Fast Fourier Transform (FFT)

Algorithms and Applications

Michael Balas

What is it?

- Efficient implementation of the Discrete Fourier Transform (DFT) or its inverse
- Many types: Split radix, prime factor, Winograd Fourier transform, vector radix
- Focus will be on the Radix-2 Decimation in time FFT Algorithm

What's the Big Deal?

Included in the Top 10 Algorithms of the 20th Century by the IEEE journal: Computing in Science & Engineering

- EKG and EEG signal processing
- Forensic Science
- Image quality measures
- Interpolation and decimation
- Magnetic Resonance Imaging (MRI)
- Noise filtering
- Optical signal processing
- Pattern Recognition
- Video/image compression
- Numerical solution of differential equations

The Discrete Fourier Transform

Discrete Fourier Transform

$$X^F(k) = \sum_{n=0}^{N-1} x(n) e^{\left(\frac{-j2\pi}{N}\right)kn}, \;\; k = 0, 1, ..., N-1$$

(Forward) Discrete Fourier Transform (DFT)

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} X^F(k) e^{\left(\frac{j2\pi}{N}\right)kn}, \;\; n = 0, 1, ..., N-1$$

Inverse Discrete Fourier Transform (IDFT)

Discrete Fourier Transform

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

$$W_N^{kn} = exp\left[\left(\frac{-j2\pi}{N}\right)kn\right]$$

$O(N^2)$ *Computations for direct DFT*

$$\left.\begin{array}{r} N \text{ complex multiplications} \\ N\text{-}1 \text{ complex additions} \end{array}\right\} \text{For each } k$$

$O(N \, log_2 \, N)$ *Computations using FFT*

Time Comparison

N	1000	10^6	10^9
N^2	10^6	10^{12}	10^{18}
$N \log_2 N$	10^4	20×10^6	30×10^9

Assuming 1 nanosecond per operation...

10^{18} nanoseconds ≅ 31.2 years

30×10^9 nanoseconds ≅ 30 seconds

The Fast and the Fourier

Image from: https://pbs.twimg.com/media/Cu4uQPZWgAAJ38H.jpg

Decimation in Time (DIT)

Decomposes an N-point sequence into two N/2 sequences (one of even and one of odd samples)

Assume N = 2^l, l = integer

Obtain the N-point DFT in terms of the DFTs of these two sequences

$$X_k^F$$

$$\{x_0, x_1, x_2, \ldots, x_{N-2}, x_{N-1}\}$$

$$\{x_0, x_2, x_4, \ldots, x_{N-2}\}$$

$$G^F(k)$$

Even Samples

$$\{x_1, x_3, x_5, \ldots, x_{N-1}\}$$

$$H^F(k)$$

Odd Samples

Math Magic

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

$$= \sum_{\substack{n \text{ even} \\ \text{integer}}} x(n) W_N^{kn} + \sum_{\substack{n \text{ odd} \\ \text{integer}}} x(n) W_N^{kn}$$

$$\left.\begin{array}{r} \textit{Even Indices} \\ \textit{Odd Indices} \end{array}\right\} \begin{array}{l} n = 2r \\ n = 2r + 1 \end{array}$$

$r = 0, 1, \ldots, N/2 - 1$

$$= \sum_{r=0}^{(N/2)-1} x(2r) W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{(2r+1)k}$$

Math Magic

$$= \sum_{r=0}^{(N/2)-1} x(2r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1)(W_N^2)^{rk}$$

Note that

$$W_N^2 = [\frac{-j2(2\pi)}{N}] = [\frac{-j2\pi)}{N/2}] = W_{N/2}$$

$$X^F(k) = \sum_{r=0}^{(N/2)-1} x(2r)W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1)W_{N/2}^{rk}$$

$$= G^F(k) + W_N^k H^F(k),\ k = 0, 1, \ldots, \frac{N}{2} - 1$$

Data-Flow Diagrams

Image from:
https://zibbet.s3.amazonaws.com/uploads/photo/file/7833065/il_fullxfull.130323217.jpg

