ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 570 028

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

NOTES

COURSE TITLE: SIGNALS & DIGITAL SIGNAL PROCESSING

COURSE CODE: BEE502

SEMESTER: V

MODULE-3: FAST FOURIER TRANSFORMS ALGORITHM

INSTITUTIONAL VISION AND MISSION

VISION:

- Development of academically excellent, culturally vibrant, socially responsible, and globally competent human resources.

MISSION:

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional, and moral foundations and shine as torchbearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence.

Department Vision and Mission

Vision:

To produce Electrical & Electronics Engineers through greatest quality of technical education, technical skill training and intellectual capacity building of individuals.

Mission:

- To provide knowledge to students that builds a strong foundation in the basic principles of electrical engineering, problem solving abilities, analytical skills, soft skills and communication skills for their overall development.
- To offer outcome based technical education.
- To encourage faculty in training & development and to offer consultancy through research & industry interaction.

MODULE-3: FAST-FOURIER-TRANSFORM(FFT) ALGORITHMS

Structure

Objectives

1. To study the difference between the direct computation and fast fourier computation of DFT
2. To study the computation fo DFT using FFT method

3.1 Direct Computation of DFT

Given signal samples: x[0], . . . , x[N - 1] (some of which may be zero), develop a procedure to compute

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn}$$

for k = 0, . . . , N - 1 where

$$W_N = e^{-j\frac{2\pi}{N}}.$$

We would like the procedure to be fast, simple, and accurate. Fast is the most important, so we will sacrifice simplicity for speed, hopefully with minimal loss of accuracy

3.2 Need for efficient computation of DFT

Let us start with the simple way. Assume that W_N^{kn} has been precompiled and stored in a table for the N of interest. How big should the table be? W_N^m is periodic in m with period N, so we just need to tabulate the N values:

$$W_N^m = \cos\left(\frac{2\pi}{N}m\right) - j\sin\left(\frac{2\pi}{N}m\right)$$

(Possibly even less since Sin is just Cos shifted by a quarter periods, so we could save just Cos when N is a multiple of 4.)

Why tabulate? To avoid repeated function calls to Cos and sin when computing the DFT. Now we can compute each X[k] directly form the formula as follows

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} = x[0]\, W_N^0 + x[1]\, W_N^k + x[2]\, W_N^{2k} + \cdots + x[N-1]\, W_N^{(N-1)k}.$$

For each value of k, there are N complex multiplications, and (N-1) complex additions. There are N values of k, so the total number of complex operations is

$$N \cdot N + N(N-1) = 2N^2 - N \equiv O(N^2).$$

Complex multiplies require 4 real multiplies and 2 real additions, whereas complex additions require just 2 real additions N^2 complex multiplies are the primary concern.

N^2 increases *rapidly* with N, so how can we reduce the amount of computation? By exploiting the following properties of W:

- Symmetry property: $W_N^{k+N/2} = -W_N^k = e^{j\pi} W_N^k$
- Periodicity property: $W_N^{k+N} = W_N^k$
- Recursion property: $W_N^2 = W_{N/2}$

The first and third properties hold for even N, *i.e.*, when 2 is one of the prime factors of N. There are related properties for other prime factors of N.

Divide and conquer approach

We have seen in the preceding sections that the DFT is a very computationally intensive operation. In 1965, Cooley and Tukey published an algorithm that could be used to compute the DFT much more efficiently. Various forms of their algorithm, which came to be known as the Fast Fourier Transform (FFT), had actually been developed much earlier by other mathematicians (even dating back to Gauss). It was their paper, however, which stimulated a revolution in the field of signal processing.

It is important to keep in mind at the outset that the FFT is *not* a new transform. It is simply a very efficient way to compute an existing transform, namely the DFT. As we saw, a straight forward implementation of the DFT can be computationally expensive because the number of multiplies grows as the square of the input length (i.e. N^2 for an N point DFT). The FFT reduces this computation using two simple but important concepts. The first concept, known as divide-and-conquer, splits the problem into two smaller problems. The second concept, known as recursion, applies this divide-and-conquer method repeatedly until the problem is solved.

3.3 Additional Problems

1. A designer has available a number of eight point FFT chips. Show explicity how he should interconnect three such chips in order to compute a 24-point DFT.

Solution:-

Create three subsequences of 8-pts each

$$\begin{aligned} Y(k) &= \sum_{n=0,3,6,\ldots}^{21} y(n)W_N^{kn} + \sum_{n=1,4,7,\ldots}^{22} y(n)W_N^{kn} + \sum_{n=2,5,\ldots}^{23} y(n)W_N^{kn} \\ &= \sum_{i=0}^{7} y(3i)W_{\frac{N}{3}}^{ki} + \sum_{i=0}^{7} y(3i+1)W_{\frac{N}{3}}^{ki}W_N^{k} + \sum_{i=0}^{7} y(3i+2)W_{\frac{N}{3}}^{ki}W_N^{2k} \\ &\triangleq Y_1(k) + W_N^k Y_2(k) + W_N^{2k} Y_3(k) \end{aligned}$$

where Y_1, Y_2, Y_3 represent the 8-pt DFTs of the subsequences.

2. Let x(n) be a real valued N-point (N=2) sequence. Develop a method to compute An N-Point DFT $X^1(k)$, which contains only the odd harmonics by using only a Real N/2 points DFT.

Solution:-

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad 0 \le k \le N-1 \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn} + \sum_{r=0}^{\frac{N}{2}-1} x(r+\frac{N}{2})W_N^{(r+\frac{N}{2})k} \\ \text{Let} X'(k') &= X(2k+1), \quad 0 \le k' \le \frac{N}{2}-1 \end{aligned}$$

$$\text{Then, } X'(k') = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) W_N^{(2k'+1)n} + x(n+\frac{N}{2}) W_N^{(n+\frac{N}{2})(2k'+1)} \right]$$

$$\text{Using the fact that } W_N^{2k'n} = W_{\frac{N}{2}}^{k'n}, \quad W_N^N = 1$$

$$X'(k') = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) W_N^n W_{\frac{N}{2}}^{k'n} + x(n+\frac{N}{2}) W_{\frac{N}{2}}^{k'n} W_N^n W_N^{\frac{N}{2}} \right]$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x(n+\frac{N}{2}) \right] W_N^n W_{\frac{N}{2}}^{k'n}$$

3. The Z-Transform of the Sequence x(n) = u(n) –u(n-7) is sampled at five points on

The unit circle as follows. X(k) = X(z) = $e^{j2\pi k/5}$ k= 0,1,2,3,4

Solution:-

$$X(z) = 1 + z^{-1} + \ldots + z^{-6}$$

$$X(k) = X(z)|_{z=e^{j\frac{2\pi}{5}}}$$

$$= 1 + e^{-j\frac{2\pi}{5}} + e^{-j\frac{4\pi}{5}} + \ldots + e^{-j\frac{12\pi}{5}}$$

$$= 2 + 2e^{-j\frac{2\pi}{5}} + e^{-j\frac{4\pi}{5}} + \ldots + e^{-j\frac{8\pi}{5}}$$

$$x'(n) = \{2, 2, 1, 1, 1\}$$

$$x'(n) = \sum_m x(n+7m), \quad n = 0, 1, \ldots, 4$$

Temporal aliasing occurs in first two points of $x'(n)$ because $X(z)$ is not sampled at sufficiently small spacing on the unit circle.

