Fast Fourier Transform and Convolution Theorem

Discrete Fourier Transform

principal

R commutative ring with n'th root of unity ω.

Column vector

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Matrix

$$A_{i,j} = \omega^{ij} \quad , \quad i,j \in [0 : n-1]$$

Discrete Fourier Transform

principal

R commutative ring with n'th root of unity ω.

Column vector

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Matrix

$$A_{i,j} = \omega^{ij} \quad , \quad i, j \in [0 : n-1]$$

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \\ \vdots & & & & \\ 1 & & & & \end{pmatrix}$$

Fourier Transform

$$f_n(a) = A * a \quad \text{vector product}$$

Discrete Fourier Transform

principal

R commutative ring with n'th root of unity ω.

Column vector

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Matrix

$$A_{i,j} = \omega^{ij} \quad , \quad i,j \in [0:n-1]$$

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \\ \vdots & & & & \\ 1 & & & & \end{pmatrix}$$

Fourier Transform

$$f_n(a) = A * a \quad \text{vector product}$$

$$\begin{aligned} f_n(a) &= (f_0, \dots, f_{n-1}) \\ f_{n,i} &= \sum_{j=0}^{n-1} \omega^{ij} a_j \end{aligned}$$

Back transformation:

matrix A' such that matrix product AA' is identity matrix

$$A * A' = I^n \quad , \quad I^n_{i,j} = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Discrete Fourier Transform

principal

R commutative ring with n'th root of unity ω.

Column vector

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Matrix

$$A_{i,j} = \omega^{ij} \quad , \quad i,j \in [0:n-1]$$

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \\ \vdots & & & & \\ 1 & & & & \end{pmatrix}$$

Fourier Transform

$$f_n(a) = A * a \quad \text{vector product}$$

$$f_n(a) = \cancel{(f_0, \dots, f_{n-1})} \quad (f_{n,0}, \dots, f_{n,n-1})$$

$$f_{n,i} = \sum_{j=0}^{n-1} \omega^{ij} a_j$$

Back transformation:

matrix A' such that matrix product AA' is identity matrix

$$A * A' = I^n \quad , \quad I^n_{i,j} = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

$$\boxed{\begin{array}{l} \textbf{Lemma 4.}\ \textit{If } \omega^{-1} \textit{ and } n^{-1} \textit{ exists in } R \textit{ and} \\ \qquad A'_{i,j} = \frac{1}{n} \cdot \omega^{-ij} \\ \textit{then } A' \textit{ is inverse Fourier Transform} \end{array}}$$

Discrete Fourier Transform

principal

R commutative ring with n'th root of unity ω.

Column vector

$$a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Matrix

$$A_{i,j} = \omega^{ij} \quad , \quad i,j \in [0:n-1]$$

Lemma 4. *If ω^{-1} and n^{-1} exists in R and*

$$A'_{i,j} = \frac{1}{n} \cdot \omega^{-ij}$$

then A' is inverse Fourier Transform

$$\begin{aligned}
(AA')_{i,j} &= \frac{1}{n}\sum_{k=0}^{n-1} \omega^{ik}\omega^{-kj} \\
&= \frac{1}{n}\sum_{k=0}^{n-1} \omega^{k(i-j)} \\
&= \begin{cases} 1 & i=j \\ 0 & i>j \quad (\omega \quad \text{pricipal root of unity}) \end{cases} \\
i<j \rightarrow \sum_{k=0}^{n-1} \omega^{k(i-j)} &= \omega^{kn}\sum_{k=0}^{n-1} \omega^{k(i-j)} \\
&= \sum_{k=0}^{n-1} \omega^{k(n+i-j)} \\
&= 0 \quad (n+i-j \in [1:n-1])
\end{aligned}$$

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

Reduce to 2 problems of half the size.

Even indices $2i$, $i \in [0 : n/2-1]$:

$$
\begin{aligned}
g_i(a) &= f_{n,2i}(a) \\
&= \sum_{j=0}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{2ij} a_j + \sum_{j=n/2}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2i(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2ij} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j + a_{n/2+j}) \\
&= f_{n/2}(b) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
b_j &= a_j + a_{n/2+j} \quad j \in [0, n/2-1]
\end{aligned}
$$

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

Reduce to 2 problems of half the size.