Butterfly Computations

N = 8

$$X^F(k) = G^F(k) + W_N^k H^F(k),\ \ k = 0, 1, ..., \frac{N}{2} - 1$$

$X^F[0]$

$X^F[2]$

$X^F[4]$

$X^F[6]$

$X^F[1]$

$X^F[3]$

$X^F[5]$

$X^F[7]$

Butterfly Computations

N = 8

$$X^F(k) = G^F(k) + W_N^k H^F(k),\ k = 0, 1, ..., \frac{N}{2} - 1$$

$X^F[0] \longrightarrow$

$X^F[2] \longrightarrow$

$X^F[4] \longrightarrow$

$X^F[6] \longrightarrow$

$X^F[1] \longrightarrow$

$X^F[3] \longrightarrow$

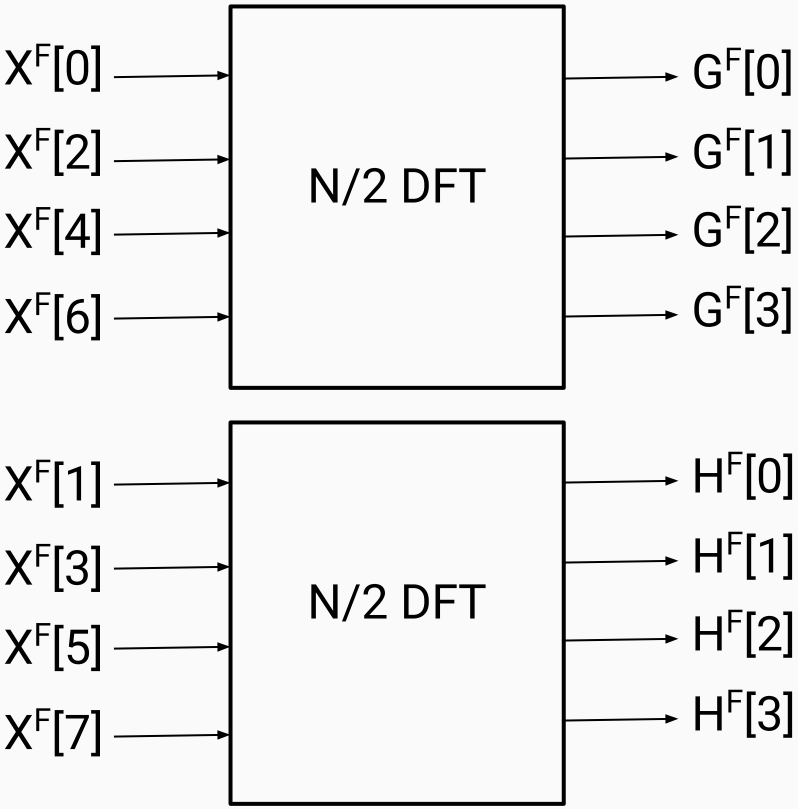
$X^F[5] \longrightarrow$

$X^F[7] \longrightarrow$

Butterfly Computations

N = 8

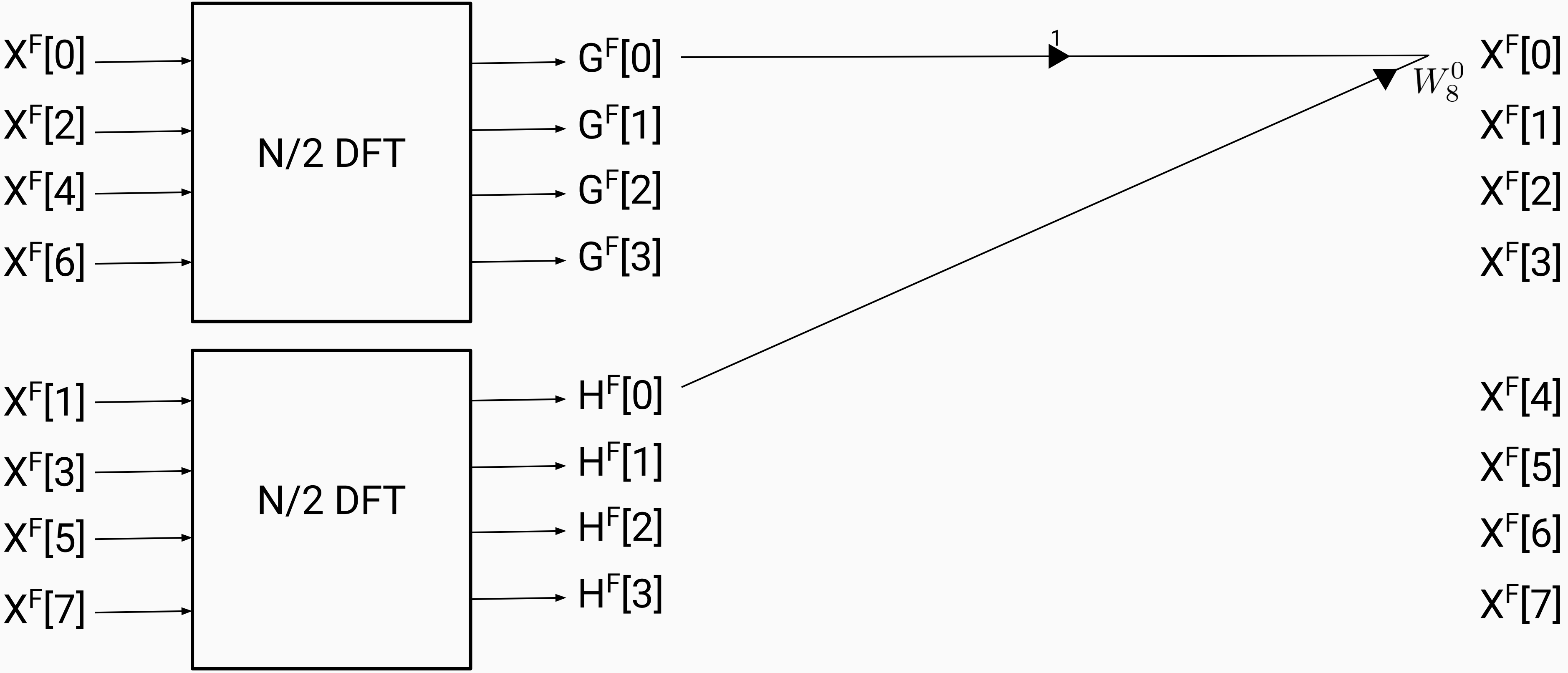
$$X^F(k) = G^F(k) + W_N^k H^F(k),\ k = 0, 1, ..., \frac{N}{2} - 1$$



