# DSP Lecture 11: Radix-2 Fast Fourier transforms



DFT → numerical thing in Matlab ~~to~~ compute a

FT

When doing an FFT in Matlab, you are computing the DTFT (ideally this continuous function of $\omega$) → sampling at these equal intervals. If asking for a 10 point DFT, we get the DTFT sampled at $\frac{2\pi}{10}$. If a more finely spaced DTFT, ask for a longer FFt. Give me 200 points or 1000 points

→ Today efficients methods for computing DFT

Matlab command is FFT
↓
Fast

we want to do Fourier transforms, we need a lot of arithmetic

if we compute DFT of matrices having 1000's of elements,

→ would like to have a more efficient method. Old computers couldn't handle ~~this~~.

The DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n} \qquad \begin{array}{l} n = 0, 1 \ldots N-1 \\ k = 0, 1 \ldots N-1 \end{array}$$

Take N numbers in, get N numbers out, those N numbers gotten are samples of the DTFT (discrete representation) of frequency rewrites

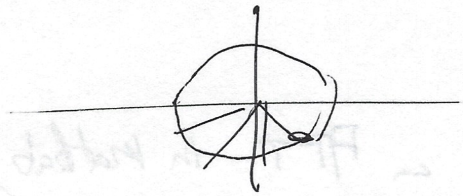
(2)

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

$n = 0, 1 \dots N-1$

$k = 0, 1 \dots N-1$

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



$N^{th}$ root of 1

$W_N, W_N^2, W_N^3, \dots W_N^N$ — $N^{th}$ roots of 1 (evenly spaced around the unit circle)

$e^{-j\frac{2\pi}{N}}, e^{-j\frac{2\pi}{N}(2)}, e^{-j\frac{2\pi}{N}(3)}, \dots e^{-j\frac{2\pi}{N}(N)}$

Input, output could be general complex numbers

So think about this,

To get $X[k]$ we

$$x[0] W_N^{k(0)} + x[1] W_N^{k} + x[2] W_N^{2k} + x[3] W_N^{3k}$$

$$+ \dots x[N-1] W_N^{k[N-1]}$$

we have N products

N values computing → repeat for all k's

$$\boxed{\begin{array}{l} N^2 \text{ complex multiplies} \\ N(N-1) \text{ complex adds} \end{array}}$$

well, when $N = 1000$, $10^6$ operations, → a lot

$N^2$ matrix

with regular random entry matrix, we can't simplify any more,

→ but we have a special matrix

→ recognizing a pattern of sorts we can drive down the computational costs

→ call it FFT (Not a single monolithic algorithm

→ Any scheme to make the ~~DFT~~ DFT faster

Reduce # of operations to $O(N\log_2 N)$

big O

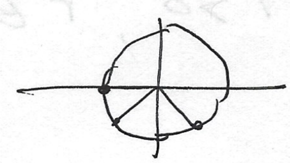
$N = \cancel{20}\ 2^{10} \approx 1000$

Naive way: 1000000 operations

Smart way $10(1000) = 10{,}000$

Say $N = 6$

$W_6^{3(\text{odd } k)} = e^{-j\pi}$



- Decompositions into smaller DFTs
- simplifications relating to $W_N^{(kN)} = 1$

$$W_N^{(N/2)(\text{odd } k)} = -1$$

- Remember, when we do the DFT, we assume $X(k)$ and $x[n]$ are periodic, $N$ is period

- $W_N^{n(k+N)} = W_N^{k(n+N)} = W_N^{kn}$ Periodicity

- $W_N^{k(N-n)} = W_N^{-kn} = \left(W_N^{kn}\right)^*$

'Symmetries'

④ Decimation in time. (N is an even ~~num~~ number)