$$X(k) = X(z)|_{z=z_k}$$

$$= \sum_{n=0}^{7} x(n) \left[0.8 e^{j\left[\frac{2\pi k}{8} + \frac{\pi}{8}\right]} \right]^{-n}$$

$$s(n) = x(n)\, 0.8 e^{-j\frac{\pi}{8}n}$$

RADIX-2 FFT ALGORITHM FOR THE COMPUTATION OF DFT AND IDFT

Standard frequency analysis requires transforming time-domain signal to frequency domain and studying Spectrum of the signal. This is done through DFT computation. N-point DFT computation results in N frequency components. We know that DFT computation through FFT requires N/2 $\log_2 N$ complex multiplications and N $\log_2 N$ additions. In certain applications not all N frequency components need to be computed (an application will be discussed). If the desired number of values of the DFT is less than 2 $\log_2 N$ than direct computation of the desired values is more efficient that FFT based computation.

4.2 Radix-2 FFT

Useful when N is a power of 2: $N = r^v$ for integers r and v. ‘r’ is called the **radix**, which comes from the Latin word meaning .a root, and has the same origins as the word radish.

When N is a power of r = 2, this is called **radix-2**, and the natural .divide and conquer approach. is to split the sequence into two sequences of length N=2. This is a very clever trick that goes back many years.

4.2.1 Decimation in time

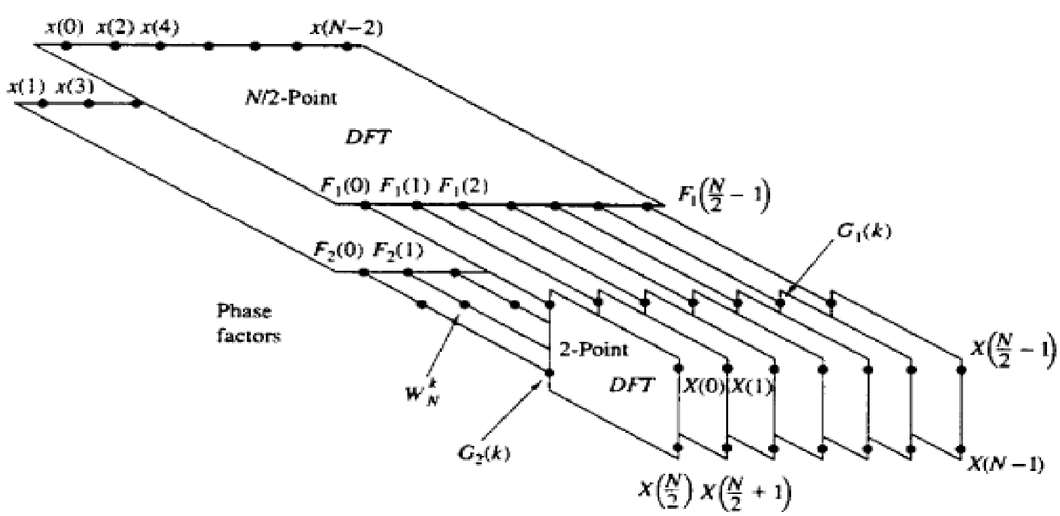
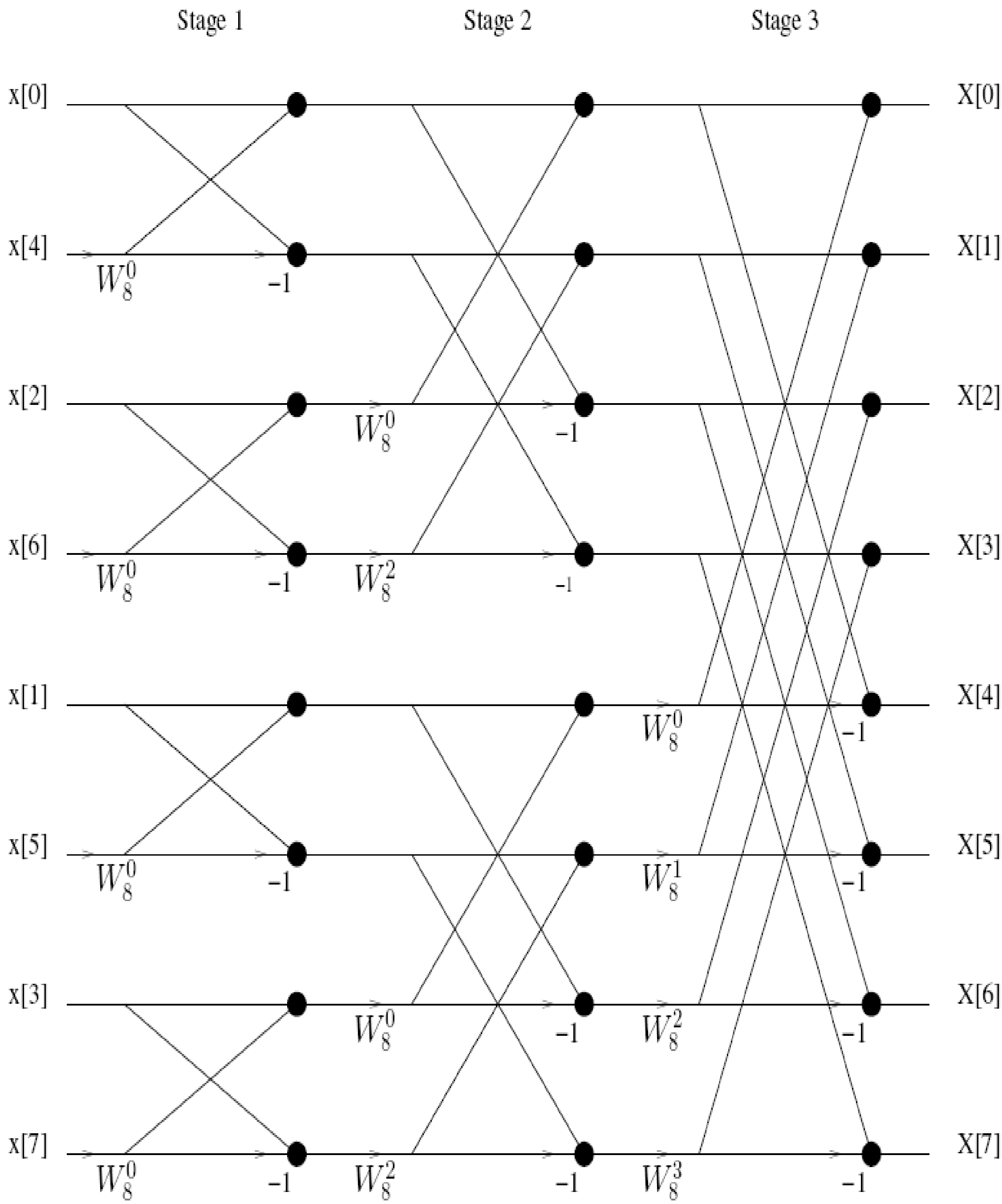


Fig 4.1 First step in Decimation-in-time domain Algorithm

$N = 8$-point decimation-in-time FFT algorithm



$$W_N = e^{-j\frac{2\pi}{N}}$$

Each dot represents a complex addition.