Even indices $2i$, $i \in [0 : n/2-1]$:

$$
\begin{aligned}
g_i(a) &= f_{n,2i}(a) \\
&= \sum_{j=0}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{2ij} a_j + \sum_{j=n/2}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2i(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2ij} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j + a_{n/2+j}) \\
&= f_{n/2}(b) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
b_j &= a_j + a_{n/2+j} \quad j \in [0, n/2-1]
\end{aligned}
$$

Odd indices $2i+1$,$i \in [0 : n/2-1]$:

$$
\begin{aligned}
h_i(a) &= f_{n,2i+1}(a) \\
&= \sum_{j=0}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{(2i+1)j} a_j + \sum_{j=n/2}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)j} \omega^{n/2} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j - \omega^{2ij} a_{n/2+j}) \omega^j \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j - a_{n/2+j}) \omega^j \\
&= f_{n/2}(c) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
c_j &= (a_j - a_{n/2+j}) \omega^j \quad j \in [0, n/2-1]
\end{aligned}
$$

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

Reduce to 2 problems of half the size.

Even indices $2i$, $i \in [0 : n/2-1]$:

$$\begin{aligned}
g_i(a) &= f_{n,2i}(a) \\
&= \sum_{j=0}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{2ij} a_j + \sum_{j=n/2}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2i(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2ij} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j + a_{n/2+j}) \\
&= f_{n/2}(b) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
b_j &= a_j + a_{n/2+j} \quad j \in [0, n/2-1]
\end{aligned}$$

