MODULE 2

MODULE 2

Linear filtering methods based on the DFT: Use of DFT in Linear Filtering, Filtering of Long data Sequences.

Fast-Fourier-Transform (FFT) algorithms: Efficient Computation of the DFT: Radix-2 FFT algorithms for the computation of DFT and IDFT–decimation-in-time and decimation-in-frequency algorithms.

Text Book/Reference Books:

1. Proakis & Monalakis, “Digital signal processing – Principles Algorithms & Applications”, 4th Edition, Pearson education, New Delhi, 2007. ISBN: 81-317-1000-9.
2. Oppenheim & Schaffer, “Discrete Time Signal Processing” , PHI, 2003.
3. D.Ganesh Rao and Vineeth P Gejji, “Digital Signal Processing” Cengage India Private Limited, 2017, ISBN: 9386858231

LINEAR FILTERING METHODS BASED ON DFT

- USE OF DFT IN LINEAR FILTERING:

- Let a finite duration sequence $x(n)$ of length L excites a FIR filter of length M.
- Let $x(n) = 0\ , n < 0\ and\ n \geq L$ and $\mathrm{h}(n) = 0\ , n < 0\ and\ n \geq M$, where $\mathrm{h}(n)$ is the impulse response of the filter.
- The output sequence $\mathrm{y}(n)$ of the FIR filter represents the time-domain convolution of $x(n)$ and $\mathrm{h}(n)$, $\mathrm{y}(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$.
- Since $x(n)$ and $\mathrm{h}(n)$ are finite duration sequences, $\mathrm{y}(n)$ is also finite of length L+M-1.
- In frequency domain $Y(\omega) = X(\omega)H(\omega)$.
- In terms of DFT, $Y(k) = X(k)H(k) \qquad k = 0,1,2 \ldots .N - 1$.
- N-point circular convolution of $x(n)$ and $\mathrm{h}(n)$ is equivalent to their linear convolution.

...LINEAR FILTERING METHODS BASED ON DFT...

- FILTERING OF LONG DATA SEQUENCES:

- In some real time processing of applications involving linear filtering of signals, the input sequence $x(n)$ is a very long sequence.
- Linear filtering on such signals involving DFT imposes severe memory constraints.
- In such filtering the sequences are processed block-wise and the output blocks are fitted together to form the overall sequence.
- There are two methods for this:

- Overlap Save Method:

- Size of input data block N=L+M-1 and size of DFT and IDFT is N.

...LINEAR FILTERING METHODS BASED ON DFT...

- Each data block consists of M-1 points of previous block and N new points.
- N-point DFT is computed per block.
- Impulse response of filter is increased by appending L-1 zeros and its DFT computed.
- For mth block we have, $\hat{Y}_m(k) = H(k)X_m(k)$, multiplication of two N-point DFTs $H(k)$ and $X_m(k)$.
- Then N-point IDFT is computed to yield, $\hat{Y}_m(n) = \{\hat{y}_m(0)\ \hat{y}_m(1)\ldots\ \hat{y}_m(M-1)$ $\hat{y}_m(M)\ \ldots \hat{y}_m(N-1)\}$.
- For a N-length record, the first M-1 datapoints of $y_m(n)$ are corrupted by aliasing and discarded.
- Last L points of $y_m(n)$ are that similar to linear convolution.

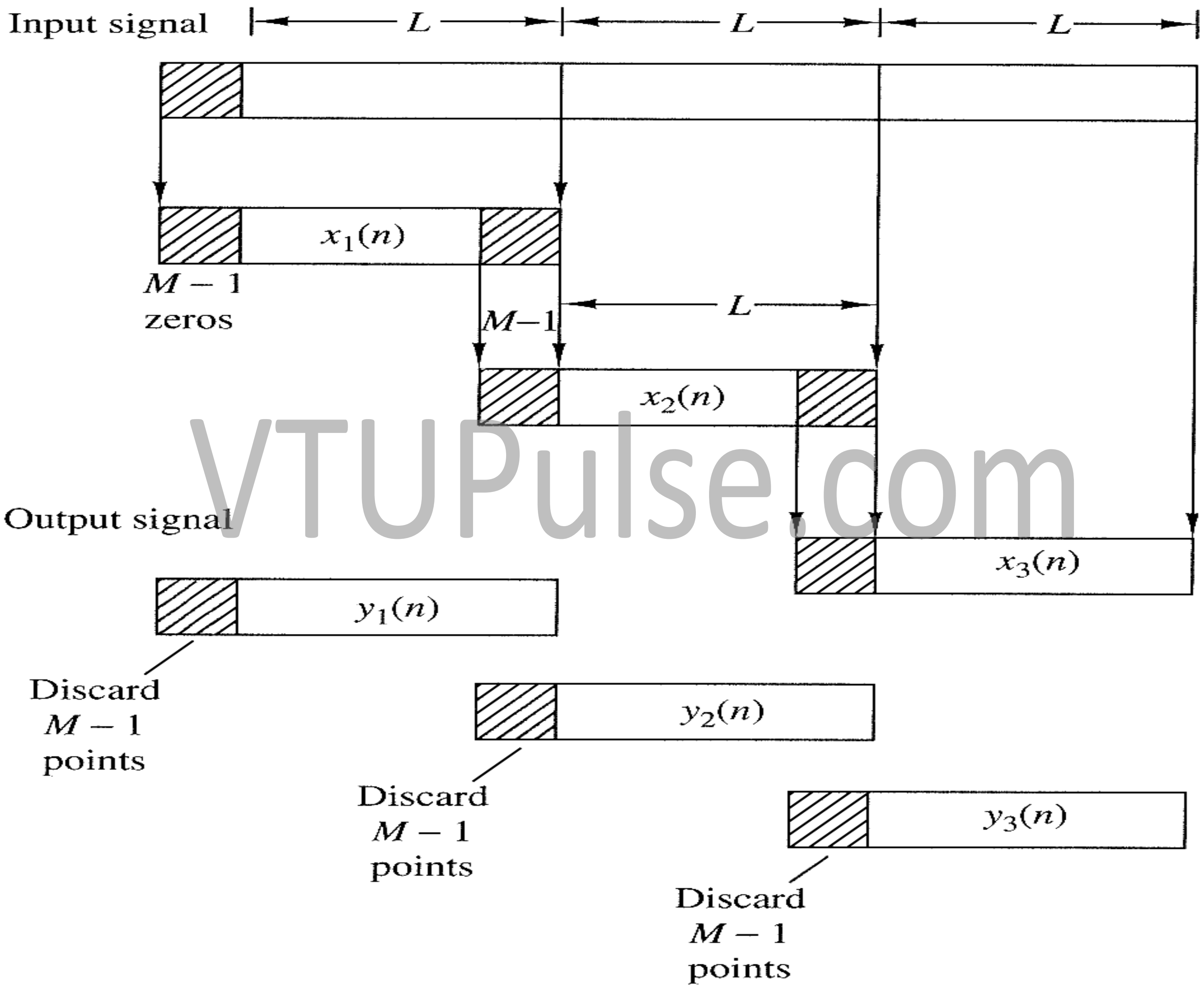
...LINEAR FILTERING METHODS BASED ON DFT...

- To avoid data loss due to aliasing, the last M-1 points of each record are made the first M-1 points of subsequent record.
- To process the data, the first M-1 points of first block is set to zero.
- The block of data is as follows:

$$x_1(n) = \{\underbrace{0, 0, \ldots, 0,}_{M-1 \text{ points}} x(0), x(1), \ldots, x(L-1)\}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots, x(L-1),}_{M-1 \text{data points from } x_1(n)} \underbrace{x(L), \ldots, x(2L-1)}_{L \text{ new data points}}\}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots, x(2L-1),}_{M-1 \text{ data points from } x_2(n)} \underbrace{x(2L), \ldots, x(3L-1)}_{L \text{ new data points}}\}$$


Input signal
L
L
L
x1(n)
M − 1
zeros
M−1
L
x2(n)
Output signal
x3(n)
y1(n)
Discard
M − 1
points
y2(n)
Discard
M − 1
points
y3(n)
Discard
M − 1
points

OVERLAP ADD EXAMPLE

Given ***x[n]={3,-1,0,1,3,2,0,1,2,1}*** & ***h[n]={1,1,1}***

Let, L=5

Length of h[n], M= 3 , Therefore, M-1= 2

We know, L=(N+M-1)

5=N+3-1

N=3

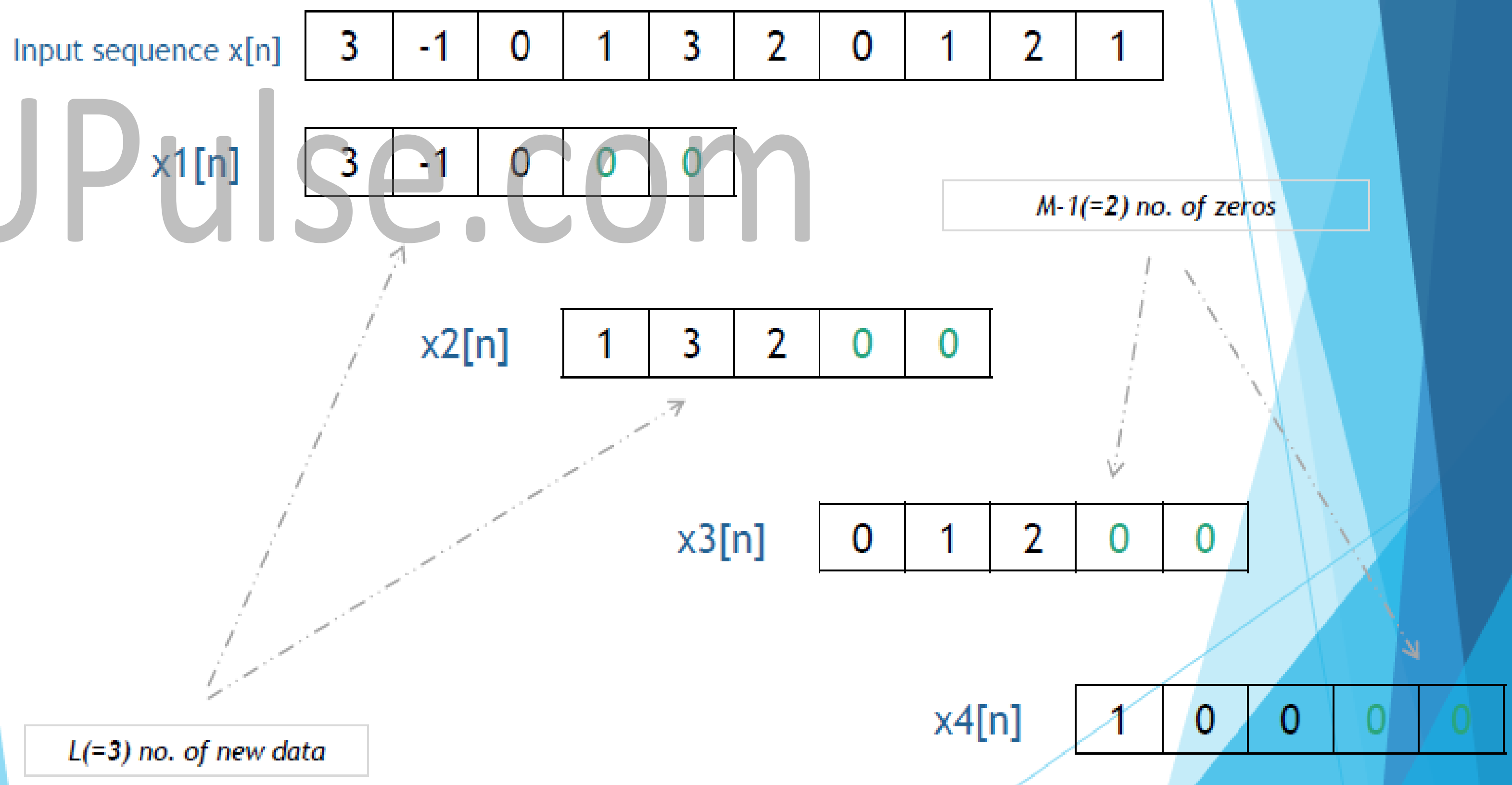
∴Pad M-1=2 zeros with h[n]

