Fast Furier Transform

- Fast Fourier transform (FFT) is an efficient algorithm to compute the DFT with reduced computations.
- Due to the efficiency offered by FFT, the DFT is widely used for the spectrum analysis, convolutions, correlations, and for linear filtering.
- FFT is only a computational algorithm and not another transform.
- FFT algorithm is developed by Cooley and Tukey in 1965.
- Two FFT algorithms are known as decimation-in-time (DIT) and decimation-in-frequency (DIF) algorithms.

196

Number of Calculations in N-point DFT

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

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

The number of calculation to calculate $\mathbf{X(k)}$ for one value of k are

$\mathbf{N}$ number of complex multiplications

$\mathbf{(N-1)}$ number of additions

The number of calculations to calculate all the $\mathbf{X(k)}$ are

$\mathbf{N \times N = O(N^2)}$ number of complex multiplications

$\mathbf{N \times (N-1) = O\left(N(N-1)\right)}$ number of complex additions.

197

Number of Caculation in Radix-2 FFT

- **The simplest and perhaps best-known method for computing the FFT is the Radix-2 Decimation in Time algorithm.**

- **The main limitation of the radix-2 method is that it only works if $N = 2^m$, where m is an integer. If $N = 37$ (for example), this method cannot be used.**

- **In decimation in time (DIT) algorithm, time sequence $x(n)$ is decimated and smaller point DFTs are combined to get the result of N point DFT.**

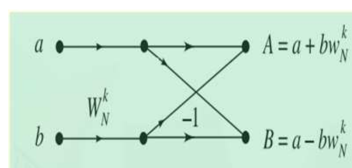
- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**

- **The decimation can be performed up to m times, where $\mathrm{N} = 2^m$ and $m = \log_2(N)$**

198

Number of Caculation in Radix-2 FFT

In radix-2 FFT, $\mathbf{N = 2^m}$, and so there will be m stages of computations where $\mathbf{m = \log_2(N)}$ with each stage having $\mathbf{N/2}$ butterflies



The number of calculation in one butterflies are

1 number of complex multiplication

2 number of complex additions

There are N/2 butterflies in each stage

$\frac{N}{2} \times 1 = O\left(\frac{N}{2}\right)$ number of complex multiplications

$\frac{N}{2} \times 2 = O(N)$ number of complex additions

The N-point DFT involves m stages of computations.

$\frac{N}{2} \times m = O\left(\frac{N}{2} \times \log_2(N)\right)$ number of complex multiplications

$N \times m = O(N \times \log_2(N))$ number of complex additions

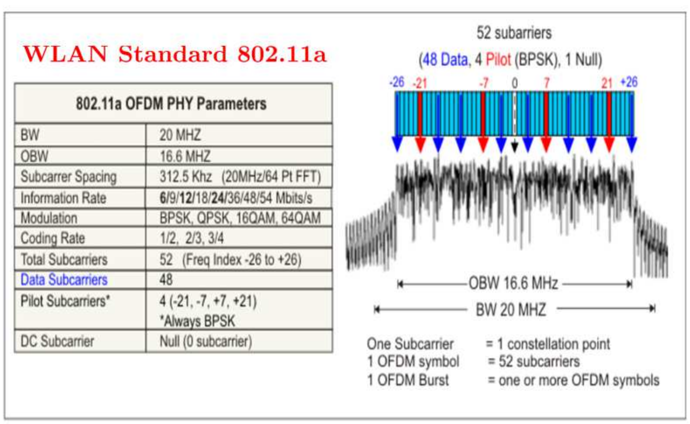
199

Number of computation in DFT and FFT

Number of points N	Direct Computation: Complex additions $N(N-1)$	Direct Computation: Complex Multiplications N^2	Radix-2 FFT: Complex additions $N\log_2 N$	Radix-2 FFT: Complex Multiplications $(N/2)\log_2 N$
$4\,(=2^2)$	12	16	$4\times\log_2 2^2=4\times2=8$	$\frac{4}{2}\times\log_2 2^2=\frac{4}{2}\times2=4$
$8\,(=2^3)$	56	64	$8\times\log_2 2^3=8\times3=24$	$\frac{8}{2}\times\log_2 2^3=\frac{8}{2}\times3=12$
$16\,(=2^4)$	240	256	$16\times\log_2 2^4=16\times4=64$	$\frac{16}{2}\times\log_2 2^4=\frac{16}{2}\times4=32$
$32\,(=2^5)$	992	1,024	$32\times\log_2 2^5=32\times5=160$	$\frac{32}{2}\times\log_2 2^5=\frac{32}{2}\times5=80$
$64\,(=2^6)$	4,032	4,096	$64\times\log_2 2^6=64\times6=384$	$\frac{64}{2}\times\log_2 2^6=\frac{64}{2}\times6=192$
$128\,(=2^7)$	16,256	16,384	$128\times\log_2 2^7=128\times7=896$	$\frac{128}{2}\times\log_2 2^7=\frac{128}{2}\times7=448$

200

Use of FFT



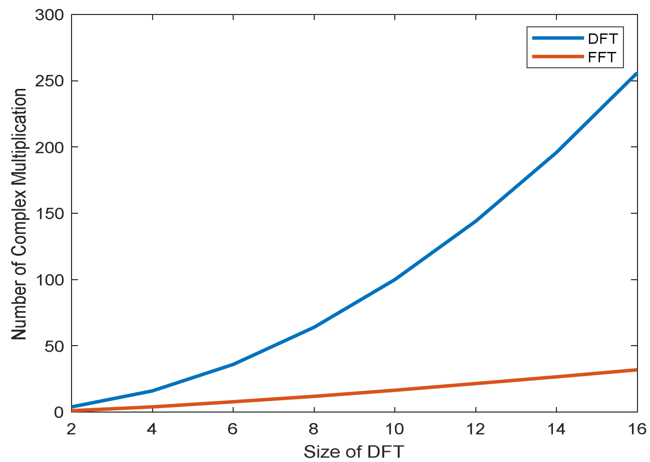
802.11a OFDM Physical Parameters

Keysight OFDM Overview

- Wi-Fi
- LTE
- 5G
- Wi-Max
- DVB-T
- Digital Audio Broadcasting

201

Number of computation in DFT and FFT



202

Decimation in Time (DIT) Radix-2 FFT

There are N/2 butterflies in each stage

❖ **If** $N = 8$, **the decimation can be performed up to** $m = \log_2(N) = \log_2(8) = 3$

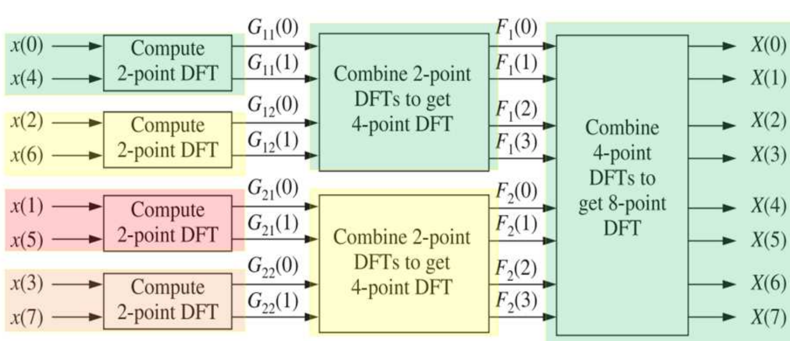


Figure 7.5 Three stages of computation in 8-point DFT.

203

Fast Furier Transform

There are N/2 butterflies in each stage

❖ **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

How to arrange inputs?

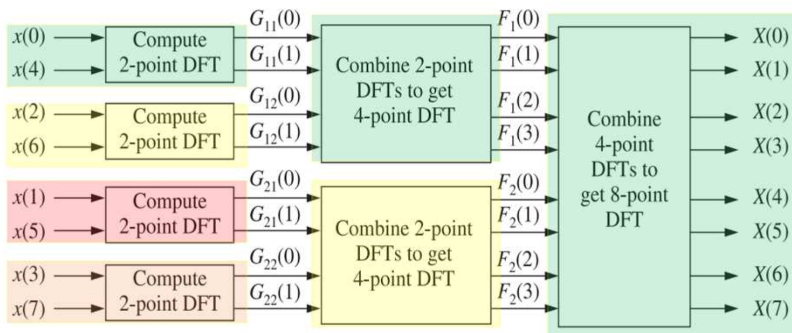


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

❖ **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**

❖ **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000		
1	001		
2	010		
3	011		
4	100		
5	101		
6	110		
7	111		

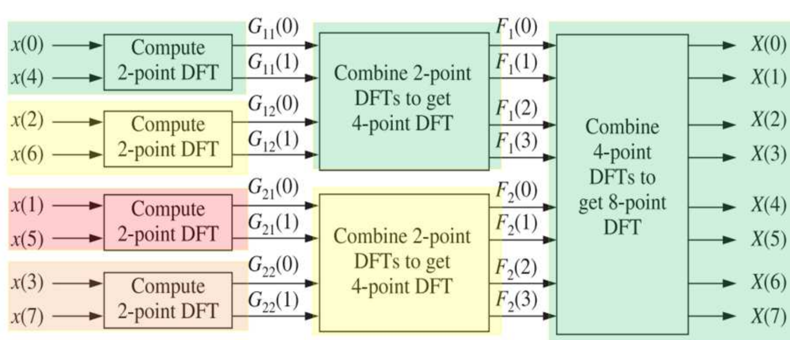


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**
- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000		
1	001		
2	010		
3	011		
4	100		
5	101		
6	110		
7	111		

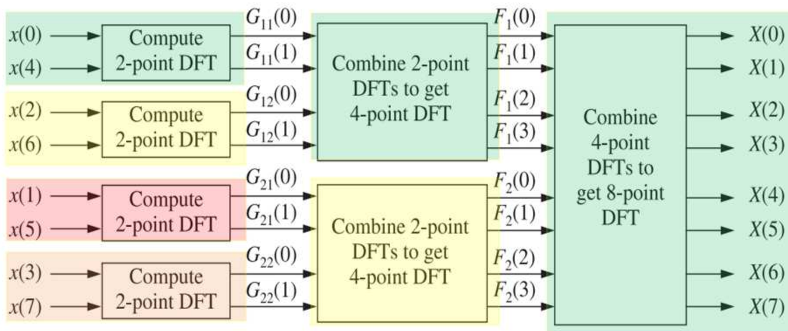


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**
- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000	000	
1	001		
2	010		
3	011		
4	100		
5	101		
6	110		
7	111		

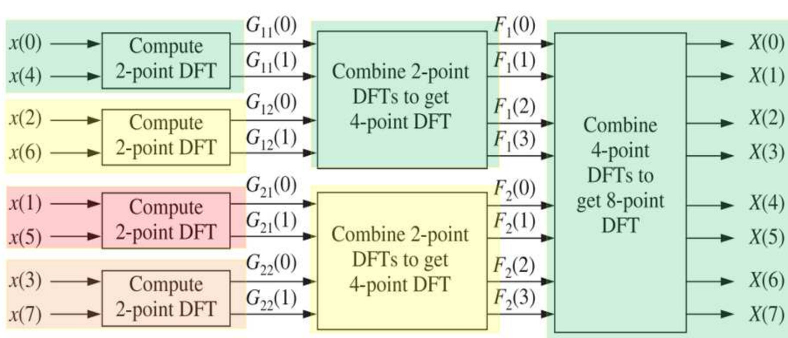


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**

- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000	000	
1	001	100	
2	010		
3	011		
4	100		
5	101		
6	110		
7	111		

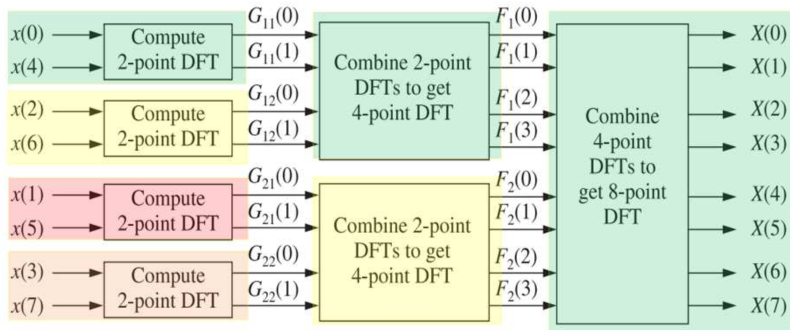


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**

- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000	000	
1	001	100	
2	010	010	
3	011		
4	100		
5	101		
6	110		
7	111		

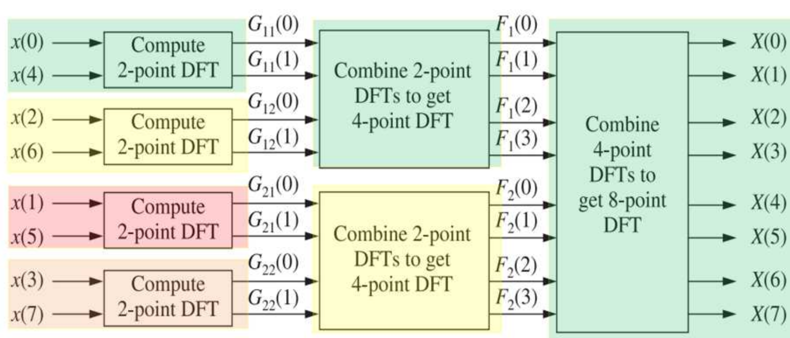

x(0)
x(4)
Compute 2-point DFT
x(2)
x(6)
Compute 2-point DFT
x(1)
x(5)
Compute 2-point DFT
x(3)
x(7)
Compute 2-point DFT
G11(0)
G11(1)
G12(0)
G12(1)
G21(0)
G21(1)
G22(0)
G22(1)
Combine 2-point DFTs to get 4-point DFT
Combine 2-point DFTs to get 4-point DFT
F1(0)
F1(1)
F1(2)
F1(3)
F2(0)
F2(1)
F2(2)
F2(3)
Combine 4-point DFTs to get 8-point DFT
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)

Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.

