

# Computational efficient Fractional Sampling-Rate Conversion architectures for the Tied Array Distribution Unit of the Westerbork Synthesis Radio Telescope

Ronald de Wild, Jerome Dromer, Arie Doorduyn

**Abstract**—In the Westerbork Synthesis Radio Telescope (WSRT), one of the key functions in the Tied Array Distribution Unit (TADU) is the conversion of the sampling frequency of the signal that originates from the radio telescope system into sampling frequencies that meet the requirements of various astronomical user groups. For the sampling-rate conversion (SRC) function, two candidate concepts have been analyzed and compared with respect to circuit-size, processing clock-rate, scalability and numerical accuracy: the Time-Variant Filter (TVF) and the Poly-Phase Filter (PPF). A structured top-down design methodology that makes use of the hardware description language VHDL has led to a fully synthesizable, technology-independent model for both candidates. The scaling parameters enable re-configurability for a wide range of sampling-rate conversion ratios. With respect to circuit-size, processing clock-rate and scalability, the TVF is the better candidate. With respect to numerical accuracy, the more conventional PPF is to be preferred.

**Index Terms**—radio astronomy, multi-rate signal processing, system architecting, hardware optimization.

## I. INTRODUCTION

THE Tied Array Distribution Unit (TADU) is the most recent hardware upgrade of the Westerbork Synthesis Radio Telescope (WSRT). It provides a new interfacing system between the existing telescope-specific receiver and the astronomy application-specific correlators, existing as well as new [6],[7]. For the specification of the TADU-subsystems, the digital functional block that has been identified for frequency conversion is the so-called “TADU VLBI Re-sampler” (TVR), where VLBI stands for Very Long Baseline Interferometer. One of the TVR-functions is sampling-rate conversion. Signals sampled at a rate of 40 MHz are converted into signals sampled at a rate of 32 MHz, 16 MHz, 8 MHz, 4 MHz, 2 MHz, 1 MHz, or 0.5 MHz - with a minimum amount of non-linear distortion and with a minimum loss of information. The other TVR-functions are

Ronald de Wild ( [wild@astron.nl](mailto:wild@astron.nl) ), Jerome Dromer ( [dromer@astron.nl](mailto:dromer@astron.nl) ) and Arie Doorduyn ( [doorduyn@astron.nl](mailto:doorduyn@astron.nl) ) are with the Netherlands Foundation for Radio Astronomy (ASTRON), P.O.Box 2, 7990AA, Dwingeloo, The Netherlands; tel 0031 521 595100, URL [www.astron.nl](http://www.astron.nl) .

power-scaling and re-quantization. Here, the momentary power-level of the celestial signal determines the functionality [9]. This paper focuses on the Sampling-Rate Conversion (SRC) function of the TVR. For this function, two candidate concepts have been analyzed and compared with respect to circuit-size, processing clock-rate, scalability and numerical accuracy: the Time Variant Filter (TVF) and the Poly-Phase Filter (PPF).

This paper is structured as follows. In section II, we review some background theory on fractional re-sampling. Next, we sketch the design space that encompasses computational efficient SRC-architectures. In section III, we present two candidate architectures, the “Poly-Phase Filter” (PPF) and the “Time-Variant Filter” (TVF). Their functional properties are examined with respect to computational accuracy and scalability. For the PPF, the presence of a high interpolation sampling clock-rate is a condition for the preservation of full scalability. For the TVF, we avoid the presence of a high interpolation sampling clock-rate by applying direct re-sampling. By doing so, we also introduce non-linear distortion and loss of information. In section IV, we match the TADU-specific filter-specifications to the technology-bounds rendered by a pre-selected FPGA-device. In section V, we capture the scaling properties for the PPF in a table as a set of parameters. The same is done for the TVF. Each set of parameters corresponds to a hardware-platform that maximally fills the same TADU-specific board. In section VI, we conclude with a comparison of both architectures.

## II. FRACTIONAL RE-SAMPLING

### A. Background Theory

A digital frequency conversion system is based on the concept of fractional re-sampling. It is a SRC-system with rational factor  $I/D$ , where the interpolation factor  $I$  and the decimation factor  $D$  are both integers. See Fig. 1 for the concept architecture. The anti-imaging filter and the anti-aliasing filter are merged into a single low-pass filter. For this filter, we apply a Finite Impulse Response (FIR-) structure of order  $M$ .

A FIR-filter structure lends itself well to poly-phase decomposition, for which several cases exist, depending on

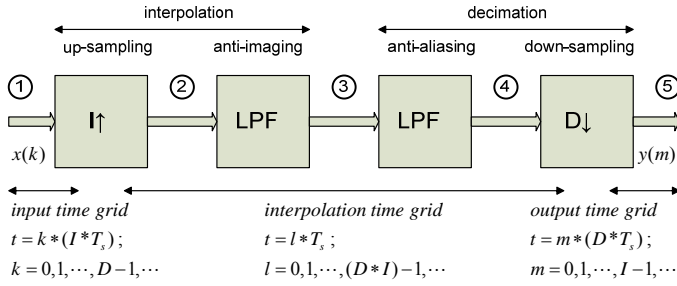


Fig. 1. Fractional re-sampling concept: block diagram with the combination of interpolation and decimation. LPF = Low-Pass Filter. For integer ratio I/D, the architecture reduces to interpolation-only; for integer ratio D/I, the architecture reduces to decimation-only.

stage	signal description	sample-index	sampling interval [s]	
			(Fig.1.)	(Fig.2.)
1	original signal	k	$I * T_s$	$T_s$
2	up-sampling (zero-insertion)	l	$T_s$	$T_s/I$
3	smoothing after up-sampling	l	$T_s$	$T_s/I$
4	smoothing before down-sampling	l	$T_s$	$T_s/I$
5	down-sampling (sample-deletion)	m	$D * T_s$	$D * T_s/I$