i.e. ***h[n]={1,1,1,0,0}***

Performing yk[n]= xk[n] h[n],

where k=1,2,3,4

1. y1[n]= {3,2,2,-1,0}
2. y2[n]= {1,4,6,5,2}
3. y3[n]= {0,1,3,3,2}
4. y4[n]= {1,1,1,0,0}



+=> add

n-	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$y_1[n]$	3	2	2	-1	0									
$y_1[n]$				1	4	6	5	2						
$y_1[n]$							0	1	3	3	2			
$y_1[n]$										1	1	1	0	0
$y[n]$	3	2	2	0	4	6	5	3	3	4	3	1	0	0

...LINEAR FILTERING METHODS BASED ON DFT...

- Overlap Add method:

- Size of input data block L and size of DFT and IDFT is N.
- Each data block is appended with M-1 zeros and DFT calculated.
- The block of data is as follows:

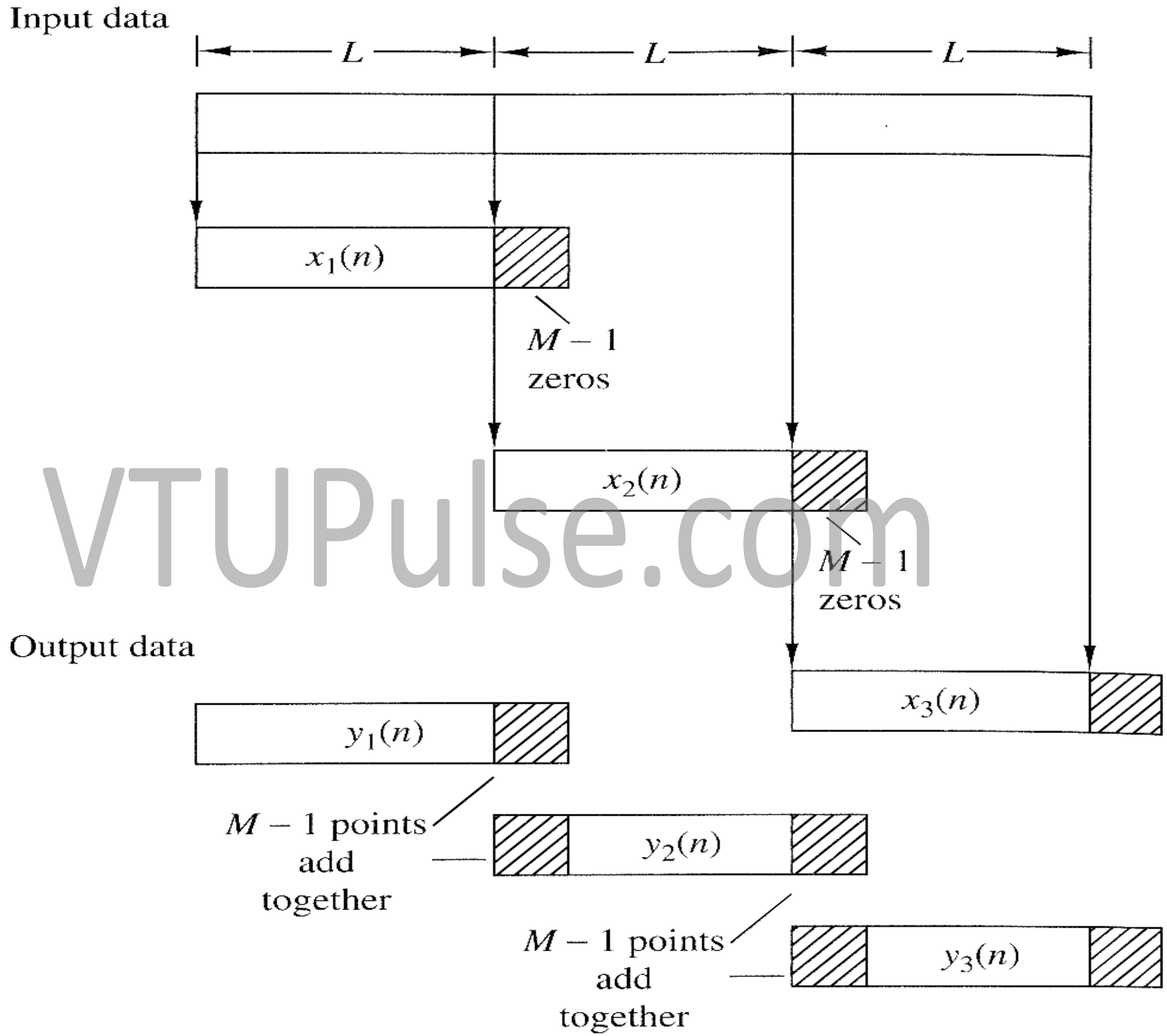
$$x_1(n) = \{x(0), x(1), \ldots, x(L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_2(n) = \{x(L), x(L+1), \ldots, x(2L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_3(n) = \{x(2L), \ldots, x(3L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

...LINEAR FILTERING METHODS BASED ON DFT...

- The two N-point DFTs are multiplied to obtain $Y_m(k) = H(k)X_m(k)$.
- IDFT yields N-points that are free of aliasing.
- Since each data block is terminated with M-1 zeros, the last M-1 points of each block is added to M-1 points of subsequent block.
- Overlapping and adding yields, $y(n) = \{y_1(0), y_1(1), .. y_1(L-1), y_1(L) + \{y_2(1), \ldots y_1(N-1) + y_2(M-1), y_1(M), \ldots\}$


Input data
L
L
L
$x_1(n)$
M − 1
zeros
$x_2(n)$
M − 1
zeros
Output data
$x_3(n)$
$y_1(n)$
M − 1 points
add
together
$y_2(n)$
M − 1 points
add
together
$y_3(n)$

OVERLAP SAVE EXAMPLE

Given ***x[n]={3,-1,0,1,3,2,0,1,2,1}*** & ***h[n]={1,1,1}***

Let, L=5

Length of h[n], M= 3, Therefore, M-1= 2

We know, L=(N+M-1)

5=N+3-1

N=3

∴Pad M-1=2 zeros with h[n] i.e. ***h[n]={1,1,1,0,0}***

x1[n]=(0 0 3 -1 0) x2[n]=(-1 0 1 3 2) x3[n]=(3 2 0 1 2) x4[n]=(1 2 1 0 0)

Performing yk[n]= xk[n] h[n], where k=1,2,3,4

1. y1[n]= {-1,0,3,2,2}
2. y2[n]= {4,1,0,4,6}
3. y3[n]= {6,7,5,3,3}
4. y4[n]= {1,3,4,3,1}

X=> discard

n-	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
$y_1[n]$	-1	0	3	2	2									
$y_1[n]$				4	1	0	4	6						
$y_1[n]$							6	7	5	3	3			
$y_1[n]$										1	3	4	3	1
$y[n]$			3	2	2	0	4	6	5	3	3	4	3	1