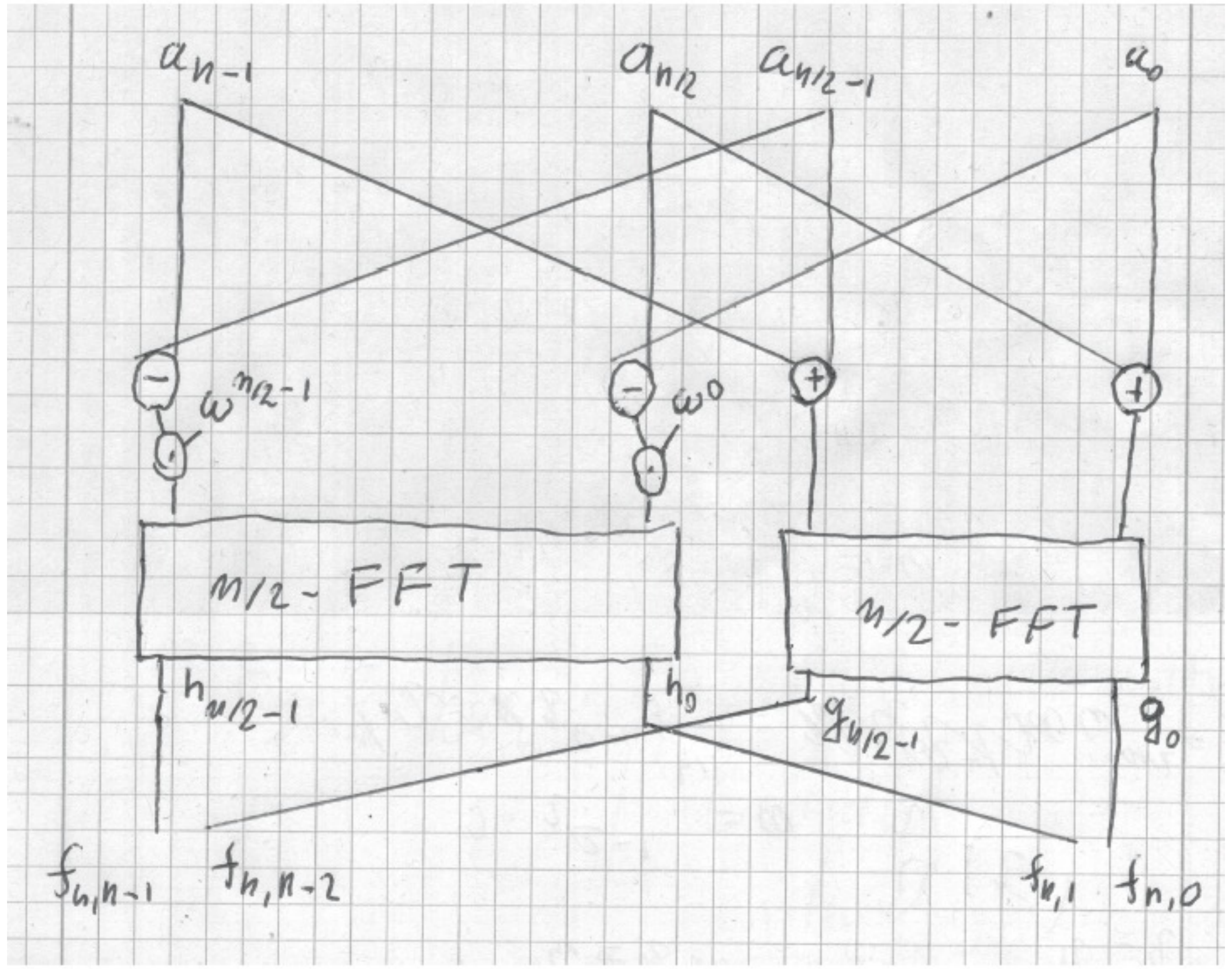


Figure 1: Recursive construction of n-FFT. Operations in 'gates' are ring operations

Odd indices $2i+1$,$i \in [0 : n/2-1]$:

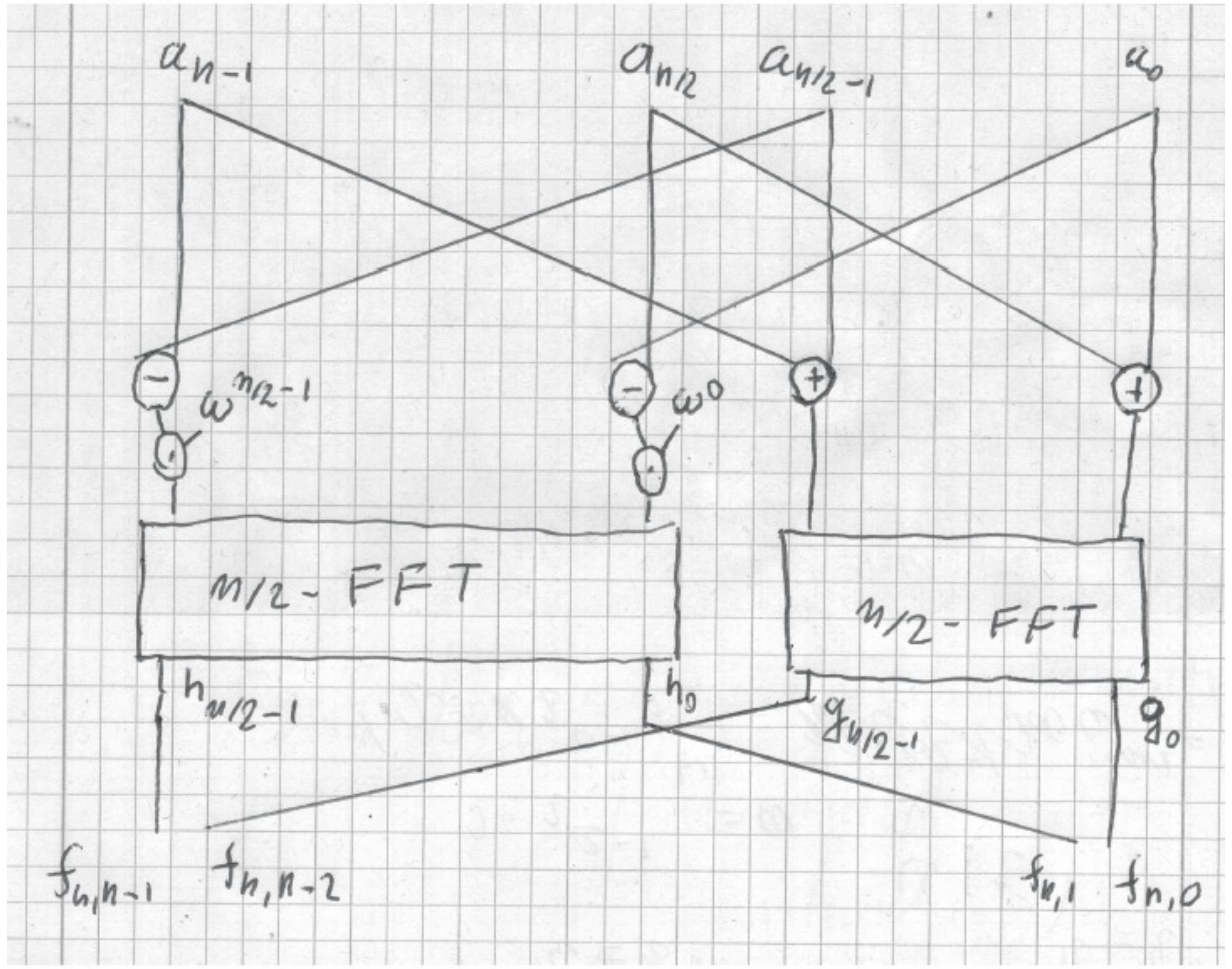
$$\begin{aligned}
h_i(a) &= f_{n,2i+1}(a) \\
&= \sum_{j=0}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{(2i+1)j} a_j + \sum_{j=n/2}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)j} \omega^{n/2} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j - \omega^{2ij} a_{n/2+j}) \omega^j \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j - a_{n/2+j}) \omega^j \\
&= f_{n/2}(c) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
c_j &= (a_j - a_{n/2+j}) \omega^j \quad j \in [0, n/2-1]
\end{aligned}$$