- If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$

Original	Binary Form	Reversed Form	Final
0	000	000	
1	001	100	
2	010	010	
3	011	110	
4	100		
5	101		
6	110		
7	111		

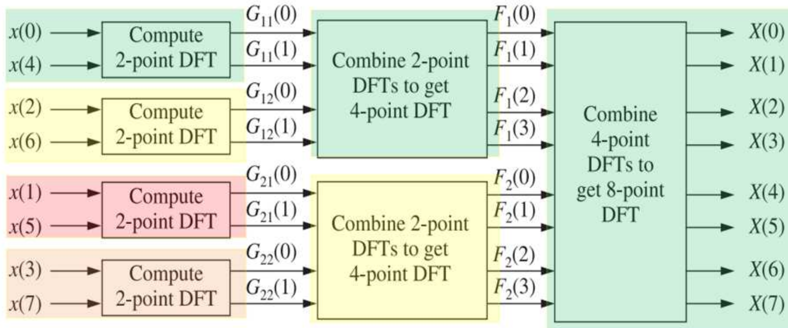


Figure 7.5 Three stages of computation in 8-point DFT.

210

Fast Furier Transform

- In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.

- If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$

Original	Binary Form	Reversed Form	Final
0	000	000	
1	001	100	
2	010	010	
3	011	110	
4	100	001	
5	101	101	
6	110	011	
7	111	111	

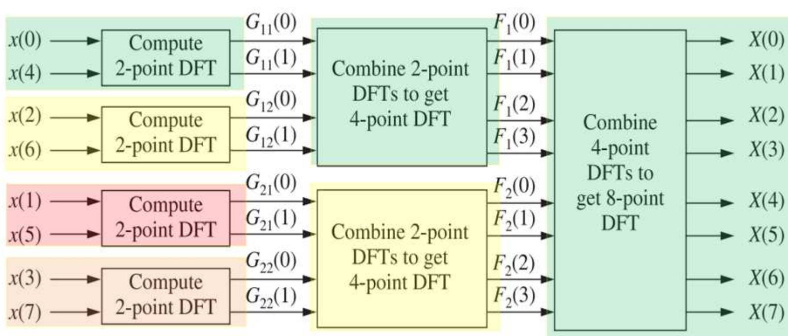


Figure 7.5 Three stages of computation in 8-point DFT.

211

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**
- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

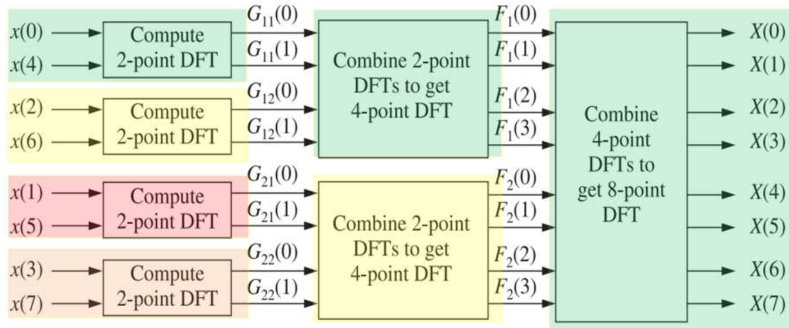


Figure 7.5 Three stages of computation in 8-point DFT.

Fast Furier Transform

- **In general, we can say that N point DFT can be realized from $N/2$ points DFT. Similarly, $N/2$ points DFT can be calculated by $N/4$ points DFT and so on.**
- **If $N = 8$, the decimation can be performed up to $m = \log_2(N) = \log_2(8) = 3$**

Original	Binary Form	Reversed Form	Final
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

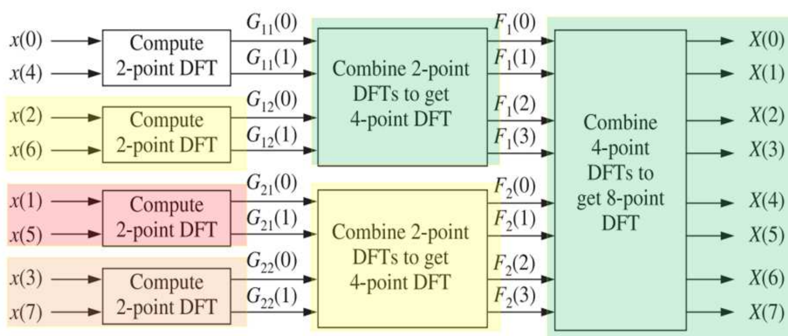
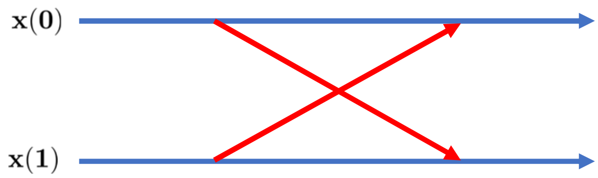


Figure 7.5 Three stages of computation in 8-point DFT.

Number of computation in DFT and FFT

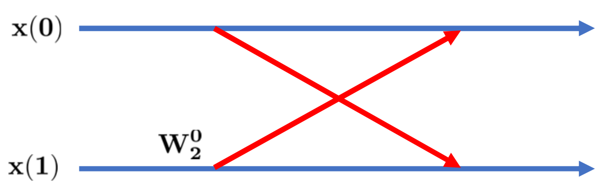
Find N=2 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(2) = 1}$



214

Number of computation in DFT and FFT

Find N=2 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(2) = 1}$

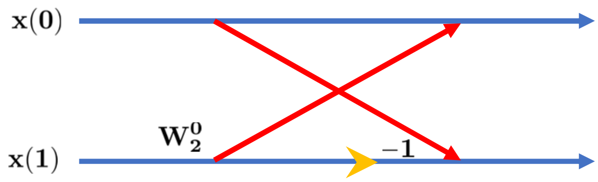


Step1: write twidel factor w_N^k at the begning of arrow going up

215

Number of computation in DFT and FFT

Find N=2 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(2) = 1}$



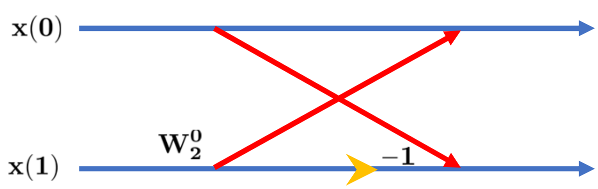
Step1: write twidel factor w_N^k at the begning of arrow going up

Step2: write -1 before the point where arrow is comming down

216

Number of computation in DFT and FFT

Find N=2 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(2) = 1}$



$$\mathbf{X(0) = x(0) + x(1)W_2^0}$$

$$\mathbf{X(1) = x(0) - x(1)W_2^0}$$

Step1: write twidel factor w_N^k at the begning of arrow going up

Step2: write -1 before the point where arrow is comming down

217

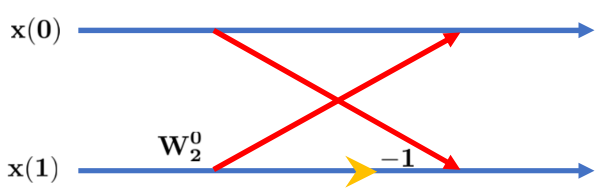
Number of computation in DFT and FFT

Find DFT with DIT FFT for x(n)={1, 2}

218

Number of computation in DFT and FFT

Find DFT with DIT FFT for x(n)={1, 2}



$$\mathbf{X(0) = x(0) + x(1)W_2^0 = 1 + 2 = 3}$$

$$\mathbf{X(1) = x(0) - x(1)W_2^0 = 1 - 2 = -1}$$

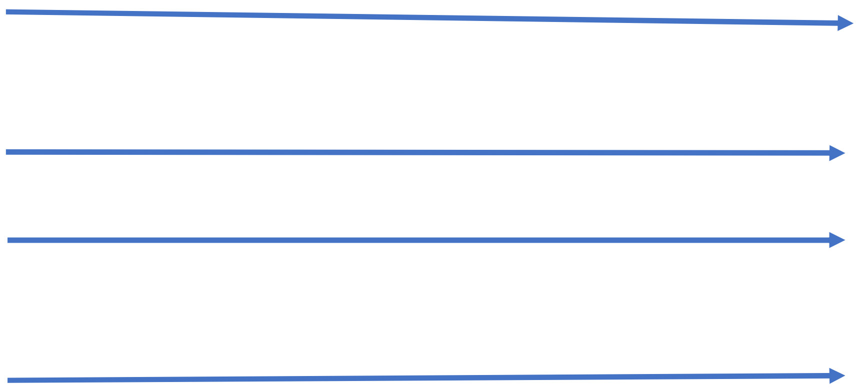
Step1: write twidel factor w_N^k at the begning of arrow going up

Step2: write -1 infront of the point where arrow is comming down

219

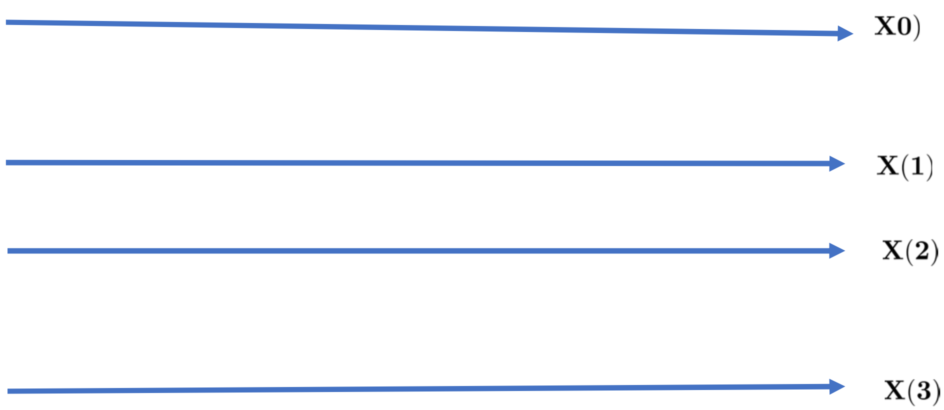
Number of computation in DFT and FFT

Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



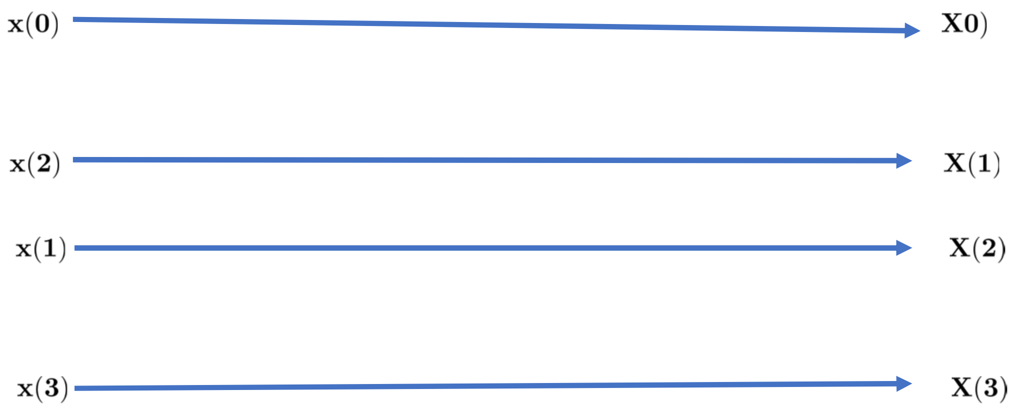
Number of computation in DFT and FFT

Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



Number of computation in DFT and FFT

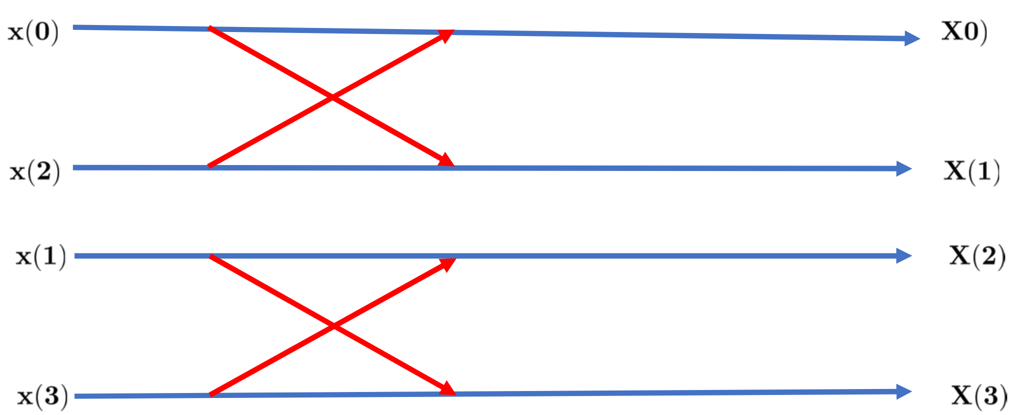
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