OVERLAP SAVE VS ADD METHOD

Overlap Save

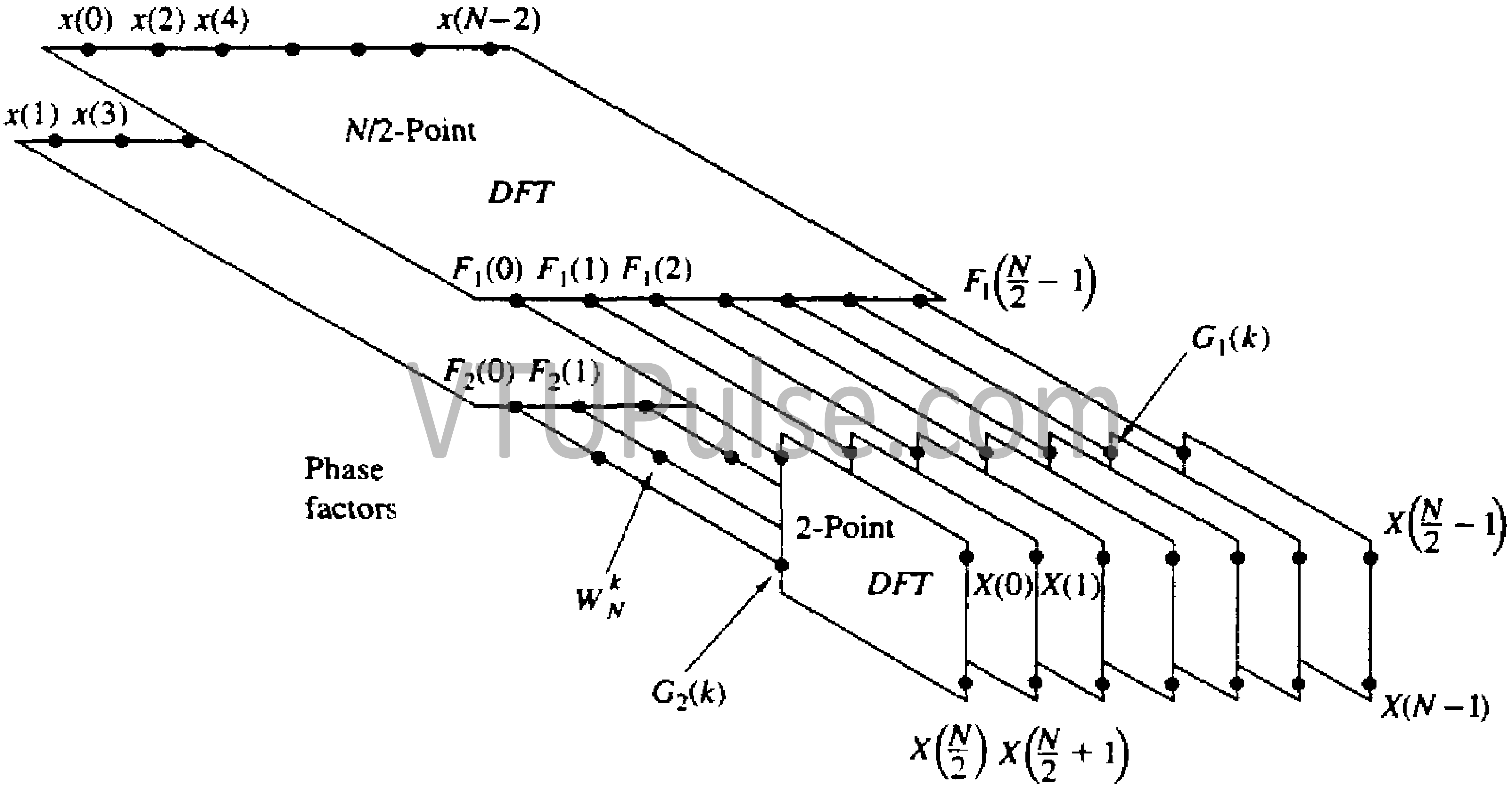
- Overlapped values has to be discarded.
- It does not require any addition.
- It can be computed using linear convolution

Overlap Add

- Overlapped values has to be added.
- It will involve adding a number of values in the output.
- Linear convolution is not applicable here.

RADIX-2FFT

- We consider N-point DFT where $N = 2^v$ and select M=N/2 and L=2.
- We split N-point data sequence into two N/2 sequences $f_1(n) = x(2n)$ and $f_2(n) = x(2n+1), \mathrm{n} = 0,1,2 \ldots \frac{\mathrm{N}}{2} - 1.$
- $f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2-decimation in time.
- DFT of decimated sequence is $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad k = 0,1,2..N-1 = \sum_{n\ even} x(n) W_N^{kn} + \sum_{n\ odd} x(n) W_N^{kn} = \sum_{m=0}^{\frac{N}{2}-1} x(2m) W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) W_N^{(2m+1)k}, W_N^2 = W_{\frac{N}{2}}.$
- $\mathrm{X(k)} = \sum_{m=0}^{\frac{N}{2}-1} f_1(m) W_{\frac{N}{2}}^{2mk} + W_N^k \sum_{m=0}^{\frac{N}{2}-1} f_2(m) W_{\frac{N}{2}}^{km} = F_1(k) + W_N^k F_2(k), \mathrm{k} = 0,1,2 \ldots \mathrm{N} - 1.$
- Periodic sequences, $F_1\left(k + \frac{N}{2}\right) = F_1(k)$ and $F_2\left(k + \frac{N}{2}\right) = F_2(k)$. Also $W_N^{k+\frac{N}{2}} = -W_N^k$.
- $X(k) = F_1(k) + W_N^k F_2(k)$ and $X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k), k = 0,1,2 \ldots \frac{N}{2} - 1.$
- Computation of $X(k)$ requires $\frac{N^2}{2} + \frac{N}{2}$. Also $G_1(k) = F_1(k)$ and $G_2(k) = W_N^k F_2(k), k = 0,1,2 \ldots \frac{N}{2} - 1.$
- $X(k) = G_1(k) + G_2(k)$ and $X\left(k + \frac{N}{2}\right) = G_1(k) - G_2(k), k = 0,1,2 \ldots \frac{N}{2} - 1.$



x(0) x(2) x(4) x(N−2)
x(1) x(3)
N/2-Point
DFT
F1(0) F1(1) F1(2)
F1(N/2 − 1)
G1(k)
F2(0) F2(1)
Phase
factors
2-Point
DFT
X(0) X(1)
X(N/2 − 1)
W_N^k
G2(k)
X(N − 1)
X(N/2) X(N/2 + 1)

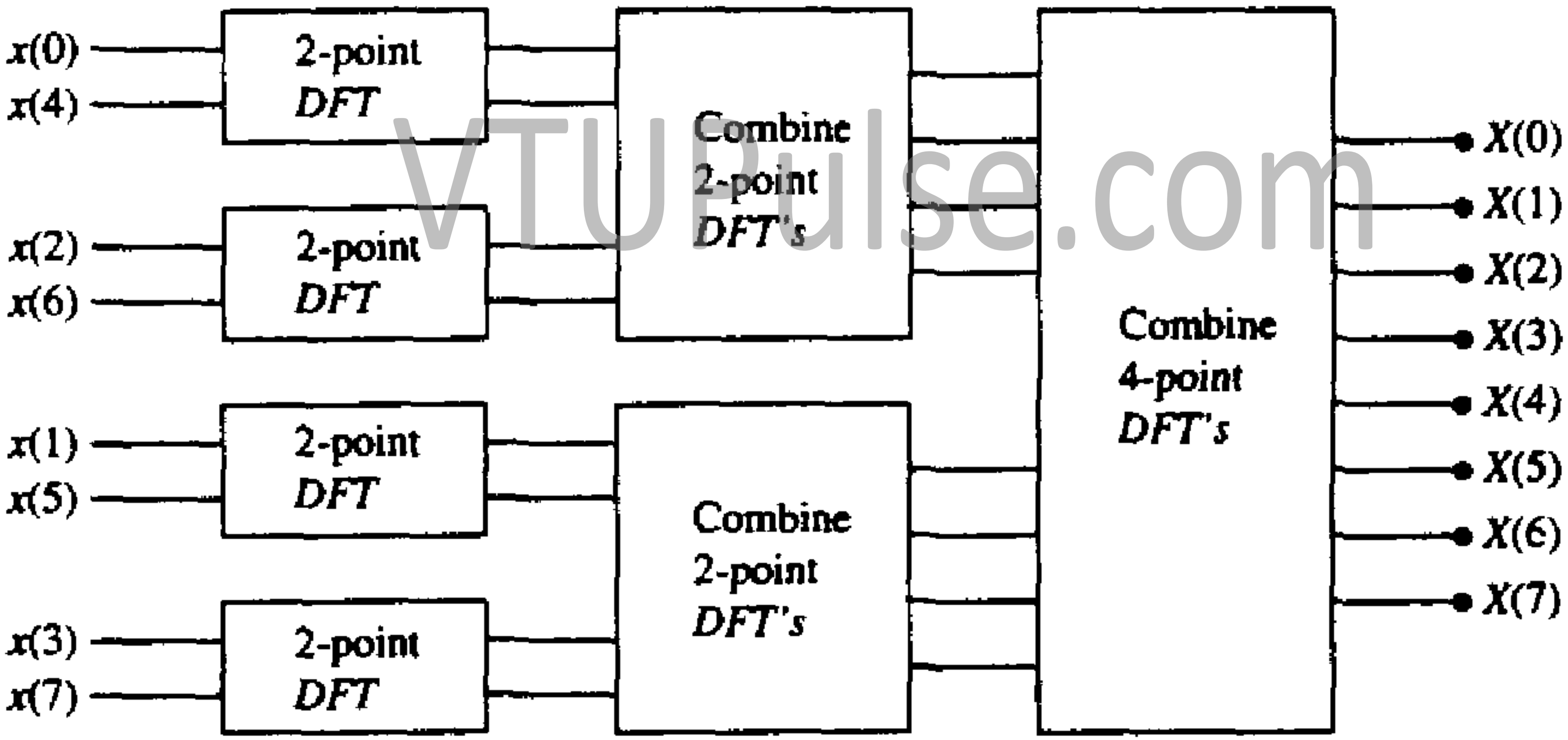
...RADIX-2FFT...

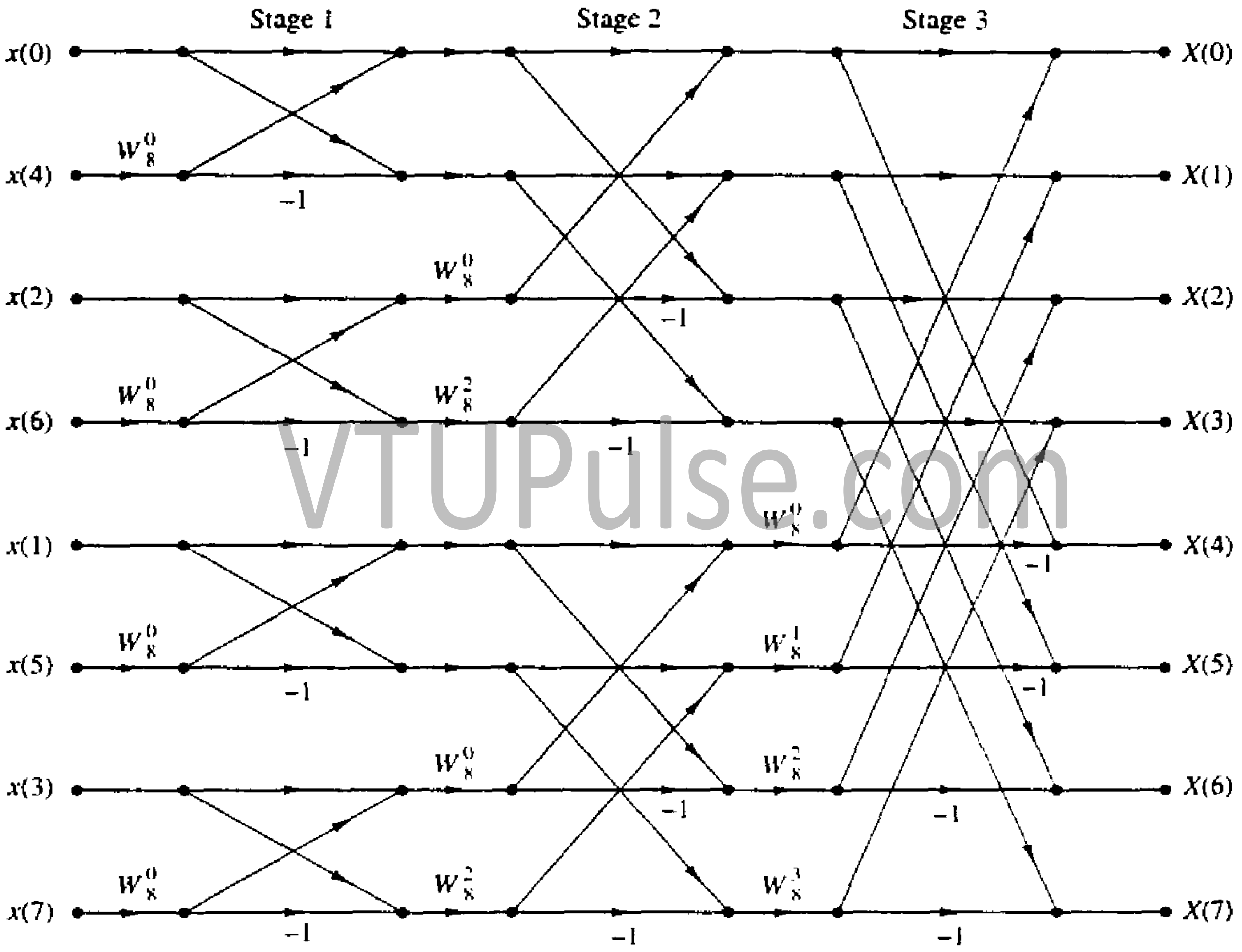
- The process is repeated for $f_1(n)$ and $f_2(n)$.
- $f_1(n)$ results in two N/4 point sequences-$v_{11}(n) = f_1(2n)$ and $v_{12}(n) = f_1(2n+1), \mathrm{k} = 0,1,2..\frac{\mathrm{N}}{4} - 1.$
- $f_2(n)$ results in two N/4 point sequences-$v_{21}(n) = f_1(2n)$ and $v_{22}(n) = f_2(2n+1), \mathrm{k} = 0,1,2..\frac{\mathrm{N}}{4} - 1.$