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

Reduce to 2 problems of half the size.

Even indices $2i$, $i \in [0 : n/2-1]$:

$$
\begin{aligned}
g_i(a) &= f_{n,2i}(a) \\
&= \sum_{j=0}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{2ij} a_j + \sum_{j=n/2}^{n-1} \omega^{2ij} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2i(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j + \omega^{2ij} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j + a_{n/2+j}) \\
&= f_{n/2}(b) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
b_j &= a_j + a_{n/2+j} \quad j \in [0, n/2-1]
\end{aligned}
$$



Odd indices $2i+1$,$i \in [0 : n/2-1]$:

$$
\begin{aligned}
h_i(a) &= f_{n,2i+1}(a) \\
&= \sum_{j=0}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} \omega^{(2i+1)j} a_j + \sum_{j=n/2}^{n-1} \omega^{(2i+1)j} a_j \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)(n/2+j)} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{(2i+1)j} a_j + \omega^{(2i+1)j} \omega^{n/2} a_{n/2+j}) \\
&= \sum_{j=0}^{n/2-1} (\omega^{2ij} a_j - \omega^{2ij} a_{n/2+j}) \omega^j \\
&= \sum_{j=0}^{n/2-1} (\omega^2)^{ij} (a_j - a_{n/2+j}) \omega^j \\
&= f_{n/2}(c) \quad (\omega^2 \text{ is } n/2\text{'th root of unity}) \\
c_j &= (a_j - a_{n/2+j}) \omega^j \quad j \in [0, n/2-1]
\end{aligned}
$$

Figure 1: Recursive construction of n-FFT. Operations in 'gates' are ring operations

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

recall:

From now on:

$$\boxed{n = 2^k \quad , \quad \omega = 2^e \quad , \quad m = \omega^{n/2} + 1 \quad , \quad k, e \in \mathbb{N}}$$

$K(n)$ = number of ring additions and subtractions

$$K(1) = 0 \quad , \quad K(n) = 2K(n/2) + n$$

$$\to K(n) = O(n \cdot \log n)$$

$O(n)$ ring multiplications with powers of two. Shifts in binary representation. But computation of results $\bmod m$ is required.