Each arrow represents a complex multiplication.

Decimation-in-frequency Domain

Another important radix-2 FFT algorithm, called decimation-in-frequency algorithm is obtained by using divide-and-conquer approach with the choice of M=2 and L= N/2.This choice of data implies a column-wise storage of the input data sequence. To derive the algorithm, we begin by splitting the DFT formula into two summations, one of which involves the sum over the first N/2 data points and the second sum involves the last N/2 data points. Thus we obtain

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + \sum_{n=N/2}^{N-1} x(n) W_N^{kn}$$
$$= \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right) W_N^{kn}$$

Since $W_N^{kN/2} = (-1)^k$, the expression (6.1.33) can be rewritten as

$$X(k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right)\right] W_N^{kn}$$

Now, let us split X(k) into the even and odd-numbered samples. Thus we obtain

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right)\right] W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \qquad (6.1.35)$$

and

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left\{\left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^{n}\right\} W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \qquad (6.1.36)$$

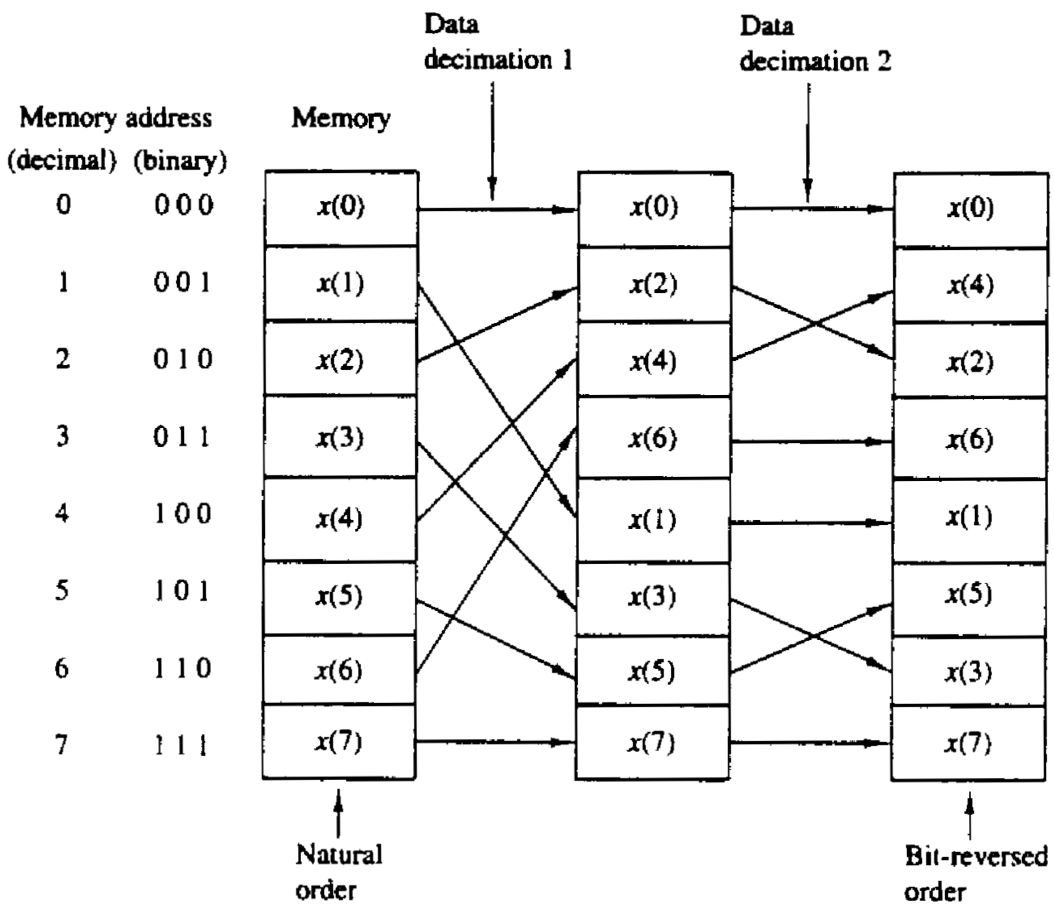
where we have used the fact that $W_N^2 = W_{N/2}$.

If we define the $N/2$-point sequences $g_1(n)$ and $g_2(n)$ as

$$g_1(n) = x(n) + x\left(n + \frac{N}{2}\right)$$
$$g_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^n \qquad n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

then

$$X(2k) = \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn}$$
$$X(2k+1) = \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn}$$



(a)

$(n_2 n_1 n_0)$	→	$(n_0 n_2 n_1)$	→	$(n_0 n_1 n_2)$
(0 0 0)	→	(0 0 0)	→	(0 0 0)
(0 0 1)	→	(1 0 0)	→	(1 0 0)
(0 1 0)	→	(0 0 1)	→	(0 1 0)
(0 1 1)	→	(1 0 1)	→	(1 1 0)
(1 0 0)	→	(0 1 0)	→	(0 0 1)
(1 0 1)	→	(1 1 0)	→	(1 0 1)
(1 1 0)	→	(0 1 1)	→	(0 1 1)
(1 1 1)	→	(1 1 1)	→	(1 1 1)

(b)

Fig 4.2 Shuffling of Data and Bit reversal

The computation of the sequences g1 (n) and g2 (n) and subsequent use of these sequences to compute the N/2-point DFTs depicted in fig we observe that the basic computation in this figure involves the butterfly operation.

The computation procedure can be repeated through decimation of the N/2-point DFTs, X(2k) and X(2k+1). The entire process involves $v = \log_2 N$ of decimation, where each stage involves N/2 butterflies of the type shown in figure 4.3.