Not a restrictive assumption

DFT is sampling underlying frequency response

→ So we are often free to choose N

$987 \rightarrow 1024$

Bump up to

$$X[K] = \sum_{n=0}^{N-1} x[n] W_N^{nK}$$

$$= \sum_{n \text{ even}} x[n] W_N^{nK} + \sum_{n \text{ odd}} x[n] W_N^{nK}$$

So $n = 2r$, integer $r$, $r > 0$, $r \in \mathbb{Z}$

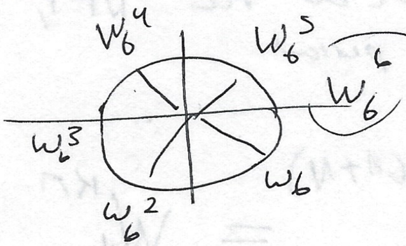
$$\sum_{r=0}^{\frac{N}{2}-1} x[2r] W_N^{2rK} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_N^{(2r+1)K}$$

0 2 4     1 3 5

$$= \sum_{r=0}^{\frac{N}{2}-1} x[2r] (W_N^2)^{rK} + W_N^K \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] (W_N^2)^{rK}$$

Length $\frac{N}{2}$ thing → smaller DFT

say $N = 6$




$W_3^3$

$W_6^4 = W_3^2$

$W_6^2 = W_3^1$

So then $W_N^2 = W_{\frac{N}{2}}^1$

So we can see

$$\underbrace{\sum_{r=0}^{N/2-1} x[2r]\, W_{N/2}^{rk}}_{\text{length } N/2 \text{ DFT of even entries}} + W_N^k \underbrace{\sum_{r=0}^{N/2-1} x[2r+1]\, W_{N/2}^{rk}}_{\text{length } N/2 \text{ DFT of odd entries}}$$

two shorter DFTs

So then

$$X[k] = G[k] + W_N^k H[k] \qquad k = 0, 1 \ldots N-1$$

for N = 6

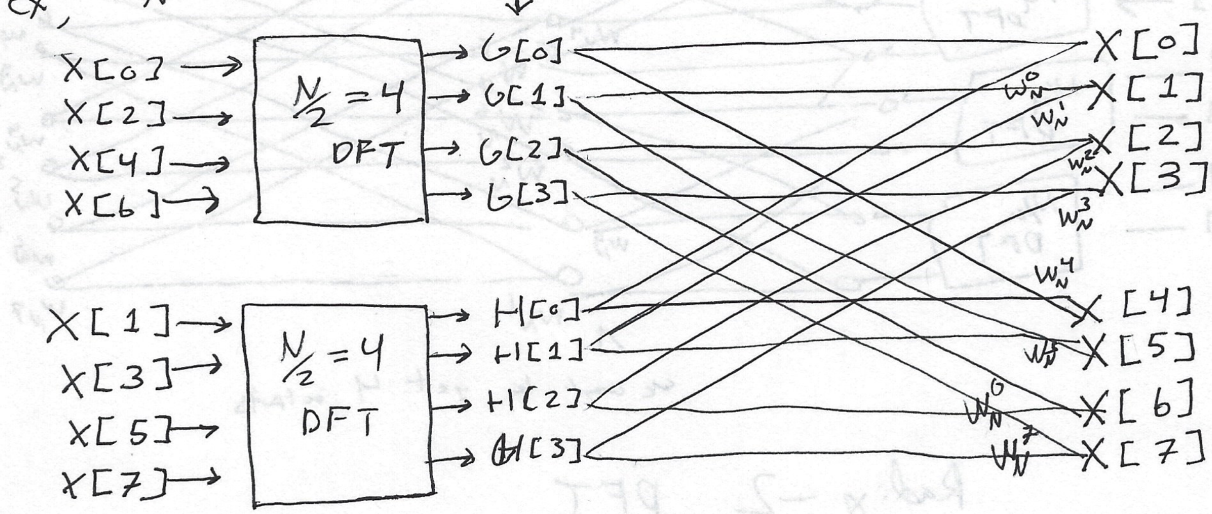
6 ranges from $0 \to 3$

$$G[0] = G[3]$$
$$G[1] = G[4]$$
$$G[2] = G[5]$$

a when we need to call a higher entry of k, than in the shorter DFT just wrap around cause this is a periodic thing

Ex, $N = 8$ DFT

Sums of the X[K]'s




For N DFT, Normally there are $N^2$ multiplications

for this example, we only needed $\left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + N = \frac{N^2 + N^2 + N(4)}{4}$

$\approx \frac{N}{2}$

we reduced multiplications by $\frac{1}{2}$ roughly.

