Nov 30, 2005

Fast Fourier Transform

Decimation in Time

Decimation in Freq.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi nk}{N}}$$

For each k $\longrightarrow$ N adds, N mult.

N values of k $\longrightarrow$ N^2 adds, N^2 mult.

Direct way of computing DFT.

$O(N^2)$

Decimation in time.

$$X(k) = \sum_{\substack{n \text{ even} \\ 0 \to N-1}} x(n) e^{-j\frac{2\pi nk}{N}} + \sum_{\substack{n \text{ odd} \\ 0 \to N-1}} x(n) e^{-j\frac{2\pi nk}{N}}$$

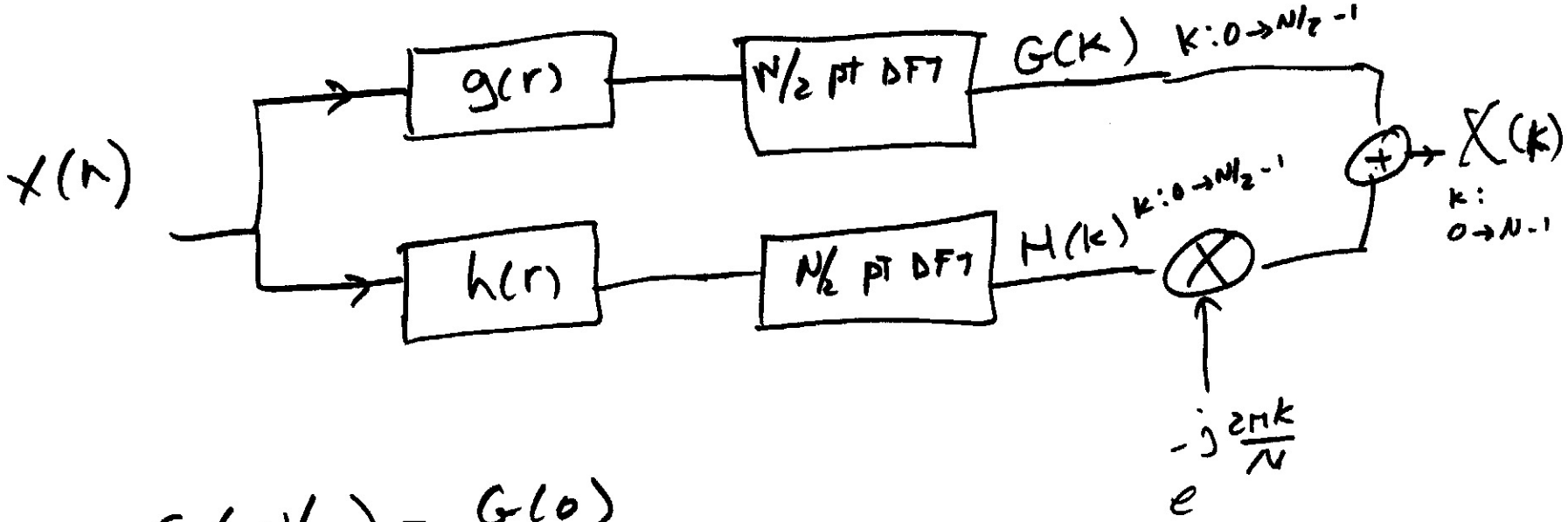
$n = 2r \quad r: 0 \to N/2 - 1$

$n = 2r+1$

$r: 0 \to N/2 - 1$

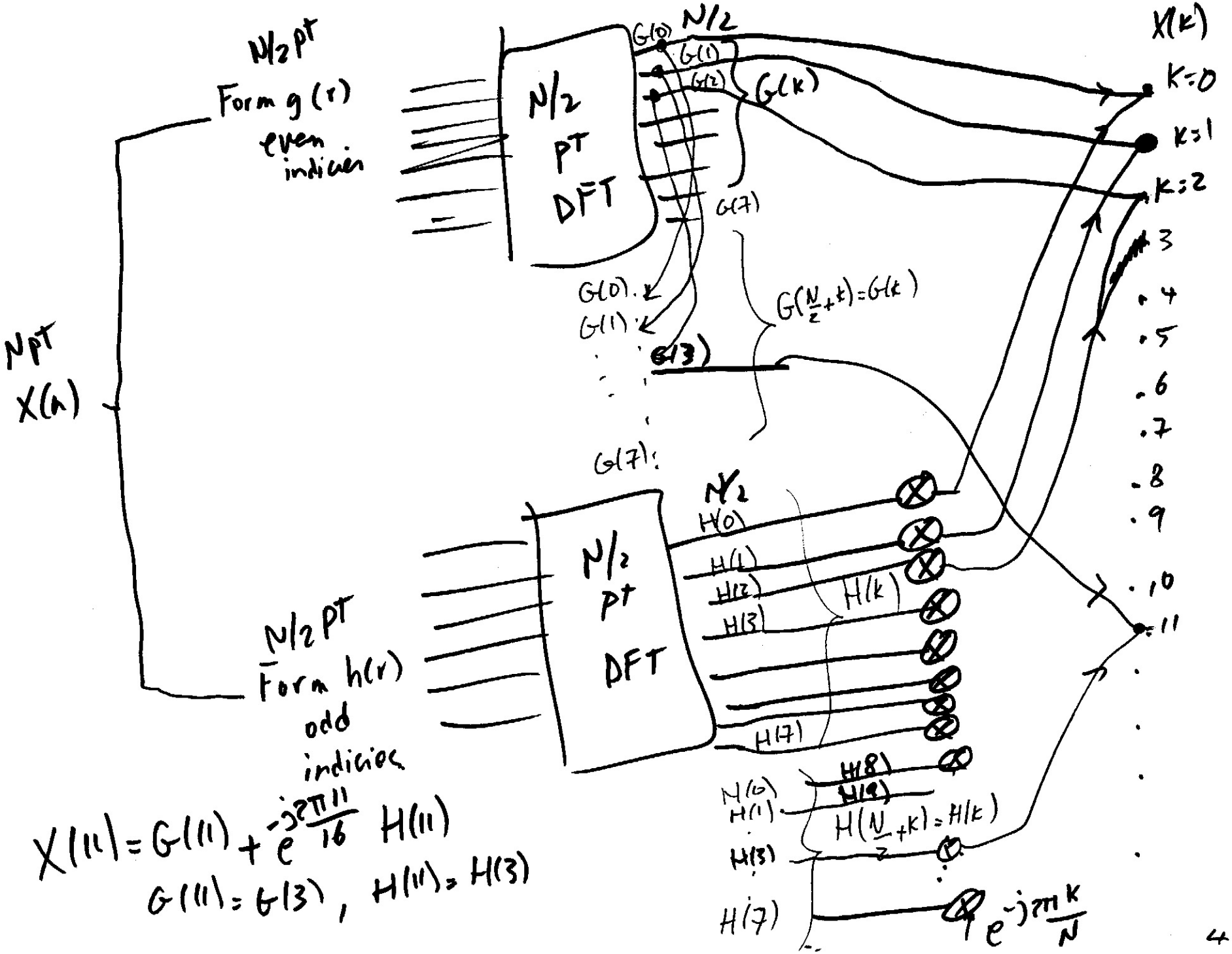
$$X(k) = \sum_{r=0}^{N/2-1} \underbrace{x(2r)}_{g(r)} e^{-j\frac{2\pi 2rk}{N}} + \sum_{r=0}^{N/2-1} \underbrace{x(2r+1)}_{h(r)} e^{-j\frac{2\pi(2r+1)k}{N}}$$