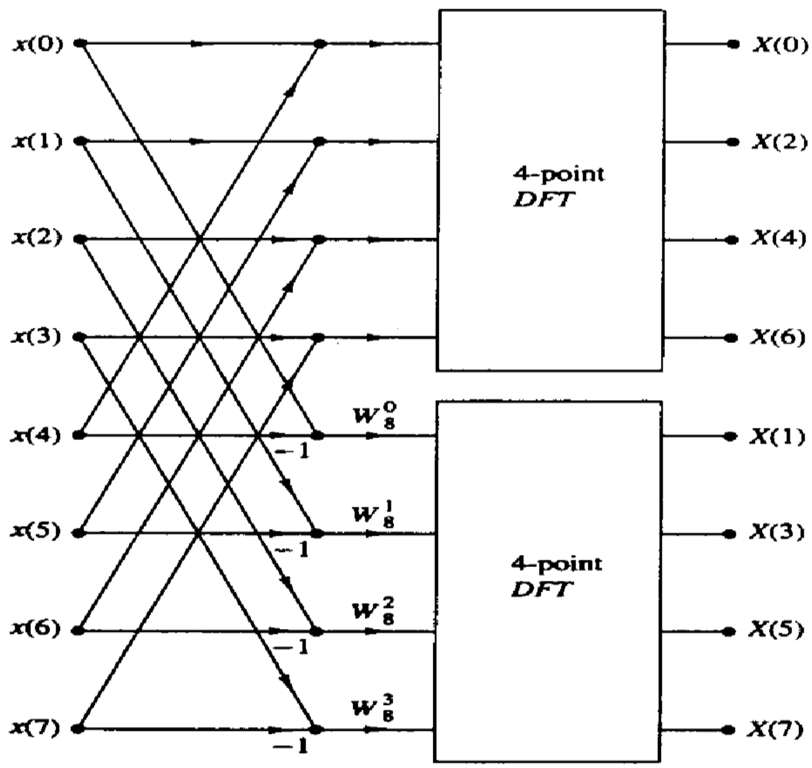


Fig 4.3 First step in Decimation-in-time domain Algorithm

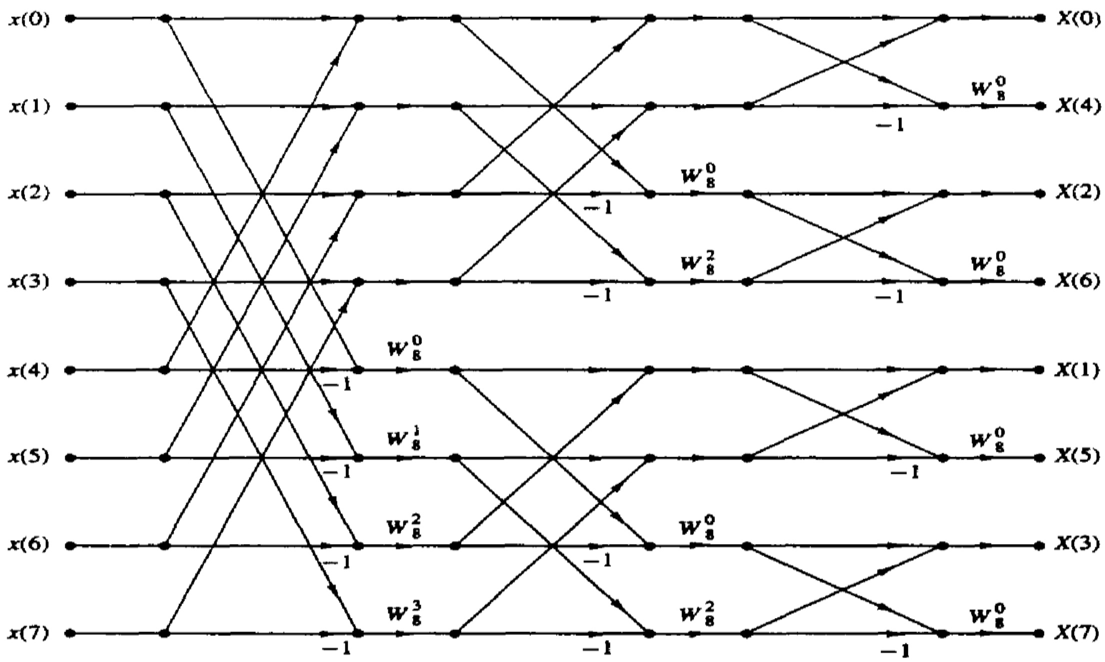


Fig 4.4 N=8 point Decimation-in-frequency domain Algorithm

Example:DTMF–DualTone Multi frequency

This is known as touch-tone/speed/electronic dialing, pressing of each button generates a unique set of two-tone signals, called DTMF signals. These signals are processed at exchange to identify the number pressed by determining the two associated tone frequencies. Seven frequenciesareusedto codethe 10 decimaldigitsand two specialcharacters (4x3 array)

In this application frequency analysis requires determination of possible seven (eight) DTMF fundamental tones and their respective second harmonics .For an 8 kHz sampling freq, the best value of the DFT length N to detect the eight fundamental DTMF tones has been found to be 205 .Not all 205 freq components are needed here, instead only those corresponding to key frequencies are required. FFT algorithm is not effective and efficient in this application. The direct computation of the DFT which is more effective in this application is formulated as a linear filtering operation on the input data sequence.

This algorithm is known as Goertzel Algorithm

This algorithm exploits periodicity property of the phase factor. Consider the DFT definition

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad (1)$$

Since W_N^{-kN} is equal to 1, multiplying both sides of the equation by this results in;

$$X(k) = W_N^{-kN} \sum_{m=0}^{N-1} x(m) W_N^{mk} = \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)} \qquad (2)$$

This is in the form of a convolution $y_k(n) = x(n) * h_k(n)$

$$y_k(n) = \sum_{m=0}^{N-1} x(m) W_N^{-k(n-m)} \qquad (3)$$

$$h_k(n) = W_N^{-kn} u(n) \qquad (4)$$

Where $y_k(n)$ is the out put of a filter which has impulse response of $h_k(n)$ and input x(n).

The output of the filter at n = N yields the value of the DFT at the freq $\omega_k = 2\pi k/N$

The filter has frequency response given by

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \qquad (6)$$

The above form of filter response shows it has a pole on the unit circle at the frequency $\omega_k = 2\pi k/N$.

Entire DFT can be computed by passing the block of input data into a parallel bank of N single-pole filters (resonators)

The above form of filter response shows it has a pole on the unit circle at the frequency $\omega_k = 2\pi k/N$.

Entire DFT can be computed by passing the block of input data into a parallel bank of N single-pole filters (resonators)

Difference Equation implementation of filter:

From the frequency response of the filter (eq 6) we can write the following difference equation relating input and output;

$$H_k(z) = \frac{Y_k(z)}{X(z)} = \frac{1}{1 - W_N^{-k} z^{-1}}$$

$$y_k(n) = W_N^{-k} y_k(n-1) + x(n) \qquad y_k(-1) = 0 \qquad (7)$$

The desired output is $X(k) = y_k(n)$ for k = 0,1,…N-1.

The phase factor appearing in the difference equation can be computed once and stored.