- By using these we obtain N/2-point DFTs: $F_1(k) = V_{11}(k) + W_{N/2}^{k} V_{12}$, $F_1\left(k + \frac{N}{4}\right) = V_{11}(k) - W_{N/2}^{k} V_{12}$, $F_2(k) = V_{21}(k) + W_{N/2}^{k} V_{22}$, $F_2\left(k + \frac{N}{4}\right) = V_{21}(k) - W_{\frac{N}{2}}^{k} V_{22}, \mathrm{k} = 0,1,2 \ldots \frac{\mathrm{N}}{4} - 1.$
- $F_1(k)$ and $F_2(k)$ can be accomplished with $\frac{N^2}{4} + N$ complex multiplications.
- Decimation is repeated again and again till one point sequence is obtained. Operation can be performed $v = \log_2 N$ times.
- Number of complex multiplications: $\frac{N}{2} \log_2 N$. Number of complex additions: $\mathrm{N} \log_2 N$.

...RADIX-2FFT...

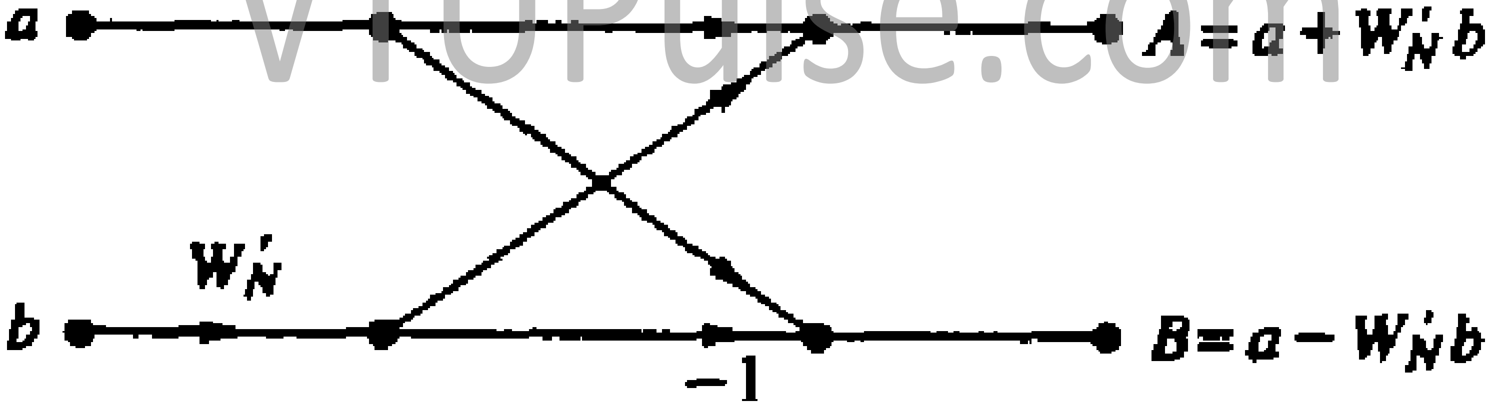
- N=8 point DFT computation is as follows.




Stage 1
Stage 2
Stage 3
x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)
W_8^0
W_8^2
W_8^1
W_8^3
−1

...RADIX-2FFT...

- Butterfly Diagram

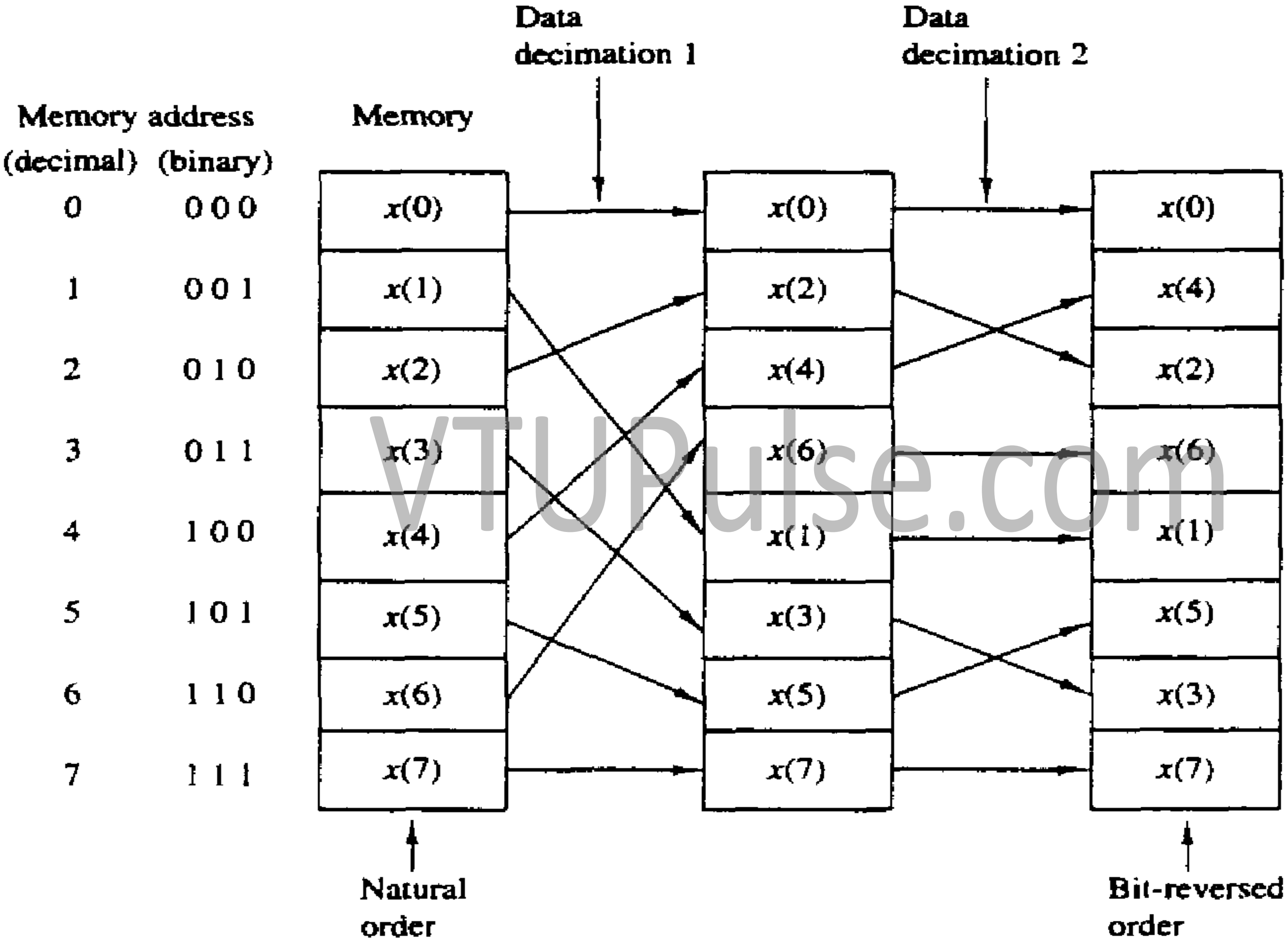


...RADIX-2FFT...

- Data points to be stored are 2N in bit reversed manner.
- Decimation in frequency FFT algorithm uses M=2 and L=N/2.
- $X(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{kn} = \sum_{n=0}^{N-1} x(n)W_N^{kn} + W_N^{\frac{kN}{2}} \sum_{n=0}^{N/2-1} x\left(n+\frac{N}{2}\right)W_N^{kn}$.

