Efficient Implementation of Interpolation Technique for Symbol Timing Recovery

CHIN-LONG WEY, SHIN-YO LIN, TSUNG-HAN TSAI, and MUH-TIAN SHIUE
Department of Electrical Engineering
National Central University
Chung-Li, Taiwan
URL: www.ee.ncu.edu.tw/~clwey

Abstract: - This study considers a non-synchronized sampling scheme for symbol timing recovery in DVB-T Transceiver design. The received signal is performed by a fixed sampling clock; the samples are not synchronized to the incoming data symbols. Timing adjustment is done after sampling using interpolation. The Lagrange interpolation for timing adjustment can be implemented FIR filter having changeable coefficients. This interpolation filter can be efficiently implemented using the **Farrow structure**. Take the advantage of symmetric property of Farrow Structure, the paper presents a low power/low-cost cubic Farrow structure, where the standard cells in TSMC .18um digial CMOS process were employed. Comparing the conventional Farrow structure, implementing with the same process, the developed structure achieves 36.8% lower power consumption and 24.3% low area cost than the conventional one.

Key-words: Interpolation technique, Cubic Interpolation, Farrow Structure, Symbol Timing Recovery, non-synchronization sampling scheme

1. Introduction

OFDM (*Orthogonal Frequency Division Multiplexing*) [1,2], a form of *multi-carrier modulation* technology [3-5], is a special case of multi-carrier transmission, where a single data stream is transmitted over a number of lower rate sub-carriers. OFDM technique has been widely implemented in high-speed digital communications to increase the robustness against frequency selective fading or narrow-band interface. It is also used for wideband data communications over mobile radio FM channels, xDSL (Digital Subscribe Lines), and DAB/DVB (Digital Audio/Video Broadcasting).

The DVB-T system consists of three major blocks: MPEG-2 source coding and multiplexing, channel coding, and modulation. In digital communication, binary information is converted by means of a modulator into a continuous-time signal which is sent over the transmission channel. A digital receiver is to extract the information sequence from a discrete signal obtained after sampling and quantizing the distorted signal presented to the demodulator. At the receiver, accurate timing recovery is critical to obtained performance close to that of the optimal receiver [6]. Timing in a data receiver must be synchronized to the symbols of the incoming data signals. The synchronization can be performed by using a feedback or feedforward loop which controls the phase of the sampling clock.

This study considers an alternative non-synchronized sampling scheme. The received signal is performed by a fixed sampling clock; the samples are not synchronized to the incoming data symbols. Timing adjustment is done after sampling using interpolation. Note that the sampling frequency does not have to be a multiple of the symbol frequency, it only has to be higher enough to avoid aliasing. There is no need for the complex PLL circuit to control the phase of the sampling clock. Figure 1 depicts our implementation [7].

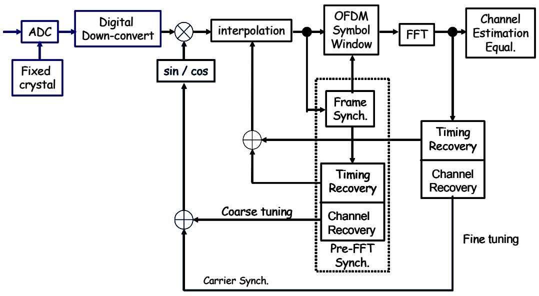

Figure 1. Non-Synchronization Sampling Scheme.

More specifically, Figure 2(a) shows the digital receiver with non-synchronized sampling [8-10]. The received signal x(t) is sampled by a fixed sampling clock. The function of the interpolation filter is to calculate one output sample $y(kT_i)$, T_i is the output interval, at a time using a set of adjacent input samples $x(mT_s)$, T_s is the sampling frequency, and the fractional interval (or timing error estimate) μ which is obtained from the control unit. Figure 2(b) illustrates the input and output samples of the interpolation filter. Given the original continuous-time signal x(t), the black dots indicate the sampled points, or the input samples of the interpolation filter, while the white dots are the interpolated points, or the output samples, determined by the adjacent sampled points and the fractional interval μ. Our goal is to develop an interpolation technique and its architecture that efficiently determines the output samples from the input samples and the fractional interval μ.