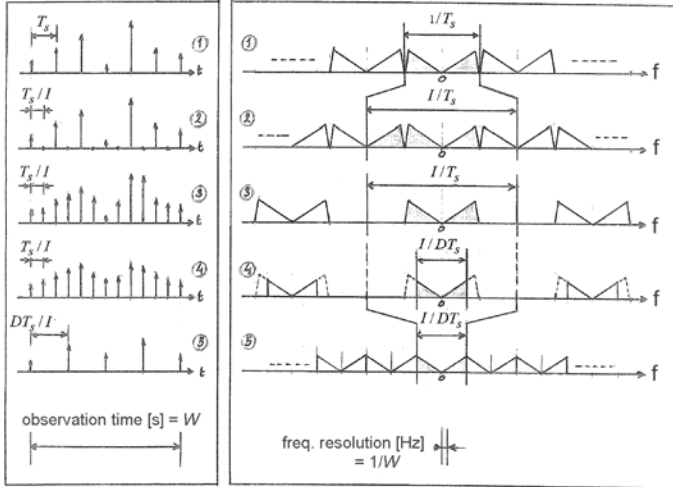


Fig. 2. Fractional re-sampling concept: time-domain and frequency domain signal processing steps. Example for I= 2 and D= 3. Note at stage 2, that time-domain insertion of zero-samples corresponds to replication of the frequency-domain spectrum.

the primary number factorization of M. The common-case decomposition splits the FIR-filter structure into a set of I parallel operating sub-filters, each of order M/I (see coefficients in Fig. 3). Its dual-case decomposition splits the FIR-filter structure into a set of M/I parallel operating sub-filters, each of order I (see coefficients in figure 4). Each sub-filter, on its turn, can be defined as a FIR-structure. Note, that by swapping the coefficient-index and the sub-filter-index, the common-case decomposition is transformed into dual-case decomposition, and vice versa. Also note two extreme cases: the common-case with a single “sub-filter” of order M - i.e. a single “horizontal” transversal structure - and its dual-case with M parallel multiplications - i.e. a single “vertical”

transversal structure.

The transversal structure of a FIR-filter exists in two forms [1], [2]. In the direct form, phase-shifted versions of the stimulus are processed by the corresponding sub-filters, before parallel addition (Fig. 3). In the transposed form, the same stimulus is processed in parallel by the sub-filters, before phase-shifted cumulative addition (Fig. 4). In Fig. 3 and Fig. 4, the impulse-response of the original combined anti-imaging/anti-aliasing filter is denoted by “h”, the impulse-responses of the sub-filters are denoted by “g”.

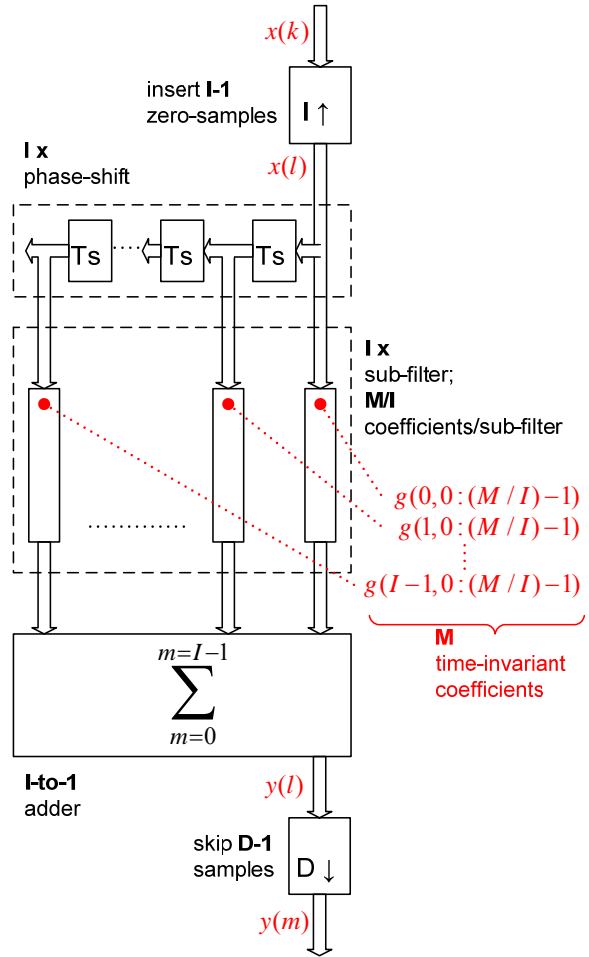


Fig. 3. Poly-phase architecture that adopts the direct-form structure. The four functional stages are: up-sampling, tapped delay-line, concurrent sub-filtering, parallel addition, down-sampling. Coefficient-indexing (according to common-case decomposition):

$$g(i_1, i_2) = h(i_2 * I + i_1)$$

with sub-filter no.  $i_1 = 0, 1, \dots, I-1$  and coefficient no.  $i_2 = 0, 1, \dots, (M/I)-1$ .

### B. Computational Efficiency

The poly-phase architectures of figures 3 and 4 are not optimized with respect to computational efficiency, i.e. circuit-size and power consumption.

The circuit-size is largely determined by the number of multipliers (and, to less degree, the number of delay-elements). A very effective way to reduce circuit-size is to apply the concept of *time-varying coefficients* [3],[4],[5]. In

the poly-phase decomposition (Fig. 3), each of the  $I$  sub-filters performs  $M/I$  multiplications with static coefficients as operands. We can suffice with 1 multiplier per sub-filter, if we perform cyclic replacement of the operand with  $M/I$  coefficients instead. The number of multipliers reduces with a factor  $I/M$  to  $I$ . This goes at the cost of some additional control circuitry and increased switching activity. The biggest disadvantage, however, is that the impulse-response becomes non-unique: each sub-filter changes its functional behaviour every clock cycle. This results in a set of  $M/I$  distinct impulse responses. Results from linear causal time-invariant system theory - including the Fourier transform - have limited validity. Therefore, simulation results of time-invariant systems should be interpreted with caution. When applying *R-to-1 time-multiplexing* on the signal processing path, the number of multipliers reduces to  $I/R$  with another factor  $R$ , but then at the same time the chip-clock (i.e. the fastest running clock in the SRC-system) increases with a factor  $R$ .