$W_N^{\frac{kN}{2}} = (-1)^k,$

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



Data decimation 1
Data decimation 2
Memory address
(decimal) (binary)
Memory
0 000
1 001
2 010
3 011
4 100
5 101
6 110
7 111
x(0)
x(1)
x(2)
x(3)
x(4)
x(5)
x(6)
x(7)
x(0)
x(2)
x(4)
x(6)
x(1)
x(3)
x(5)
x(7)
x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
Natural order
Bit-reversed order

$(n_2 n_1 n_0)$		$(n_0 n_2 n_1)$		$(n_0 n_1 n_2)$
(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)

- Shuffling of data and bit reversal

Use the 8 point redix-2 DIT-FFT algorithm to find the DFT of the sequence x(n)={0.707,1, 0.707,0,-0.707, -1,-0.707,0}

Solution:
Based on the signal flow graph it is first we have to determine the two point DFT

$$\begin{aligned} V_{11}(0) &= x(0) + W_8^0 x4 \\ &= 0.707 + 1(-0.707) = 0 \\ V_{11}(1) &= x(0) - W_8^0 x4 \\ &= 0.707 - 1(-0.707) = 1.414 \end{aligned}$$

$$\begin{aligned} V_{11}(1) &= x(0) - W_8^0 + x4 \\ &= 0.707 - 1(-0.707) = 1.414 \end{aligned}$$

$$\begin{aligned} V_{21}(0) &= x(1) + W_8^0 x5 \\ &= 1 + 1(-1) = 0 \\ V_{21}(1) &= x(1) - W_8^0 x5 \\ &= 1 - 1(-1) = 2 \end{aligned}$$

$$\begin{aligned} V_{12}(0) &= x(2) + W_8^0 x6 \\ &= 0.707 + 1(-0.707) = 0 \\ V_{12}(1) &= x(2) - W_8^0 x6 \\ &= 0.707 - 1(-0.707) = 1.414 \end{aligned}$$

$$\begin{aligned} V_{22}(0) &= x(3) + W_8^0 x7 \\ &= 0 + 1(0) = 0 \\ V_{22}(1) &= x(1) - W_8^0 x5 \\ &= 1 - 1(0) = 0 \end{aligned}$$

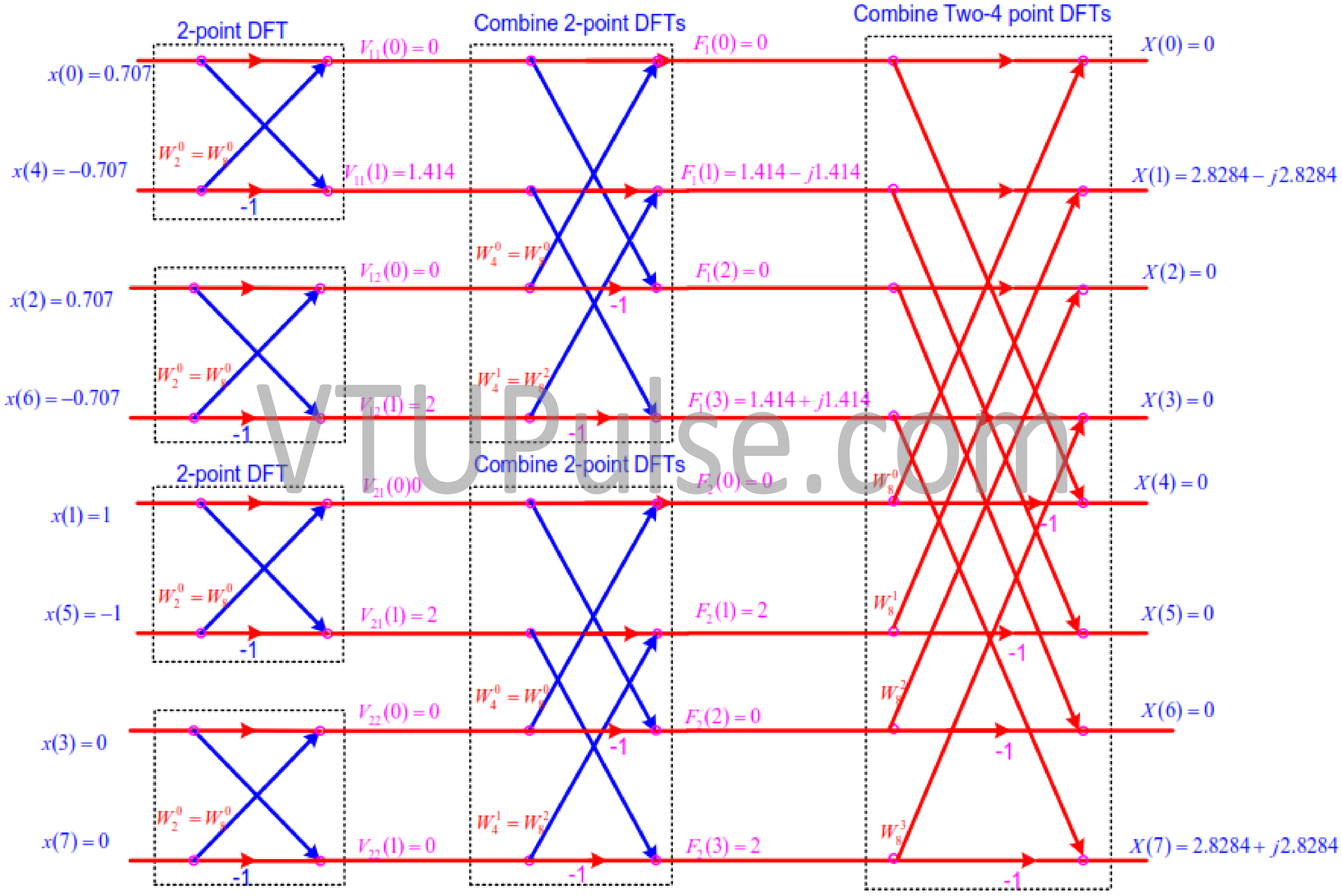
$$
\begin{aligned}
F_1(0) &= V_{11}(0) + W_8^0 V_{12}(0) \\
&= 0 + 1(0) = 0 \\
F_1(1) &= V_{11}(1) + W_8^0 V_{12}(1) \\
&= 1.414 + (-j)1.414 = 1.414 - j1.414
\end{aligned}
$$

$$
\begin{aligned}
F_1(2) &= V_{11}(0) - W_8^0 V_{12}(0) \\
&= 0 - 1(0) = 0 \\
F_1(3) &= V_{11}(1) - W_8^0 V_{12}(1) \\
&= 1.414 - (-j)1.414 = 1.414 + j1.414
\end{aligned}
$$

$$
\begin{aligned}
F_2(0) &= V_{21}(0) + W_8^0 V_{22}(0) \\
&= 0 + 1(0) = 0 \\
F_2(1) &= V_{21}(1) + W_8^0 V_{22}(1) \\
&= 2 + (-j)0 = 2
\end{aligned}
$$

$$
\begin{aligned}
F_2(2) &= V_{21}(0) - W_8^0 V_{22}(0) \\
&= 0 - 1(0) = 0 \\
F_2(3) &= V_{21}(1) - W_8^0 V_{22}(1)
\end{aligned}
$$

$$
\begin{aligned}
X(0) &= F_1(0) + W_8^0 F_2(0) \\
&= 0 + 1(0) = 0 \\
X(1) &= F_1(1) + W_8^1 F_2(1) \\
&= (1.414 - j1.414) + (0.7071 - j0.7071)2 \\
&= 2.8284 - j2.8284 \\
X(2) &= F_1(2) + W_8^2 F_2(2) \\
&= 0 + (-j)(0) = 0 \\
X(3) &= F_1(3) + W_8^3 F_2(3) \\
&= (1.414 + j1.414) + (-0.7071 - j0.7071)2 = 0 \\
X(4) &= F_1(0) - W_8^0 F_2(0) \\
&= 0 + 1(0) = 0 \\
X(5) &= F_1(1) - W_8^1 F_2(1) \\
&= (1.414 - j1.414) - (0.7071 - j0.7071)2 = 0 \\
X(6) &= F_1(2) - W_8^2 F_2(2) \\
&= 0 - (-j)(0) = 0 \\
X(7) &= F_1(3) - W_8^3 F_2(3) \\
&= (1.414 + j1.414) - (-0.7071 - j0.7071)2 \\
&= 2.8284 + j2.8284
\end{aligned}
$$


2-point DFT
Combine 2-point DFTs
Combine Two-4 point DFTs
x(0) = 0.707
x(4) = −0.707
x(2) = 0.707
x(6) = −0.707
x(1) = 1
x(5) = −1
x(3) = 0
x(7) = 0
W20 = W80
-1
V11(0) = 0
V11(1) = 1.414
V12(0) = 0
V12(1) = 2
V21(0)0
V21(1) = 2
V22(0) = 0
V22(1) = 0
W40 = W80
W41 = W82
F1(0) = 0
F1(1) = 1.414 − j1.414
F1(2) = 0
F1(3) = 1.414 + j1.414
F2(0) = 0
F2(1) = 2
F2(2) = 0
F2(3) = 2
W80
W81
W82
W83
X(0) = 0
X(1) = 2.8284 − j2.8284
X(2) = 0
X(3) = 0
X(4) = 0
X(5) = 0
X(6) = 0
X(7) = 2.8284 + j2.8284