The form shown in eq (7) requires complex multiplications which can be avoided doing suitable modifications (divide and multiply by $1 - W_N^k z^{-1}$). Then frequency response of the filter can be alternatively expressed as

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N) z^{-1} + z^{-2}} \qquad (8)$$

This is second –order realization of the filter (observe the denominator now is a second-order expression). The direct form realization of the above is given by

$$v_k(n) = 2\cos(2\pi k/N) v_k(n-1) - v_k(n-2) + x(n) \qquad (9)$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \qquad v_k(-1) = v_k(-2) = 0 \qquad (10)$$

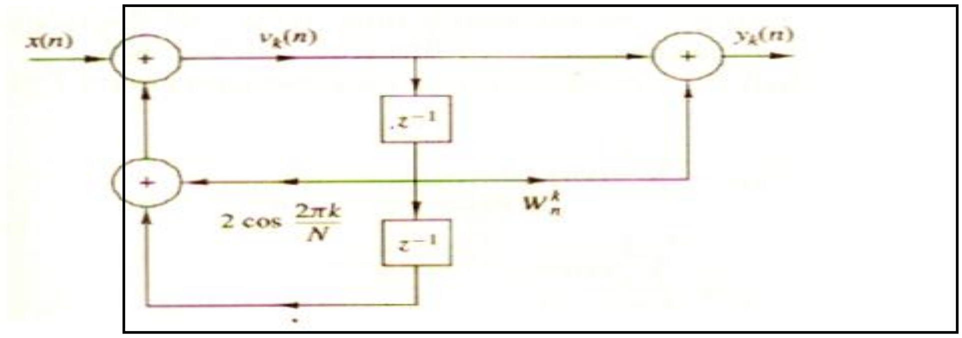


Fig 4.5: Block diagram representation

The recursive relation in (9) is iterated for n = 0,1,……N, but the equation in (10) is computed only once at time n =N. Each iteration requires one real multiplication and two additions. Thus , for a real input sequence x(n) this algorithm requires (N+1) real multiplications to yield X(k) and X(N-k) (this is due to symmetry). Going through the Goertzel algorithm it is clear that this algorithm is useful only when M out of N DFT values need to be computed where $M \le 2\log_2 N$, Otherwise, the FFT algorithm is more efficient method. The utility of the algorithm completely depends on the application and number of frequency components we are looking for.

Chirp z-Transform

Computation of DFT is equivalent to samples of the z-transform of a finite-length sequence at equally spaced points around the unit circle. The spacing between the samples is given by 2π/N. The efficient computation of DFT through FFT requires N to be a highly composite number which is a constraint. Many a times we may need samples of z-transform on contours other than unit circle or we my require dense set of frequency samples over a small region of unit circle. To understand these let us look in to the following situations:

1. Obtain samples of z-transform on a circle of radius 'a' which is concentric to unit circle The possible solution is to multiply the input sequence by a^{-n}
2. 128 samples needed between frequencies ω = -π/8 to +π/8 from a 128 point sequence From the given specifications we see that the spacing between the frequency samples is π/512 or 2π/1024. In order to achieve this freq resolution we take 1024- point FFT of the given 128-point seq by appending the sequence with 896 zeros. Since we need

only 128 frequencies out of 1024 there will be big wastage of computations in this scheme.

For the above two problems Chirp z-transform is the alternative.

Chirp z- transform is defined as:

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} \qquad k = 0,1,......L-1 \qquad (11)$$

Where z_k is a generalized contour. z_k is the set of points in the z-plane falling on an arc which begins at some point z_0 and spirals either in toward the origin or out away from the origin such that the points $\{z_k\}$ are defined as,

$$z_k = r_0 e^{j\theta_0} (R_0 e^{j\phi_0})^k \qquad k = 0,1,....L-1 \qquad (12)$$

Note that,

a. if $R_0 < 1$ the points fall on a contour that spirals toward the origin

b. If $R_0 > 1$ the contour spirals away from the origin

c. If $R_0 = 1$ the contour is a circular arc of radius

d.If $r_0=1$ and $R_0=1$ the contour is an arc of the unit circle.

(Additionally this contour allows one to compute the freq content of the sequence x(n) at dense set of L frequencies in the range covered by the arc without having to compute a large DFT (i.e., a DFT of the sequence x(n) padded with many zeros to obtain the desired resolution in freq.))

e. If $r_0= R_0=1$ and $\theta_0=0$ $\Phi_0=2\pi/N$ and L = N the contour is the entire unit circle similar to the standard DFT. These conditions are shown in the Fig 4.6.

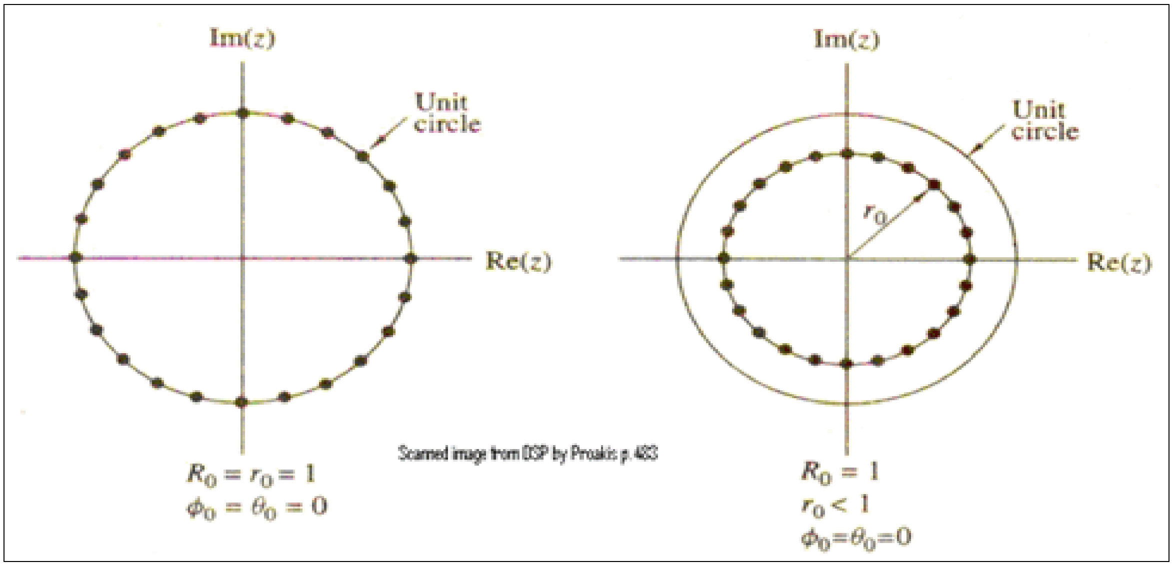
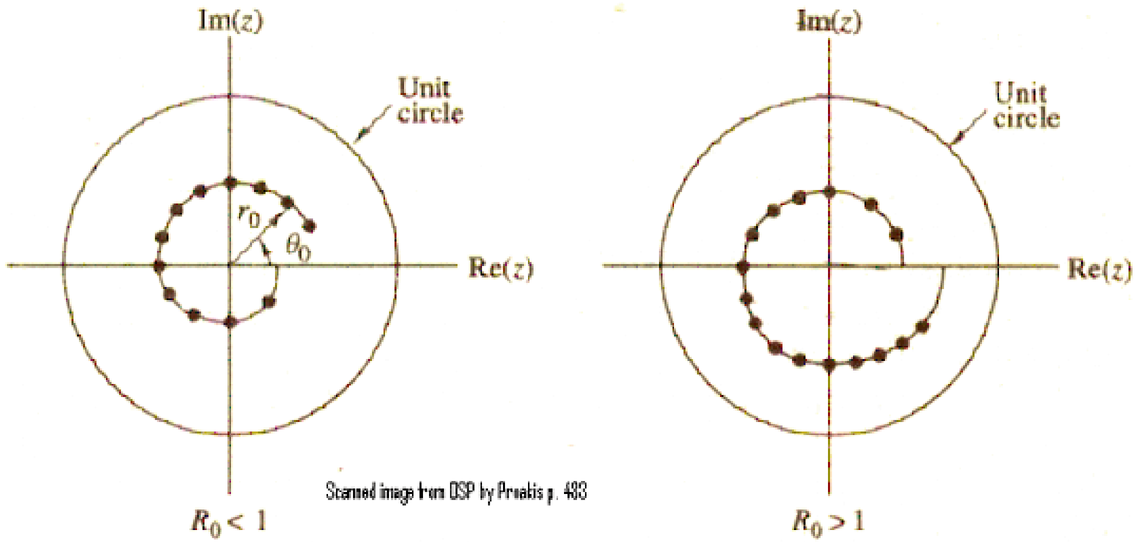


