POLYNOMIALS AND FFT

- **Evaluation & Multiplication of Polynomials I**
- **Multiplication of Large Integers**
- **Strassen's algorithm for large Matrices**
- **Zeros of Polynomials: Bisection vs Newton's method**
- **Evaluation & Multiplication of Polynomials II**
- **Fast Fourier Transforms**
- **Arithmetic, (Pseudo) Random Numbers and all that**

ADDING, MULTIPLYING AND EVALUATING POLYNOMIALS

- **Why are polynomials so interesting?**

$$P_N(x) = a_0 + a_1\, x + a_2\, x^2 \boxed{?}\, a_N\, x^N$$

- **Fit data $d_i = P_N(x_i)$ by picking a_i's**
- **Integer arithmetic base B is polynomial with x = B:**

e.g Integer decimal base (B=10)

$$P_{N-1}\,(10) = a_0 + a_1\; 10 + a_2\; 10^2 \boxed{?}\, a_{N-1}\; 10^{N-1}$$

- **Fourier transform is $P_{N-1}(x_k)$ with $x_k=\omega^k{}_N=$ exp[i k 2 π/N]**

$$y_k\ ´\ \sum_l\ e^{ik\, n\, 2\,\pi/N}\, a_n$$

- **ETC**

EVALUATION OF $P(X) = A_0 + A_1 X + \ldots + A_N X^N$

- **Method 1:**
  - **Compute $x, x^2, x^3, \ldots x^N$ with N-1-mults plus N mults and N adds**

  or 2N-1 mults and N adds

- **Horner's Method 2:**
  - **$a_0 + x(a_1 + x (a_2 + x (a_3 + .a_4\ x)))$ for $N = 4$**

    N adds and N multiplies!
  - **Evaluating it at N points is $O(N^2)$**

- **Special case: $p(x) = x^N$ do in Log(N) steps!**
  - **Consider N in binary base eg.100111001101.**

  Cal $x, x^2, x^4, x^8 \ldots$. by repeated squaring and then multiply

  to get x^N in $O(\mathrm{Log}_2(N))$ steps.

POLYNOMIAL MULTIPLICATION (COEF REP)

- $P(x) = a_0 + a_1\, x + \ldots + a_N\, x^N$ and $Q(x) = b_0 + b_1\, x + \ldots + b_N\, x^N$
- $P(x)\, Q(x) = a_0\, b_0 + (a_0\, b_1 + a_1\, b_0)x + (a_0\, b_2 + a_1\, b_1 + a_2\, b_0)x^2$

Try divide and conquer: $P_N(x) = p_L(x) + x^N/2\; p'_H(x)$
$p_L(x) = a_0 + a_1\, x + \ldots\, a_{N/2}\, x^{N/2}$ $\quad p_H(x) = a_{N/2} + \ldots\, a_{N/2}\, x^{N/2}$

$P(x)\, Q(x) = p_L(x)\, q_L(x) + (p_L(x)\, q_H(x) + p_H(x)\, q_L(x))\, x^{N/2} + p_H(x)\, q_H(x)\, x^N$

$$T(N) = 4\, T(N/2) + c_0\, N \rightarrow T(N) = O(N^2)$$

Better: define $(p_L(x) + p_H(x))(q_L(x) + q_H(x)) - p_L(x)\, q_L(x) - p_H(x)\, p_H(x)$
$T(N) = 3\, T(N/2) + c_0\, N \rightarrow T(N) = O(N^\gamma)$
$\Rightarrow \gamma = \log[3]/\log[2] \simeq 1.58496$

FFT

- ***Karl Friedrich Gauss (1777-1855)***
- ***J. W. Cooley and J. W. Tukey, 1965***
- ***Evaluation & Multiplication of Polynomials I***
- ***Evaluation & Multiplication of Polynomials II***
- ***Fast Fourier Transforms***
- ***Interpolation/splines/FEM etc***

INTERPOLATION (POINT REP):

For $P(x) = \sum_n x^n a_n$ *pick values* $p_k = P(x_k)$

- **$P(x) = \sum_k p_k L_k(x)$ where (Lagrange says?) O(N^2)**

$$L_k(x) = \prod_{k \neq i} \frac{x - x_i}{x_k - x_i}$$

- **If $x_k = e^{2\pi\ k\, i\,/N}$ then this is called a Fourier series!**

$p_k = \sum_n e^{2\pi\, k\, n\, i/N}\, a_n$ and

$a_n = \sum (1/2\ \pi)\ \exp\{-\pi\ (n+1)\ k\ i/N\}\ p_k$

MULTIPLYING POLYNOMIALS AND

- **Multiply $P_N(x)$, $Q_N(x)$**
  - **Evaluate at 2N points (k=0,? 2N-1)** **O(N)**
  - **$W(x_k) = Q(x_k)\ P(x_k)$** **O(N)**