Butterfly Computations

N = 8

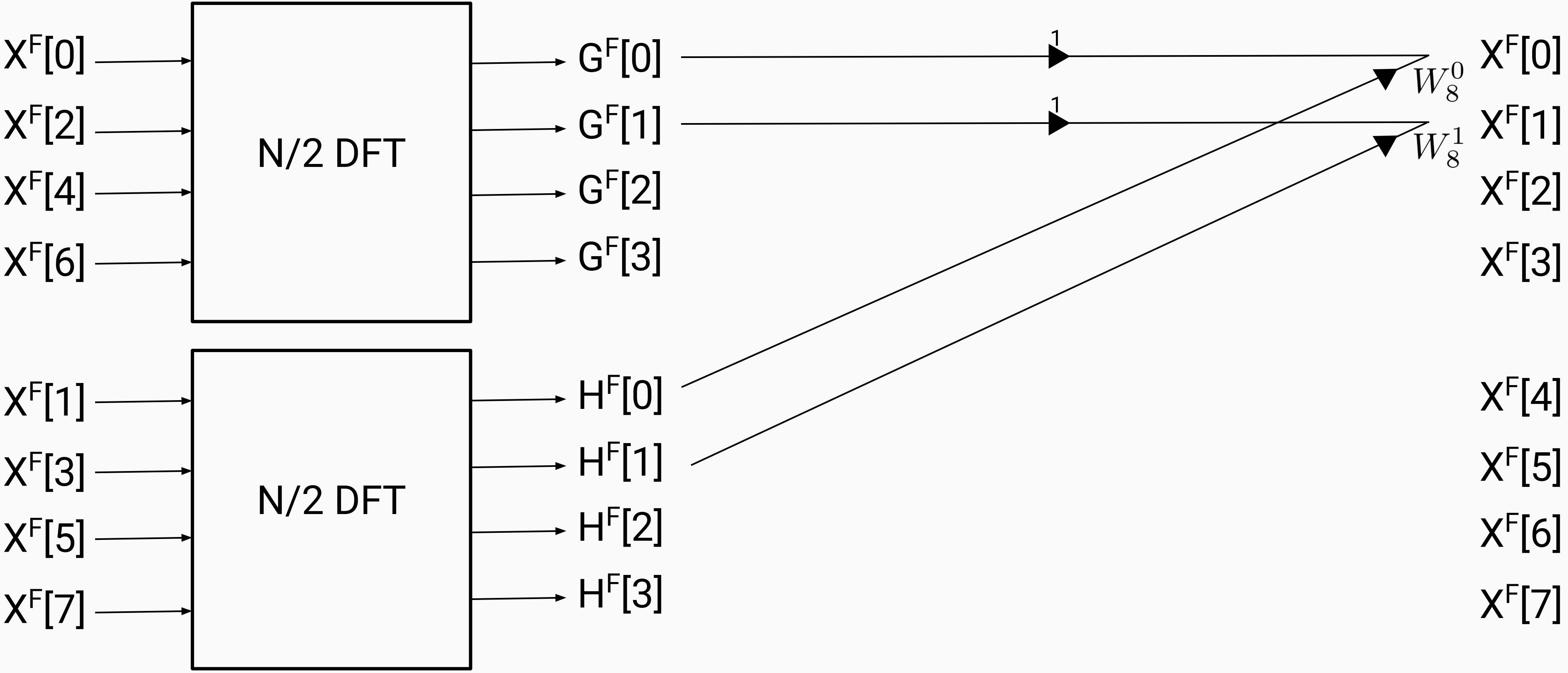
$$X^F(k) = G^F(k) + W_N^k H^F(k),\ k = 0, 1, ..., \frac{N}{2} - 1$$