222

Number of computation in DFT and FFT

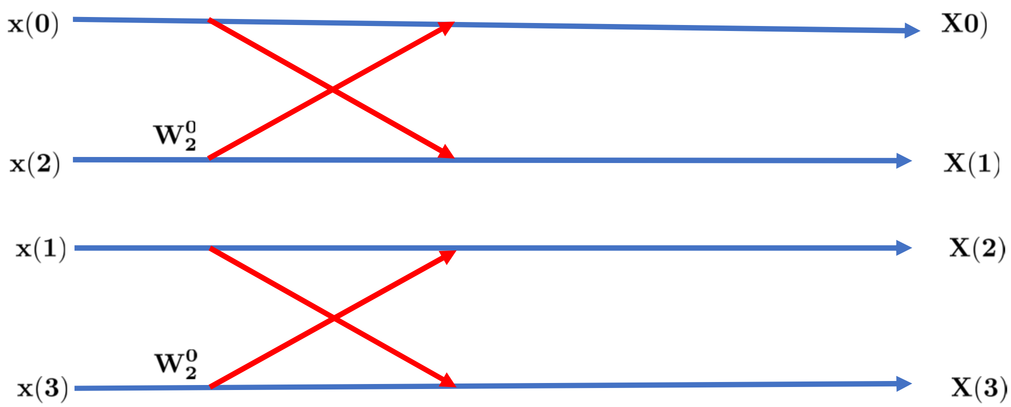
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



223

Number of computation in DFT and FFT

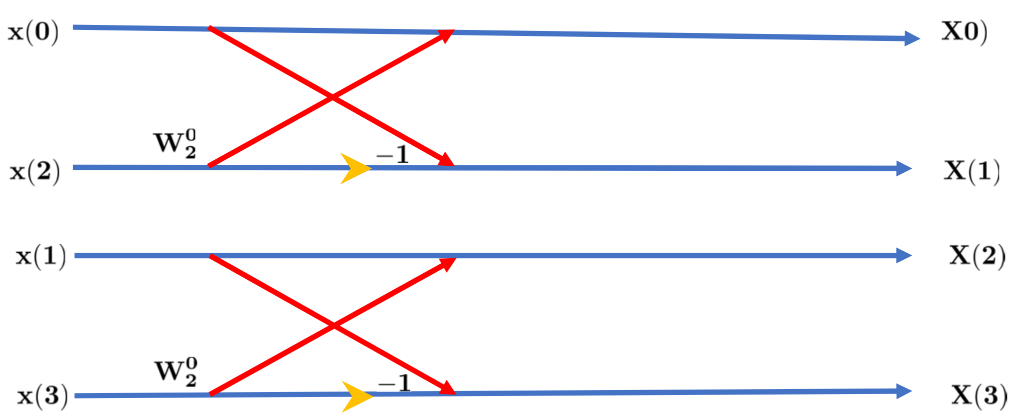
Find N=4 DFT with DIT FFT method $m = \log_2(N) = \log_2(4) = 2$



224

Number of computation in DFT and FFT

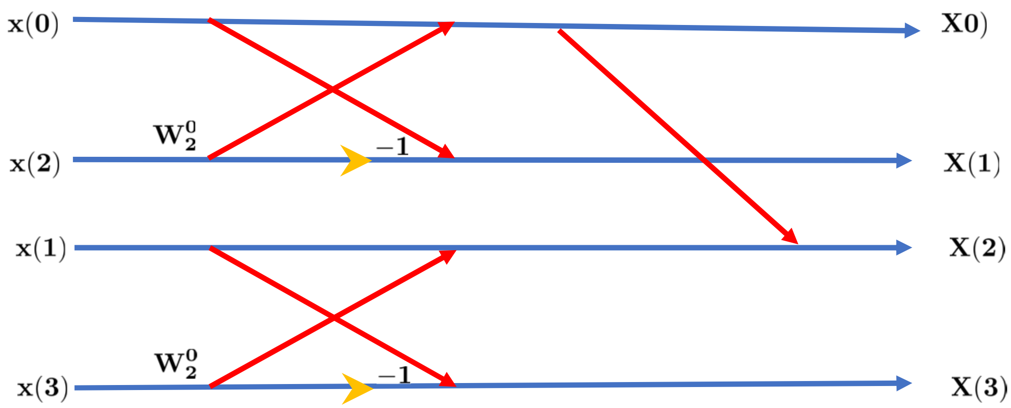
Find N=4 DFT with DIT FFT method $m = \log_2(N) = \log_2(4) = 2$



225

Number of computation in DFT and FFT

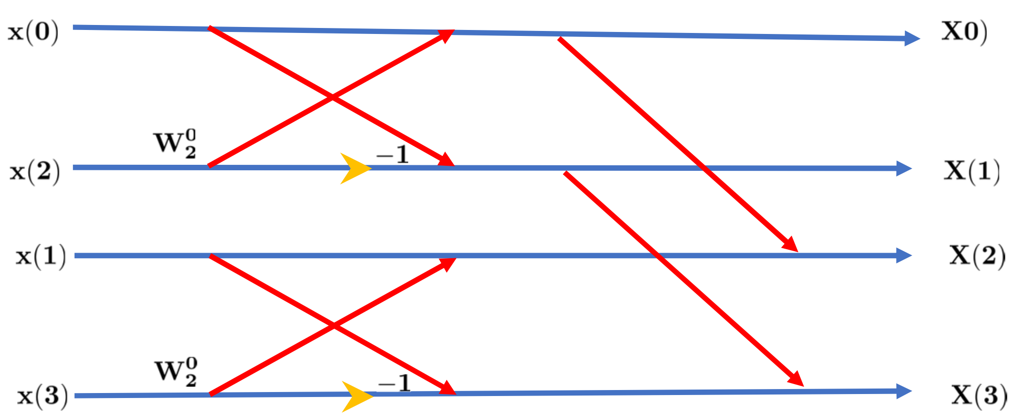
Find N=4 DFT with DIT FFT method $m = \log_2(N) = \log_2(4) = 2$



226

Number of computation in DFT and FFT

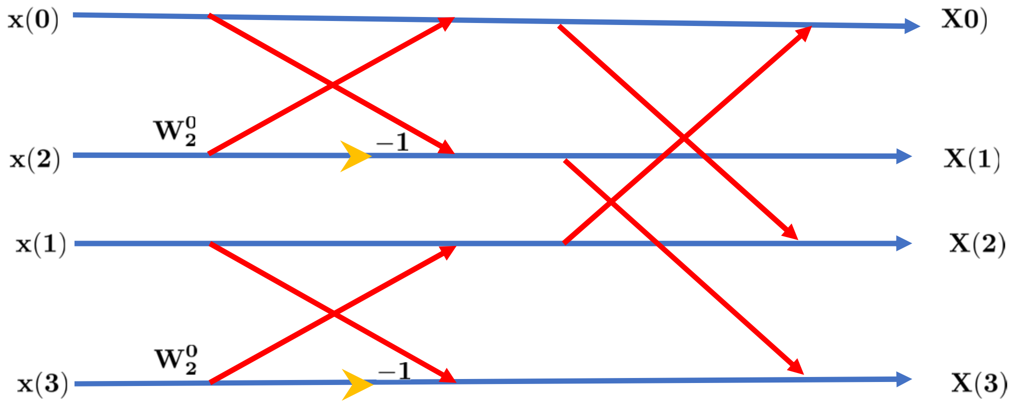
Find N=4 DFT with DIT FFT method $m = \log_2(N) = \log_2(4) = 2$



227

Number of computation in DFT and FFT

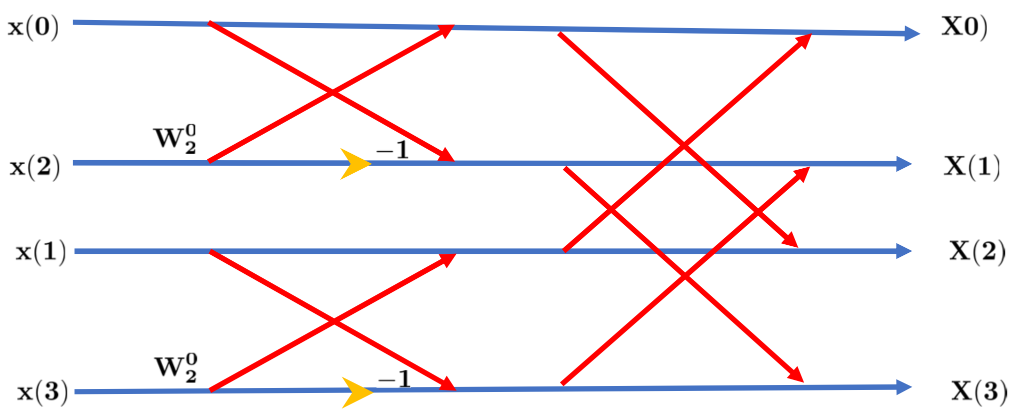
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



228

Number of computation in DFT and FFT

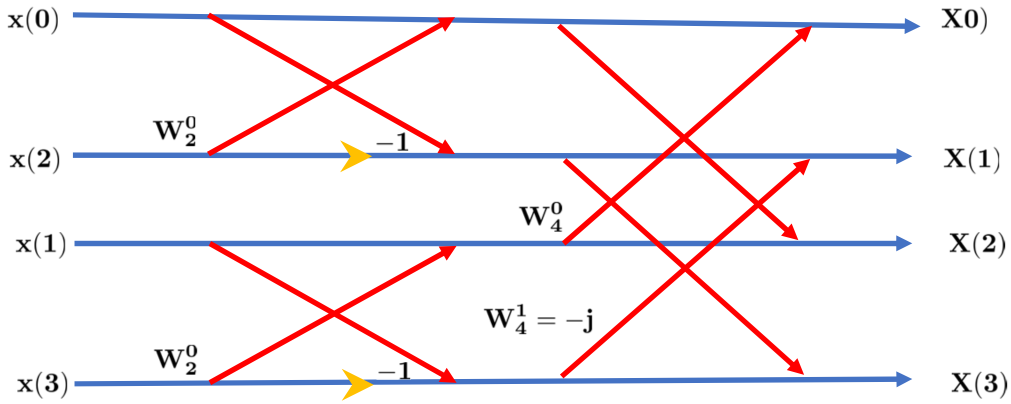
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



229

Number of computation in DFT and FFT

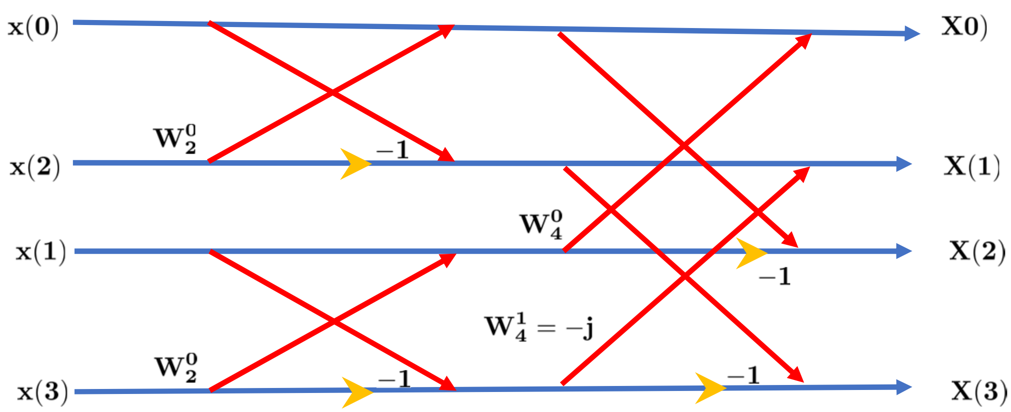
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



230

Number of computation in DFT and FFT

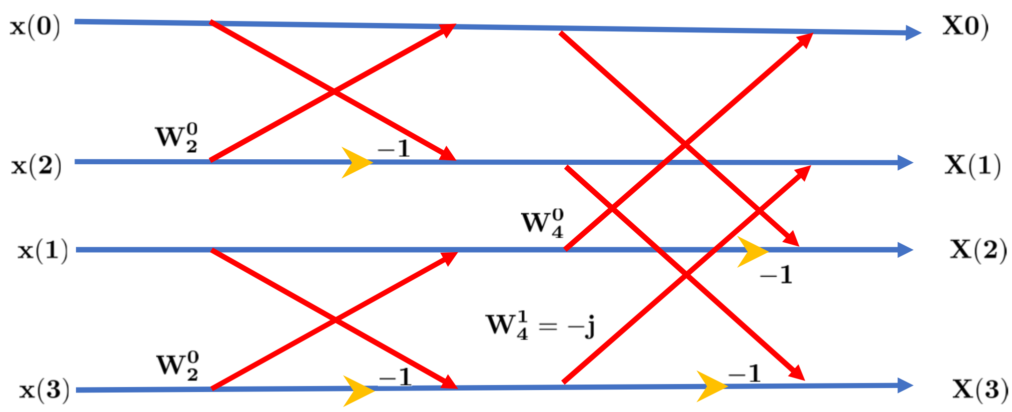
Find N=4 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(4) = 2}$