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
a_i,\ b_j \arrow[r, "O(N^2)", "\textit{convolution}"'] \arrow[d, "O(N\log(N))\ \text{FFT}"'] & c\_k \\
Q(x\_k)\ \ P(x\_k) \arrow[r, "\textit{multiply}"', "O(N)"] & W(x_k) = Q(x_k)\ P(x_k) \arrow[u, "O(N\log(N))\ \text{FFT}"']
\end{tikzcd}
$$

FTT: $x_k = \omega_N^k = e^{2\pi i k/N}$

$$y_k \equiv \mathcal{FT}_N[a_n] = \sum_{n=0}^{N-1} (\omega_N^k)^n a_n = \sum_{n=0}^{N-1} e^{i2\pi nk/N} a_n$$

The trick: $n = n_0 + 2\, n_1 + 2^2\, n_2 + \ldots + 2^p\, n_p$

$$\omega_N^n = \omega_N^{n_0}\, \omega_{N/2}^{n_1}\, \omega_{N/4}^{n_2} \cdots \omega_2^{n_p}$$

$$\sum_n \omega_N^{nk} = \sum_{n_0=0,1} \omega_N^{n_0 k} \sum_{n_1=0,1} \omega_{N/2}^{n_1 k} \cdots \sum_{n_p=0,1} \omega_2^{n_p k}$$

HIGH BIT FIRST

$$y_k \equiv \mathcal{FT}_N[a_n] = \sum_{n=0}^{N-1} (\omega_N^k)^n a_n = \sum_{n=0}^{N-1} e^{i2\pi nk/N} a_n$$

spit polynomial into low/high pieces:

$$y_k = \sum_{n=0}^{N/2-1} e^{i2\pi nk/N} a_n + e^{i\pi k} \sum_{n=0}^{N/2-1} e^{i2\pi nk/N} a_{n+N/2}$$

➔ One N = Two N/2 Fourier transforms

even k $\quad y_{2\tilde{k}} = \sum_{n=0}^{N/2-1} e^{i2\pi n\tilde{k}/(N/2)} [a_n + a_{n+N/2}]$

odd k $\quad y_{2\tilde{k}+1} = \sum_{n=0}^{N/2-1} e^{i2\pi n\tilde{k}/(N/2)} \omega_N^n [a_n - a_{n+N/2}]$

LOW BIT FIRST

$$y_k \equiv \mathcal{FT}_N[a_n] = \sum_{n=0}^{N-1} (\omega_N^k)^n a_n = \sum_{n=0}^{N-1} e^{i2\pi nk/N} a_n$$

spit polynomial into even/odd pieces:

$$y_k = \sum_{n=0}^{N/2-1} e^{i2\pi nk/(N/2)} a_{2n} + \omega_N^k \sum_{n=0}^{N/2-1} e^{i2\pi nk/(N/2)} a_{2n+1}$$

➔ One N = Two N/2 Fourier transforms

low k

$$y_k = \sum_{n=0}^{N/2-1} e^{i2\pi nk/(N/2)} [a_n + \omega_N^k a_{n+N/2}]$$

high k

$$y_{k+N/2} = \sum_{n=0}^{N/2-1} e^{i2\pi nk/(N/2)} [a_n - \omega_N^k a_{n+N/2}]$$