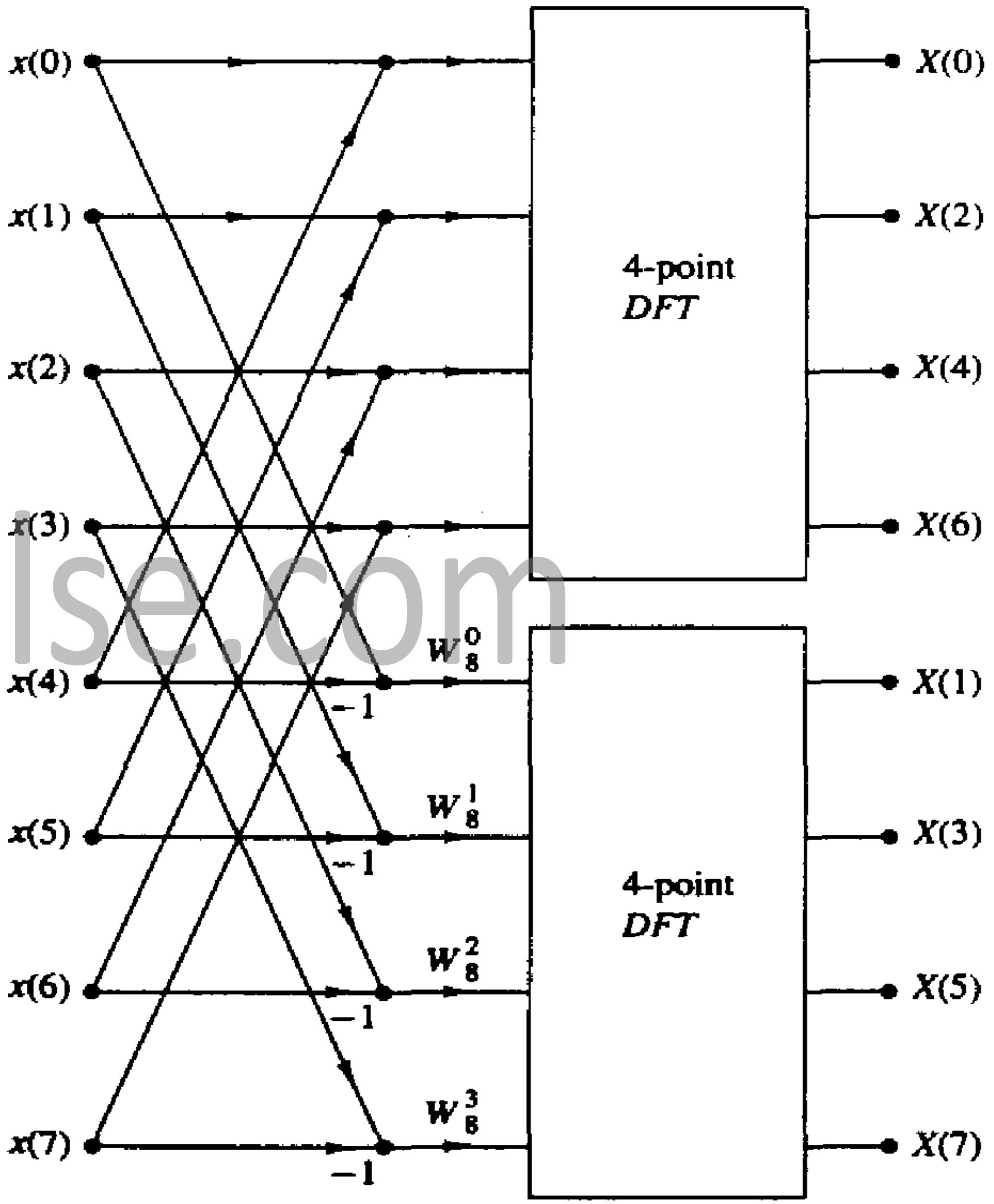
…RADIX-2FFT…

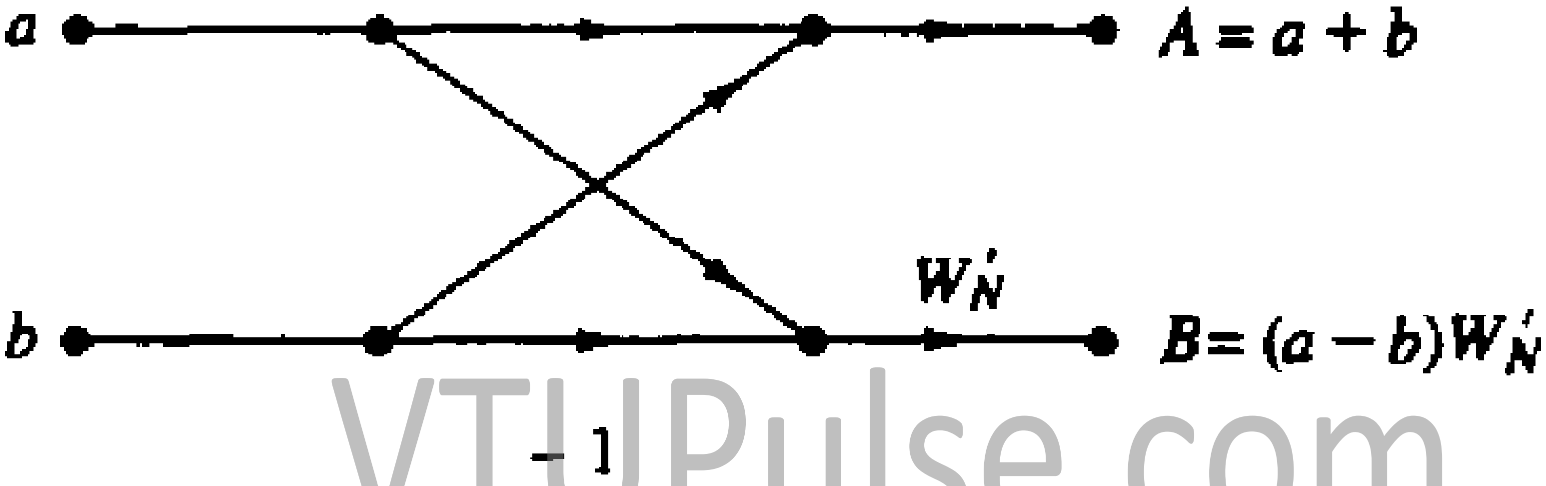
- We split $X(k)$ into odd numbered and even numbered sequence.
- $X(2k) = \sum_{n=0}^{\frac{N}{2}-1}[x(n) + x\left(n+\frac{N}{2}\right)]\, W_{\frac{N}{2}}^{kn}$ and $X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1}[x(n) - x\left(n+\frac{N}{2}\right) W_N^n\ ]\, W_{\frac{N}{2}}^{kn}$, $k = 0,1 \ldots \frac{N}{2}-1.\ W_N^2 \ = W_{\frac{N}{2}}.$
- If $g_1(n)$ and $g_2(n)$ are two N/2-point sequences, $g_1(n) = x(n) + x\left(n+\frac{N}{2}\right)$ and

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

- $X(2k) = \sum_{n=0}^{\frac{N}{2}-1} g_1(n) W_{\frac{N}{2}}^{kn}$ and $X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} g_2(n) W_{\frac{N}{2}}^{kn}$.
- The computation procedure can be repeated and the process involves $v = \log_2 N$ stages of decimation.

- First stage of decimation in frequency FFT.

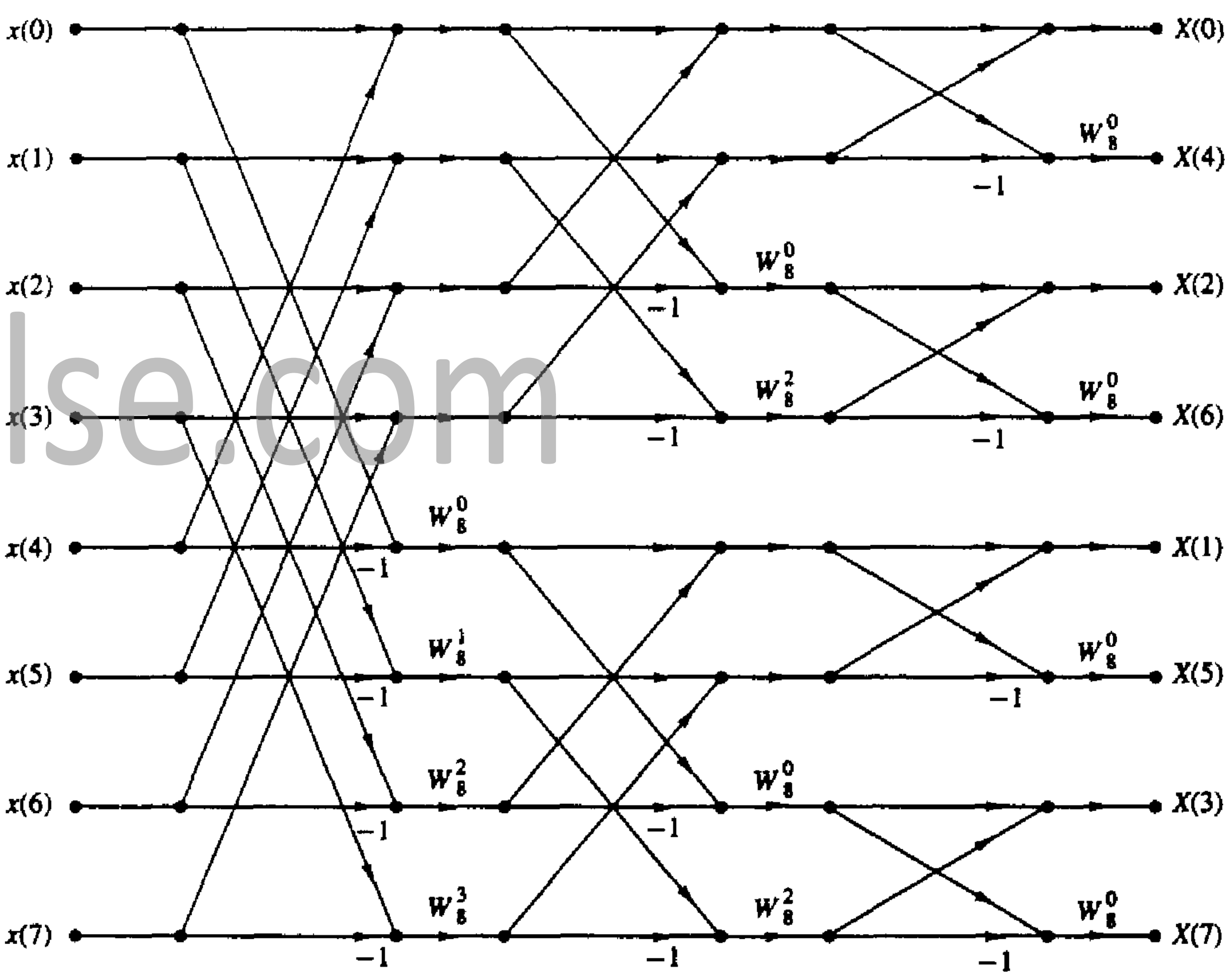




- Butterfly computation for Decimation in Frequency Algorithm

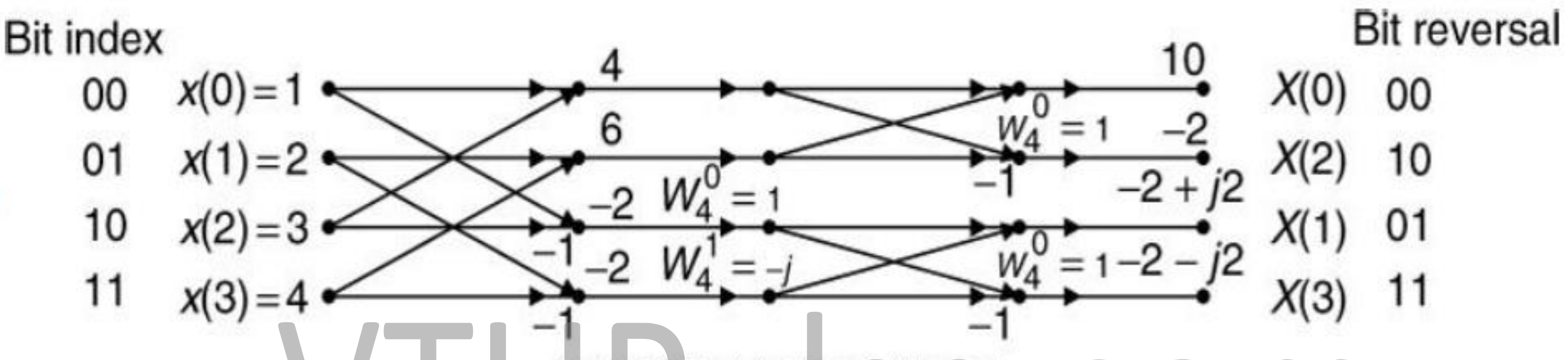
- Number of complex multiplications: $\frac{N}{2}\log_2 N$.
- Number of complex additions: $\mathrm{N}\log_2 N$.