231

Number of computation in DFT and FFT

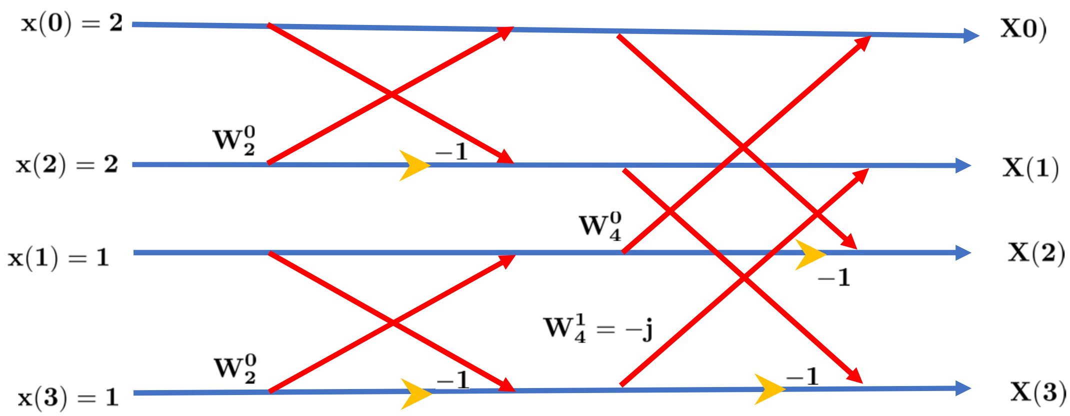
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



232

Number of computation in DFT and FFT

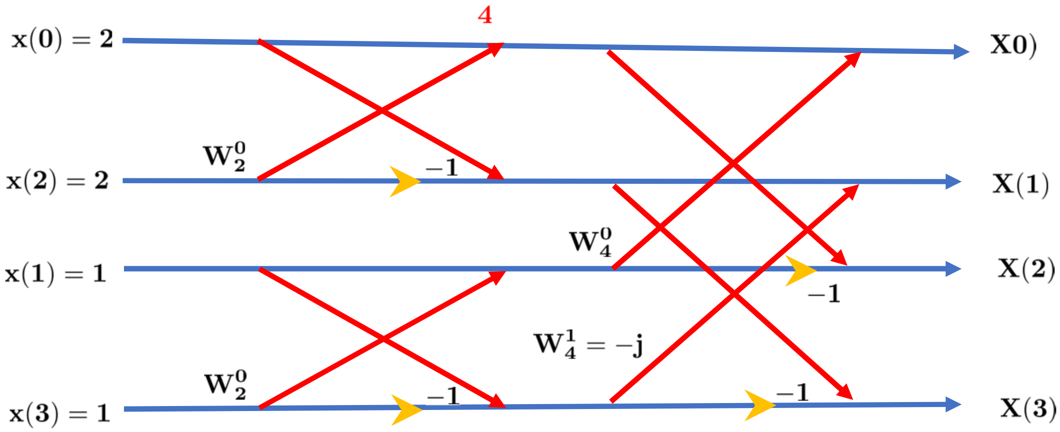
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



233

Number of computation in DFT and FFT

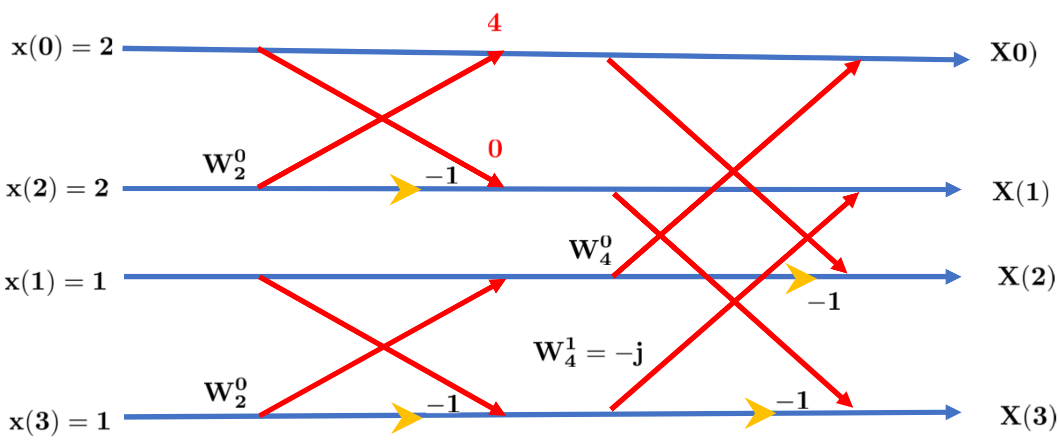
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



234

Number of computation in DFT and FFT

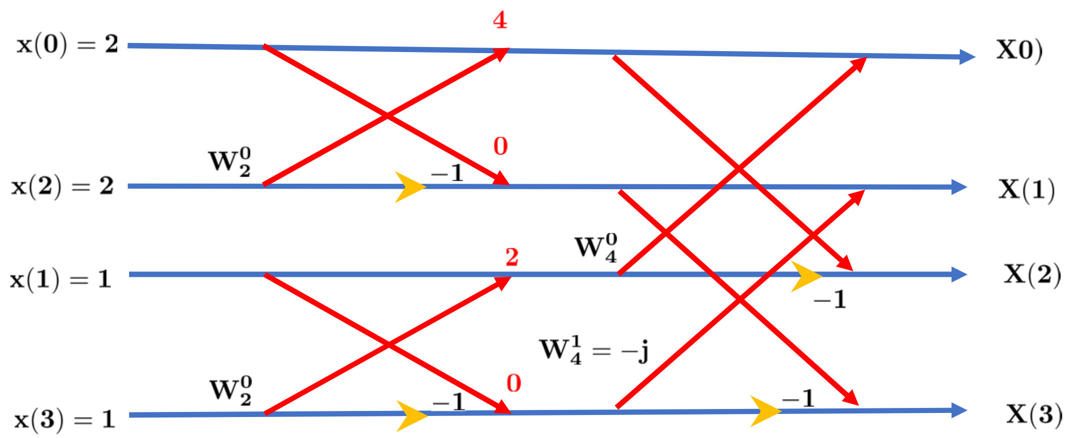
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



235

Number of computation in DFT and FFT

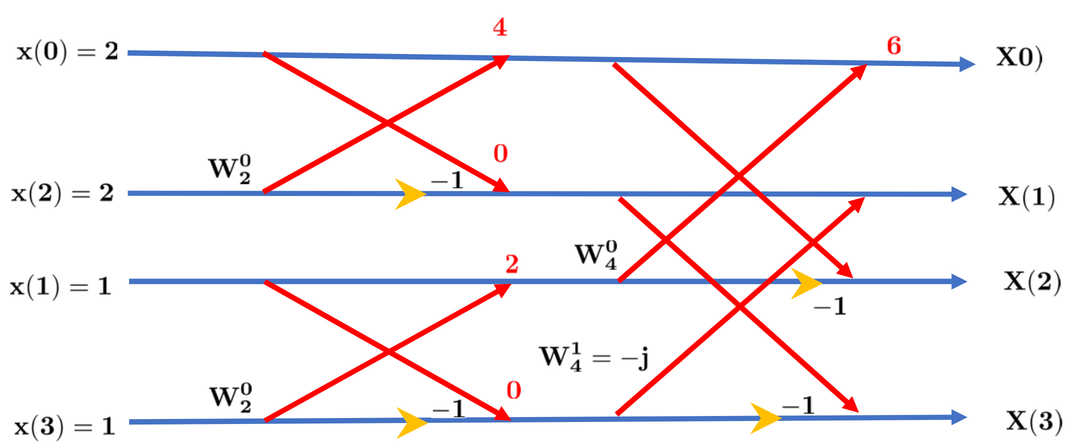
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



236

Number of computation in DFT and FFT

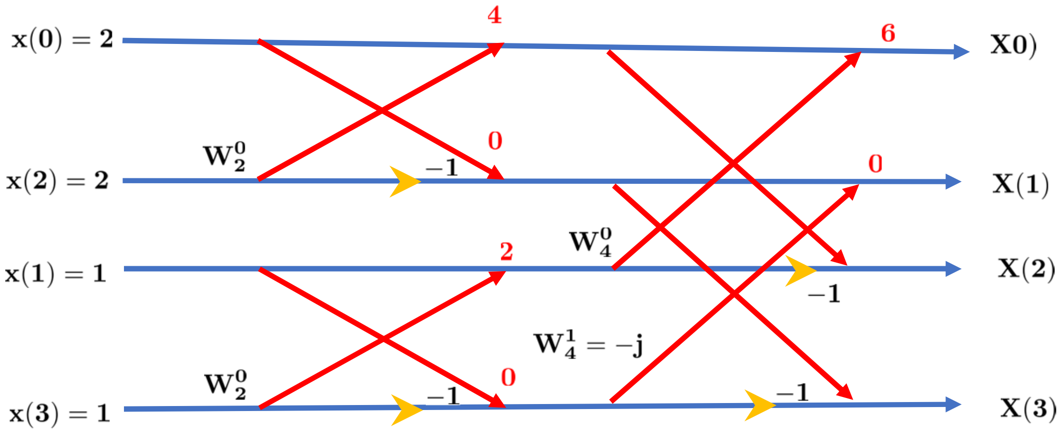
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



237

Number of computation in DFT and FFT

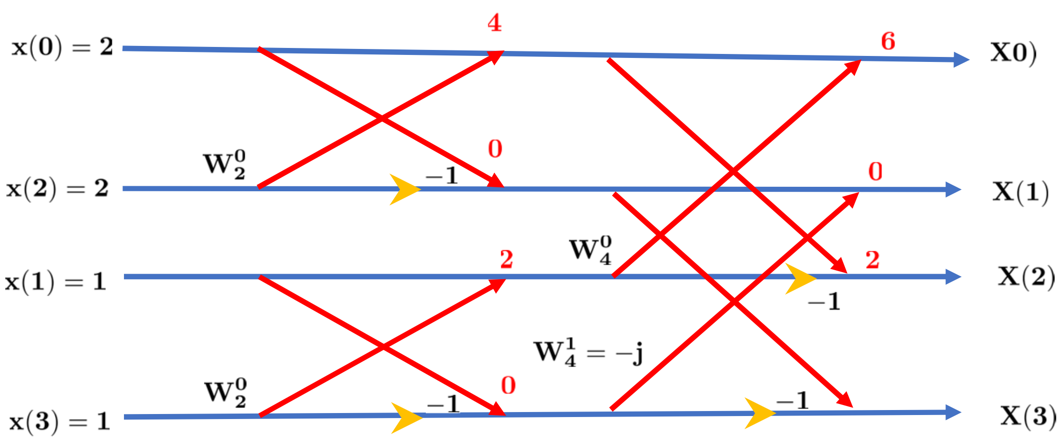
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



238

Number of computation in DFT and FFT

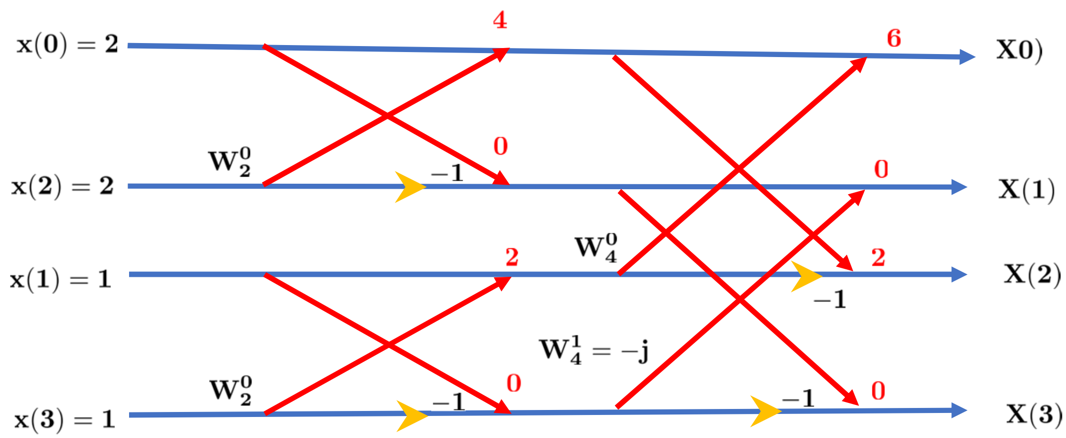
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



239

Number of computation in DFT and FFT

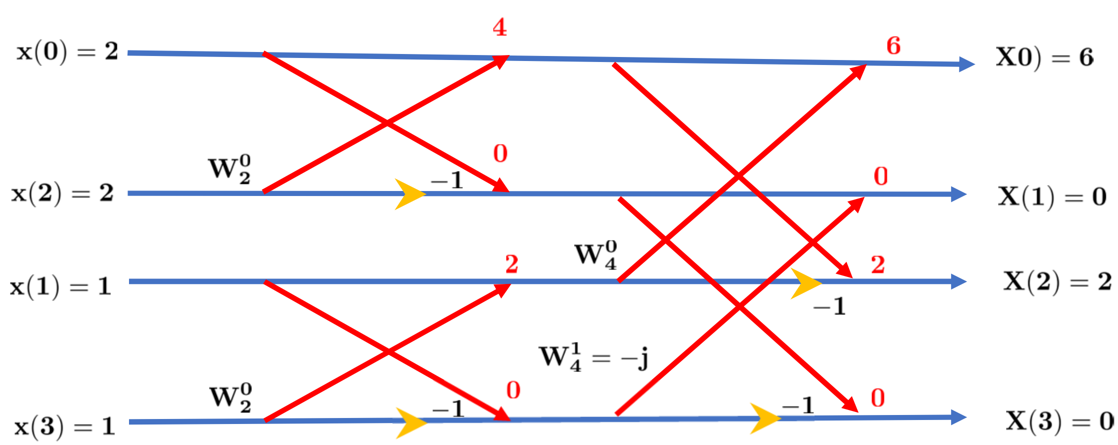
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