THUS $T(N) = 2\,T(N/2) + C_O\,N$: $T(N)$ » N

$$y_{2k} = \mathcal{FT}_{N/2}[\, a_n + a_{n+N/2}\,]$$

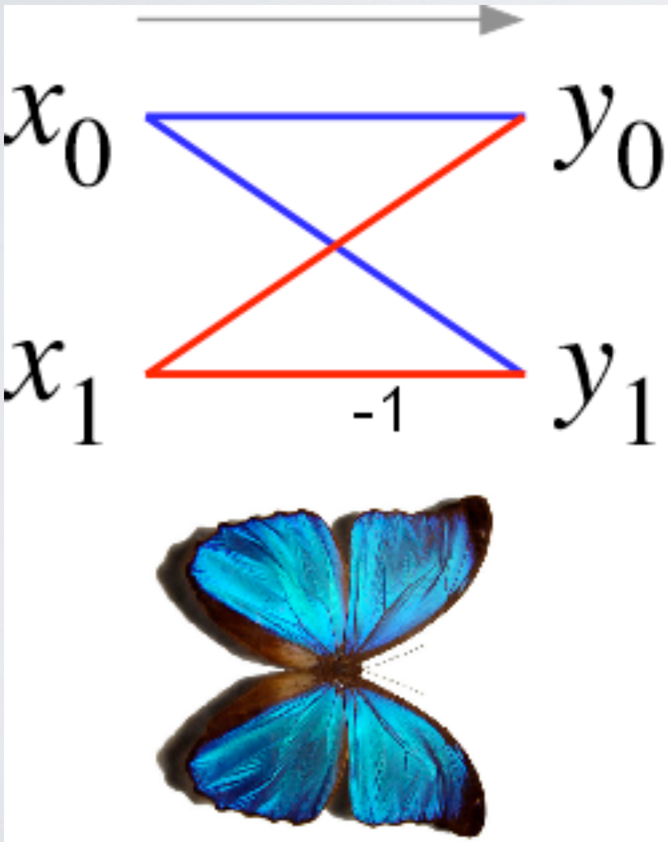
$$y_{2k+1} = \mathcal{FT}_{N/2}[\, \omega_N^n (a_n - a_{n+N/2})\,]$$