Butterfly Computations

N = 8

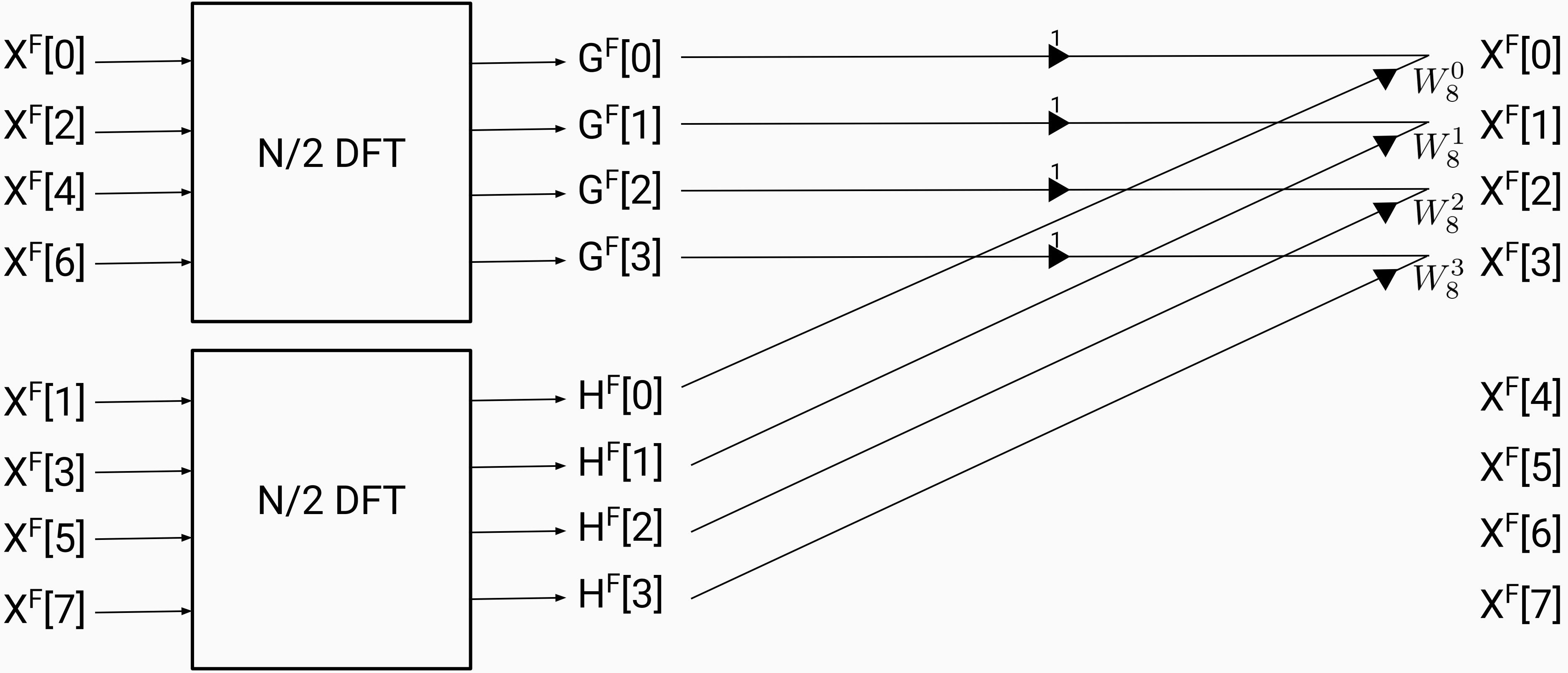
$$X^F(k) = G^F(k) + W_N^k H^F(k),\ \ k = 0, 1, ..., \frac{N}{2} - 1$$



Butterfly Computations

N = 8

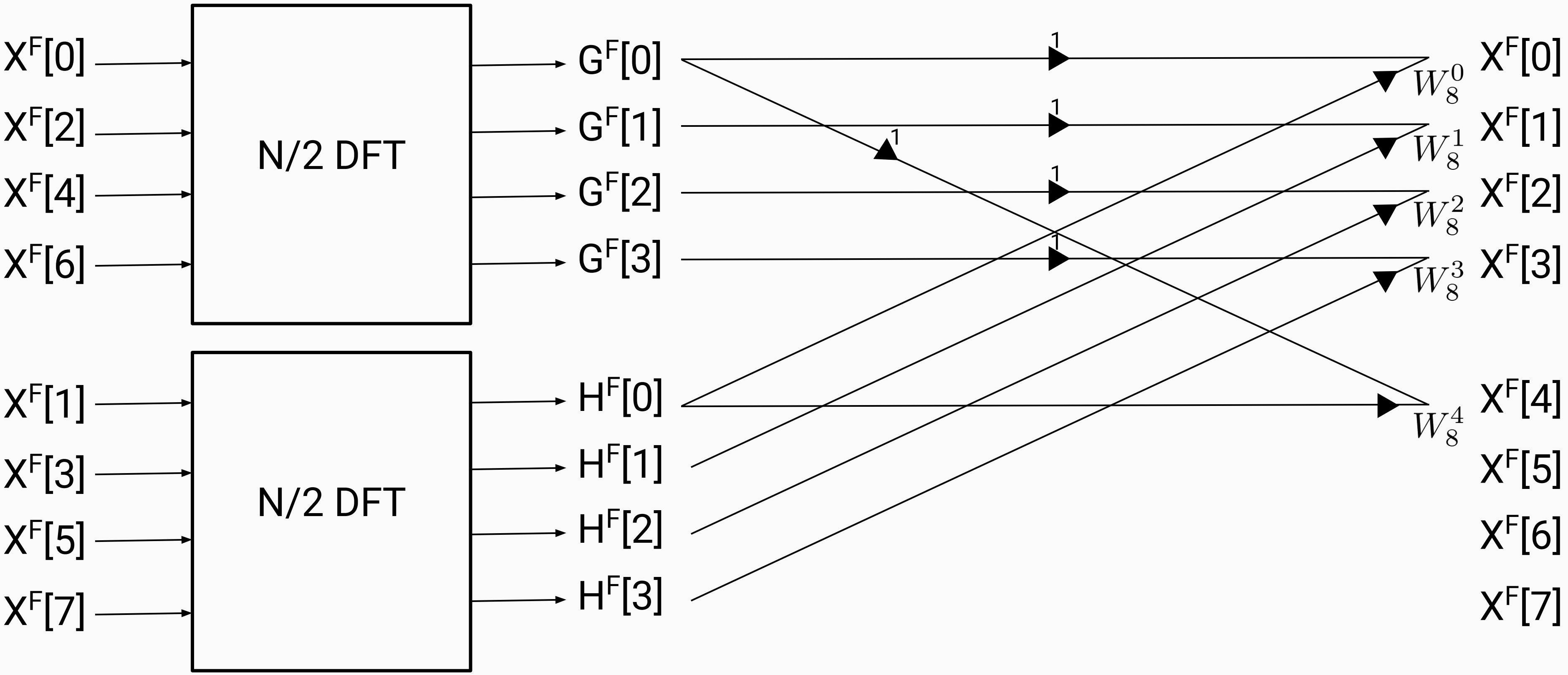
$$X^F(k) = G^F(k) + W_N^k H^F(k),\ \ k = 0, 1, ..., \frac{N}{2} - 1$$