240

Number of computation in DFT and FFT

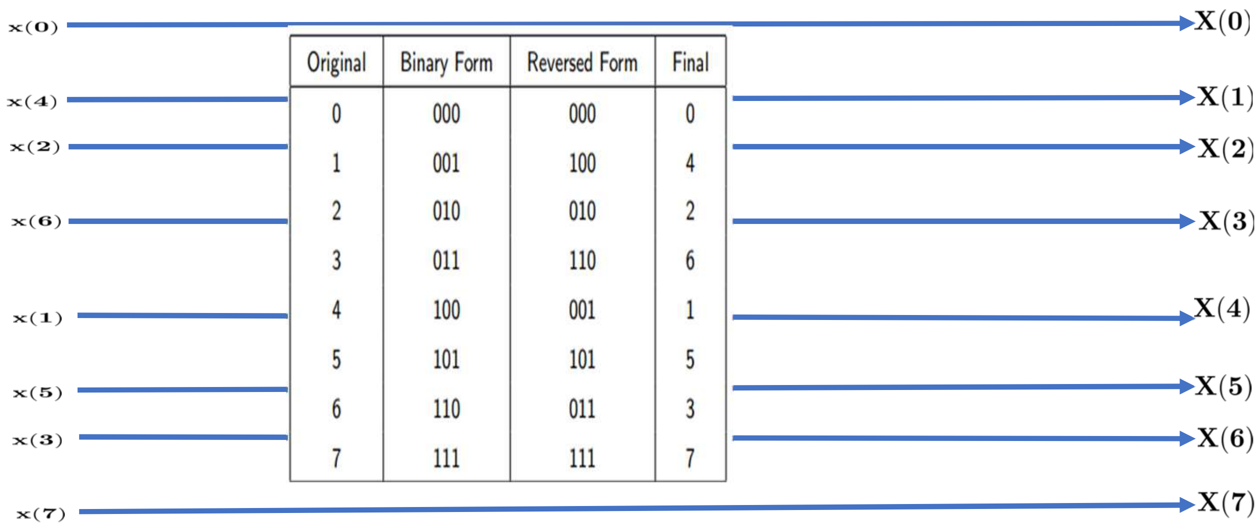
$$\mathbf{x[n] = \{2, 1, 2, 1\}}$$



$$\mathbf{X[k] = \{6, 0, 2, 0\}}$$

241

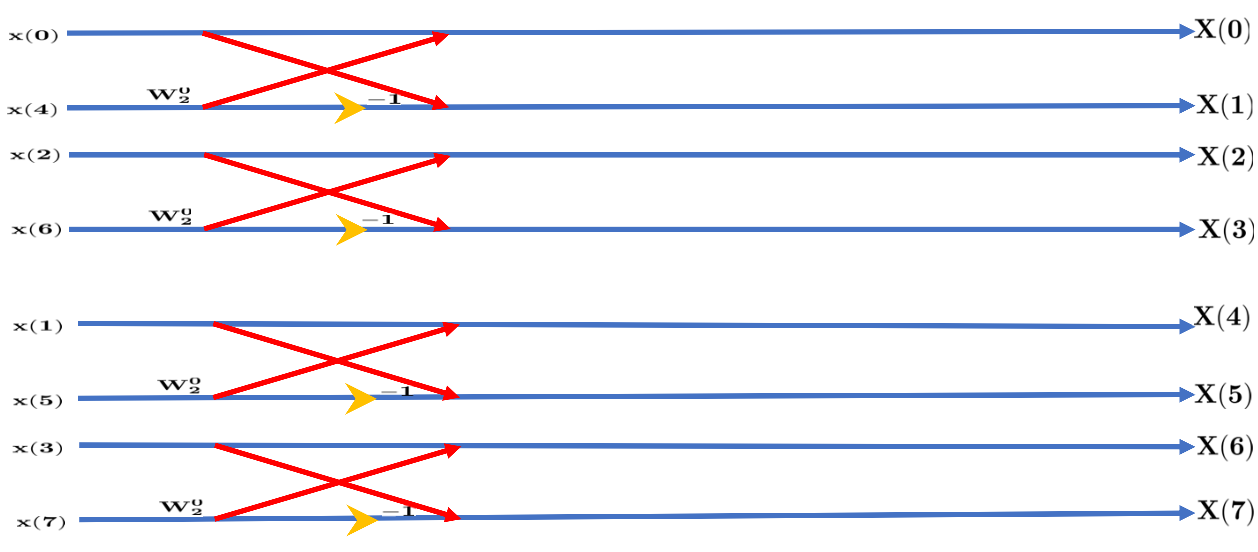
Find N=8 DFT with DIT FFT method $m = \log_2(N) = \log_2(8) = 3$



Original	Binary Form	Reversed Form	Final
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

242

Find N=8 DFT with DIT FFT method $m = \log_2(N) = \log_2(8) = 3$

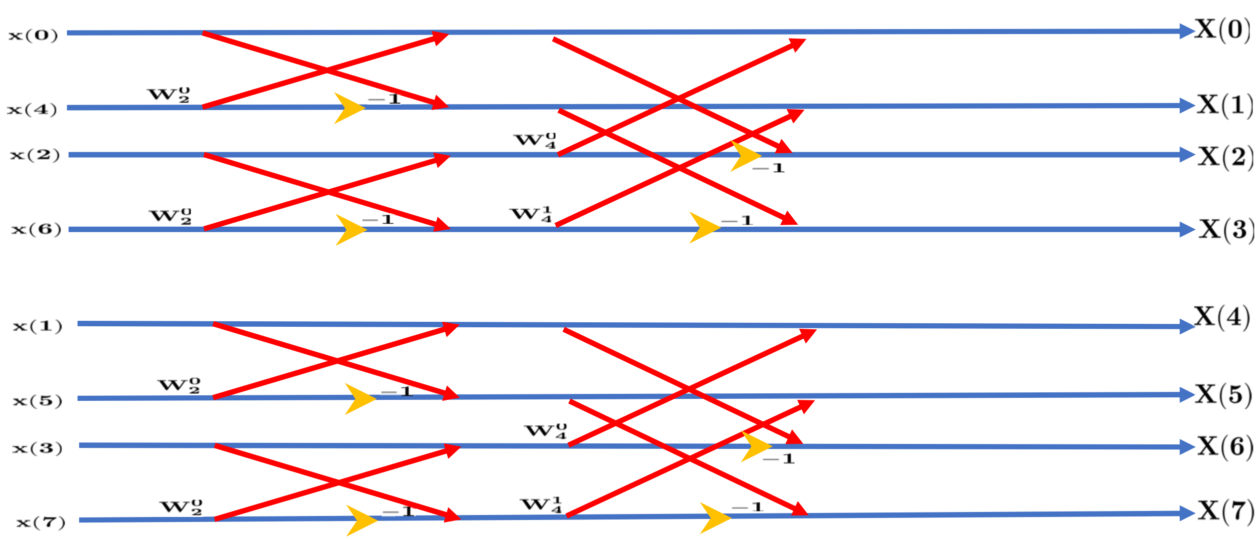


243

Find N=8 DFT with DIT FFT method $m = \log_2(N) = \log_2(8) = 3$

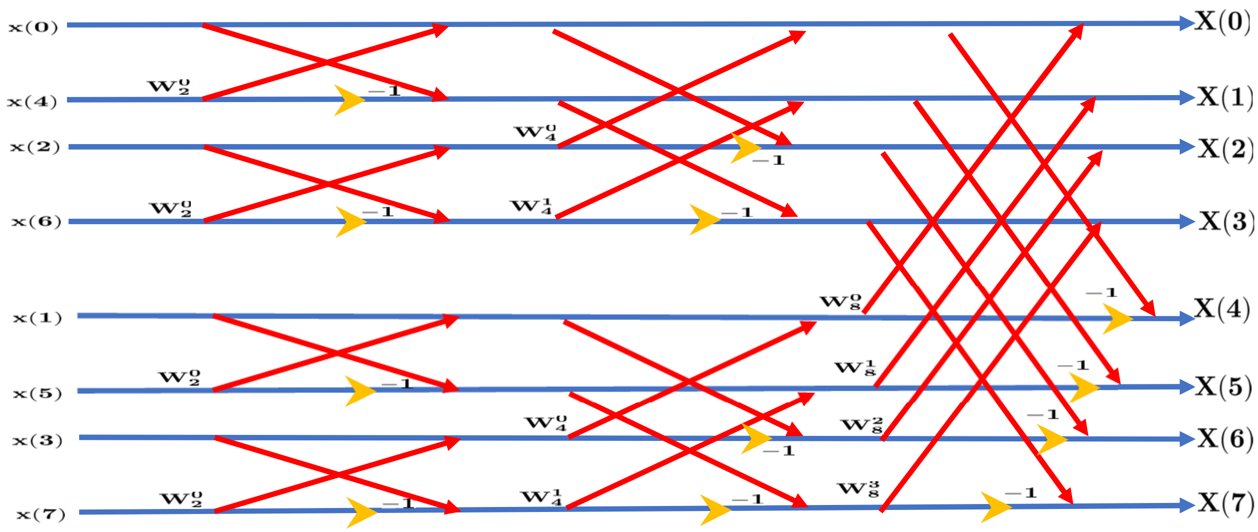
244

Find N=8 DFT with DIT FFT method $m = \log_2(N) = \log_2(8) = 3$



245

Find N=8 DFT with DIT FFT method $\mathbf{m = \log_2(N) = \log_2(8) = 3}$



246

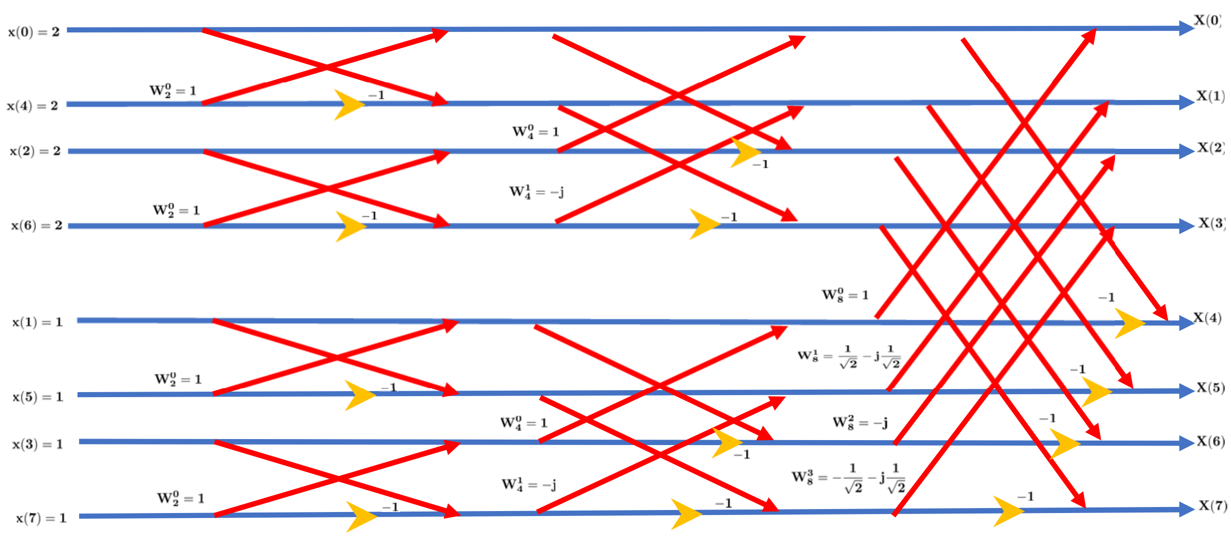
EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

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

247

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

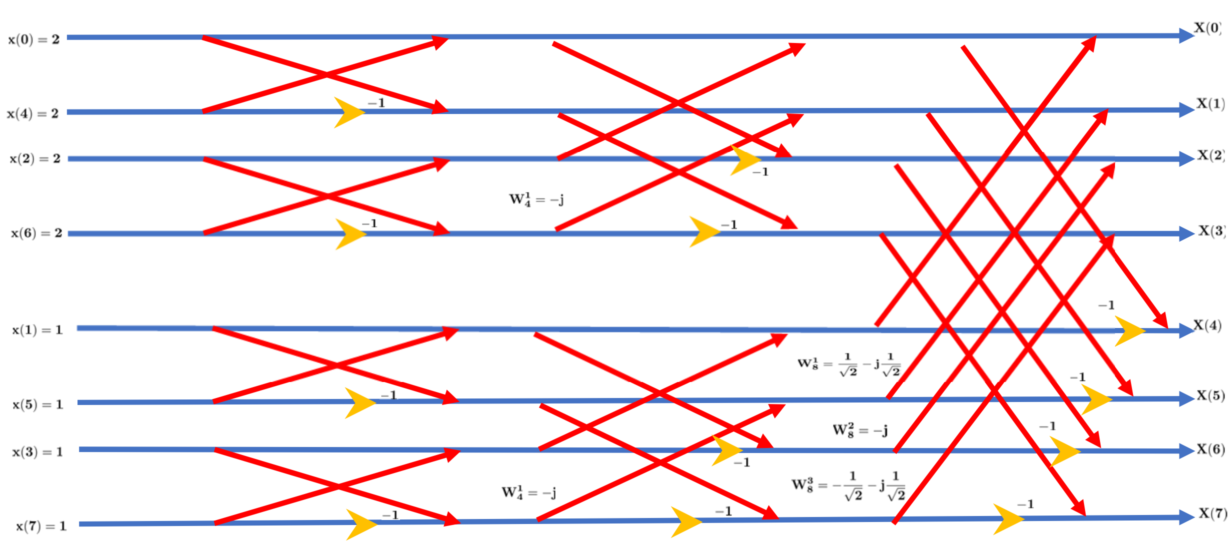
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