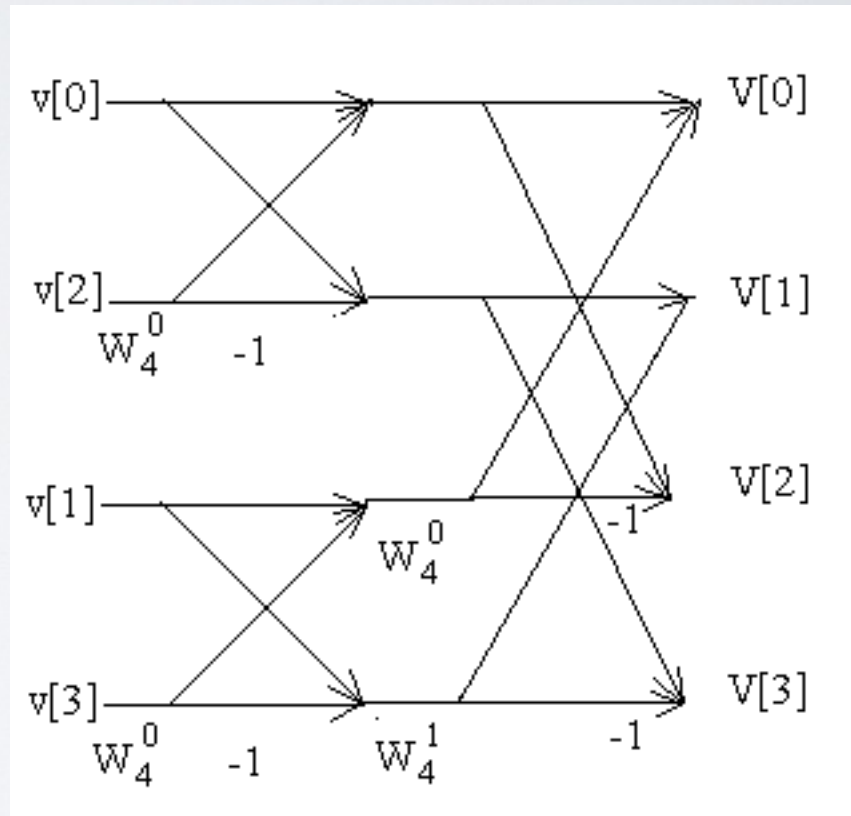
OR

$$y_k = \mathcal{FT}_{N/2}[a_{2n} + \omega_N^k a_{2n+1}]$$

$$y_{k+N/2} = \mathcal{FT}_{N/2}[a_{2n} - \omega_N^k a_{2n+1}]$$

BUTTERFLY

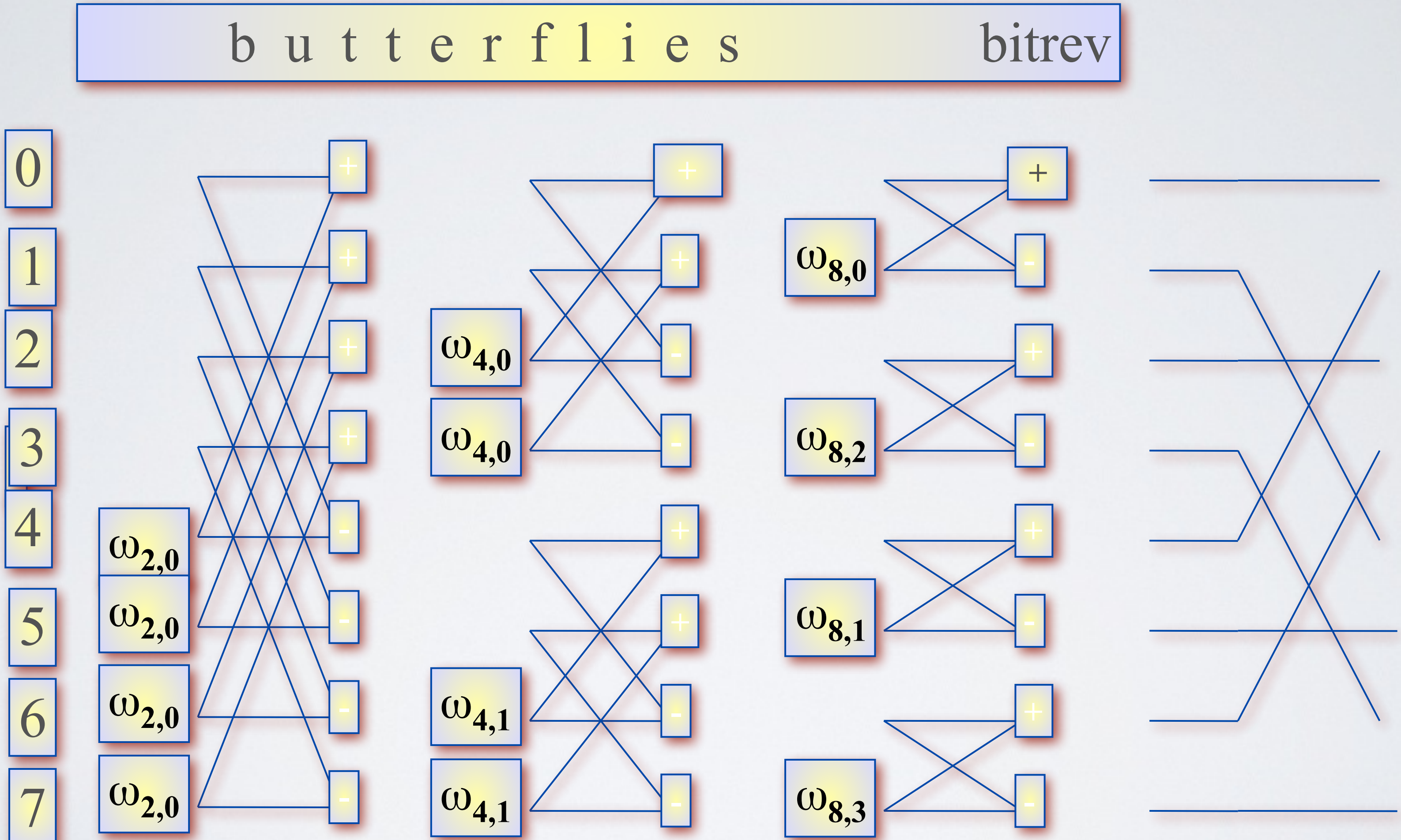


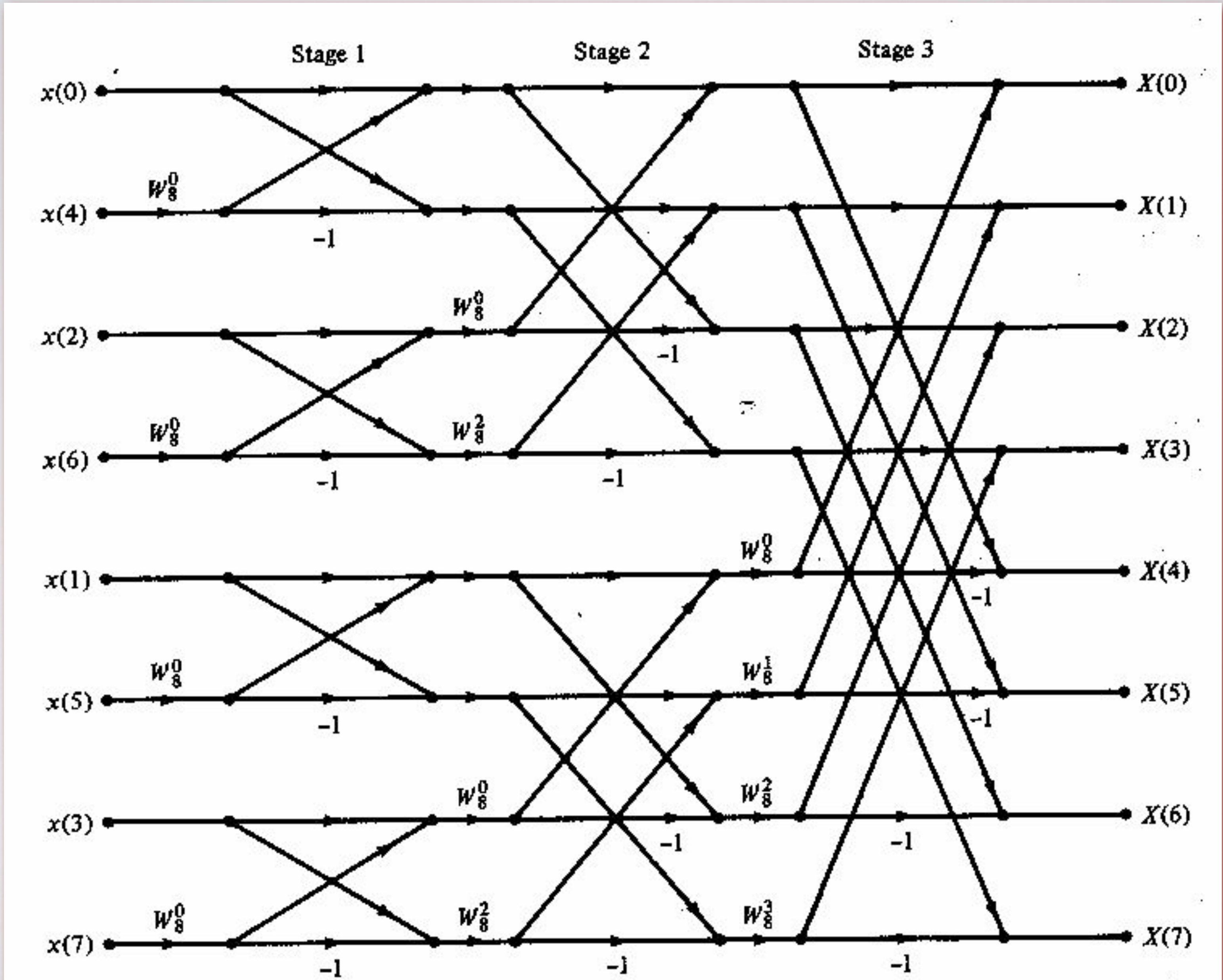


BUTTERFLIES NETWORK

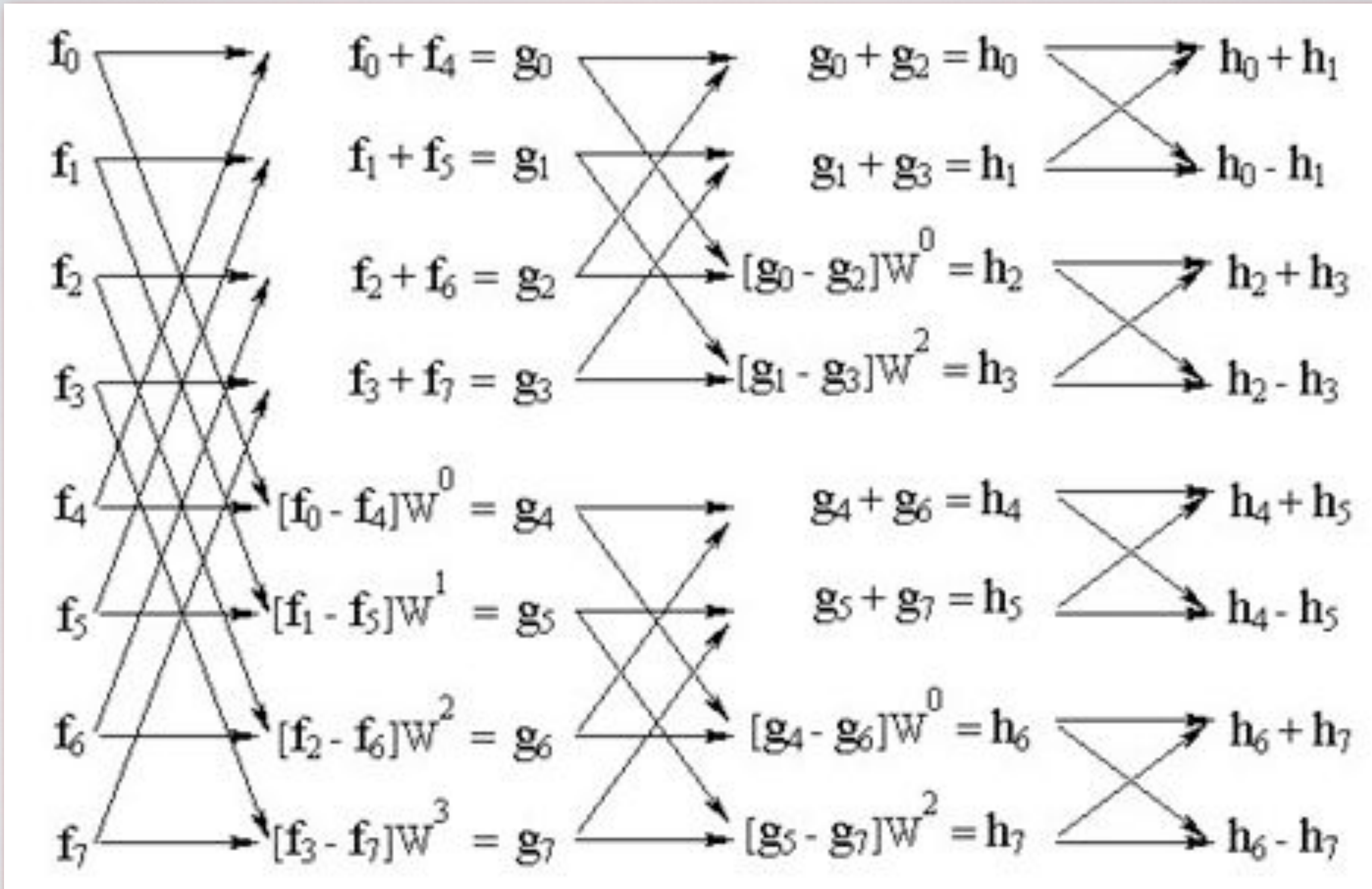
$$y_k = \mathcal{FT}_{N/2}[a_{2n} + \omega_N^k a_{2n+1}]$$

$$y_{k+N/2} = \mathcal{FT}_{N/2}[a_{2n} - \omega_N^k a_{2n+1}]$$





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_8^1
W_8^2
W_8^3
–1