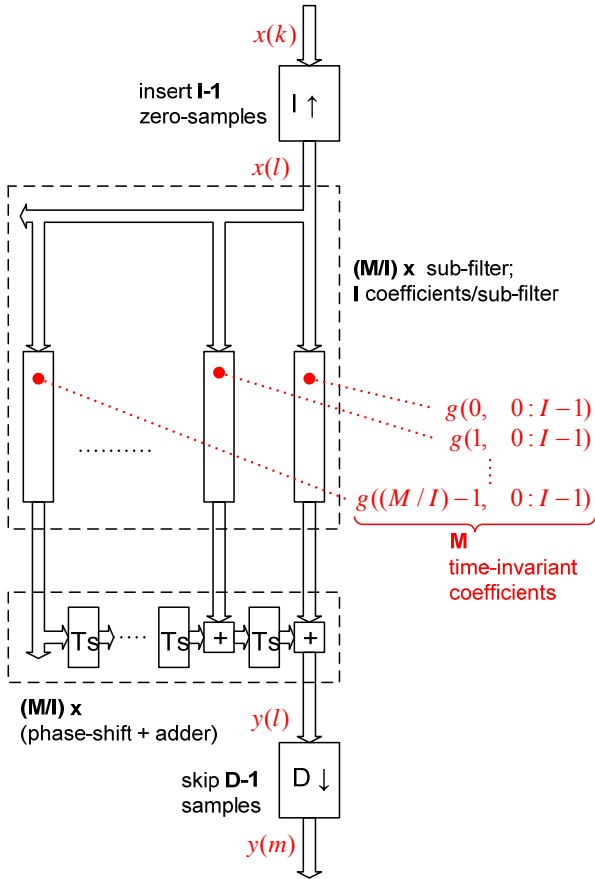


Fig. 4. Poly-phase architecture that adopts the transposed-form structure. The five functional stages are: up-sampling, tapped delay-line, concurrent sub-filtering, cumulative phase-shifted addition, down-sampling. Coefficient-indexing (according to dual-case decomposition):

$$g(i_1, i_2) = h(i_2 \cdot I + i_1)$$

with sub-filter no.  $i_1 = 0, 1, \dots, (M/I)-1$  and coefficient no.  $i_2 = 0, 1, \dots, I-1$ .

Dynamic power consumption is determined in the first place by the interpolation sampling clock-rate. Especially when the ratio  $I/D$  approaches 1 (corresponding to large and

mutually prime values for  $I$  and  $D$ ), the interpolation sampling clock-rate can become unacceptably high. The first stage, up-sampling, is equivalent with *zero-value interpolation* (signal-stage 2 in Fig. 2). Zero-value interpolation enables the identification of redundant operations (on zero-valued signal-samples). For these operations we can avoid the use of the interpolation sampling clock, which leads to a reduction in power consumption. In the frequency domain, the original spectrum is replicated " $I$ " times: up-sampling produces image-components at equal amplitudes.

If we apply  $0^{\text{th}}$ -order interpolation or  $1^{\text{st}}$ -order interpolation instead of zero-value interpolation, we lose the option of redundant operations, but in return we get additional image-suppression that precedes and thus relaxes the requirements for the FIR-filter stage.  $0^{\text{th}}$ -order (*constant-level, rectangular*) interpolation [3] creates an oversampled step-function approximation of the input-signal. In the frequency-domain, the image-components decrease according to a sinc envelope.  $1^{\text{st}}$ -order (*linear, triangular*) interpolation [5] creates an oversampled trapezoid-function approximation of the input-signal. In the frequency-domain, the image-components decrease according to squared-sinc envelope. Note that only oversampling, i.e. "*exact*" (*sinc-shape*) interpolation, creates a perfect replica of the input signal. Here, in the frequency-domain, the image-components have disappeared, i.e. all duplicated spectral periods have become zero.

### III. COMPUTATIONAL ACCURACY AND SCALABILITY

By re-arranging the structure described in the direct form of the architecture in Fig. 3, we arrive at two computational-efficient architectures: the "Time-Variant Filter" (TVF), where we have applied zero-value interpolation and time-varying coefficients, and the "Poly-Phase Filter" (PPF), where we have applied constant-value interpolation and time-invariant coefficients. For both architectures, we shall evaluate computational accuracy and scalability with respect to computational efficiency.

#### A. Poly-Phase Filter (PPF) Architecture Decisions

The PPF-architecture is directly derived from the structure of Fig. 3. In fact, it represents the extreme case of  $M$  concurrent sub-filters, with each sub-filter degenerated to a single multiplier. With respect to Fig. 3, for the PPF the coefficient-indexing degenerates to:

$$g(i_1, i_2) = h(i_2 \cdot M + i_1),$$

with sub-filter no.  $i_1 = 0, 1, \dots, M-1$  and coefficient no.  $i_2 = 0$ . For the up-sampling stage we have substituted the *sample-repetition stage*. Remark that all PPF-stages operate at the *interpolation clock-rate*.

The PPF-datapath consists of five functional stages (Fig. 3): At the Sample-Repetition stage, the 0-th order oversampling process derives a step-wise approximated version from the input-signal. This stage is implemented as a buffer, with input clocked by  $f_{s,\text{in}}$  and output clocked by  $f_{s,\text{intp}}$ . At the Tapped Delay-Line stage, clocked at  $f_{s,\text{intp}}$ ,  $M$  concurrent phase-shifted versions are created out of the incoming data-stream. At the Concurrent Multiplication stage,  $M$  parallel multipliers

with time-invariant coefficients generate the  $M$  sub-filter outputs. At the Parallel Addition stage, the  $M$  concurrent products are added via a Wallace Tree structure. At the Down-Sampling stage, the function for skipping  $D-1$  samples is implemented as a simple delay-element, clocked by  $f_{s,out}$ .

For the PPF, we have decided to *preserve scalability at the cost of computational efficiency*. In order to gain computational efficiency, we should start from the transposed form (Fig. 4). By applying the noble identities we arrive at a parallel structure, where the pairs of up-samplers and down-samplers are only separated by linearly increasing time-delays [4]. As long as up-sampler and down-sampler have not swapped position, i.e. as long as the interpolation sampling-clock (between up-sampler and down-sampler) remains present, scalability is preserved. As soon as up-sampler and down-sampler have swapped position, the interpolation sampling clock disappears. At the same time, the time-delays are split according to primary number decomposition. This split implies loss of scalability.