248

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

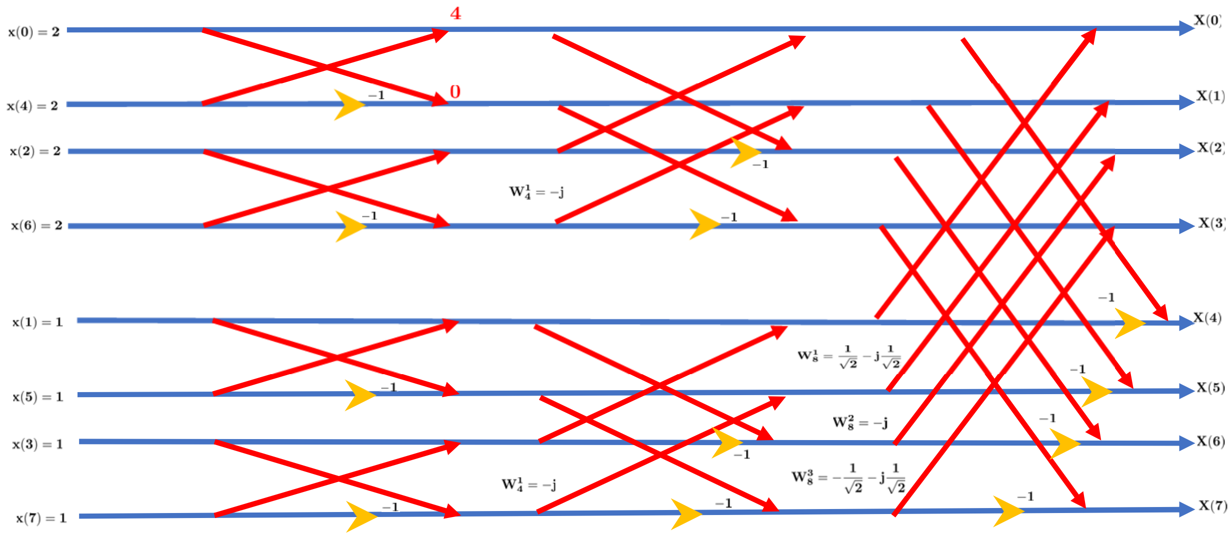
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



249

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

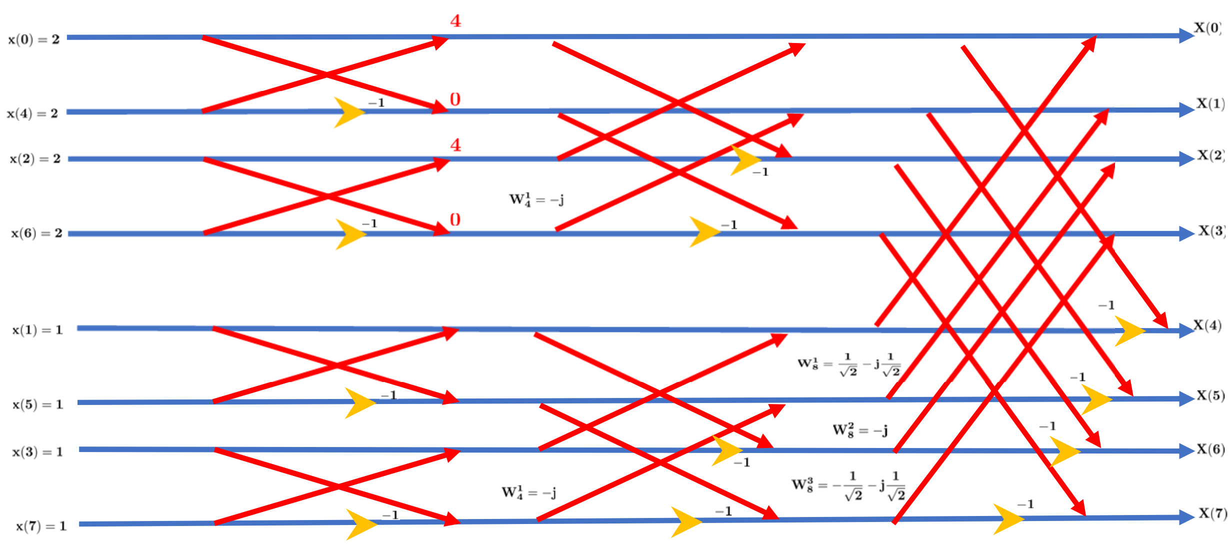
$$x(n) = \{2, 1, 2, 1, 2, 1, 2, 1\}$$



250

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

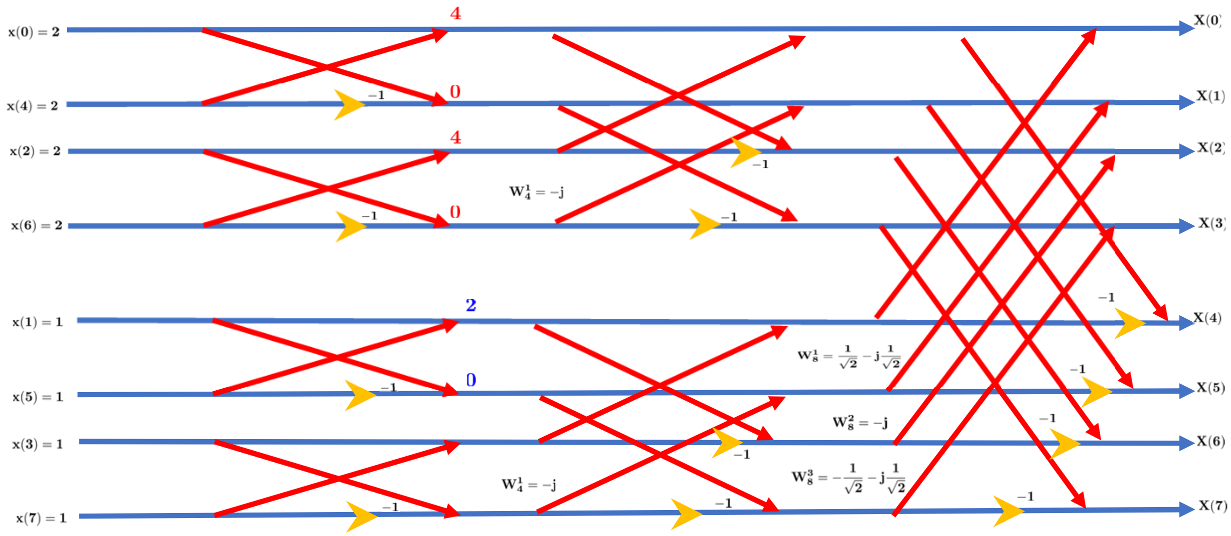
$$x(n) = \{2, 1, 2, 1, 2, 1, 2, 1\}$$



251

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

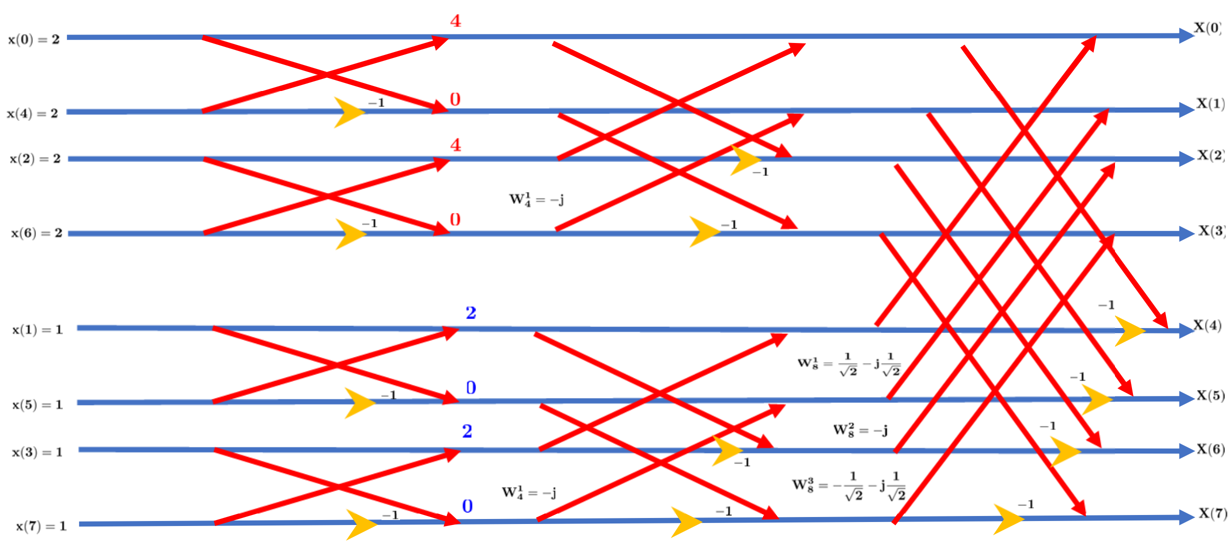
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



252

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

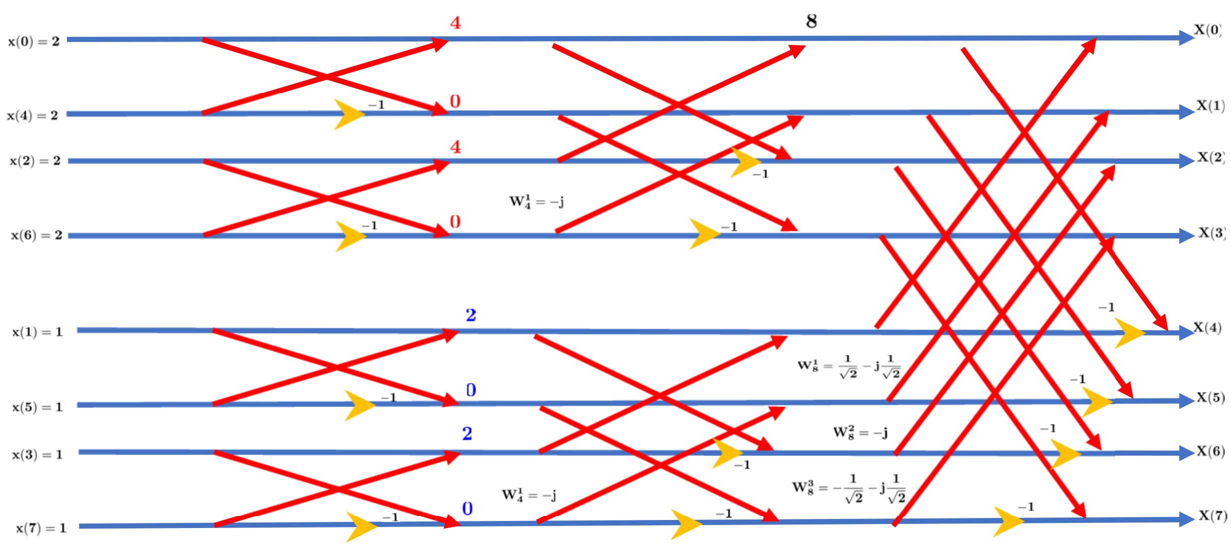
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



253

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

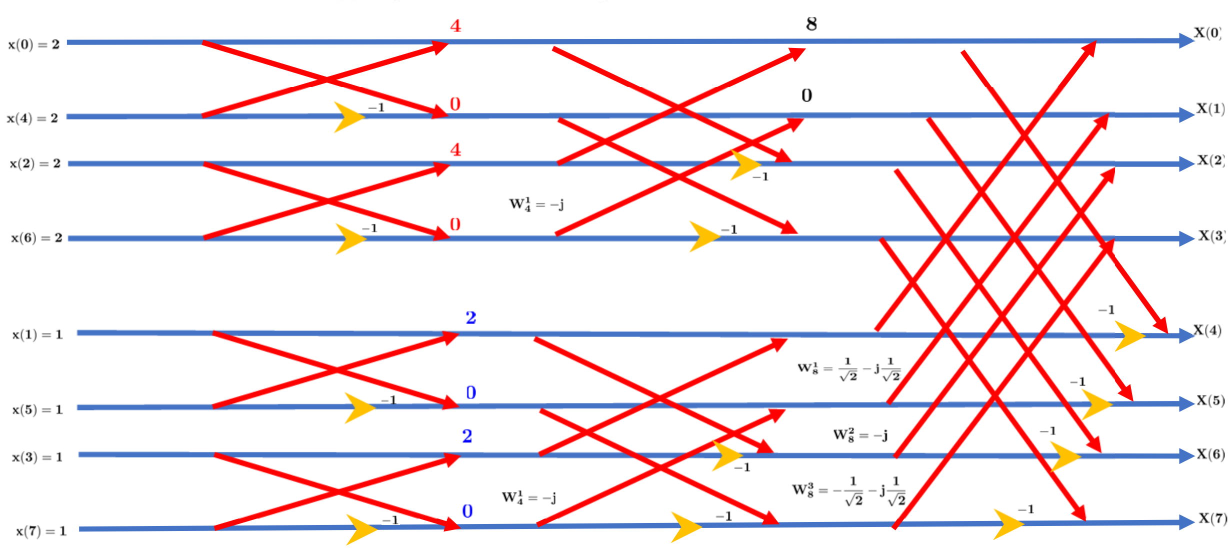
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



