It can be observed from the above figure that direct the brute force implementation not only computes the summation over zero-valued samples but also does not exploit the real-valuedness of the WVD. When we mean zero-valued samples, the data explicitly becomes zero at those points, where the indices referring them point to samples outside the range. We modify the data flow that has to be fed to the time-recursive architecture in a serial manner by noting the fact that Eqn. (6.11) can be rewritten as:

$$
\begin{aligned}
C(n_0,k) = {} & 2\sum_{\tau=-(N-1)/2}^{-1} x(n_0+\tau)\,x^*(n_0-\tau)\,e^{-j\frac{2\pi k\tau}{N}} \\
& + 2\sum_{\tau=1}^{(N-1)/2} x(n_0+\tau)\,x^*(n_0-\tau)\,e^{-j\frac{2\pi k\tau}{N}} + 2\,\left| x(n_0)\right|^2 ,
\end{aligned}
\tag{6.12}
$$

which reduces to

$$
\begin{aligned}
C(n_0,k) = {} & 4\sum_{\tau=1}^{(N-1)/2} [\, x_r(n_0+\tau)x_r(n_0-\tau)+x_i(n_0+\tau)x_i(n_0-\tau)\,]\cos(\frac{2\pi k\tau}{N}) \\
& +4\sum_{\tau=1}^{(N-1)/2} [\, x_r(n_0+\tau)x_i(n_0-\tau)-x_i(n_0+\tau)x_r(n_0-\tau)\,]\sin(\frac{2\pi k\tau}{N}) +2\left| x(n_0)\right|^2.
\end{aligned}
\tag{6.13}
$$

To implement the above equation, the data to be fed to the architecture is in the positive direction of τ and pruning cannot be done if the data flow is in this order, since time-recursive approach is an accumulate operation and utilizes the periodicity of the Fourier transform kernel. To circumvent this difficulty, we propose a different data flow which prunes the input data to avoid unnecessary computations. We start with the first nonzero value as shown in Fig. 6.4 and feed it to the time-recursive architecture for implementing the WVD as well as GTFDs shown in Fig. 6.5 in the direction of τ till $\tau = 0$.

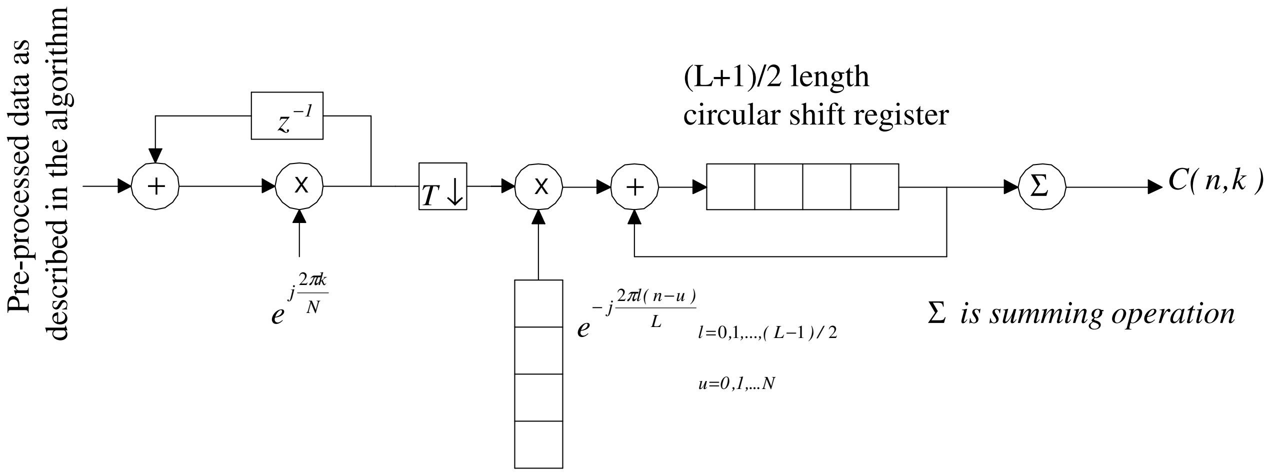

Fig. 6.5. Architecture for WVD and GTFDs

The final value is multiplied by an additional term of $e^{-j\frac{2\pi k}{N}}$. This additional multiplication is done using the same DFT module as explained later. The downsampler in the architecture is set according to the time instant we wish to compute as:

$$T = n+2, \quad 0 \leq n \leq (N+1)/2$$

$$= N-n+1, \quad (N+3)/2 \leq n \leq N-1.$$

The DFT module and the downsampler of the architecture are used for computing the WVD, while the remaining blocks are used in computing GTFDs, that will be dealt with in the next Section. Now we consider the WVD algorithm with an example:

Let the discrete time sequence *x(n)* is nonzero in the interval *[0, 4]* and we require Eqn. (6.11) to be evaluated at *n=1*. Then the data at this instant, for $\tau = -2, -1, 0, 1$ and 2, are $0, x(0)x^*(1), |x(2)|^2, x(1)x^*(0)$ and 0, respectively. Since the data is symmetric over τ, we can run the summation either in the positive direction or negative direction of τ and finally consider the real part of the resulting sum. In this case, at the data indexed by $\tau = 2$, we need not compute the summation since it is a zero-valued sample. However, if we were to compute it in a time-recursive manner, we cannot stop the summation since the phase term that every term in the summation has to undergo differs. This problem is very serious if there were more zero-valued samples. Now we start with the first nonzero samples in the negative direction of τ, i.e., $\tau = -1$ and run the summation till $\tau = 0$. After doing the summation, we have to multiply the resultant with a term $e^{-j\frac{2\pi k}{N}}$, i.e., Eqn. (6.13) becomes

$$C(2,k) = 2\,Re[\,(\,2x(0)x^*(1)\,e^{j\frac{2\pi k 2}{5}} + |\,x(2)\,|^2\,)\,e^{-j\frac{2\pi k}{N}}\,] \quad . \tag{6.14}$$

We have mentioned earlier that the multiplication of the resulting term by $e^{-j\frac{2\pi k}{N}}$ using the same DFT module can be done by resetting the data at the input side to zero and performing the accumulate operation one more time. This is stemmed from the fact that Eqn. (6.14) can equivalently be obtained from $2\,Re[\,(\,2x(0)x^*(1)\,e^{j\frac{2\pi k 2}{5}} + |\,x(2)\,|^2\,)\,e^{j\frac{2\pi k}{N}}\,]$. It can be verified that the same result can be obtained by using Eqn. (6.12) directly without pre-processing and pruning. The WVD of a linear FM signal computed using the above algorithm is shown in Fig. 6.6. It is required that the data has to be multiplied by a factor of two for all τ, except at $\tau = 0$. This modification to the data flow is expressed mathematically as:

$$C(n_0,k)= 2\,Re[(\sum_{\tau=-n_0}^{-1} 2\,x(n_0+\tau)\,x^*(n_0-\tau)\,e^{j\frac{2\pi k(|\tau|+1)}{N}} + |x(n_0)|^2\, e^{j\frac{2\pi k(\tau+1)}{N}})e^{-j\frac{2\pi k}{N}}],$$

$$0 \le n_0 \le \frac{(N-1)}{2} \tag{6.15a}$$

and

$$C(n_0,k)= 2\,Re[(\sum_{\tau=N1-n_0}^{-1} 2x(n_0+\tau)\,x^*(n_0-\tau)\,e^{j\frac{2\pi k(|\tau|+1)}{N}} + |x(n_0)|^2\, e^{j\frac{2\pi k(\tau+1)}{N}})e^{-j\frac{2\pi k}{N}}],$$