### B. Time-Variant Filter (TVF) Architecture Decisions

The TVF-architecture is derived from the structure of Fig. 3 by re-arranging the up-sampling and down-sampling functional blocks through the structure, according to the noble identities. Because the up-sampling and down-sampling stages have merged into the direct re-sampling stage, *the interpolation time-grid has disappeared* [3].

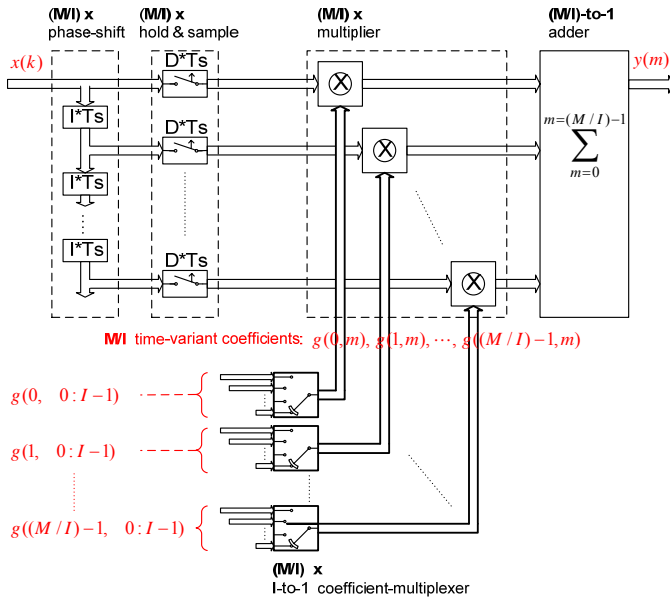


Fig. 5. The “Time-Variant Filter” (TVF) architecture applies direct re-sampling and consists of four functional stages: tapped delay-line, direct re-sampling, concurrent multiplication, parallel addition.

Coefficient-indexing:

$$g(k, m) = h(k.I + [m.D]_{MOD(I)})$$

with sub-filter no.  $k = 0, 1, \dots, (M/I)-1$  and coefficient no.  $m = 0, 1, \dots, I-1$ .

The TVF-datapath consists of four functional stages (Fig. 5). At the Tapped Delay-Line stage, at clock-rate  $f_{s,in}$ ,  $M/I$  concurrent phase-shifted versions are created out of the

incoming data-stream. At the Direct Re-sampling stage, a Hold & Sample (H&S-) process maps the  $M/I$  concurrent data-streams directly from input time-grid onto output time-grid, without any intervening interpolation time-grid. At the Concurrent Multiplication stage,  $M/I$  parallel multipliers, with  $I$  time-variant coefficients per multiplier, generate the  $M/I$  sub-filter outputs. At the Parallel Addition stage, the  $M/I$  concurrent products are added via a Wallace Tree structure.

The H&S-process performs *direct re-sampling*, i.e. direct mapping of the signal-samples from the input time-grid onto the output time-grid. As a consequence, the interpolation clock is no longer present, which implies a reduction in power consumption. (Actually, the interpolation clock *is* present implicitly: in the fixated phase-relationship between  $f_{s,in}$  and  $f_{s,out}$ .) Furthermore, the TVF is a time-invariant structure with *time-varying coefficients*. Each sub-filter is represented by  $I$  instead of  $I$  multiplications, where for the multiplicand the  $I$  filter-coefficients are cyclically substituted. Thus, the required number of multipliers is reduced with a factor  $I$  (from  $M$  to  $M/I$ ), which implies a reduction in the amount of circuitry.

The TVF-functionality can be captured in a single formula, which is in fact a convolution sum [3]:

$$y(m) = \sum_{k=0}^{(M/I)-1} h(kI + [mD]_{MOD(I)}) * x(\lfloor mD/I \rfloor - k)$$

Here,  $x(\lfloor mD/I \rfloor - k)$  represents the  $k$ -th sample, based on the input time-grid, displaced to the  $m$ -th position of the output time-grid. The effect of the “ $\lfloor \cdot \rfloor$ ”-operation is rounding to the next higher integer. The expression “ $h(kI + [mD]_{MOD(I)})$ ” represents the  $m$ -th coefficient of the  $k$ -th sub-filter.

We illustrate this formula with an example. Let’s take  $I = 8$ ,  $D = 5$  and  $M = 32$ , then the  $M = 32$  filter-coefficients are partitioned in  $32/8 = 4$  sub-sets (indices  $k = 0, 1, 2, 3$ ), each sub-set consisting of 8 filter-coefficients (indices  $m = 0, 1, \dots, 7$ ). The coefficients  $g(k, m) = h(8k + [5m]_{MOD(8)})$  are now partitioned as follows:

sub-filter no.	sub-set of 8 filter-coefficients
$k = 0$	$h(0) h(5) h(2) h(7) h(4) h(1) h(6) h(3)$
$k = 1$	$h(8) h(13) h(10) h(15) h(12) h(9) h(14) h(11)$
$k = 3$	$h(16) h(21) h(18) h(23) h(20) h(17) h(22) h(19)$
$k = 4$	$h(24) h(29) h(26) h(31) h(28) h(25) h(30) h(27)$

Each column represents the 4 filter-coefficients that appear as operands of the 4 parallel multipliers at a specific output-sampling moment.

For the TVF, we have to tolerate a *limitation in computational accuracy*. The Hold & Sample stage performs direct re-sampling: the input-sample is held (“repeated”) until the next edge of the output-sampling clock. Thus, the Hold & Sample stage performs 0<sup>th</sup>-order interpolation (at a virtual, non-present interpolation sampling clock), immediately followed by down-sampling. This process causes *non-linear distortion*. The phenomenon can already be observed when simulating the initial TVF-architecture at system level. Appendix A describes how the non-linear distortion pattern can be predicted. The Parallel Addition stage is able to

“smooth” the resulting waveform and thus reduces the level of the sub-harmonics to a certain degree - but the sub-harmonics will not disappear. We have to accept a certain level of non-linear distortion.