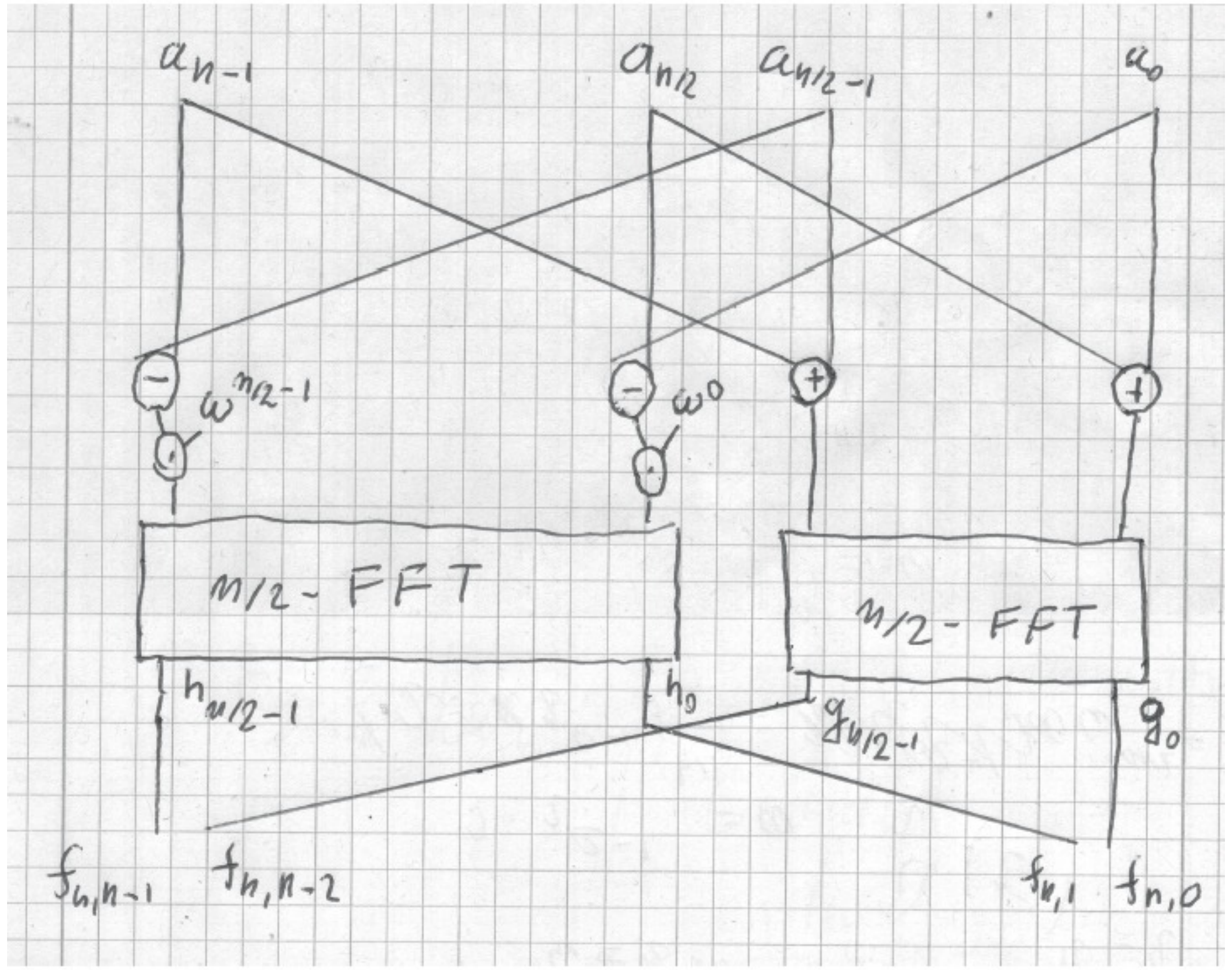


Figure 1: Recursive construction of n-FFT. Operations in 'gates' are ring operations

Fast Fourier Transform (FFT) and Fast Inverse Fourier Transform (FIFT)

recall:

From now on:

$$\boxed{n = 2^k \quad , \quad \omega = 2^e \quad , \quad m = \omega^{n/2} + 1 \quad , \quad k, e \in \mathbb{N}}$$