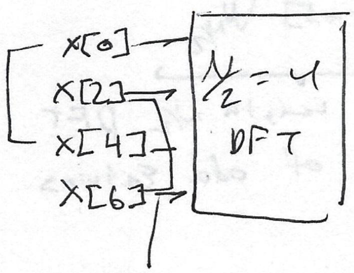
So, why stop there.

6

## Breakdown again:

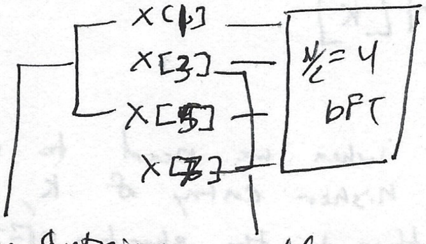
Focus on the Top

Take the even parts of these inputs



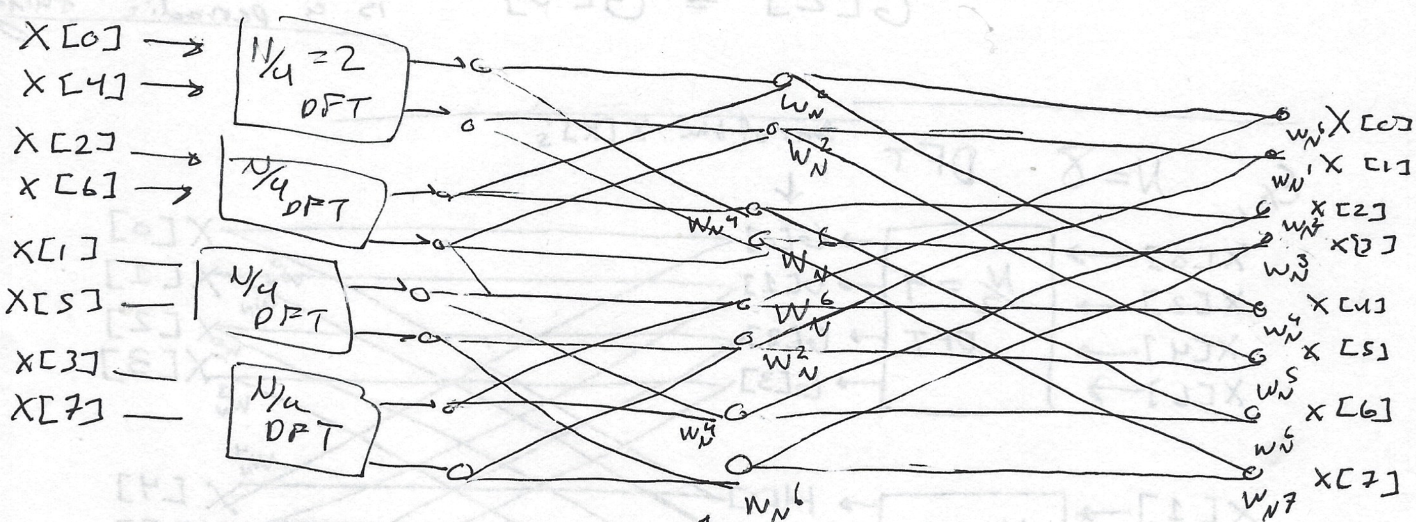

Take the odd entries

Radix 4 DFT is in the text




Even Entries odd Entries

So reorder the input in a slightly different way




we want to get 4 outputs

Radix-2 DFT

Length 2 DFT

$x[0], x[1] \longrightarrow X[0], X[1]$

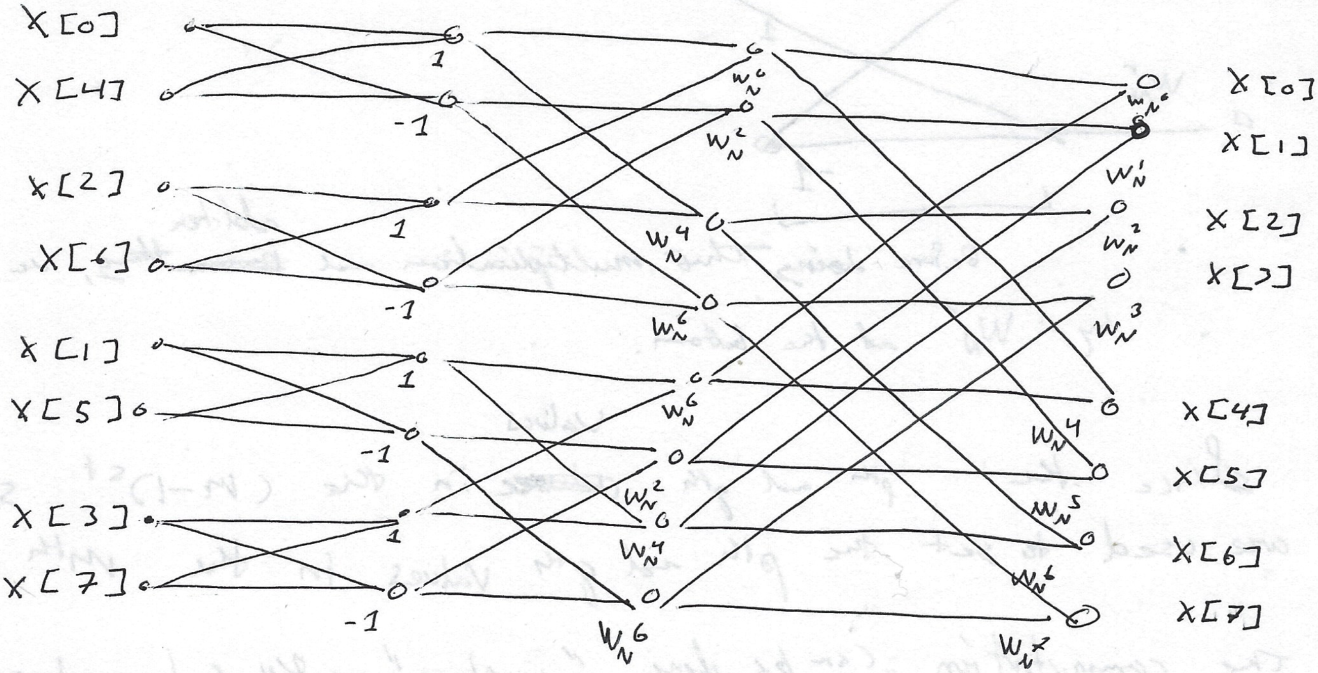
$$X[k] = \sum_{n=0}^{1} x[n] W_2^{nk}$$

$$= \sum_{n=0}^{1} x[n] (-1)^{nk}$$

$X[0] = x[0] + x[1]$