Butterfly Computations

N = 8

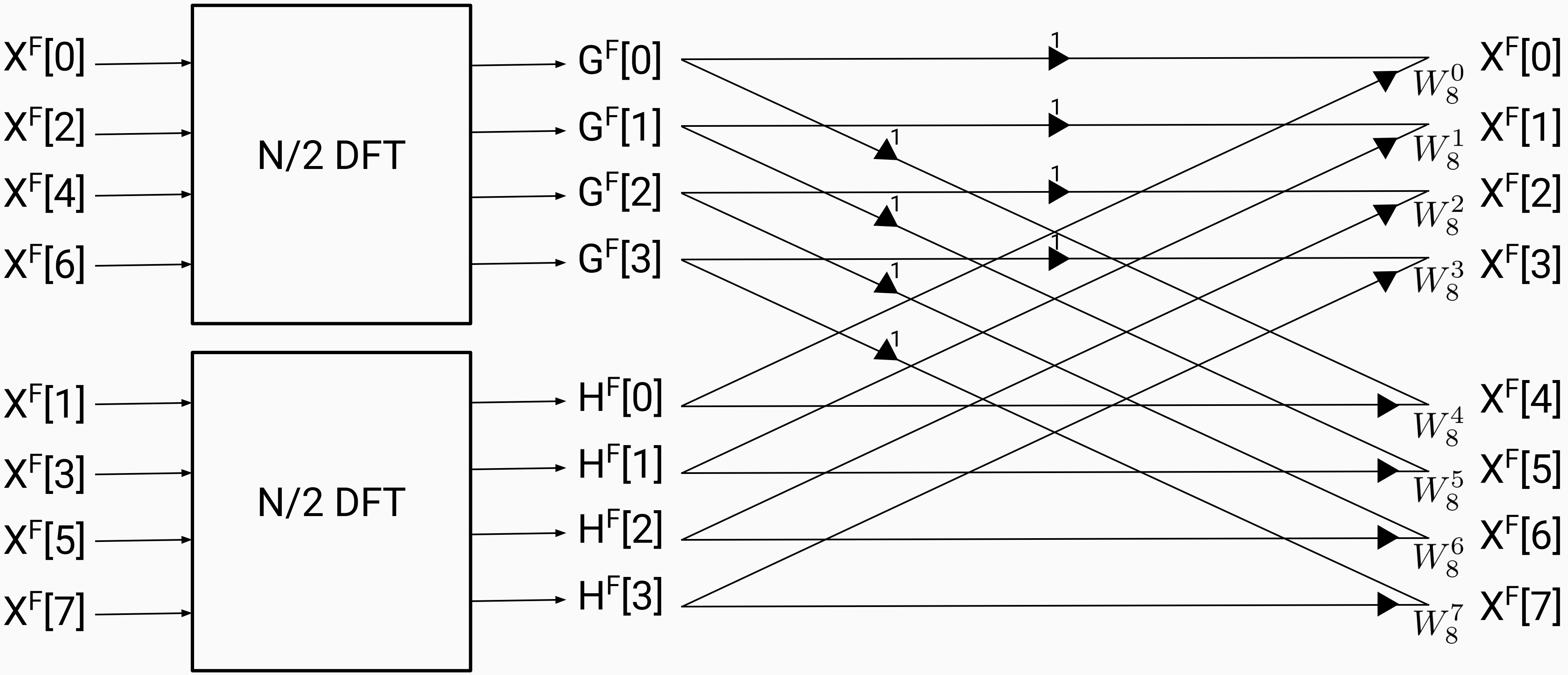
$$X^F(k) = G^F(k) + W_N^k H^F(k),\ \ k = 0, 1, \dots, \frac{N}{2} - 1$$