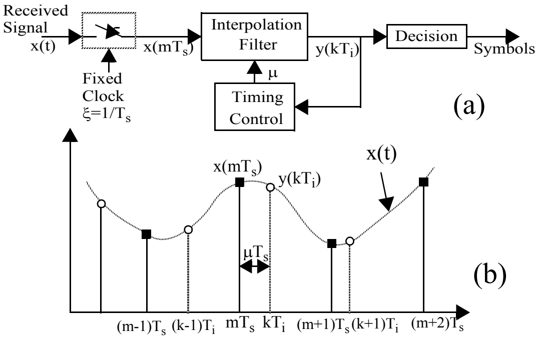

Figure 2. (a) Non-Synchronization Sampling Structure; and (b)Input/output Samples of Interpolation Filter.

The Lagrange interpolation for timing adjustment has been proposed in [11,12]. The polynomial-based interpolation filter can be implemented finite impulse response (FIR) filter having changeable coefficients. The values of the coefficients are determined by the fractional interval μ. The filter coefficients can be calculated on-line by simply evaluating the values of the polynomials for the given value of μ. This filter can be efficiently implemented using the **Farrow structure**.

Interestingly, the filter coefficients have the following property because the interpolation filter is symmetric, (will be discussed shortly)

$$C_i(\mu)=C_{-(i+1)}(1-\mu),$$
$$C_i(1-\mu)=C_{-(i+1)}(\mu);\quad i=0,1,..L. (1)$$

where N=2L+1 is the degree of the Lagrange polynomial. For the cubic Lagrange interpolation filter, N=3 and L=1, $C_0(\mu)=C_{-1}(1-\mu)$ and $C_1(\mu)= C_{-2}(1-\mu)$. Similarly, $C_0(1-\mu)=C_{-1}[1-(1-\mu)]=C_{-1}(\mu)$, and $C_1(1-\mu)=C_{-2}(\mu)$.

Taking the advantage of the symmetric relation in (1), the conventional Farrow structure is modified in this study. A significant reduction in hardware cost is resulted in this study.

In the next section, the basic design and implementation of the Farrow structure are discussed. Section 3 presents the developed low-power/low-cost Farrow structure for cubic interpolation. Finally, a concluding remark is given in Section 4.

2. Farrow Structure

As shown in Figure 2(b), the time-continuous output of the interpolation filter can be represented as

$$y(kT_i) = y[(m_k+\mu_k)T_s]$$
$$= \sum_{i=I_1}^{I_2} x[(m_k - i)T_s]h_I[(i+\mu_k))T_s] \quad (2)$$

where {x(m)} is a sequence of signal samples taken at intervals T_s, and $h_I(t)$ is the finite-duration impulse response of a fictitious, time-continuous, analog interpolating filter. Eqn. (2) derives interpolants y(k) at adjustable intervals T_i; T_i is in general incommensurate with T_s.

Classical polynomial interpolation of an N-point basepoint set can be performed by the Lagrange formulas

$$y(t) = \sum_{i=I_1}^{I_2} C_i x(I_1+I_2-i) \quad (3)$$

where

$$C_i = \prod_{j=I_1, j\neq i}^{I_2} (t-t_j)/(t_i-t_j)$$

For even value N, we set I_1=-N/2 and I_2=N/2-1, respectively. With this normalization, t = $(i+\mu)T_s$ and

$$h_I[(i+\mu)T_s]=C_i(\mu) \quad (4)$$

For N=4, i.e., cubic interpolation, the output sample $y(kT_i)$ is expressed as

$$y(kT_i)(\mu) = x(m_k+2)C_{-2}(\mu) + x(m_k+1)C_{-1}(\mu) +$$
$$x(m_k)C_{-0}(\mu) + x(m_k-1)C_1(\mu) \quad (5)$$

where

$$C_{-2}(\mu)=(\mu+1)\mu(\mu-1)/6=(1/6)\mu^3+(-1/6)\mu$$
$$C_{-1}(\mu)=(-1/2)\mu^3+(1/2)\mu^2+\mu$$
$$C_0(\mu)=(1/2)\mu^3+(-1)\mu^2+(-1/2)\mu+1 \quad (6)$$
$$C_1(\mu)=(-1/6)\mu^3+(1/2)\mu^2+(-1/3)\mu$$

Figure 3(a) shows the Farrow structure of the cubic interpolation which implements the above coefficient directly.