254

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

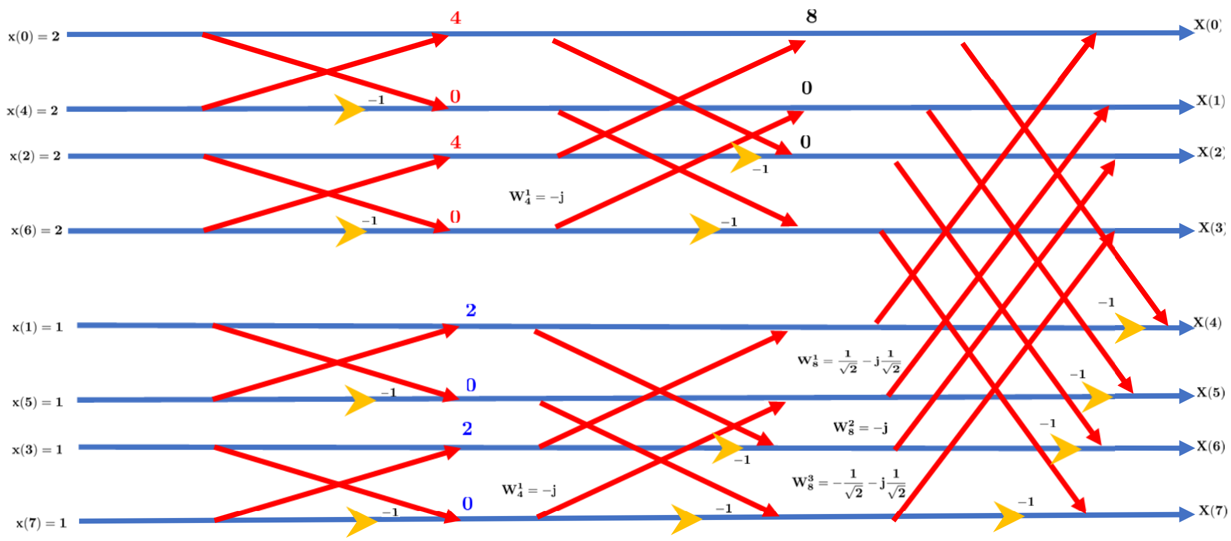
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



255

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

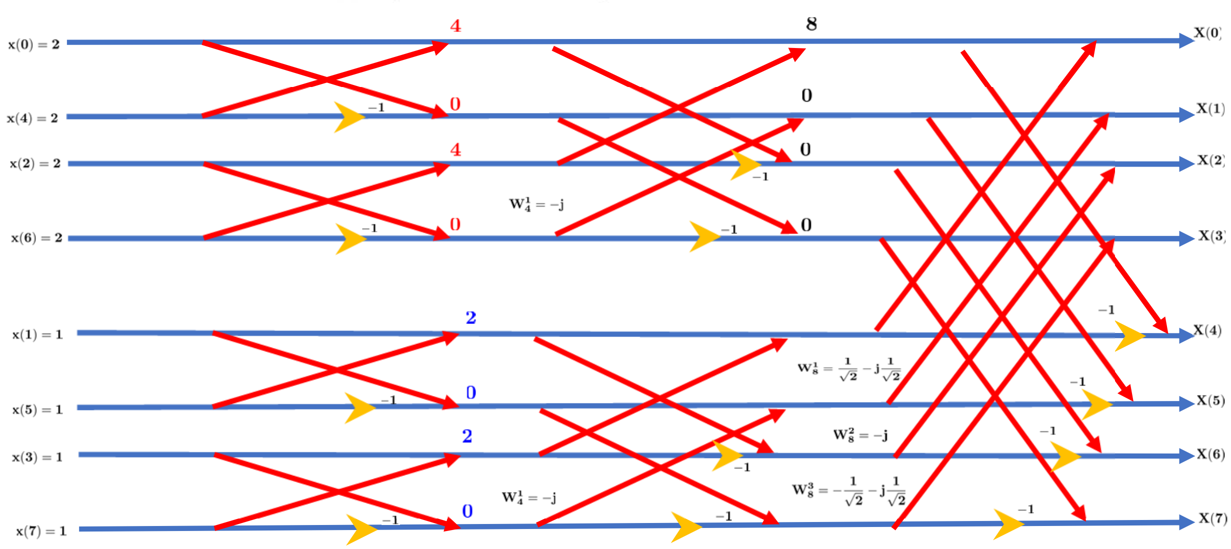
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



256

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

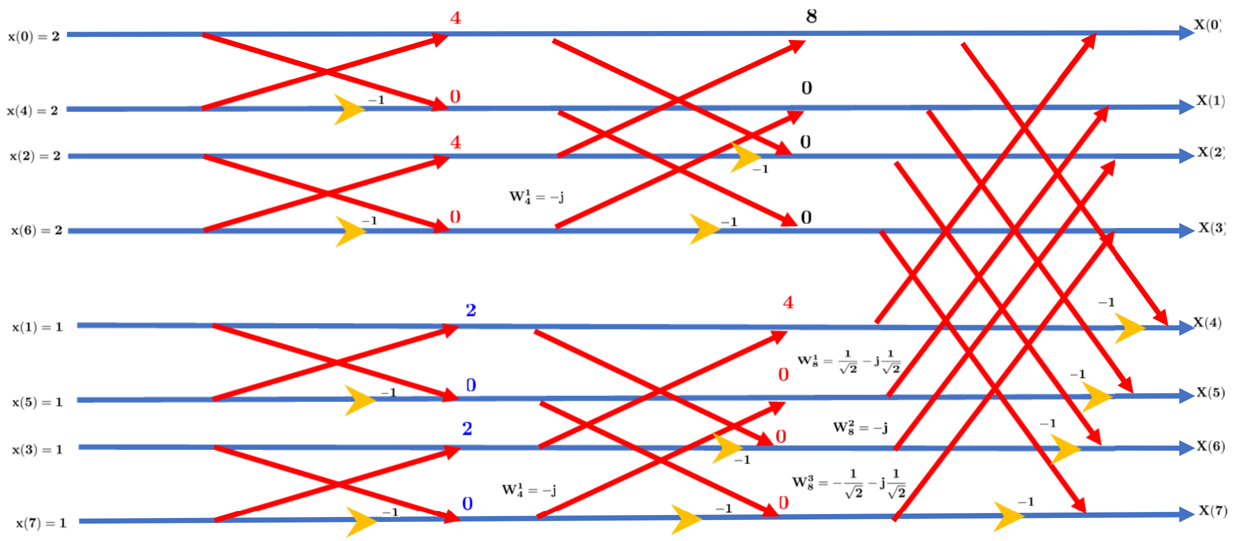
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



257

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

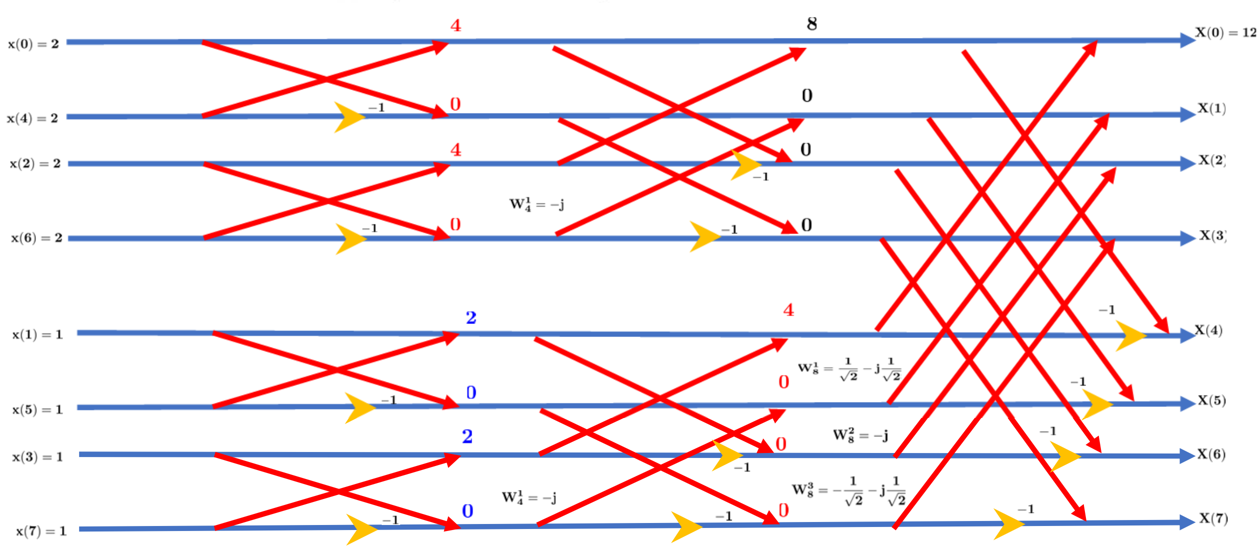
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



258

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

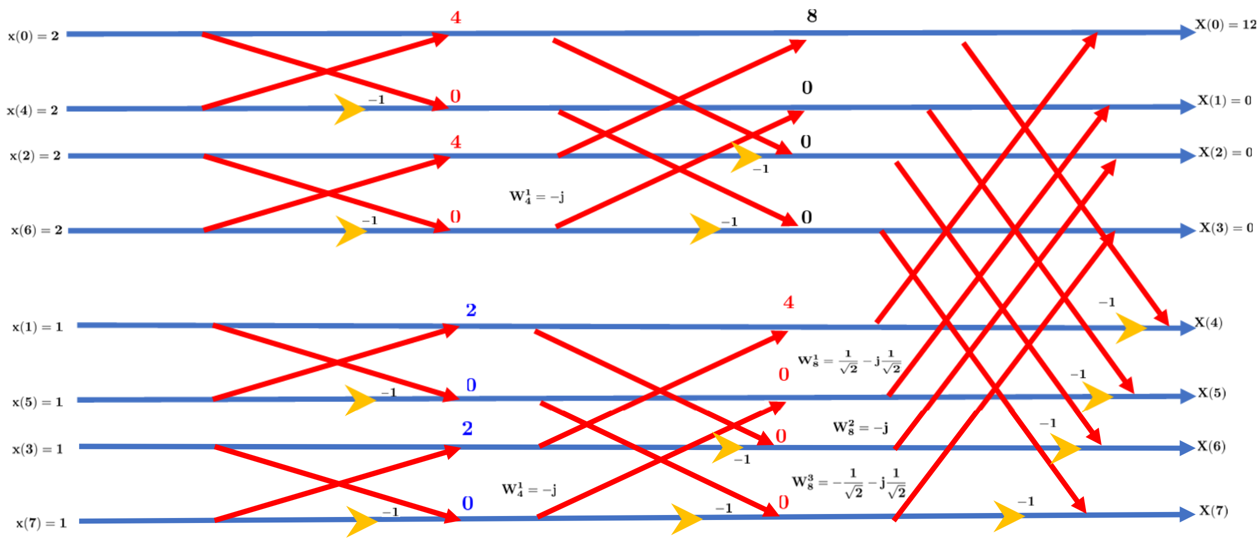
$$x(n) = \{2,\ 1,\ 2,\ 1,\ 2,\ 1,\ 2,\ 1\}$$