Fig. 4.6: If r0= R0=1 and θ0=0 Φ0=2π/N and L = N



Substituting the value of z_k in the expression of $X(z_k)$

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} = \sum_{n=0}^{N-1} x(n)(r_0 e^{j\theta_0})^{-n} W^{-nk} \qquad (13)$$

where $\quad W = R_0 e^{j\phi_0} \qquad (14)$

Expressing computation of $X(z_k)$ as linear filtering operation:

By substitution of

$$nk = \frac{1}{2}(n^2 + k^2 - (k-n)^2) \qquad (15)$$

we can express $X(z_k)$ as

$$X(z_k) = W^{-k^2/2} y(k) = y(k)/h(k) \qquad k = 0,1,\dots\dots\dots L-1 \qquad (16)$$

Where

$$h(n) = W^{n^2/2} \qquad g(n) = x(n)(r_0 e^{j\theta_0})^{-n} W^{-n^2/2}$$

$$y(k) = \sum_{n=0}^{N-1} g(n) h(k-n) \qquad (17)$$

both g(n) and h(n) are complex valued sequences

Why it is called Chirp z-transform?

If $R_0 = 1$, then sequence h(n) has the form of complex exponential with argument $\omega n = n^2\Phi_0/2 = (n\ \Phi_0/2)\ n$. The quantity $(n\ \Phi_0/2)$ represents the freq of the complex exponential signal, which increases linearly with time. Such signals are used in radar systems are called chirp signals. Hence the name chirp z-transform.

How to Evaluate linear convolution of eq(17)

1. Can be done efficiently with FFT
2. The two sequences involved are g(n) and h(n). g(n) is finite length seq of length N and h(n) is of infinite duration, but fortunately only a portion of h(n) is required to compute L values of X(z), hence FFT could be still be used.

3. Since convolution is via FFT, it is circular convolution of the N-point seq g(n) with an M -point section of h(n) where M > N

4. The concepts used in overlap –save method can be used

5. While circular convolution is used to compute linear convolution of two sequences we know the initial N-1 points contain aliasing and the remaining points are identical to the result that would be obtained from a linear convolution of h(n) and g(n), In view of this the DFT size selected is M = L+N-1 which would yield L valid points and N-1 points corrupted by aliasing. The section of h(n) considered is for $-(N-1) \leq n \leq (L-1)$ yielding total length M as defined

6. The portion of h(n) can be defined in many ways, one such way is,

   $h_1(n) = h(n-N+1) \quad n = 0,1,\ldots..M-1$

7. Compute H1(k) and G(k) to obtain

   $Y_1(k) = G(K)H_1(k)$

8. Application of IDFT will give $y_1(n)$, for

n =0,1,…M-1. The starting N-1 are discarded and desired values are $y_1(n)$ for

N-1 ≤n ≤ M-1 which corresponds to the range 0 ≤n ≤ L-1 i.e.,

$y(n)= y_1(n+N-1)$ n=0,1,2,…..L-1

9. Alternatively $h_2(n)$ can be defined as

$$h_2(n) = h(n) \qquad 0 \le n \le L-1$$
$$= h(n-(N+L-1)) \qquad L \le n \le M-1$$

10. Compute $Y_2(k) = G(K)H_2(k)$, The desired values of $y_2(n)$ are in the range

0 ≤n ≤L-1 i.e.,

$y(n) = y_2(n)$ n=0,1,….L-1

11. Finally, the complex values $X(z_k)$ are computed by dividing y(k) by h(k)
For k =0,1,……L-1

4.3 Computational complexity

In general the computational complexity of CZT is of the order of M $\log_2 M$ complex multiplications. This should be compared with N.L which is required for direct evaluation. If L is small direct evaluation is more efficient otherwise if L is large then CZT is more efficient.

Advantages of CZT

a. Not necessary to have N =L

b.Neither N or L need to be highly composite

c.The samples of Z transform are taken on a more general contour that includes the unit circle as a special case.

Example to understand utility of CZT algorithmin freq analysis

(ref: DSP by Oppenheim Schaffer)

CZT is used in this application to sharpen the resonances by evaluating the z-transform off the unit circle. Signal to be analyzed is a synthetic speech signal generated by exciting a five-pole system with a periodic impulse train. The system was simulated to correspond to a sampling freq. of 10 kHz. The poles are located at center freqs of 270,2290,3010,3500 & 4500 Hz with bandwidth of 30, 50, 60,87 & 140 Hz respectively.

Solution: Observe the pole-zero plots and corresponding magnitude frequency response for different choices of |w|. The following observations are in order:

- The first two spectra correspond to spiral contours outside the unit circle with a resulting broadening of the resonance peaks
- |w| = 1 corresponds to evaluating z-transform on the unit circle
- The last two choices correspond to spiral contours which spirals inside the unit circle and close to the pole locations resulting in a sharpening of resonance peaks.