#### IV. FILTER SPECIFICATION AND IMPLEMENTATION

##### A. Technology Bounds

The number of available hardware-multipliers (to be realized in pre-located DSP-blocks) and the maximum rating for the chip-clock provide specific bounds for implementation on Altera’s field-programmable gate-array (FPGA) technology. In order to enable the portability of the pre-manufactured TOM-board (see figure 6), we select Altera-devices with identical footprints, i.e. 1020-pin FBGA (fine-line ball-grid array). See Table 1 [11]. Taking the worst-case device (EPS1S60) as a reference and

Table 1. FPGA device-specification of the number of hardware-multipliers

Altera device technology	Stratix - EP1S60 150 nm CMOS 1.5 V	Stratix II – EP2S60 90 nm CMOS 1.2 V
amount & bit-size of embedded multipliers	(18 DSP-blocks →) 72 18 bit-based 144 9 bit-based	(36 DSP-blocks →) 36 36 bit-based 144 18 bit-based 288 9 bit-based

assuming a reliable upper bound for the chip-clock of about 200 MHz, we specify for the PPF (Table 2):

128 9-bit-based multipliers, operating at 160 MHz,  
and for the TVF (Table 3):

64 18-bit-based multipliers, operating at max.128 MHz.

##### B. Filter Specification



Fig. 6. Implementation platform: the TOM-board (Timing Output Module), with 8 FPGAs from Altera’s Stratix-family

The combined anti-imaging/anti-aliasing filter is specified for  $I = 4$  and  $D = 5$  as a low-pass FIR-filter with an equiripple characteristic. For other values of  $I$  and  $D$  we apply this frequency characteristic as a reference with re-scaled frequency-axis (Fig. 7).

For the TADU-system, the FIR-filter has to fulfill the following requirements:

$r_{pb} = 10^{-0.001}$  (i.e. -0.01 dB) and  $r_{sb} = 10^{-4}$  (i.e. -40 dB). Since  $r_{pb} \ll 1$ , the required dynamic range is approximately equal to  $1/r_{sb}$ . With  $n$  bits, we are able to cover a dynamic range of  $2^{2n}$ .

Thus, by solving the equation “ $1/r_{sb} = 2^{2n}$ ”, we arrive at about 7 bits per filter-coefficient required. Taking into account the hardware-bound for the multiplier bit-size (Table 1), we have to specify at least a DSP-block configuration with *9 bit based multipliers*.

For the TADU-specification, where  $I < D$  (Table 2 and

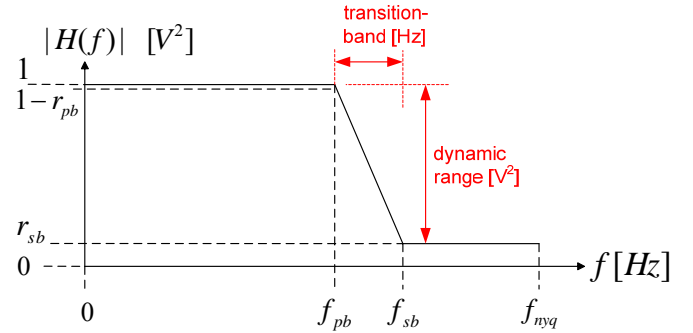


Fig. 7. Frequency characteristic specification:

- $r_{pb}$  = pass-band ripple
- $r_{sb}$  = stop-band attenuation (= quantization noise level)
- $f_{pb}$  = netto bandwidth (= effective bandwidth)
- $f_{sb}$  = bruto bandwidth
- $f_{nyq}$  = nyquist-frequency
- $M$  = filter-order = number of coefficients
- $n$  = number of bits per coefficient
- dynamic range =  $10 * \log(1/r_{sb})$  [dB] (=  $2^{2n}$ )
- slope =  $10 * \log(1/r_{sb}) / (f_{sb} - f_{pb})$  [dB/Hz] (proportional to  $M$ )

Table 3), we can state that  $f_{sb} = f_{s,out}/2 = (I/D) * f_{s,in}/2$ . With  $f_{s,in} = 40$  MHz and  $(I,D) = (4,5)$  the bruto bandwidth becomes:  $f_{sb} = 16$  MHz. The netto bandwidth (the effective bandwidth) should approach the bruto bandwidth as close as possible. If we specify a “channel occupation-efficiency” of  $f_{pb} / f_{sb} = 90\%$ , we get a netto bandwidth of:  $f_{pb} = 14.4$  MHz.

The stop-band runs between  $f_{nyq}$  and  $f_{sb}$ . For the PPF, where the filter operates at the interpolation sampling clock, we get:  $f_{nyq} = f_{s,intp}/2 = I * f_{s,in}/2 = 80$  MHz.

For the TVF, where the filter operates at the output sampling clock, we get:  $f_{nyq} = f_{sb} = 16$  MHz. (In this case, no stop-band exists: the bruto bandwidth occupies the entire spectral period.)

The transition-band runs between  $f_{sb}$  and  $f_{pb}$ . For an order- $M$  FIR-filter, the transition-slope (the “roll-off” factor), expressed in dB/Hz, is proportional to  $M$ :

$$10 * \log(1/r_{sb}) / (f_{sb} - f_{pb}) = \text{constant} * M$$

If on the one hand we keep  $f_{sb}$  and  $r_{sb}$  fixed, this expression defines a 1:1-relationship between  $f_{pb}$  and  $M$ : the larger  $M$  becomes, the more  $f_{pb}$  approaches  $f_{sb} = 16$  MHz, i.e. the larger the “channel occupation-efficiency” becomes. If on the other hand we keep a fixed transition band (i.e.  $f_{sb}$  and  $f_{pb}$  fixed), this expression defines a 1:1-relationship between  $r_{sb}$  and  $M$ : a larger  $M$  corresponds to a larger dynamic range, i.e. a larger number of bits.