Butterfly Computations

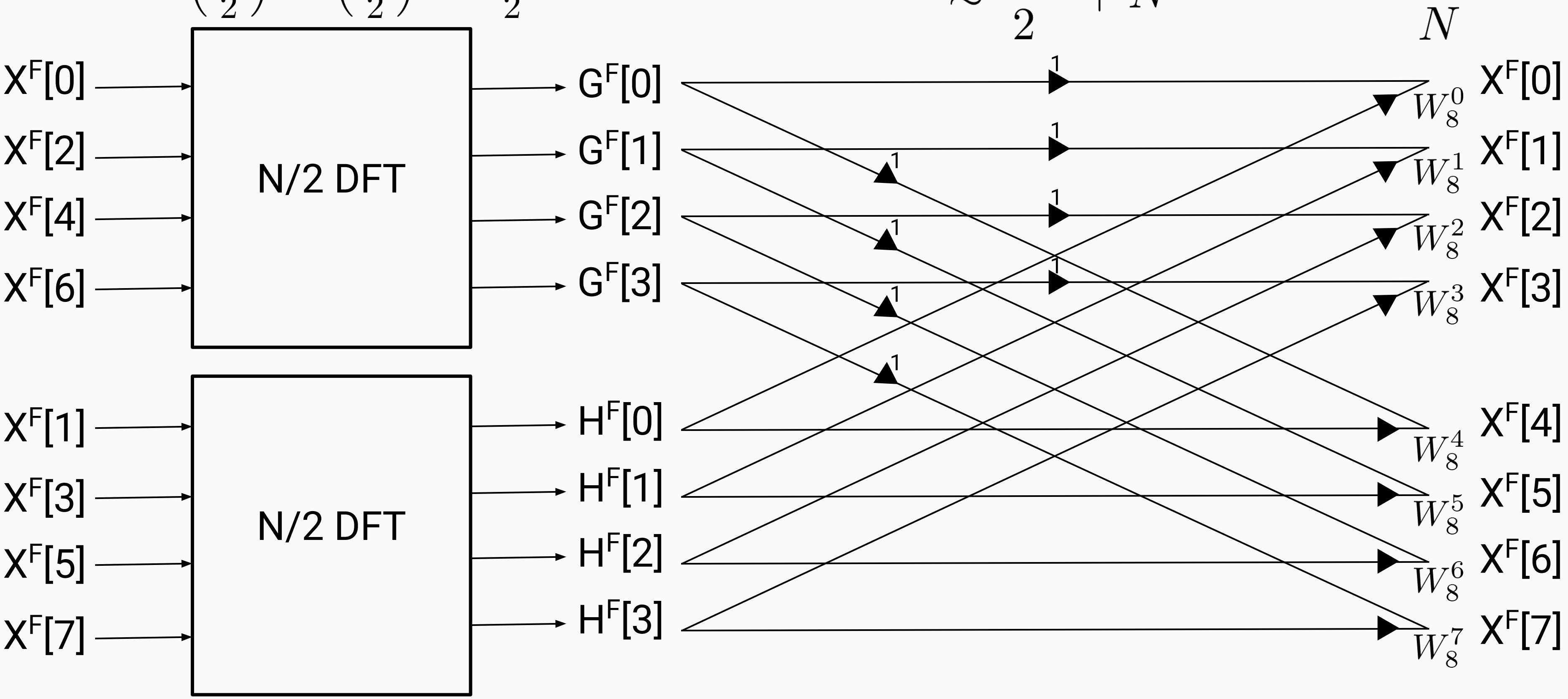
N = 8

$$X^F(k) = G^F(k) + W_N^k H^F(k),\ \ k = 0, 1, \ldots, \frac{N}{2} - 1$$



Butterfly Computations

N = 8 $\left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 = \frac{N^2}{2}$ $\approx \frac{N}{2}^2 + N$



Why Stop Here?

Let's keep splitting: $each\ \frac{N}{2}pt \rightarrow 2\ \frac{N}{4}pt\ DFTs$

How many times? $\frac{N}{2}, \frac{N}{4}, \ldots, \frac{N}{2^{v-1}}, \frac{N}{2^v} = 1, \quad v = log_2 N$

Cost?

$$1: \frac{N}{2} \to 2\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N$$

$$2: \frac{N}{4} \to 2\left(2\left(\frac{N}{4}\right)^2 + \frac{N}{2}\right) + N = \frac{N^2}{4} + 2N$$

$$3: \frac{N}{8} \to 2\left[2\left(2\left(\frac{N}{8}\right)^2 + \frac{N}{4}\right) + \frac{N}{2}\right] + N = \frac{N^2}{8} + 3N$$

$$\vdots$$

Cost?