$X[1] = x[0] - x[1]$

now describe




Σ Not as much complex multiplies

If N is a power of 2, we can recursively break the DFT into $\underline{\log_2 N \text{ stages}}$

$\Longrightarrow$ $N \log_2 N$ multiplies

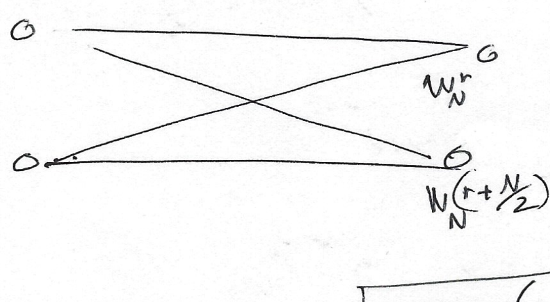
So ~~N=20~~ $N = 2^{10} = 1024$

$N^2 = 1000,000$ vs. $\underline{N \log N = 10,240}$ $\underline{100X \text{ faster}}$

Recognize ~~that~~ in each of these stages, there is a pattern,

Take 2 elements get 2 other elements

This case, we are combining the same two,



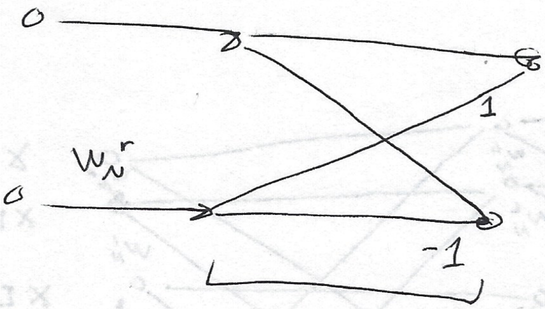

$W_N^1$, $W_N^5$ separated by 4.