  - 8-point Decimation in frequency FFT.



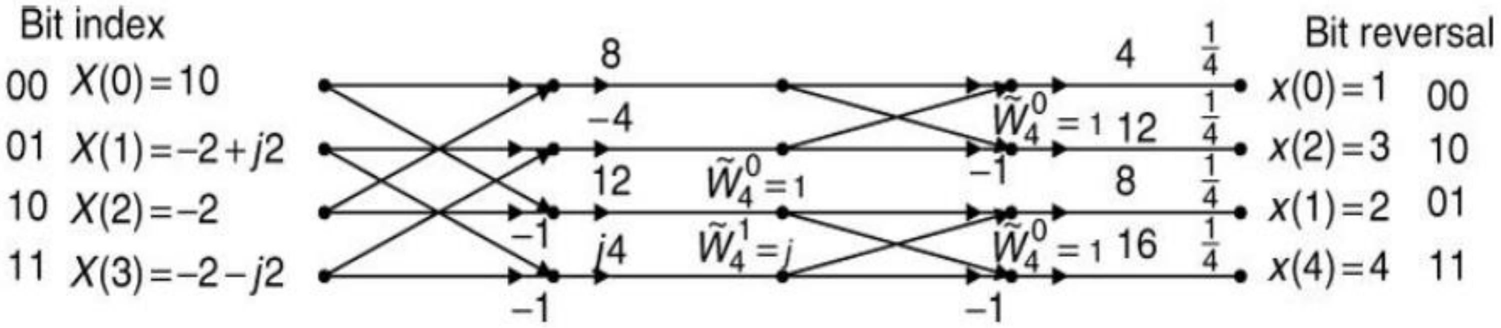
FFT and IFFT Examples

FFT



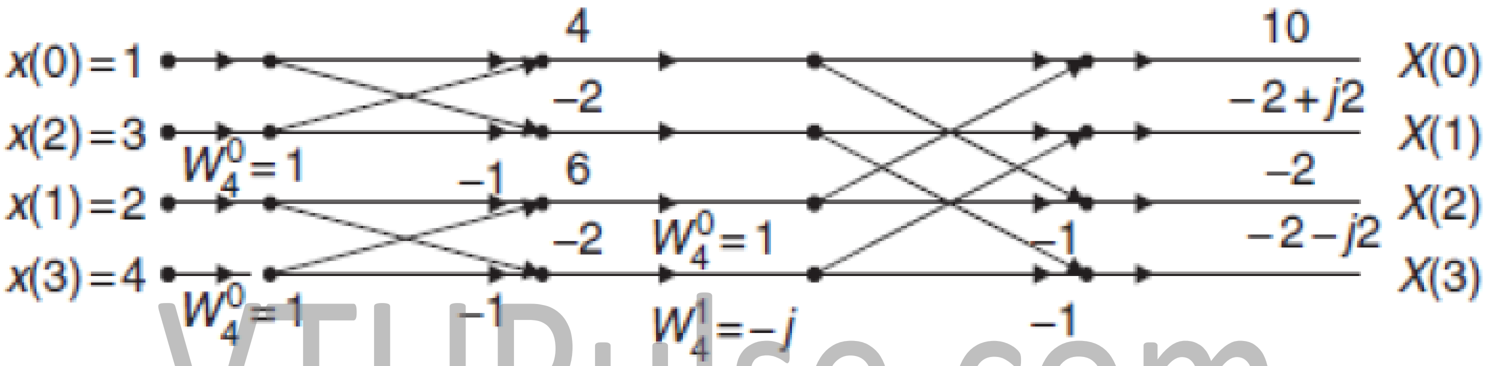
Number of complex multiplication = $\frac{N}{2}\log_2(N) = \frac{4}{2}\log_2(4) = 4.$

IFFT

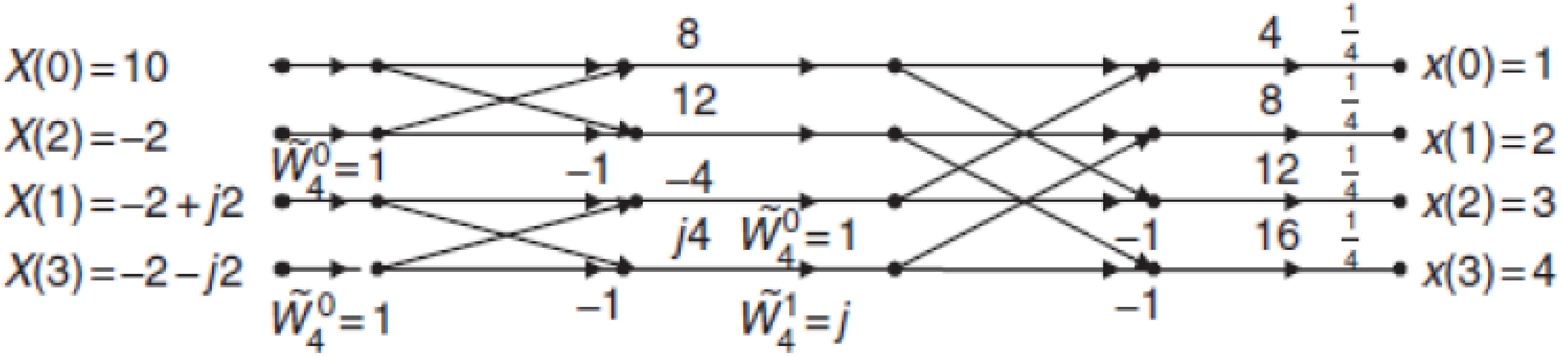


FFT and IFFT Examples

FFT



IFFT



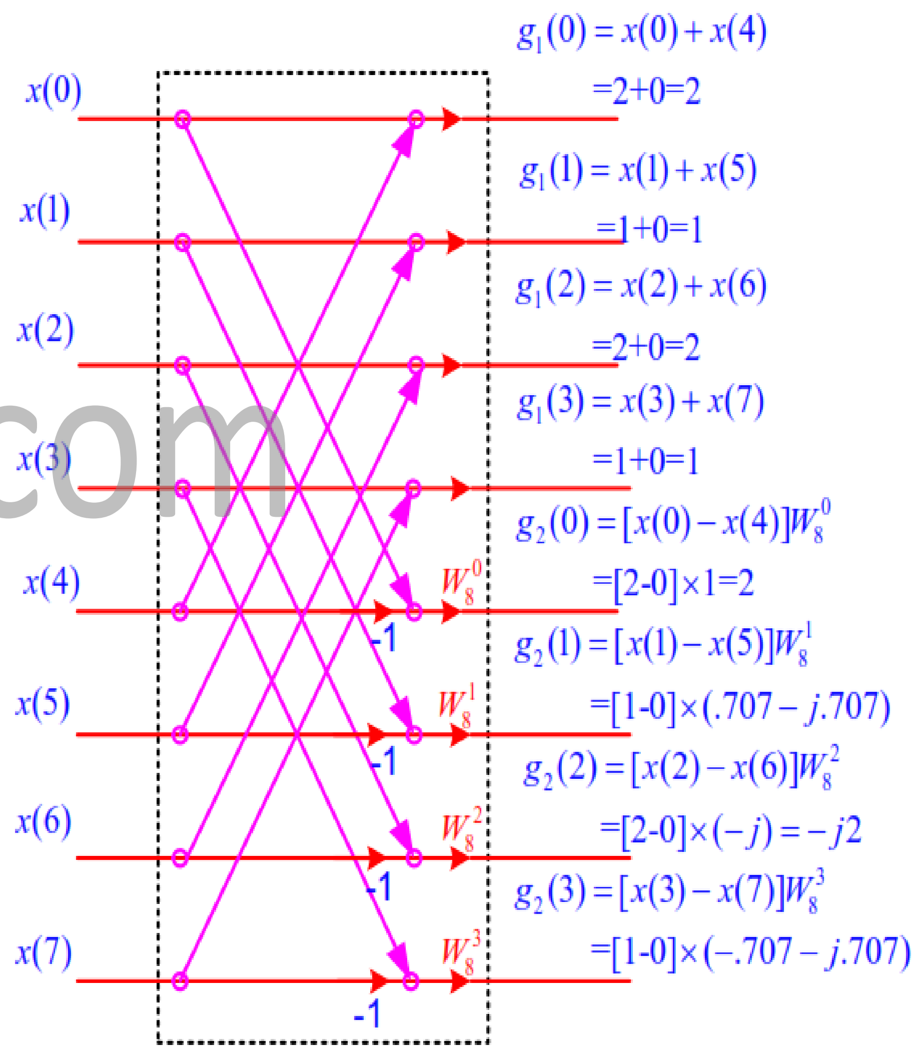
Obtain the 8 point DFT of the following sequence using redix-2 DIF-FFT algorithm. Show all the results along signal flow graph x(n)={2,1,2,1}

Solution:

$$W_8^0 = 1$$
$$W_8^1 = 0.707 - j0.707$$
$$W_8^2 = -j$$
$$W_8^3 = -0.707 - j0.707$$

$$\begin{aligned} g_1(0) &= x(0) + x(4) \\ &= 2 + 0 = 2 \\ g_1(1) &= x(1) + x(5) \\ &= 1 + 0 = 1 \\ g_1(2) &= x(2) + x(6) \\ &= 2 + 0 = 2 \\ g_1(3) &= x(3) + x(7) \\ &= 1 + 0 = 1 \end{aligned}$$

$$\begin{aligned} g_2(0) &= [x(0) - x(4)]W_8^0 \\ &= [2 - 0] \times 1 = 2 \\ g_2(1) &= [x(1) - x(5)]W_8^1 \\ &= [1 - 0] \times [0.707 - j0.707] \\ &= 0.707 - j0.707 \\ g_2(2) &= [x(2) - x(6)]W_8^2 \\ &= [2 - 0] \times [-j] = -j2 \\ g_2(3) &= [x(3) - x(7)]W_8^3 \\ &= [1 - 0] \times [-0.707 - j0.707] \\ &= -0.707 - j0.707 \end{aligned}$$