For both the PPF and the TVF, the filter-characteristic was

computed with the help of Matlab’s signal processing toolbox (routine “fircls1 ( M-1 , f<sub>pb</sub> , r<sub>pb</sub> , r<sub>sb</sub> )” ) [10]. Calculus for M = 128, succeeded by quantization to 9 bit of the real-valued coefficients, produces a filter-characteristic that just meets the specifications for the netto/bruto bandwidth and the stop-band attenuation. Remind that, in order to enable a valid poly-phase decomposition, M must be a multiple of I (i.e. quotient M/I must be integer).

## V. PLATFORM SPECIFICATION AND IMPLEMENTATION

The specifications of PPF and TVF are summarized in Table 2 and Table 3 respectively. The parameter-values guarantee optimal fitting of their implementations on the FPGA-technology of the pre-manufactured TOM-board (Fig. 6). The chip-clock and the number of DSP-blocks put a limit to the range of multiplexing factor R. For the TVF this range is a factor 4 wider than for the PPF. Thus for the TVF, more multiplications can be executed with fewer multipliers.

### A. Poly-Phase Filter (PPF) parameter valuation

The PPF-*specification* is captured in four parameters, see Table 2 [8]: the input sampling-rate (f<sub>s,in</sub>), the interpolation factor (= I), the decimation factor (= D), the number of filter-coefficients (= M). From these, we derive: the interpolation sampling-rate (= f<sub>s,interp</sub>), the output sampling-rate (= f<sub>s,out</sub>), the number of parallel signal-paths (= M). For PPF-*implementation*, three parameters are added: the multiplexing factor (= R), the number of bits per input-sample, the number of bits per filter-coefficient. From these, we derive: the multiplexed interpolation sampling-rate (= the chip-clock = R\*f<sub>s,interp</sub>) and the number of multiplexed parallel signal-paths (= M/R).

The number of multiplications is equal to the number of parallel signal-paths (= M). Thus, the number of multipliers is equal to the number of multiplexed parallel signal-paths (= M/R). Each multiplier is re-used for multiplication R times, at R times the interpolation sampling-rate.

For the PPF we have selected 9 bits per coefficient. Note, that the number of multipliers (= 128) is below the Stratix-bound of 144, and the chip-clock (= 160 MHz) is below the soft bound of 200 MHz.

Table 2. Summary of the PPF-specification for implementation on Altera’s Stratix (and Stratix-II ) families.

fs,in	I	D	fs,out = (I/D)*fs,in	M	fs,interp = I*fs,in	M/R	R	fs,interpMUX = R*fs,interp
40	4	5	32	128	160	<b>128</b>	1	<b>160</b>
40	2	5	16	256	80	<b>128</b>	2	<b>160</b>
40	1	5	8	512	40	<b>128</b>	4	<b>160</b>
40	1	10	4	512	40	<b>128</b>	4	<b>160</b>
40	1	20	2	512	40	<b>128</b>	4	<b>160</b>
40	1	40	1	512	40	<b>128</b>	4	<b>160</b>
40	1	80	0,5	512	40	<b>128</b>	4	<b>160</b>

### B. Time-Variant Filter (TVF) parameter valuation

The TVF-*specification* is captured in four parameters, see Table 3 [8]: the input sampling-rate (= f<sub>s,in</sub>), the interpolation factor (= I), the decimation factor (= D), the number of filter-coefficients (= M). From these, we derive: the output sampling-rate (= f<sub>s,out</sub>), the number of parallel signal-paths (= M/I), the number of coefficients per sub-filter (= I). For TVF-*implementation*, three parameters are added: the multiplexing factor (= R), the number of bits per input-sample, the number of bits per filter-coefficient. From these, we derive: the multiplexed output sampling-rate (= the chip-clock = R\*f<sub>s,out</sub>) and the number of multiplexed parallel signal-paths (= (M/I)/R).

The number of multiplications is equal to the number of parallel signal-paths (= M/I). Thus, the number of multipliers is equal to the number of multiplexed parallel signal-paths (= (M/I)/R). Each multiplier is re-used for multiplication R times, at R times the output sampling-rate.

For the TVF we have selected 18 bits per coefficient. Note, that the number of multipliers (= 64) is below the Stratix-bound of 72, and the chip-clock (max. 128 MHz) is below the soft bound of 200 MHz.

Table 3. Summary of the TVF-specification for implementation on Altera’s Stratix (and Stratix-II ) families.

fs,in	I	D	fs,out = (I/D)*fs,in	M	M/I	(M/I)/R	R	fs,outMUX = R*fs,out
40	4	5	32	1024	256	<b>64</b>	4	<b>128</b>
40	2	5	16	1024	512	<b>64</b>	8	<b>128</b>
40	1	5	8	1024	1024	<b>64</b>	16	<b>128</b>
40	1	10	4	1024	1024	<b>64</b>	16	<b>64</b>
40	1	20	2	1024	1024	<b>64</b>	16	<b>32</b>
40	1	40	1	1024	1024	<b>64</b>	16	<b>16</b>
40	1	80	0,5	1024	1024	<b>64</b>	16	<b>8</b>

## VI. COMPARISON BETWEEN TVF AND PPF

The PPF-architecture offers undiminished computational accuracy, at the cost of the presence of a high interpolation clock-rate and a larger circuit. The TVF-architecture offers a higher computational efficiency, at the cost of non-linear distortion. Because of their difference in computational efficiency, the TVF comprises more filter-functionality than the PPF when both are mapped onto FPGA-technology of the same TOM-board. The architectures of PPF and TVF possess equally scalable structures, according to 7 parameters. The architectures of PPF and TVF have been transformed into VHDL-models fit for logic synthesis. Interpolation factor I and decimation factor D can be modified during the operation, so that real-time adaptation of the SRC-functionality is possible.

For consumer applications, such as audio electronics, non-linear distortion is accepted up to a certain level, in

order to keep the amount of circuitry and the power consumption to a minimum. For high-precision applications, such as radio astronomy, non-linear distortion is to be avoided - even this would lead to an increase in the amount of circuitry and in power consumption. Therefore, the PPF is preferred above the TVF.