$$\frac{(N+1)}{2} \le n_0 \le N-1 \tag{6.15b}$$

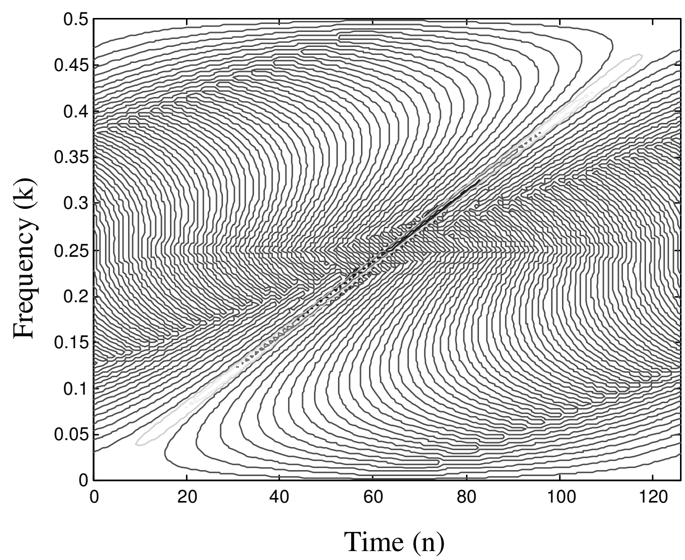

Fig. 6.6. WVD of a linear FM signal computed using the time-recursive approach

When this approach is followed, the complex multiplications required to compute the WVD at each instant increase linearly till $n=(N-1)/2$ and then start decreasing linearly till $n=(N-2)$ besides another complex multiplication required to obtain the phase offset at all time instants. We now make conservative estimate of the *OPs* required to obtain the WVD at different time instants.

$$\begin{aligned} OPs &= \sum_{i=1}^{(N+1)/2} i \;+\; \sum_{i=(N-1)/2}^{1} i \;+\; N \\ &= 2\sum_{i=1}^{(N-1)/2} i \;+\; \frac{(N+1)}{2} \;+\; N \\ &= (N^2+6N+1)/4 \;. \end{aligned} \tag{6.16}$$

The *OPs* obtained using the FFT approach are $\frac{N}{2}log_2(N)$, with direct brute force are N^2 and by exploiting symmetry conditions, the *OPs* are $N(N+1)/2$. Hence, we have obtained a balance between the FFT and time-recursive approach, i.e., it is less complex in terms of hardware when compared FFT and is computationally more efficient than a direct time-recursive approach. As the FFT based computation is a parallel-in and parallel-out operation, the throughput to obtain WVD at each time is constant and WVD at all time instants is known after $log_2(N)\,OPs$. In the brute force time-recursive approach that exploits symmetry conditions, the throughput is $\frac{(N+1)}{2}OPs$. In the method proposed, as we are not computing the summation over all possible combinations, the OPs required for each time instant are dependent on the pruning we can do. A comparison of the computational complexity in terms of OPs for the butterfly structure, time-recursive approach with symmetry conditions and our approach has been made in Table. 6.3

Table 6.3: Comparison of various approaches for computing WVD

	Butterfly structure (Radix-2)	Time-recursive approach with symmetry conditions	Proposed method
Complex multipliers	$\frac{N}{2} log_2(N)$	$(N-1)$	$(N-1)$
OPs required to obtain WVD at all time instants	$\frac{N}{2} log_2(N)$	$N(N+1)/2$	$(N^2+6N+1)/4$
Throughput (in *OPs*)	$log_2(N)$	$(N+1)/2$	$n+1,\ 0 \le n \le (N-1)/2$ $N-n,\ (N+1)/2 \le n \le N-1$

6.6. GENERALIZED TFDs

Any distribution that belongs to the Cohen's class can be represented by

$$C(t,\ \omega) \ = \ \iiint x(u+\frac{\tau}{2})\, x^*(u-\frac{\tau}{2})\ \phi(\theta,\tau)\, e^{-j\omega\tau}\ e^{j\theta u}\ e^{-j\theta t}\ d\theta\ d\tau du\,, \tag{6.17}$$

where $\phi(\theta,\tau)$ is the kernel that determines the properties of the distribution. When $\phi(\theta,\tau)$=1, Eqn. (6.17) reduces to the WVD expression, as discussed in the previous Section. In general, most of the distributions are real-valued, for example, Choi-Williams, Born-Jordan, Page, Generalized Exponential Distributions, etc. This property of real-valuedness gets reflected as a constraint on the kernel as conjugate symmetry of the kernel. We are interested in the TFDs of this type, eventhough the same can be extended to any kernel, which obviously reduces the throughput. When put in a mathematical form,