$g_1(0) = 2$

$g_1(1) = 1$

$g_1(2) = 2$

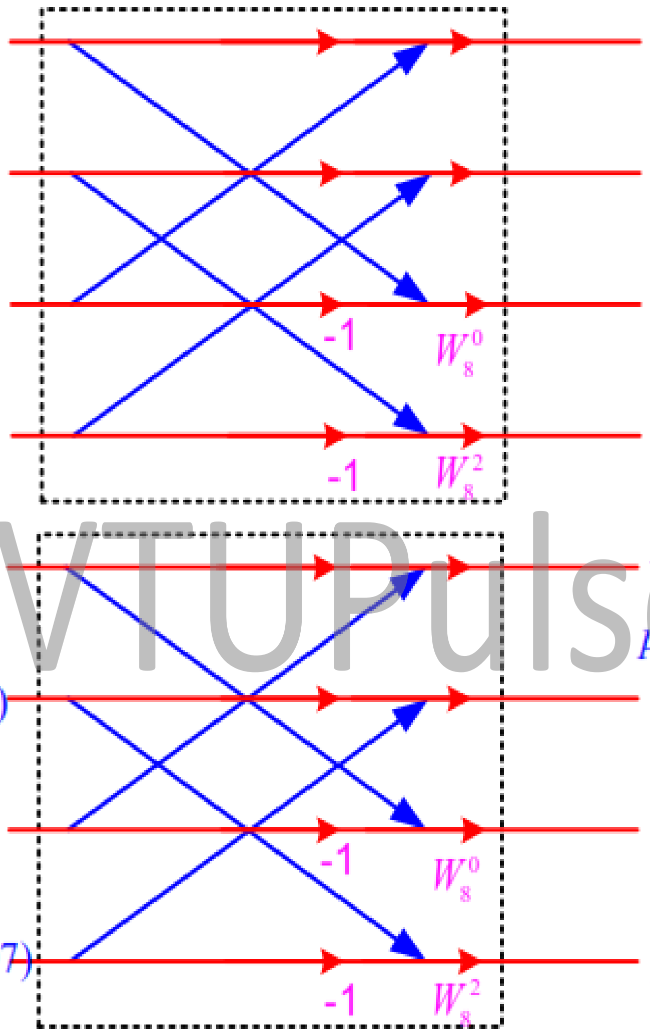
$g_1(3) = 1$

$g_2(0) = 2$

$g_2(1) = (.707 - j.707)$

$g_2(2) = -j2$

$g_2(3) = (-.707 - j.707)$



$$p_{11}(0) = g_1(0) + g_1(2) = 2 + 2 = 4$$

$$p_{11}(1) = g_1(1) + g_1(3) = 1 + 1 = 2$$

$$p_{12}(0) = [g_1(0) - g_1(2)]W_8^0 = (2-2)\times 1 = 0$$

$$p_{12}(1) = [g_1(1) - g_1(3)]W_8^0 = (1-1)\times(-j) = 0$$

$$p_{21}(0) = g_2(0) + g_2(2) = 2 - j2$$

$$p_{21}(1) = [g_2(1) + g_2(3)] = (.707 - j.707 - .707 - j.707) = -j1.414$$

$$p_{22}(0) = [g_2(0) - g_2(2)]W_8^0 = [2 - (-j2)]\times -j = 2 + j2$$

$$p_{22}(1) = [g_2(1) - g_2(3)]W_8^1 = (.707 - j.707 + .707 + j.707)\times -j = -j1.414$$

2 point DFT

$p_{11}(0) = 4$

$p_{11}(1) = 2$

$p_{12}(0) = 0$

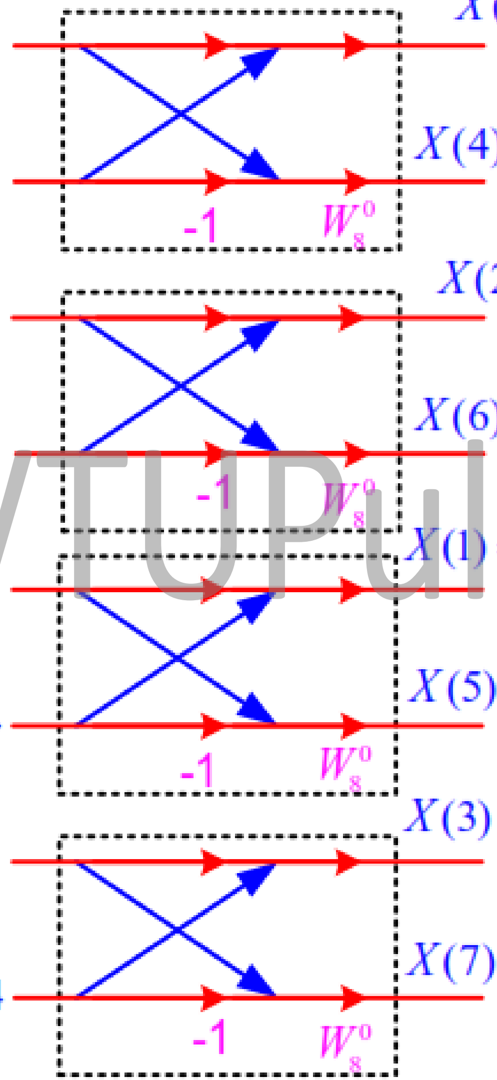
$p_{12}(1) = 0$

$p_{21}(0) = 2 - j2$

$p_{21}(1) = -j1.414$

$p_{22}(0) = 2 + j2$

$p_{22}(1) = -j1.414$



$X(0) = p_{11}(0) + p_{11}(1)$

4+2=6

$X(4) = [p_{11}(0) - p_{11}(1)]W_8^0$

(4-2)1=2

$X(2) = p_{12}(0) - p_{12}(1)$

0+0=0

$X(6) = [p_{11}(0) + p_{11}(1)]W_8^0$

(0-0)1=0

$X(1) = p_{21}(0) + p_{21}(1)$

2-j2-j1.414=2-j3.414

$X(5) = [p_{21}(0) + p_{21}(1)]W_8^0$

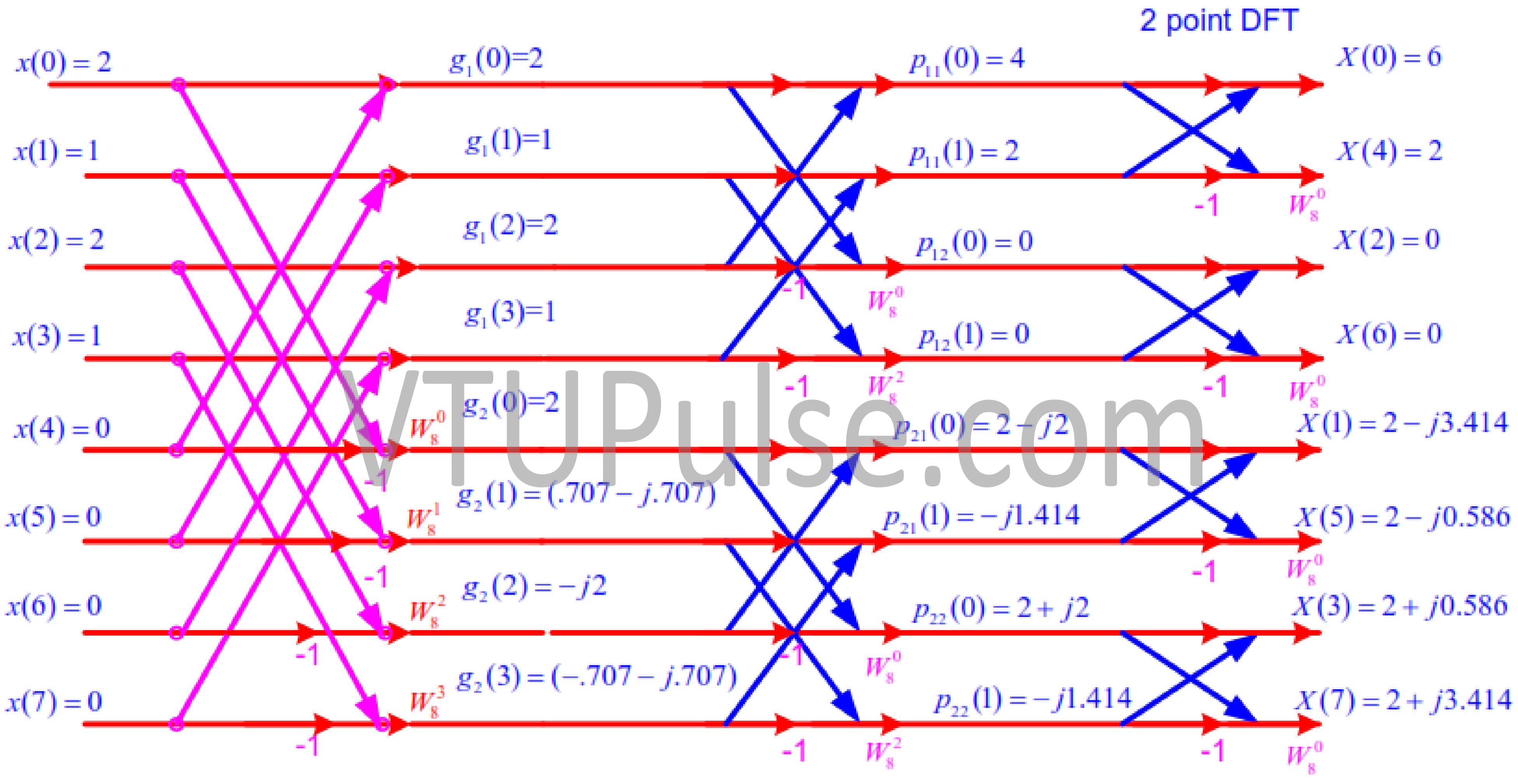
[2-j2-(-j1.414)]1=2-j0.566

$X(3) = p_{22}(0) + p_{22}(1)$

2+j2-j1.414=2+j0.586

$X(7) = [p_{22}(0) - p_{22}(1)]W_8^0$

[2+j2-(-j1.414)]1=2+j3.414



2 point DFT
x(0) = 2
x(1) = 1
x(2) = 2
x(3) = 1
x(4) = 0
x(5) = 0
x(6) = 0
x(7) = 0
g1(0)=2
g1(1)=1
g1(2)=2
g1(3)=1
g2(0)=2
g2(1) = (.707 − j.707)
g2(2) = −j2
g2(3) = (−.707 − j.707)
p11(0) = 4
p11(1) = 2
p12(0) = 0
p12(1) = 0
p21(0) = 2 − j2
p21(1) = −j1.414
p22(0) = 2 + j2
p22(1) = −j1.414
X(0) = 6
X(4) = 2
X(2) = 0
X(6) = 0
X(1) = 2 − j3.414
X(5) = 2 − j0.586
X(3) = 2 + j0.586
X(7) = 2 + j3.414