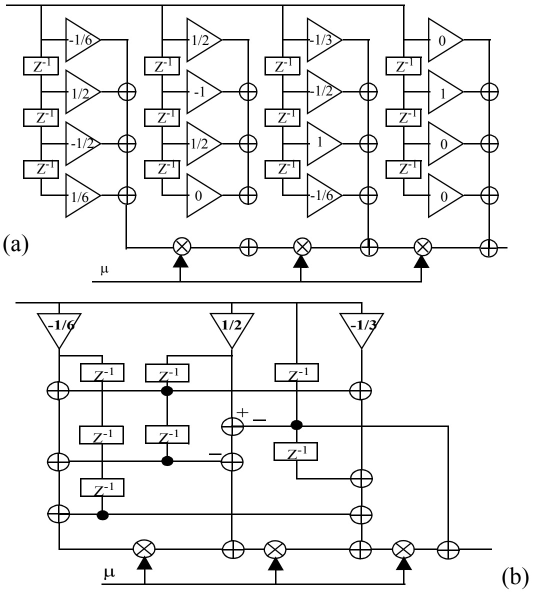

Figure 3: Farrow Structures for Cubic Interpolation: (a) Direct Implementation; and (b) Simplified.

The Farrow structure can be simplified by removing those cells with multiplying by the counts 0, 1, or -1. The structure also can be simplified by sharing the results of the scalar multiplications, as shown in Figure 3(b), where the structure requires 7 Delay cells, 3 scalar multiplications, 3 multiplications, and 11 additions/subtractions. The structure can be further improved in the following section.

3. Development

Figure 4(a) was derived directly from Figure 3(a) by disconnecting the cells with 0 coefficient and directly connecting the cells with 1 coefficient. Figure 4(a) can be further simplified as follows. Consider the Eqn. (6) for the coefficients of cubic interpolation, where

$$C_{-2}(\mu)=(\mu+1)\mu(\mu-1)/6=(1/6)\mu^3+(-1/6)\mu$$

$$C_{-1}(\mu)=(-1/2)\mu^3+(1/2)\mu^2+\mu$$

$$C_0(\mu)=C_{-1}(1-\mu)=(-1/2)(1-\mu)^3+(1/2)(1-\mu)^2+(1-\mu)$$

$$C_1(\mu)=C_{-2}(1-\mu)=(1/6)(1-\mu)^3+(-1/6)(1-\mu)$$

This verifies what discussed in Eqn. (1) due to the symmetric of the interpolation filter. Thus, the upper half of the Farrow structure in Figure 4(a) can be replaced by its low half with the input of (1-μ), as shown in Figure 4(b).

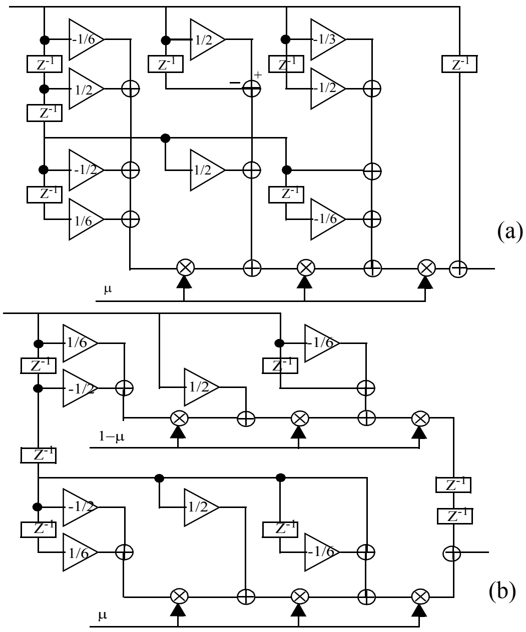

Figure 4. Modification Process.

In order to show the operating sequence, Table 1 lists the time sequence of receiving the input sample points.