259

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

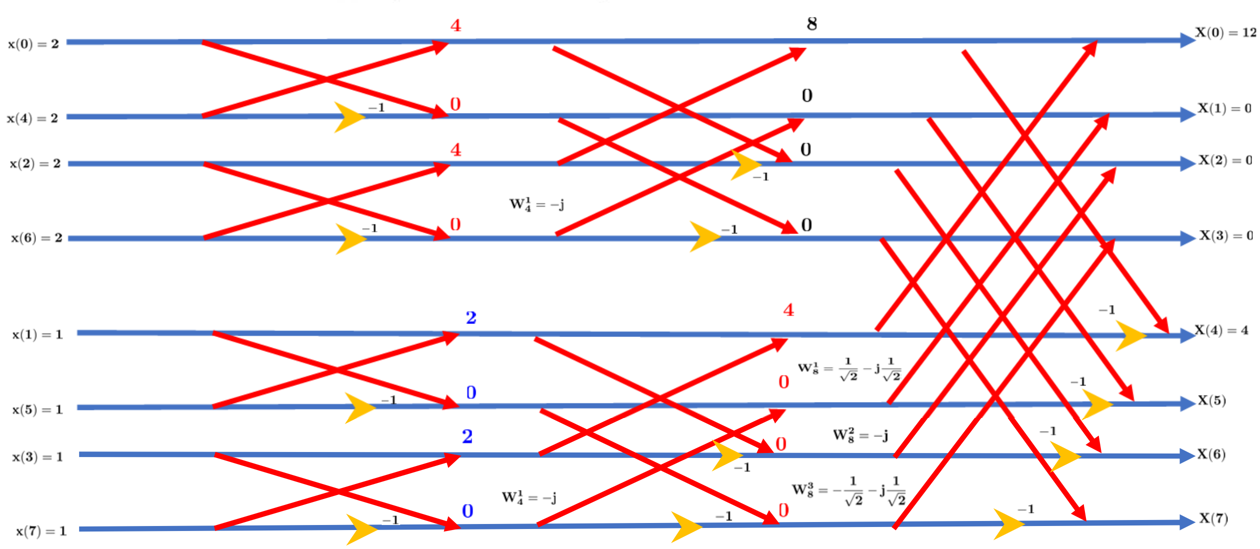
$$x(n) = \{2, 1, 2, 1, 2, 1, 2, 1\}$$



260

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

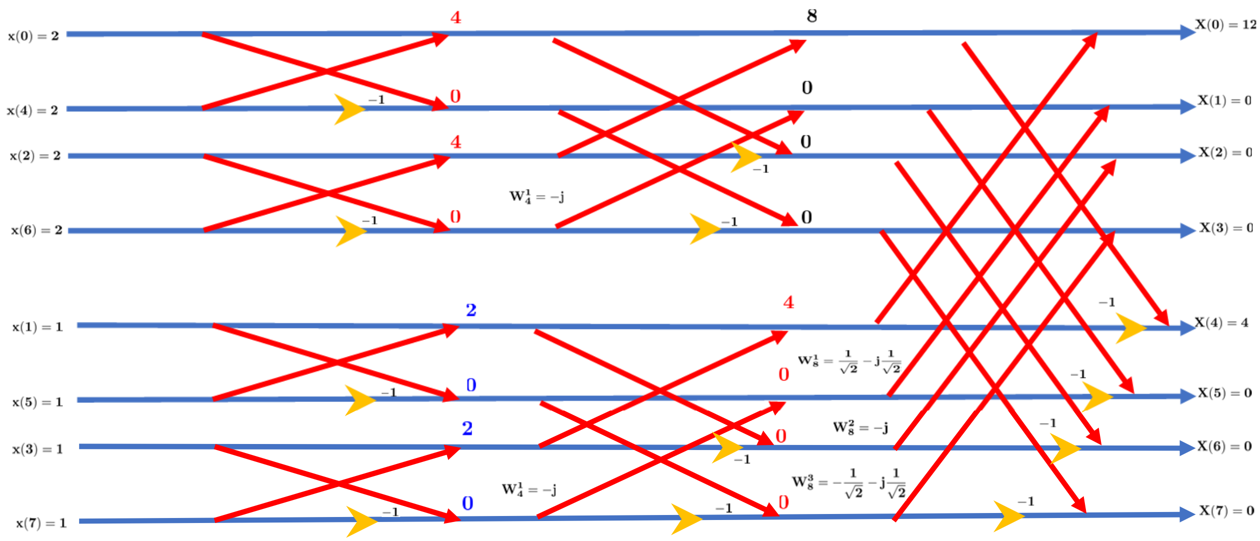
$$x(n) = \{2, 1, 2, 1, 2, 1, 2, 1\}$$



261

EXAMPLE 7.13 Find the 8-point DFT by radix-2 DIT FFT algorithm.

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

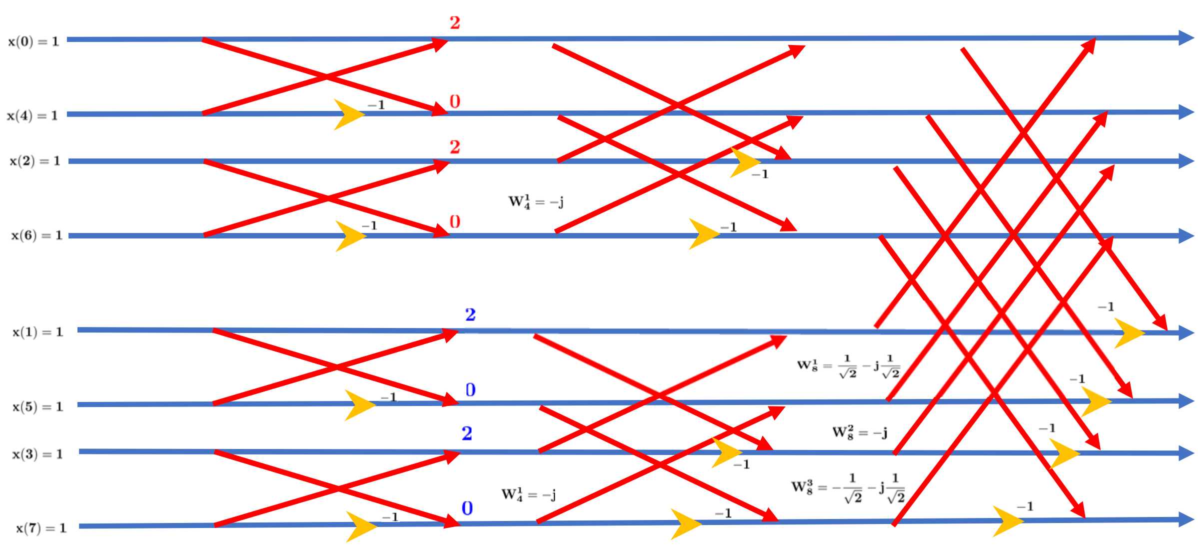


262

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.

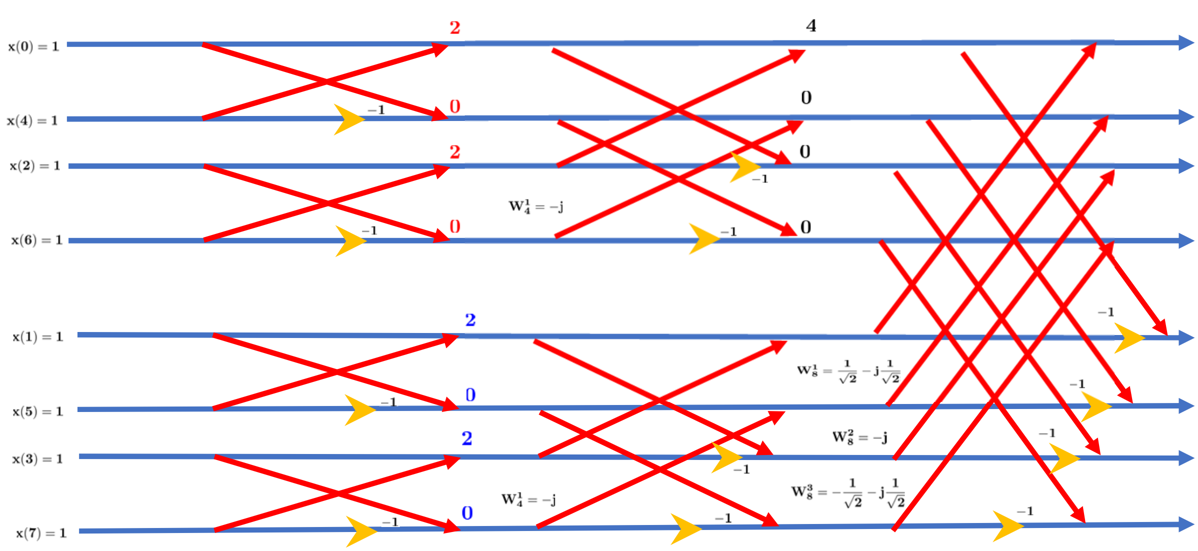
263

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.



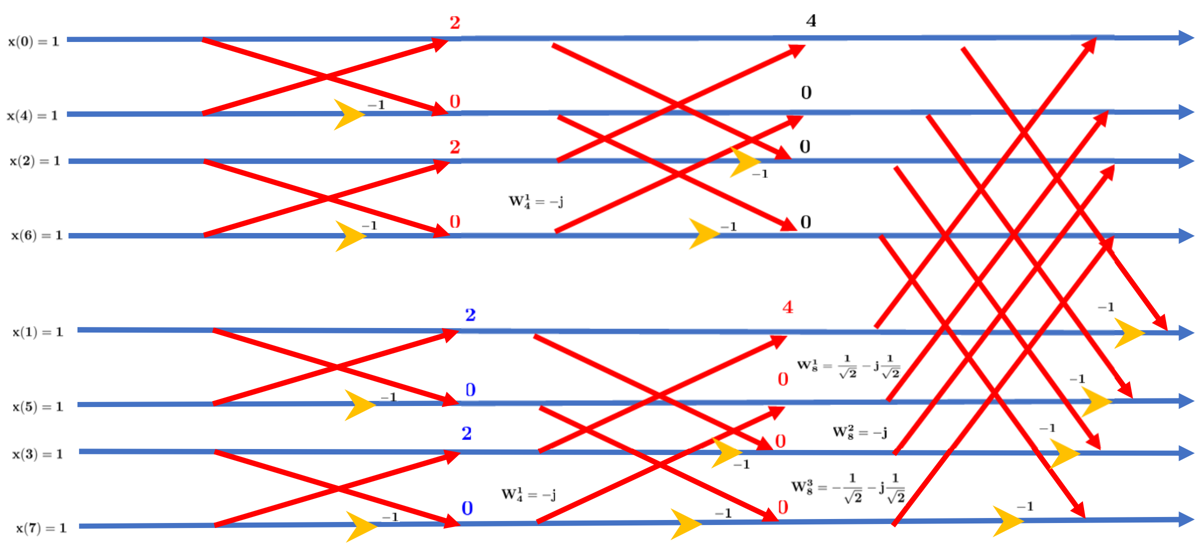
264

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.



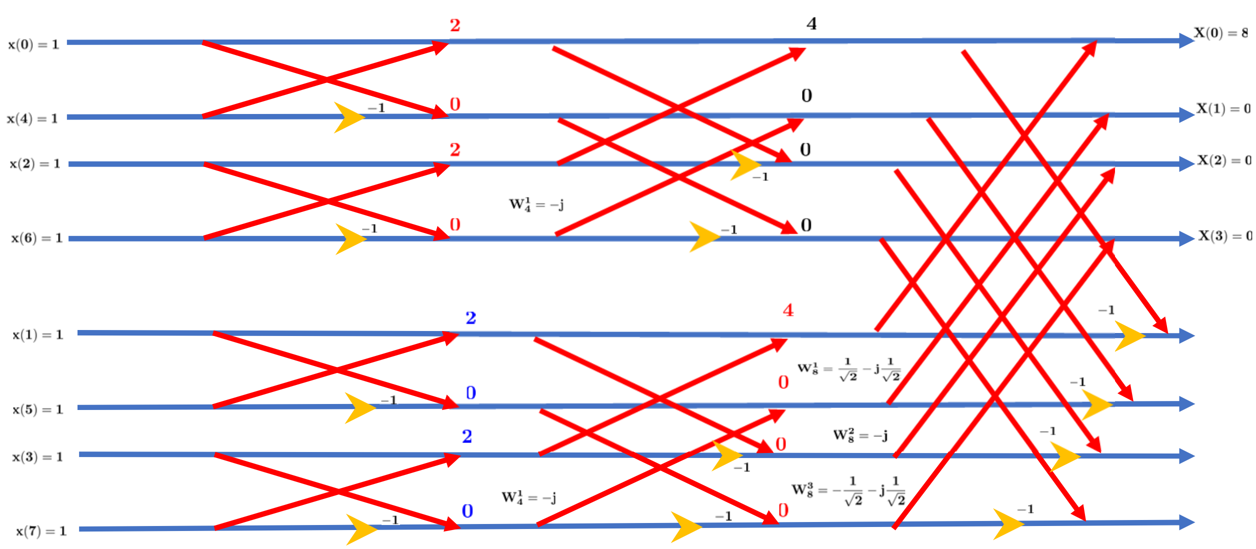
265

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.



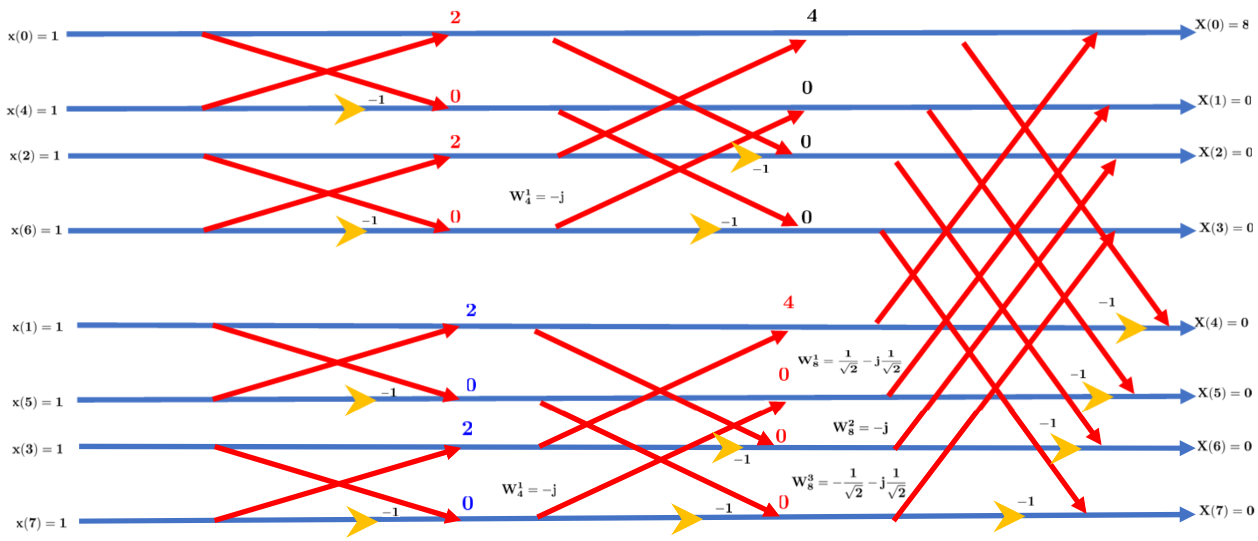
266

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.



267

EXAMPLE 7.14 Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.



268