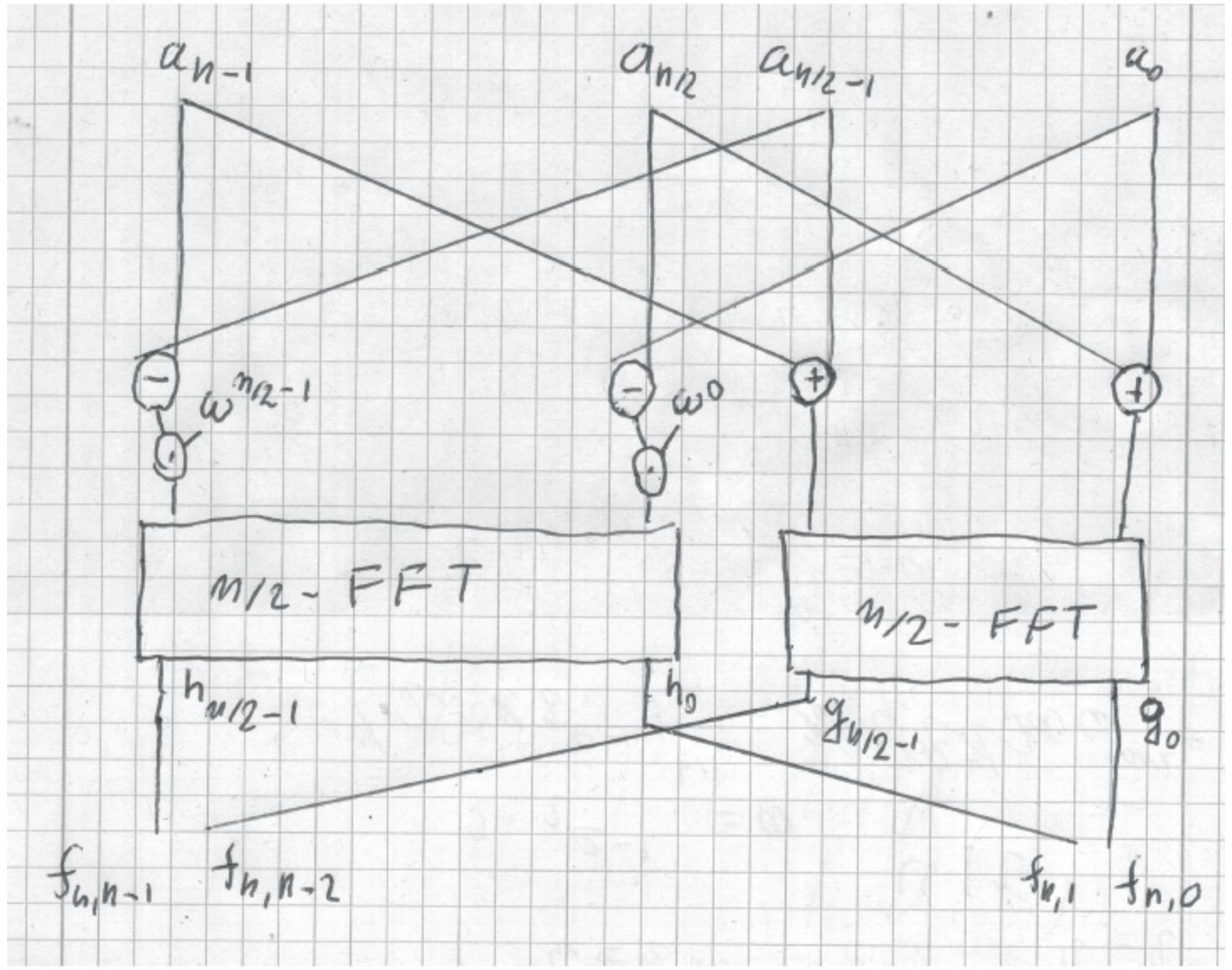


Figure 1: Recursive construction of n-FFT. Operations in 'gates' are ring operations

$K(n)$ = number of ring additions and subtractions

$$K(1) = 0 \quad , \quad K(n) = 2K(n/2) + n$$

$$\rightarrow K(n) = O(n \cdot \log n)$$

$O(n)$ ring multiplications with powers of two. Shifts in binary representation. But computation of results $\bmod m$ is required.