Table 1.
Timing Sequence of Cubit Farrow Structure

m_k	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	-	1	2	3	4	5	6	7	8
3	-	-	1	2	3	4	5	6	7
4	-	-	-	1	2	3	4	5	6
w_1	-	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}	s_{91}
w_2	-	-	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}	s_{81}
w_3	-	-	-	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}
w_4	-	-	-	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}
out	-	-	-	s_2	s_3	s_4	s_5	s_6	s_7

The first row depicts the clock sequence, while the upper part of the first column presents the four points to be considered for the interpolation. The corresponding entries are the input sample points applied to the cubic Farrow structure. Finally, the symbols in the lower part of the first column are the labels marked in Figure 6. Note that s_{j1}, j=2,..,9, is the computation result of the upper part in Figure 5(b), while s_{j2} is that of the lower part. The result at w2 is one clock delay of that at w1, while the result at w3 is one clock delay of that at w2. Both results at w3 and w4 are then summed as the final result s_j, or $s_j=s_{j1}+s_{j2}$.

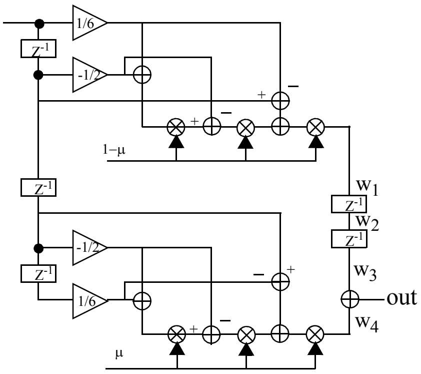

Figure 5. Modification.

Since both halves of the Farrow structure in Figure 5 are symmetric, but with two different inputs μ and (1-μ), respectively, it can be implemented using the multiplexer to select different value of μ and (1-μ), as shown in Figure 6.

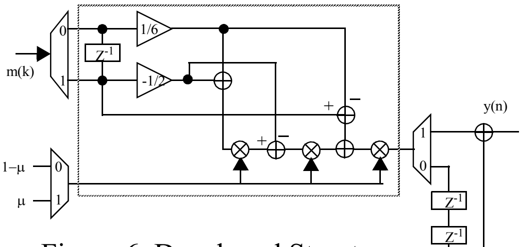

Figure 6. Developed Structure.

A multiplexer (MUX) circuit was used in Figure 6 for switching the input to either (1-μ) for the upper part, or μ for the lower part. One demultiplexer (DMUX) circuit was used for switching the input to either the upper part or the lower part. More specifically, the first two clock cycles, the DMUX is configured in such a way that the first two input sampled points are entered, respectively. The results are calculated with the input (1-μ), and held in the delay cells (in Figure 6), where another DMUX selects this mode. On the other hand, the next two clock cycles, the DMUX circuit selects to calculates the next two input sample points with the input μ, and the final result is then summed with that generated by the last two clock cycles. The clock sequence is exactly the same as that in Table 1. Thus, the hardware required for this structure is one scalar division, one shift, five additions/subtractions, three multiplication, three delay cells, one MUX, and two DMUX.

Results show that the structure requires 2 Delay cells, 2 scalar multiplications, 3 multiplications, and 5 additions/subtractions. It also requires one multiplexer and two demultiplexers. Note that the scalar multiplication with (1/2) is nothing but a shift. Thus, the developed structure is much simpler than the conventional Farrow structures for cubit interpolation.

The developed (Figure 6) structures has been implemented and synthesized with SynopsysDV using the standard cells in TSMC 0.18μm digital CMOS process. The performance of dan interpolation filter is measured by its signal to sampling noise ratio (SSNR), which is defined as

$$SSNR = E\{y(kT)^2\}/(E\{y(kT) - y_{rec}(kT))^2\})$$

where y(kT) is the ideal sample and y_{rec}(kT) is the reconstructed sample at correct sample time. In this simulation,

$u = \mu_k = 8/2^4 = 0.5$

$\{y(kT)\} = 2.5;\ y_{rec}(kT)=2+2032/2^{12} = 2.49609$

$\text{SSNR} = 10 * \log((2.5)^2 / (2.5-2.49609)^2) = 56$ db

Table 2 compares the simulation results of both conventional (Figure 3b) (which has been simplified, but without using the symmetry property) and the developed (Figure 6) structure (with symmetry property).