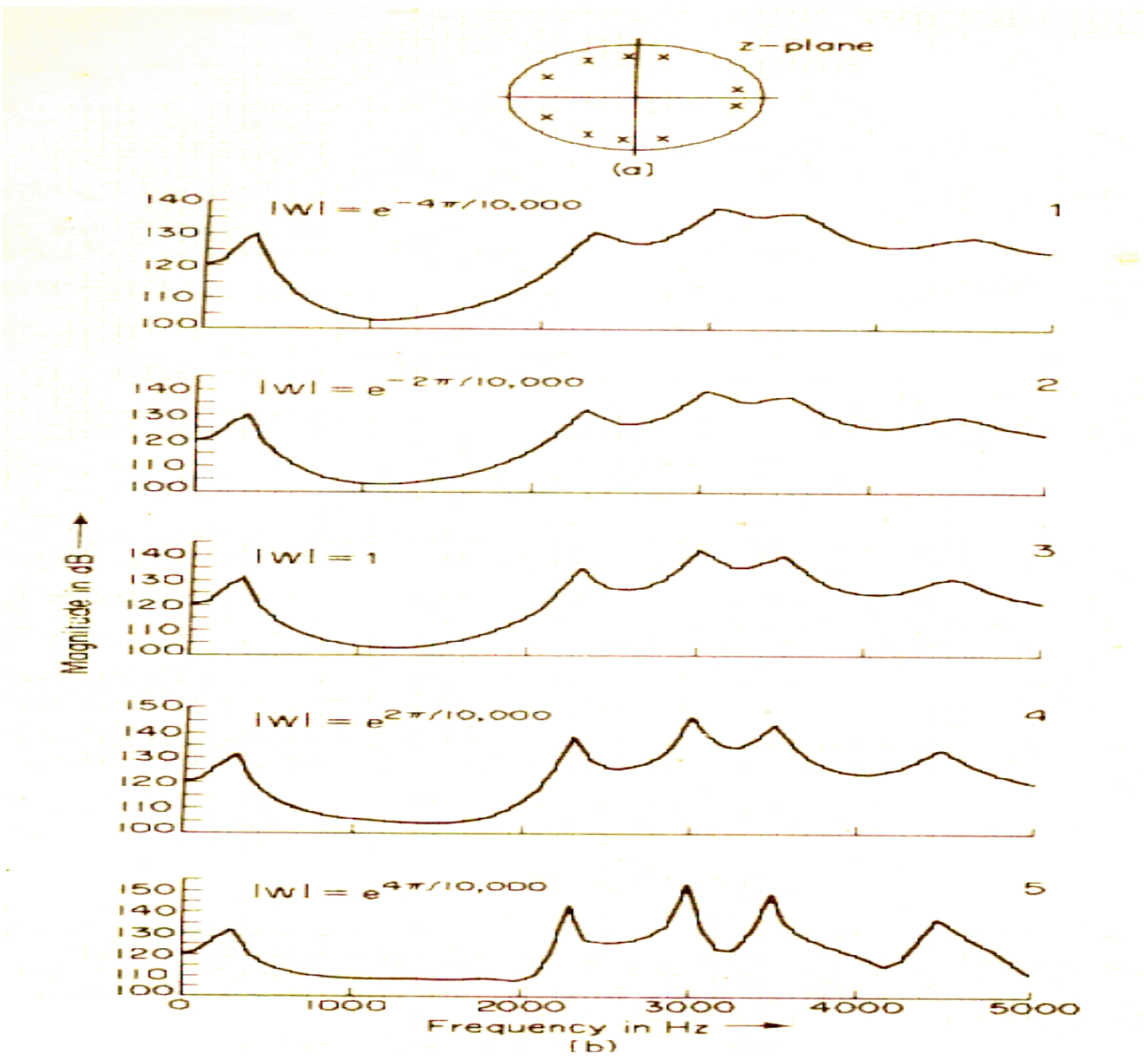


Fig. 4.7: magnitude frequency response for different choices of |w|

Implementation of CZT in hardware to compute the DFT signals

The block schematic of the CZT hardware is shown in down fig. 4.8. DFT computation requires $r_0 = R_0 = 1$, $\theta_0 = 0$ $\Phi_0 = 2\pi/N$ and $L = N$.

The cosine and sine sequences in h(n) needed for pre multiplication and post multiplication are usually stored in a ROM. If only magnitude of DFT is desired, the post multiplications are unnecessary,

In this case |X(zk)| = |y(k)| k =0,1,….N-1

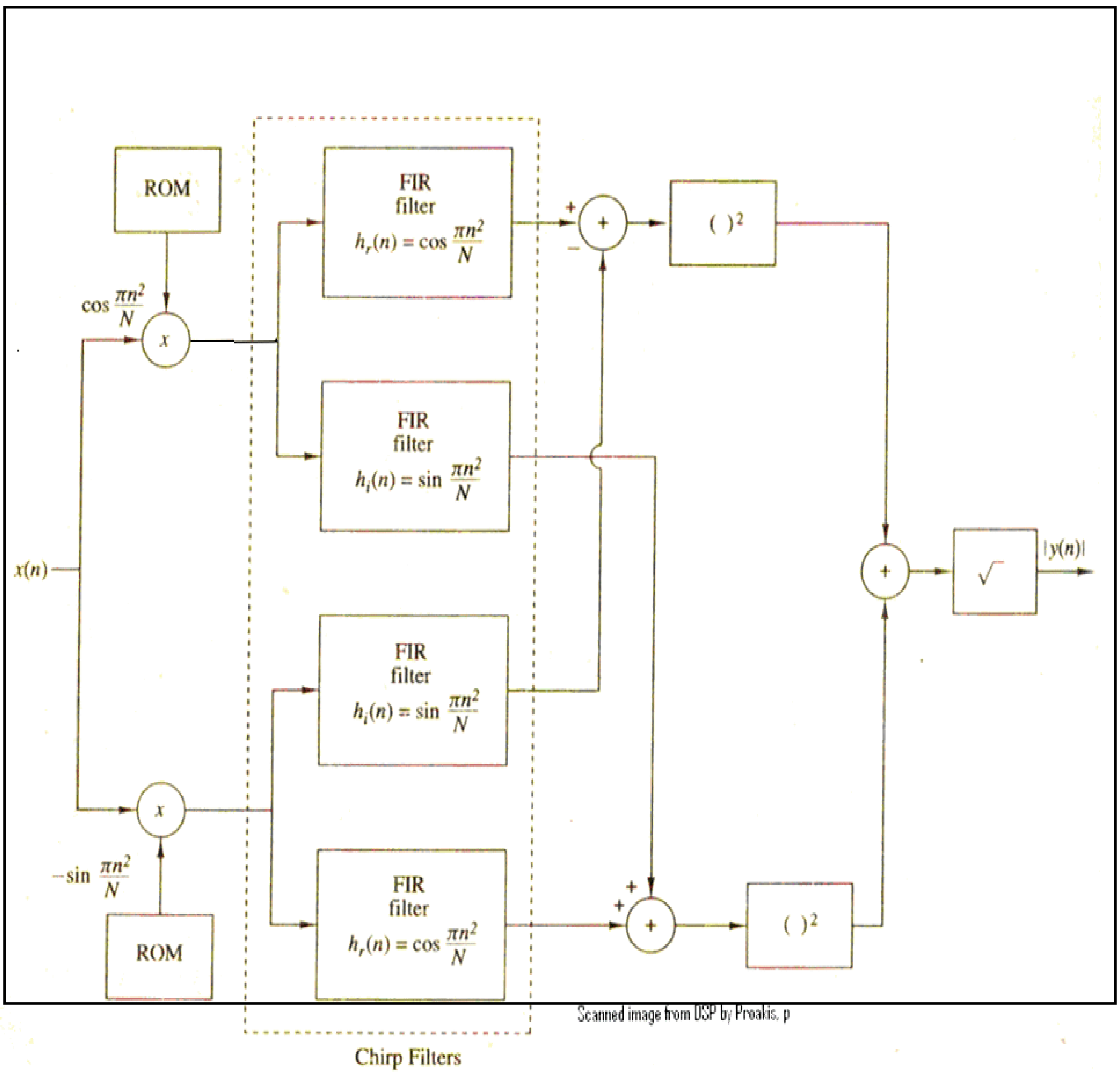


Fig. 4.8: block schematic of the CZT hardware

4.5:Additional Problems

1. Draw the flow graph for the decimation in frequency (DIF) SRFFT algorithm for N=16. What is the number of nontrivial multiplications?