$4 = 8/2$

So then $W_n^2$, $W_n^6$ are also separated by 4

⑧ We can see ~~that~~

$$W_N^{r+(N/2)} = W_N^r W_N^{N/2} = W_N^r e^{-j\frac{2\pi}{N}\frac{N}{2}} = -W_N^r$$




Before doing this multiplication and ~~[illegible]~~ addition ~~things~~, we multiply by $W_N^r$ at the bottom.

Since the $p$th and $q$th ~~values~~ values in the $(m-1)^{st}$ stage are used to set the $p$th and $q$th values in the $m$th stage, the computation can be done "in place" – no extra storage

"Bit Reversed order" of input

Reverse

000 ← 000 = x[0]
001 ← 100 = x[4]

010 — 010 = x[2]
011 ~~110~~ ← 110 = x[6]

100 ← 001 = x[1]
101 — 101 = x[5]

110 ← 011 = x[3]
111 — 111 = x[7]

$W_N^{-2} = \frac{4\pi}{8}$ , $\frac{-12}{8}\pi \approx W_N^{6}$

$W_8^{-2} = -W_8^{2}$ ; $j \to -j$

X[k]'s

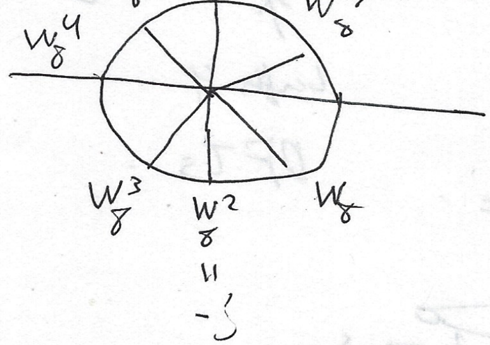
$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \\ X[4] \\ X[5] \\ X[6] \\ X[7] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W_8 & W_8^2 & W_8^3 & -1 & -W_8 & -W_8^2 & -W_8^3 \\ 1 & W_8^2 & -1 & W_8^{-2} & 1 & W_8^2 & -1 & W_8^{-2} \\ 1 & W_8^3 & j & W_8 & -1 & -W_8^3 & -j & -W_8 \\ \hline 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & W_8^5 = -W_8 & W_8^2 & -W_8^3 & -1 & W_8 & W_8^6 = -W_8^2 & W_8^3 \\ 1 & -W_8^2 & -1 & W_8^2 & 1 & -W_8^2 & -1 & W_8^2 \\ 1 & -W_8^3 & -W_8^2 & -W_8 & -1 & W_8^3 & W_8^2 & W_8 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}$$

x[n]'s

Our $W_8^6 = -W_8^2$, $W_8^6 = j$

$-W_8 = W_8^5$

$W_8^7 = -W_8^3$




→ cut the matrix in half, horizontally, then divide the columns

the top even columns are the same (0 → 6) as the bottom.

→ The odd columns are negations of each other

Think

$$\underbrace{\begin{bmatrix} I_{4\times4} \\ I_{4\times4} \end{bmatrix}}_{8\times4} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_8^2 & -1 & -W_8^2 \\ 1 & -1 & 1 & -1 \\ 1 & -W_8^2 & -1 & W_8^2 \end{bmatrix}}_{\text{DFT for } N=4}$$

$$\begin{bmatrix} I \\ I \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & -1 & 1 & -1 \\ 1 & -W_4 & -1 & W_4 \end{bmatrix}$$

"Think Decomposition of the matrix into a smaller DFT"

(10)

