit also prevents errors which could occur when rising edges of both clock signals arrive nearly simultaneously<sup>2</sup>. Assuming we had a register clocked by `src clk` followed by a register clocked by `tgt clk`, in this case undefined data could propagate to approximator, because setup and hold times of flip-flops could not be made allowance for. With the introduction of the ring buffer, once a signal is latched in one of its registers, it is guaranteed to remain stable for  $3 \cdot T_{\text{src clk}}$ . As elaborated in Sec. 3.3 this amount of time is sufficient to switch to the register, process the data contained therein and switch to the next in time to make the register available for incoming data.

## 4 SIMULATION RESULTS

In order to test the presented extension to the Farrow structure we described the circuit in VHDL. Little attention has thereby so far been paid to the optimization of the FIR and approximator blocks. We used the same set of coefficients as provided in [3], which leads to a block of four 8th order filters. Except where noted all results presented in this paper were obtained by simulating this description using a commercial VHDL simulator.

To verify the proper function the circuit has additionally been synthesized for various bit-widths using a commercial synthesis tool and simulated on

<sup>2</sup>We do not make any assumptions about the implementation of the approximator. It could e. g. have a pipeline stage (clocked by `tgt clk`) at its input.

transistor level using PowerMill for several clock period ratios. Both for synthesis and simulation the technology data of a  $0.18\mu\text{m}$  standard cell library operated at  $V_{dd} = 1.6\text{V}$  were used. Neither placement nor routing information has been taken into account.

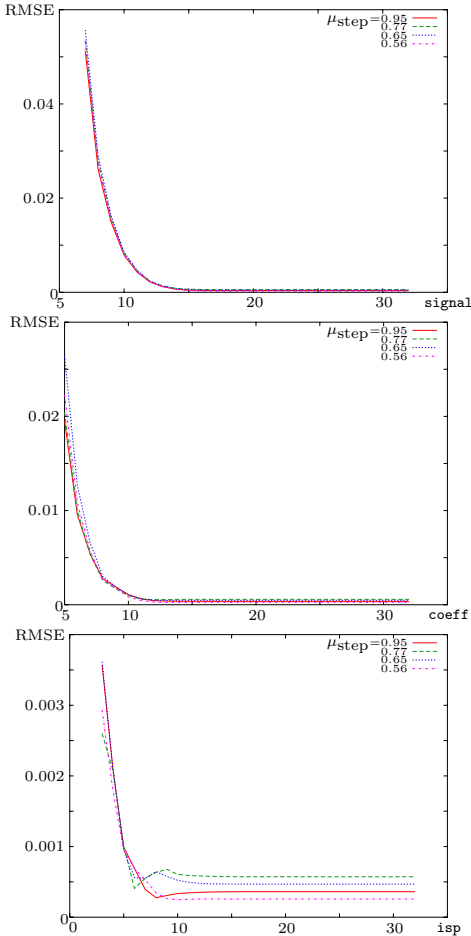


Figure 5: RMSE for varying values of **isp** and **signal** and **coeff** and **mustep**.

We use RMSE obtained by taking samples computed using floating-point arithmetic based on perfect knowledge of the input signal as reference output values for the data provided by the simulated circuit. As input we use a single RRC impulse with roll-off factor of 0.35 and length of 30 symbols.

The diagrams in Fig. 5 provide RMSE for various combinations of bit-widths and sampling ratios. If not specified bit-widths were set to 48,  $\mu_{\text{step}}$

to 0.95. Without discussing the results in detail here, we conclude that the values in the quadruple (**isp**, **signal**, **coeff**, **mustep**)=(7,16,11,10) each for itself attain RMSE values which do not improve significantly for higher bit-widths (for this particular combination of parameters an RMSE of  $6.59 \cdot 10^{-4}$  was obtained from simulation, as opposed to  $3.07 \cdot 10^{-4}$  for (48,48,48,48)). The parameters denote the number of bits used for representing  $\mu_m$ , signals ( $x$ ,  $\tilde{x}$ ,  $y$ ), filter coefficients and  $\mu_{\text{step}}$  (also the precision used for computing  $\mu_m$ ), respectively. For this set of parameters the synthesis tool reports an area of approx.  $0.55 \text{ mm}^2$ . According to PowerMill the power consumption of the circuit averages 27 mW for  $T_{\text{tgt clk}}^1 = 50 \text{ MHz}$ , 3% of which account for the presented extension.

For particular values of  $\mu_{\text{step}}$  the period of the sequence  $\mu_m$  is much longer than the simulation time, so that not all effects may have been captured by our simulations in this case. The most important issue here appears to be the impact of the inaccuracy of computation of  $\mu_m$  on proper synchronization of read and write addressing, and consequently on the accuracy of output data.

## 5 CONCLUSIONS

We presented an extension to the Farrow structure enabling its robust operation. Based on simulation results we estimated minimum bit-widths required for high-quality SRC.

Following the study of the performance for longer input sequences, we are going to investigate ways of optimizing the circuit, in particular reducing power consumption, both in this fully parallel implementation and in alternative variants based on programmable processor cores.

## References

- [1] D. Babic, J. Vesma, T. Saramäki, and M. Renfors. Implementation of the transposed Farrow structure. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 2002.
- [2] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice Hall, 1983.
- [3] C. W. Farrow. A continuously variable digital delay element. In *Proc. IEEE Int. Symp. on Circuits and Systems*, 1988.
- [4] T. Hentschel and G. Fettweis. Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals. In *Proc. European Wireless Conference*, 2000.