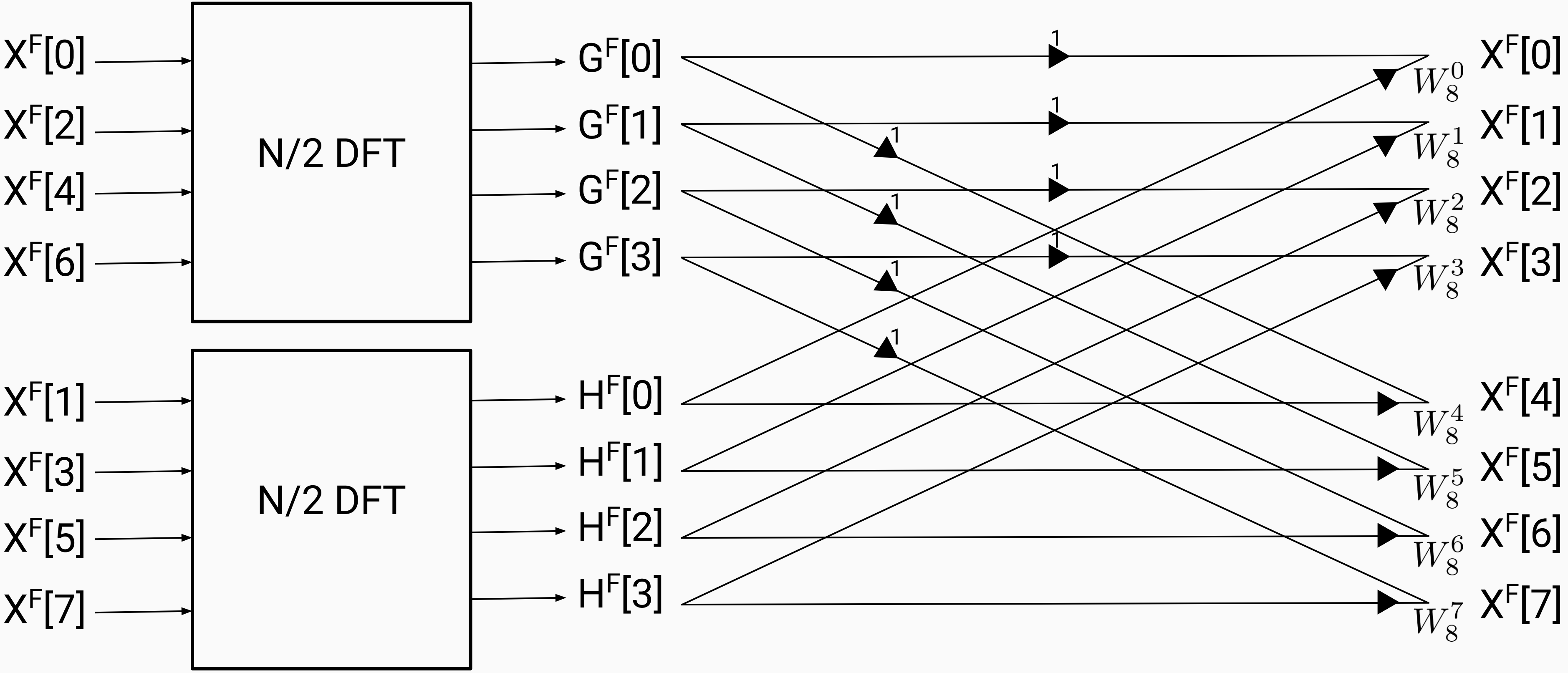
$$v : \frac{N}{2^v} = 1 \rightarrow \frac{N^2}{2^v} + vN = \frac{N^2}{N} + Nlog_2N$$

$$\approx O(Nlog_2N) \ for \ N \ large$$

Butterfly Computations

N = 8

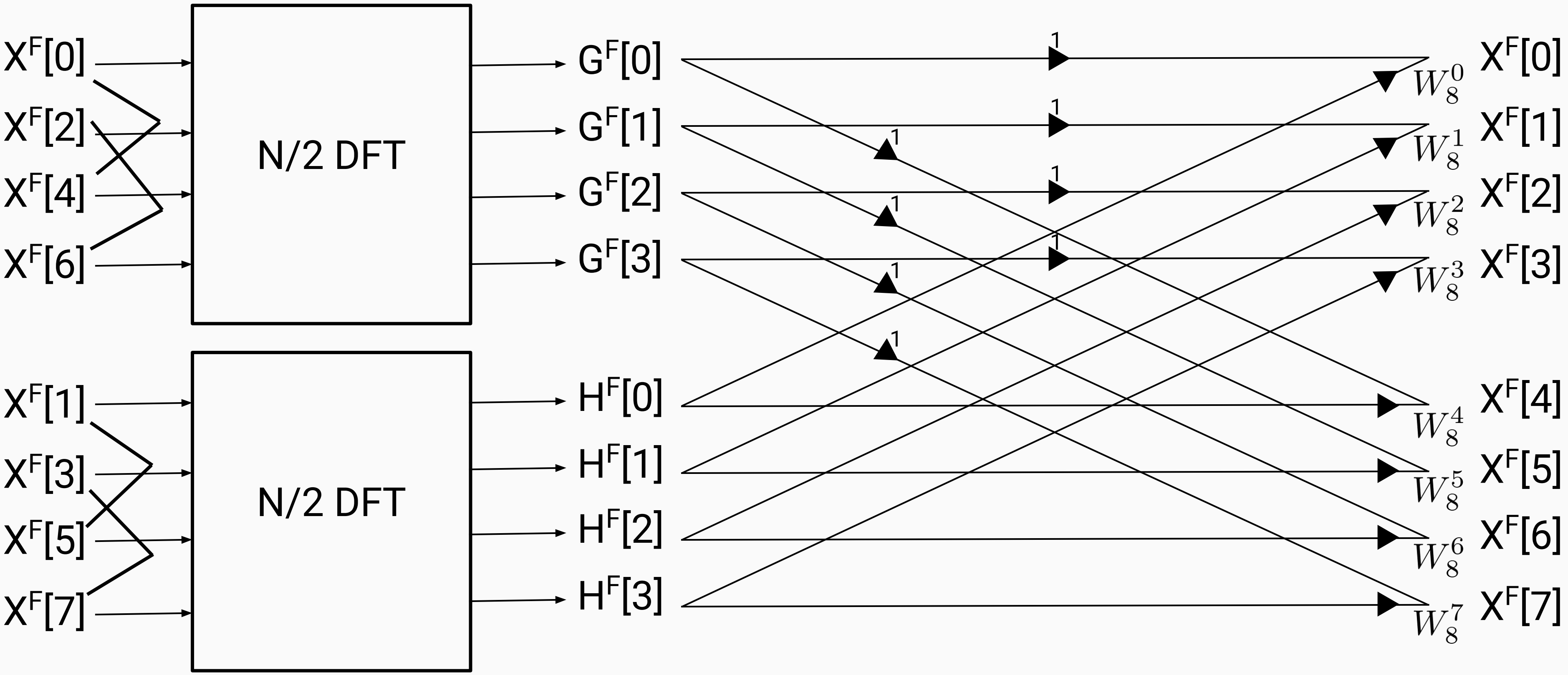
$$X^F(k) = G^F(k) + W_N^k H^F(k), \; k = 0, 1, ..., \frac{N}{2} - 1$$