f0
f1
f2
f3
f4
f5
f6
f7
f0 + f4 = g0
f1 + f5 = g1
f2 + f6 = g2
f3 + f7 = g3
[f0 - f4]W0 = g4
[f1 - f5]W1 = g5
[f2 - f6]W2 = g6
[f3 - f7]W3 = g7
g0 + g2 = h0
g1 + g3 = h1
[g0 - g2]W0 = h2
[g1 - g3]W2 = h3
g4 + g6 = h4
g5 + g7 = h5
[g4 - g6]W0 = h6
[g5 - g7]W2 = h7
h0 + h1
h0 - h1
h2 + h3
h2 - h3
h4 + h5
h4 - h5
h6 + h7
h6 - h7

PSEUDORANDOM NUMBERS
LINEAR CONGRUENTIAL GENERATOR (LCG)

- **x[n+1] = (a x[n] + c) % m**
  - **m, modulus**
  - **a multiplier**
  - **c increment (c = 0, Park-Miller RNG)**
- **All m if gcd(m,c) = 1, a-1 has prime factors of m, a-1 is a multiple of 4 if m is a multiple of 4**

see http://en.wikipedia.org/wiki/Linear_congruential_generator

EXAMPLES OF CLG

	m	a	c
Numerical Recipes	2^{32}	1664525	1013904223
Borland C/C++	2^{32}	22695477	1 Ω
glibc (GCC)[4]	2^{32}	1103515245	12345
ANSI C:	2^{32}	1103515245	12345
Borland,Pascal	2^{32}	134775813	1
VisualC/C++	2^{32}	214013	2531011
Apple CarbonLib	$2^{31} - 1$	16807	0
MMIX, Donald Knuth	2^{64}	6364136223846793005	1442695040888963407
VAX's	2^{32}	69069	1
Java API	2^{48}	25214903917	11