## VII. CONCLUSION

In this article we have presented two SRC-architectures, the PPF and the TVF, and explored their computational efficiency, scalability and computational accuracy. Out of both architectures, we have developed synthesizable VHDL-models that fit into pre-selected FPGA-technology.

## APPENDIX

### DIRECT RE-SAMPLING AND NON-LINEAR DISTORTION

At the Hold & Sample (H&S-) stage of the TVF, the M/I input-signals are directly re-sampled, without an intermediate stage that applies a - usually high-rate - interpolation time-grid [3]. For each input-signal, the length of the sequence of samples changes with a factor D/I according to the following time-domain process (Fig. A.1):

- First, for each sub-sequence of D input-samples, we skip (I-D) samples (if quotient D/I > 1) or duplicate (I-D) samples (if quotient I/D > 1).

- Next, we replace the resulting set of I output-samples from the input time-grid onto the output time-grid.

Applying time-displacement on an input-sample is equivalent to introducing an amplitude-error in the corresponding output-sample [5]. Thus, the second step in this process introduces *non-linear distortion*. If the input-signal is a single harmonic, then the output-signal does not represent a single harmonic any more, although it is still periodic. For an input-signal consisting of a single harmonic, the pattern of sub-harmonics appearing in the discrete output-spectrum depends on the re-sampling ratio I/D. For a fixed ratio I/D, each harmonic that is present in the input-signal generates its own specific set of sub-harmonics in the output-signal. The essential flaw of the Time-Variant Filter concept is that these sets of sub-harmonics can neither be prevented nor removed by any filtering action, whether preceding or succeeding the H&S-stage. (In fact, the H&S-stage performs image-suppression with bad accuracy. Out of an arbitrary input-signal, a step-function approximation is created which serves as an input for immediate down-sampling.)

What does such a pattern of sub-harmonics look like in the frequency-domain? How many sub-harmonics are created? At what frequencies do they appear?

We shall illustrate the answer by means of an example with I = 5 and D = 7 (i.e. I < D and I,D mutually prime).

Let  $f_{s,in} = 40.0$  MHz, then  $f_{s,out} = (I/D) * f_{s,in} = 200.0/7$  MHz .

The input-signal is related to an input time-grid with resolution:  $T_{s,in} = 1/f_{s,in} = 25.0$  ns.

The output-signal is related to an output time-grid with resolution:  $T_{s,out} = (D/I)*T_{s,in} = 35.0$  ns.

Of each sub-sequence of 7 input-samples, 5 samples (i.e. sample-numbers 0, 1, 2, 4, 5) are mapped onto the output time-grid, 2 samples (i.e. sample-numbers 3, 6) are discarded.

In the *time-domain*, the shortest interval for which the output time-grid coincides with the input time-grid (the “grid-coincidence” period) is:  $D*T_{s,in} = I*T_{s,out} = 175.0$  ns .

Note, that this is identical to the “smallest common multiple” (SCM) of  $T_{s,in}$  and  $T_{s,out}$ :

$$SCM(T_{s,in}, T_{s,out}) = SCM(1, D/I) * T_{s,in} = (1 * D/I) * T_{s,in} .$$

See Fig. A.1 for the time-domain representations of the input- and output-signals.

(A more intuitive reasoning that leads to the same value for the shortest interval:

$$X \text{ periods of an } X \text{ MHz harmonic wave} = 1 \text{ us} \quad \Leftrightarrow$$

$$X \text{ periods of an } X*Y \text{ MHz harmonic wave} = 1/Y \text{ us} \quad \Leftrightarrow$$

$$D \text{ periods of } (D*f_{s,in}) \text{ MHz} = I \text{ periods of } (I*f_{s,in}) \text{ MHz} \\ = 1/f_{s,in} \text{ us} \quad \Leftrightarrow$$

$$D \text{ periods of } (f_{s,in}) \text{ MHz} = I \text{ periods of } (I*f_{s,in}/D) \text{ MHz} \\ = D / f_{s,in} \text{ us} = I / f_{s,out} \text{ us} )$$

In the *frequency-domain*, this corresponds to a fundamental wave of  $f_{s,in}/D = f_{s,out}/I = 40.0/7$  MHz .

This fundamental wave becomes visible in 6 (= I + 1) equidistant frequency-positions  $f_c$  in the basic spectral period  $(-f_{s,out}/2 \dots +f_{s,out}/2)$  of the output-signal, i.e. in the spectral positions

$$f_c = -100.0/7, -60.0/7, -20.0/7, \\ +20.0/7, +60.0/7, +100.0/7 \text{ MHz} .$$

See Fig A.2 for the frequency-domain representations of the input- and output-signals.

(Note that for I is odd - as in this example - the frequency-positions  $f_c$  form a set of exclusively odd-valued harmonics. In the time-domain, this corresponds to a symmetric waveform. For I is even, the frequency-positions  $f_c$  form a set of exclusively even-valued harmonics. In the time-domain, this corresponds to an anti-symmetric waveform.)

Up to this point, the pattern still does not depend upon the frequency of the harmonic input-signal! Let’s now choose this frequency:  $f_x = 8.0$  MHz. Within each period of the output-spectrum, 10 (= I \*2) spectral lines appear, according to a double-sideband (dsb) modulation-pattern around each of the frequency-positions  $f_c$  . The “modulation-depth”  $\Delta f_{dsb}$  is determined by the distance of signal frequency  $f_x$  to the nearest frequency-position  $f_c$ :

$$\Delta f_{dsb} = \min(\text{abs}(f_x - f_c)) = |8.0 - 60.0/7| = 4.0/7 \sim 0.57 \text{ MHz} .$$

In the time-domain, this corresponds to an “envelope” with period  $1 / \Delta f_{dsb} = 1750.0$  ns. This is the maximum period that occurs in the output-signal. Note, that this is identical to twice the “smallest common multiple” (SCM) of the signal period and the “grid-coincidence” period:

$$SCM(T_x, D*T_{s,in}) = SCM(T_x/T_{s,in}, D) * T_{s,in} = \\ = ((T_x/T_{s,in}) * D) * T_{s,in} = ((125.0/25.0) * 7) * 25.0 \text{ ns} = 875.0 \text{ ns} .$$