Solution :- There are 20 real , non trial multiplications

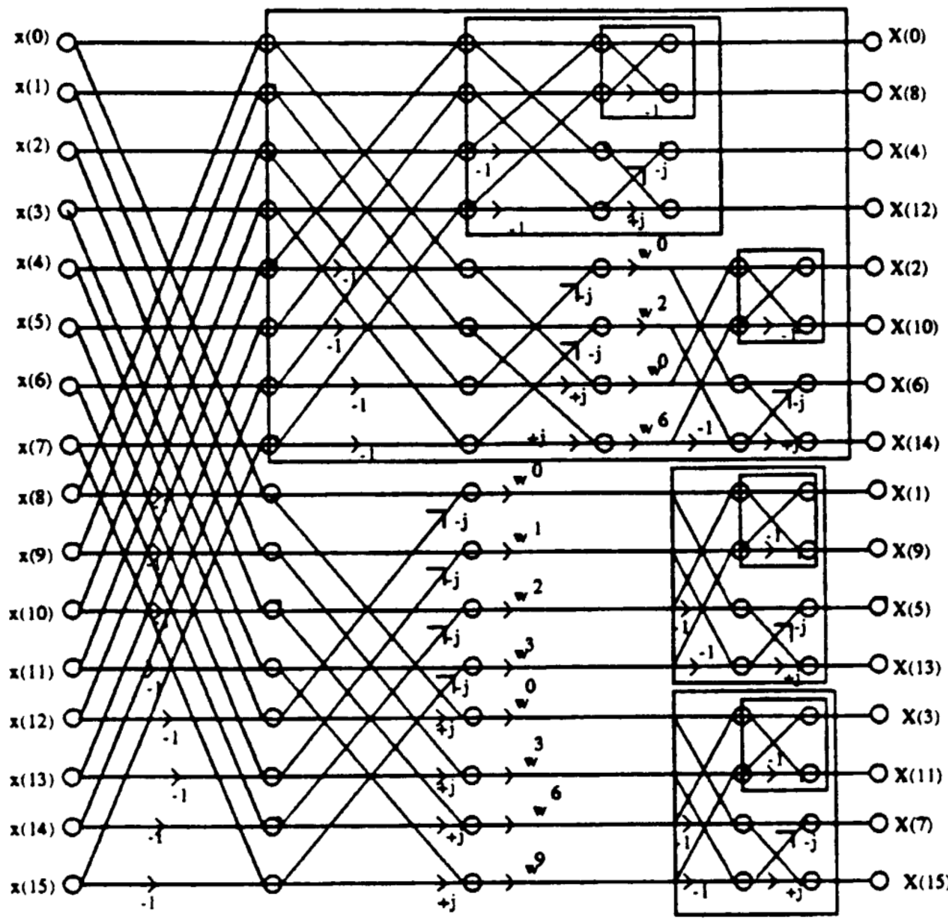


Fig. 4.9 DIF Algorithm for N=16

2. Explain how the DFT can be used to compute N equispaced samples of the Z-transform, of an N-point sequence, on a circle of radius r.

Solution:-

$$
\begin{aligned}
X(z) &= \sum_{n=0}^{N-1} x(n) z^{-n} \\
\text{Hence, } X(z_k) &= \sum_{n=0}^{N-1} x(n) r^{-n} e^{-j\frac{2\pi}{N}kn}
\end{aligned}
$$

where $z_k = re^{-j\frac{2\pi}{N}k}, k = 0, 1, \ldots, N-1$ are the N sample points. It is clear that $X(z_k), k = 0, 1, \ldots, N-1$ is equivalent to the DFT (N-pt) of the sequence $x(n)r^{-n}, n \in [0, N-1]$.

<u>Q3.</u>

Let $X(k)$ be the N-point DFT of the sequence $x(n)$, $0 \leq n \leq N-1$. What is the N-point DFT of the sequence $s(n) = X(n)$, $0 \leq n \leq N-1$?

Solution:-

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

Let $F(t)$, $t = 0, 1, \ldots, N-1$ be the DFT of the sequence on k $X(k)$.

$$
\begin{aligned}
F(t) &= \sum_{k=0}^{N-1} X(k) W_N^{tk} \\
&= \sum_{k=0}^{N-1} \left[\sum_{n=0}^{N-1} x(n) W_N^{kn} \right] W_N^{tk} \\
&= \sum_{n=0}^{N-1} x(n) \left[\sum_{k=0}^{N-1} W_N^{k(n+t)} \right] \\
&= \sum_{n=0}^{N-1} x(n) \delta(n+t)_{\text{mod N}} \\
&= \sum_{n=0}^{N-1} x(n) \delta(N-1-n-t) \qquad t = 0, 1, \ldots, N-1 \\
&= \{x(N-1), x(N-2), \ldots, x(1), x(0)\}
\end{aligned}
$$

4. **Develop a radix-3 decimation-in-time FFT algorithm for N=3 and draw the corres Ponding flow graph for N=9. What is the number of required complex multiplications? Can the operations be performed in place.**

Solution:-

$$\begin{aligned} Y(k) &= \sum_{n=0}^{8} y(n) W_9^{nk} \\ &= \sum_{n=0,3,6} y(n) W_9^{nk} + \sum_{n=1,4,7} y(n) W_9^{nk} + \sum_{n=2,5,8} y(n) W_9^{nk} \\ &= \sum_{m=0}^{2} y(3m) W_9^{3km} + \sum_{m=0}^{2} y(3m+1) W_9^{(3m+1)k} + \sum_{m=0}^{2} y(3m+2) W_9^{(3m+2)k} \\ &= \sum_{m=0}^{2} y(3m) W_3^{km} + \sum_{m=0}^{2} y(3m+1) W_3^{mk} W_9^{k} + \sum_{m=0}^{2} y(3m+2) W_3^{mk} W_9^{2k} \end{aligned}$$

5. **Determine the system function H(z) and the Difference equation for the system That uses the geortzel algorithm to compute the DFT value X(N-k).**

Solution:-

$$\begin{aligned} X(k) &= \sum_{m=0}^{N-1} x(m) W_N^{km} \\ &= \sum_{m=0}^{N-1} x(m) W_N^{km} W_N^{-kN} \text{ since } W_N^{-kN} = 1 \\ &= \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)} \end{aligned}$$

This can be viewed as the convolution of the N-length sequence x(n) with implulse response of a linear filter

$$\begin{aligned} h_k(n) &\triangleq W_N^{kn} u(n), \text{ evaluated at time N} \\ H_k(z) &= \sum_{n=0}^{\infty} W_N^{kn} z^{-n} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{1 - W_N^k z^{-1}} \\
&= \frac{Y_u(z)}{X(z)} \\
y_k(n) &= W_N^k y_k(n-1) + x(n), \qquad y_k(-1) = 0 \\
y_k(N) &= X(k)
\end{aligned}
$$

Outcomes

Apply Fast Fourier Transforms Algorithm for computing DFT and inverse DFT of a given sequence

Further Reading

1. https://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm

2. https://www.youtube.com/watch?v=Dz8EgbjHN8w

3. https://www.youtube.com/watch?v=EsJGuI7e_ZQ