Table 2. Comparison of Hardware Implementation.

	Conventional	Developed	Improvement
Number of ports	288	177	
Number of nets	712	1059	
Number of cells	53	18	
Number of references	30	18	
Net interconnect area	1125	974	13.4%
Total cell area	192498	145706	24.3%
Total area (um^2)	193621	146678	24.3%
Power Consumpton (mW)	421.5	258.9	38.6%

Result shows that both structure have the same speed performance and SSNR. However, the developed structure takes a total area of 146,678 um^2 and a power consumption of approximately 258.9 mW, where the standard cell multipliers were used and the scaled multiplier/divider (or reciprocal multiplier) was efficiently implemented using shifters and adders. On the other hand, with the same design style and implementation, the structure in Figure 4c takes 193,621 um^2, and consume nearly 421.5 mW in power. In other words, the developed structure with symmetry property consume approximately 38.6% less power than the conventional one and also requires about 24.3% less area.

4. Conclusions

This study considers a non-synchronized sampling scheme. The received signal is performed by a fixed sampling clock; the samples are not synchronized to the incoming data symbols. Timing adjustment is done after sampling using interpolation. The Lagrange interpolation for timing adjustment can be implemented FIR filter having changeable coefficients. This interpolation filter can be efficiently implemented using the Farrow structure.

This paper presents the efficient implementation of the Farrow structure for cubit interpolation. Results show that the hardware implementation of the developed structure requires less hardware and power consumption than the conventional structure for cubic interpolation. It is possible to develop the synthesis process that automatic generate the efficient structures. This leads to a interesting problem for future study.

Acknowledgment:
This work was supported in part by the National Science Council, Taiwan, under the grant numbers NSC94-2220-E-008-001, NSC94-2220-E-008-005, NSC94-2220-E-008-008.

References

1. R. van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, 2000.
2. W.-Y. Zou and Y. Wu, "COFDM: an Overview," IEEE Trans. on Broadcasting, pp.1-8, March 1995.
3. I. Kalet, "The Multiton Channel," IEEE Trans. on Communications, pp.119-124, Feb. 1989.
4. P. S. Chow, J.C. Tu, and J. Cioffi, "Performance Evaluation of a Multichannel Transceiver System for ADSL and VHDSL Services," IEEE Journal on Communications, pp.909-919, August 1991.
5. N.Al-Dhahir, and J.M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers," IEEE Trans. on Communications, pp. 56-64, January 1996.
6. T. Pollet and M. Peeters, "Synchronization with DMT Modulation," IEEE Communication Magazine, pp.80-86, April 1999.
7. C.L. Wey, "SoC Design of Digital Video Broadcasting Receiver and Its Platform Development," Technical Report to Research Project of National Science Council, Taiwan, May 2005.

8 V. Tuukkanen, J. Vesma, and M. Renfors, “Combined Interpolation and Maximum Likelihood Symbol Timing Recovery in Digital Receivers,” Proc. IEEE International Conference on Universal Personal Communication, pp.698-702, Oct. 1997.

9. J. Vesma, M. Renfors, and J. Rinne, “Comparison of Efficient Interpolation Techniques for Symbol Timing Recovery,” Proc. IEEE Globecom 96, London, UK, pp.953-957, Nov, 1996
10. A.S.H. Ghadam and M. Renfors, “Farrow Structure Interpolators Based on Even Order Shaped Lagrange Polynomial,” Proc. Int’l Symp. on Image and Signal Processing and Analysis, pp.745-748, Sept. 2003
11. F.M. Gardner, “Interpolation in Digital Modems -- Part I: Fundamentals,” IEEE Trans. on Communications, pp.501-507, March 1993.
12. L. Eruo, F.M. Gardner, and R.A. Harris, “Interpolation in Digital Modems -- Part II: Implementation and Performance,” IEEE Trans. on Communications, pp.998-1008, June 1993.