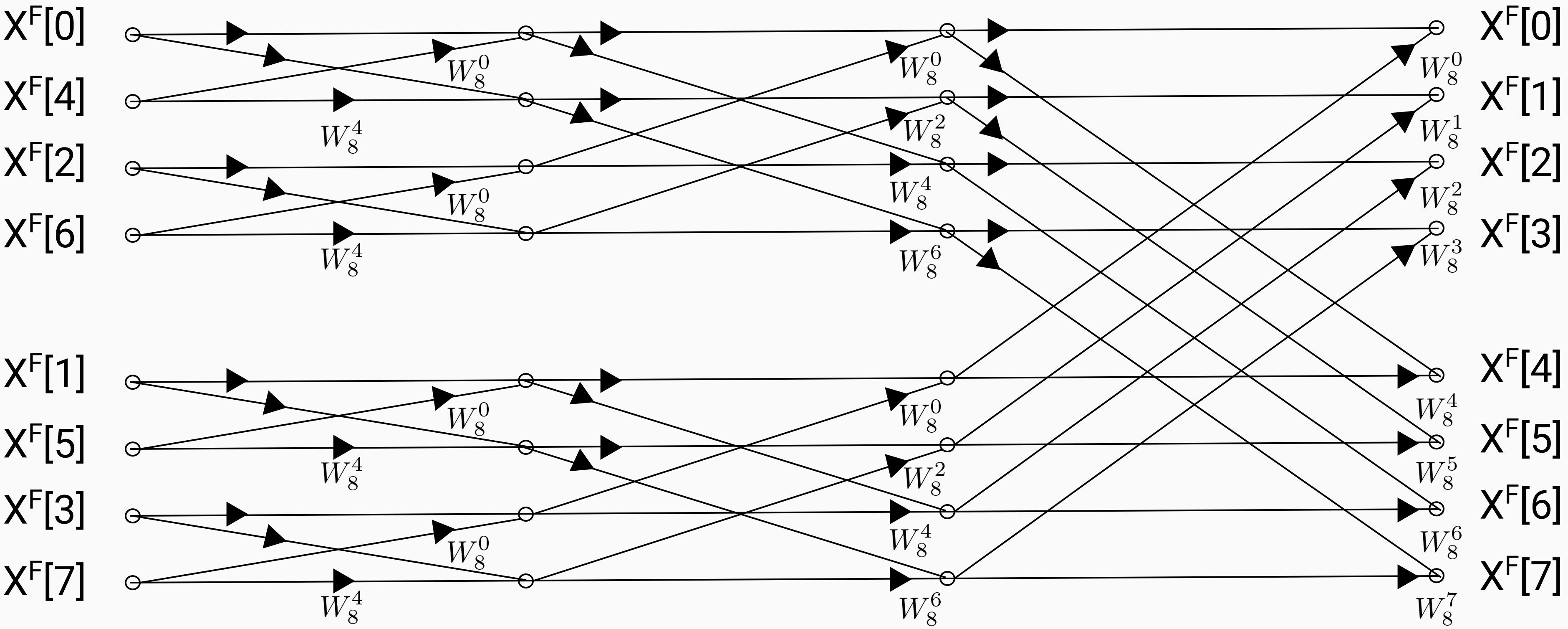
Butterfly Computations

N = 8

$$X^F(k) = G^F(k) + W_N^k H^F(k),\ k = 0, 1, ..., \frac{N}{2} - 1$$



Butterfly Computations



Thank You

References

Jörg Arndt. Matters Computational: ideas, algorithms, source code. Springer Science & Business Media, 2010.

Eleanor Chu and Alan George. Inside the FFT black box: serial and parallel fast Fourier transform algorithms. CRC Press, 1999.

James W Cooley, Peter AW Lewis, and Peter D Welch. The fast fourier transform and its applications. IEEE Transactions on Education, 12(1):27–34, 1969.

Jack Dongarra and Francis Sullivan. Guest editors' introduction: The top 10 algorithms. Computing in Science & Engineering, 2(1):22–23, 2000.

Madhavi S Kapale and Nilesh P Bodne. Design & simulation parallel pipelined radix-2ˆ 2 fft architecture for real valued signals.

References

Douglas A Lyon. The discrete fourier transform, part 2: Radix 2 fft. journal of object technology, 8(5), 2009.

Brad Osgood. The fourier transform and its applications. Lecture Notes for EE, 261:20, 2009.

Kamisetty Ramamohan Rao, Do Nyeon Kim, and Jae Jeong Hwang. Fast Fourier transform - algorithms and applications. Springer Science & Business Media, 2011.

Ali Saidi. Decimation-in-time-frequency fft algorithm. In Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on, volume 3, pages III–453. IEEE, 1994.