Here,  $T_x$  is expressed as a fractional multiple of  $T_{s,in}$ . Thus, the non-linear distortion pattern of the output-signal consists of the following set of harmonics (“ $f_c \pm \Delta f_{dsb}$ ”):

$f_c$	$\pm \Delta f_{dsb}$		[MHz]
-100/7	+ 4/7	= - 96/7	~ -13.7
- 60/7	- 4/7	= - 64/7	~ - 9.14
- 60/7	+ 4/7	= - 56/7	= - <b>8.00</b>
- 20/7	- 4/7	= - 24/7	~ - 3.43
- 20/7	+ 4/7	= - 16/7	~ - 2.29
+ 20/7	- 4/7	= + 16/7	~ + 2.29
+ 20/7	+ 4/7	= + 24/7	~ + 3.43
+ 60/7	- 4/7	= + 56/7	= + <b>8.00</b>
+ 60/7	+ 4/7	= + 64/7	~ + 9.14
+100/7	- 4/7	= + 96/7	~ +13.7

Note, that +/- 8.00 MHz represents  $f_x$ , the frequency of the input-signal.

This example illustrates a mere receipt in order to arrive at the frequency-positions of the harmonics. The receipt that delivers the amplitudes of the harmonics (a complicated combination of sinc-function envelopes) falls outside the scope of this appendix.

#### ACKNOWLEDGEMENTS

We would like to thank Albert-Jan Boonstra for his comments and Albert Bos for his measurement contributions.

#### REFERENCES

- [1] A.V. Oppenheim, R.W. Schaffer, "Digital Signal Processing", Prentice Hall, 1975
- [2] A.W.M. van den Enden, N.A.M. Verhoeckx, "Discrete-time Signal Processing", Prentice Hall, 1989
- [3] J.G. Proakis, D.G. Manolakis, "Digital Signal Processing - principles, algorithms and applications", Prentice Hall, 1996 (3<sup>rd</sup> ed.), chapter 10.5
- [4] N.J. Fliege, "Multirate Digital Signal Processing - multirate systems, filter banks, wavelets", Wiley, 1994, chapter 4.5
- [5] A.W.M. van den Enden, "Efficiency in Multirate and Complex Digital Signal Processing", Delta Press, 2001, chapter 4
- [6] R. Millenaar, "Towards a Digital TADU-MAX System", ASTRON-RP (vs.0.2), 11-2003
- [7] J. Dromer, "The TADU system: a Digital Fractional Re-sampler and

- Multi-bits Power Scaler", ASTRON-RP-015, 04-2005
- [8] R. de Wild, "Realization of a Time Variant Filter (TVF) for the Tied Array Distribution Unit (TADU) of the Westerbork Synthesis Radio Telescope (WSRT)", ASTRON-RP-110 (vs.1.0), 26-07-2006
- [9] A. Doorduyn, "Design of the TADU TOM-board", ASTRON-RP-120 (vs.1.0), 28-07-2006
- [10] Matlab (Filter design Toolbox):  
<http://www.mathworks.com/products/filterdesign>
- [11] Altera (Stratix I & II device-families):  
<http://www.altera.com/support/devices/dvs-index.html>

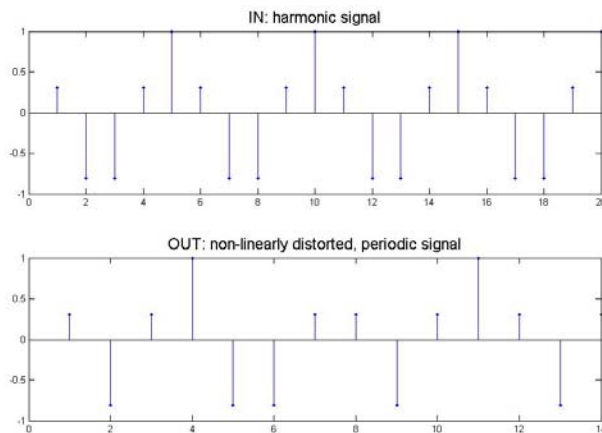


Fig. A.1. Direct re-sampling in the time-domain ( $I = 5$ ,  $D = 7$ ):  
input time-grid resolution = 25.0 ns  
output time-grid resolution =  $(7/5) * 25.0 = 35.0$  ns  
→ grid-coincidence interval =  $7 * 25.0 = 5 * ((7/5) * 25.0) = 175.0$  ns  
harmonic input-signal: period = 125.0 ns  
→ output-signal:  
fundamental wave =  $2 * S.C.M. (125.0 \text{ ns} \cdot 7 * 25.0 \text{ ns}) = 1750.0$  ns

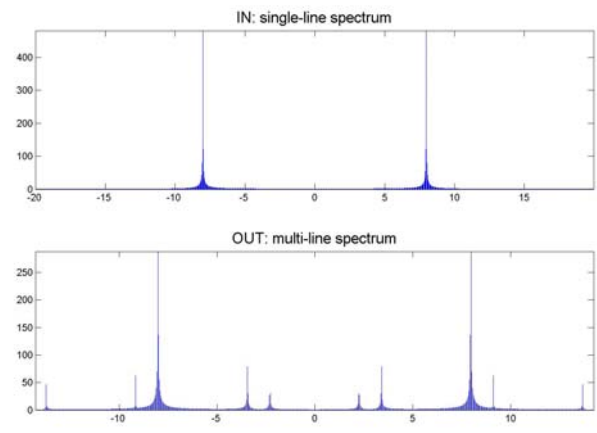


Fig. A.2. Direct re-sampling in the frequency-domain ( $I = 5$ ,  $D = 7$ ):  
input spectral period = 40.0 MHz  
output spectral period =  $(5/7) * 40.0 = 28.5714$  MHz  
→ "carrier"-spacing =  $40.0/7 = ((5/7) * 40.0) / 5 = 5.7143$  MHz  
harmonic input signal: frequency = 8.0 MHz  
→ output-signal:  
"modulation-depth" =  $(8.0 - (+3) * (40.0/7)) / 2 = 4.0/7 = 0.57$  MHz