Odd Columns

$$\begin{bmatrix} I_{4\times4} \\ -I_{4\times4} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ W_8 & W_8^3 & -W_8 & W_8^{7} \\ W_8^2 & -W_8^2 & W_8^2 & -W_8^2 \\ W_8^3 & W_8 & -W_8^3 & -W_8 \end{bmatrix}$$

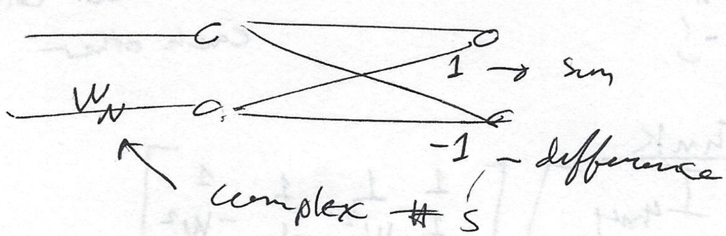
$$= \begin{bmatrix} I \\ -I \end{bmatrix} \begin{bmatrix} 1 & & & \\ & W_8 & & \\ & & W_8^2 & \\ & & & W_8^3 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_8^2 & -1 & -W_8^2 \\ 1 & -1 & 1 & -1 \\ 1 & -W_8^2 & -1 & W_8^2 \end{bmatrix}}_{F_4}$$

$$F_8 = \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} I & \\ & \begin{bmatrix} 1 & & & \\ & W_8 & & \\ & & W_8^2 & \\ & & & W_8^3 \end{bmatrix} \end{bmatrix} \begin{bmatrix} F_4 & \\ & F_4 \end{bmatrix} \begin{bmatrix} X_{even} \\ X_{odd} \end{bmatrix}$$

Sums and Differences in "Butterfly"

"Twiddle factors" → multiplication by complex #

Length 4 DFTs




This is called a 'Decimation in time strategy'

Taking time domain signal and chunking it up into little pieces (smaller and smaller subunits)

Decimation in frequency FFT

Instead of keeping the output order fixed and stuffing around the input, keep input order fixed and shuffle around the output

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

Even samples of $X[k]$ $X[2r]$ $r = 0, 1, \ldots \frac{N-1}{2}$

$$X[2r] = \sum_{n=0}^{N-1} x[n] W_N^{2nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[n] W_N^{2nr} + \sum_{n=\frac{N}{2}}^{N-1} x[n] W_N^{2nr}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[n] W_{N/2}^{nr} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{2(n+N/2)r}$$

We can reindex this

we may collect terms like this

$$= \sum_{n=0}^{N/2-1} (x[n] + x[n+N/2]) W_{N/2}^{nr}$$

Get the even entries of the output

Take Top half / bottom half of input and take a shorter DFT of that.

Like An $N/2$ DFT of Summed input
(Top half + Bottom half)

for odd entries of $X[k]$, can show that these odd entries look like a shorter DFT applied to the difference between top half and Bottom half

(12) We show this

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

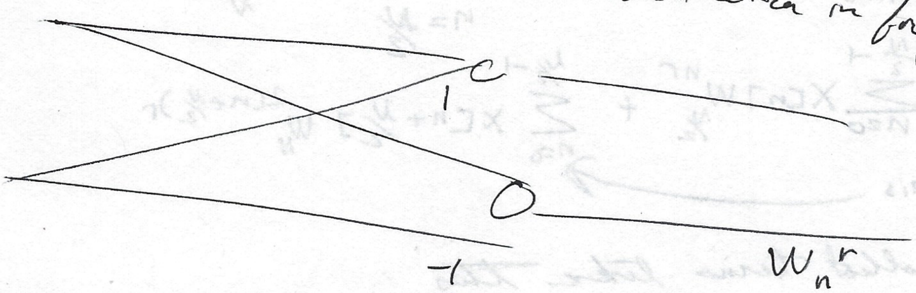
we have this Twiddle factor ~~that~~ comes into play

We are now getting the output in terms of Even Entries and odd Entries.

→ we can keep the input in the same order if we

want

Butterfly looks like (decimation in frequency) FFT




DFT chapter 7