Fast Inverse Fourier Transform:

- ω^{-ij} instead of ω^{ij}
- multiply results with n^{-1}

$\cancel{L'(n) = 2n}$

Again multiplications only with powers of two.

Convolution Theorem

$$a = (0, \ldots, 0, a_{n-1}, \ldots, a_0) \in R^{2n}$$
$$b = (0, \ldots, 0, b_{n-1}, \ldots, b_0) \in R^{2n}$$

Define *convolution*

$$a \otimes b \in R^{2n}$$
$$(a \otimes b)_j = \sum_{k=0}^{j} a_k b_{j-k}$$

Convolution Theorem

$$a = (0, \ldots, 0, a_{n-1}, \ldots, a_0) \in R^{2n}$$
$$b = (0, \ldots, 0, b_{n-1}, \ldots, b_0) \in R^{2n}$$

Define *convolution*

$$a \otimes b \in R^{2n}$$
$$(a \otimes b)_j = \sum_{k=0}^{j} a_k b_{j-k}$$

Algorithm/Convolution Theorem

- transform operands:

$$c = f_{2n}(a) \quad , \quad d = f_{2n}(b)$$

- multiply componentwise

$$g \in R^{2n} \quad , \quad g_e = c_e d_e \quad e \in [0 : 2n-1]$$

- transforming back gives the convolution

$$a \otimes b = f_{2n}^{-1}(g)$$

Convolution Theorem

$$a = (0, \ldots, 0, a_{n-1}, \ldots, a_0) \in R^{2n}$$
$$b = (0, \ldots, 0, b_{n-1}, \ldots, b_0) \in R^{2n}$$

Define *convolution*

$$a \otimes b \in R^{2n}$$
$$(a \otimes b)_j = \sum_{k=0}^{j} a_k b_{j-k}$$

Algorithm/Convolution Theorem

- transform operands:

$$c = f_{2n}(a) \quad , \quad d = f_{2n}(b)$$

- multiply componentwise

$$g \in R^{2n} \quad , \quad g_e = c_e d_e \quad e \in [0 : 2n-1]$$

- transforming back gives the convolution

$$a \otimes b = f_{2n}^{-1}(g)$$

$$(f_{2n}(a))_e \cdot (f_{2n}(b))_e = \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} a_j b_k \omega^{e(j+k)}$$

Convolution Theorem

$$a = (0,\dots,0,a_{n-1},\dots,a_0) \in R^{2n}$$
$$b = (0,\dots,0,b_{n-1},\dots,b_0) \in R^{2n}$$

Define *convolution*

$$a \otimes b \in R^{2n}$$
$$(a \otimes b)_j = \sum_{k=0}^{j} a_k b_{j-k}$$

Algorithm/Convolution Theorem

- transform operands:

$$c = f_{2n}(a) \quad , \quad d = f_{2n}(b)$$

- multiply componentwise

$$g \in R^{2n} \quad , \quad g_e = c_e d_e \quad e \in [0 : 2n-1]$$

- transforming back gives the convolution

$$a \otimes b = f_{2n}^{-1}(g)$$

$$(f_{2n}(a))_e \cdot (f_{2n}(b))_e = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} a_j b_k \omega^{e(j+k)}$$

$$\begin{aligned}
(f_{2n}(a \otimes b))_e &= \sum_{p=0}^{2n-1} \Big(\sum_{j=0}^{p} a_j b_{p-j}\Big)\omega^{ep} \\
&= \sum_{p=0}^{2n-1}\sum_{j=0}^{2n-1} a_j b_{p-j}\omega^{ep} \quad \text{(with } b_s = 0 \text{ for } s < 0) \\
&= \sum_{j=0}^{2n-1}\sum_{p=0}^{2n-1} a_j b_{p-j}\omega^{ep} \quad \text{now transform } k = p - j \\
&= \sum_{j=0}^{2n-1}\sum_{k=-j}^{2n-j-1} a_j b_k \omega^{e(j+k)} \\
&= \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} a_j b_k \omega^{e(j+k)}
\end{aligned}$$