$$X(k) = \underbrace{\sum_{r=0}^{N/2-1} g(r) e^{-j\frac{2\pi kr}{N/2}}}_{N/2 \text{ pt DFT of } g(r)} + e^{-j\frac{2\pi k}{N}} \underbrace{\sum_{r=0}^{N/2-1} h(r) e^{-j\frac{2\pi kr}{N/2}}}_{N/2 \text{ pt DFT of } h(r)}$$



$$G(N/2) = G(0)$$
$$G(N/2+1) = G(1)$$
$$G(N/2+k) \triangleq G(K)$$

$$H(K) = H\left(\frac{N}{2}+K\right)$$



$$X(11) = G(11) + e^{-j\frac{2\pi 11}{16}} H(11)$$

$$G(11) = G(3), \quad H(11) = H(3)$$

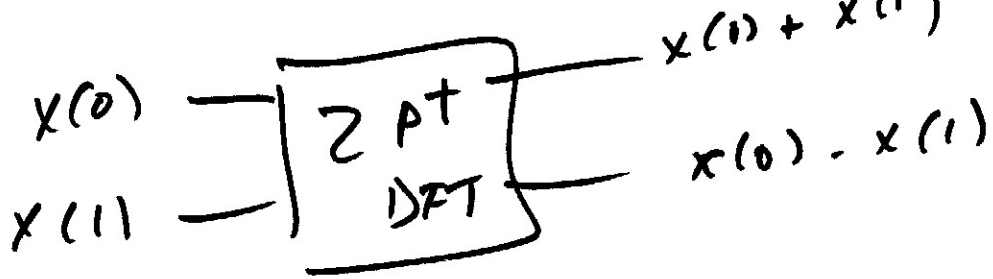
Repeat the above process.
until I get to a 2 pt DFT.

$$\sum_{n=0}^{1} x(n) e^{-j\frac{2\pi n k}{2}} = x(0) e^{-j(0)} + x(1) e^{-j\frac{2\pi k}{2}}$$

$$= x(0) + x(1) e^{-j\pi k}$$

$k=0$: $e^{-j\pi k} = 1 \longrightarrow X(0) = x(0) + x(1)$

$k=1$: $e^{-j\pi k} = -1 \Longrightarrow X(1) = x(0) - x(1)$



New Notation

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

Twiddle factor.

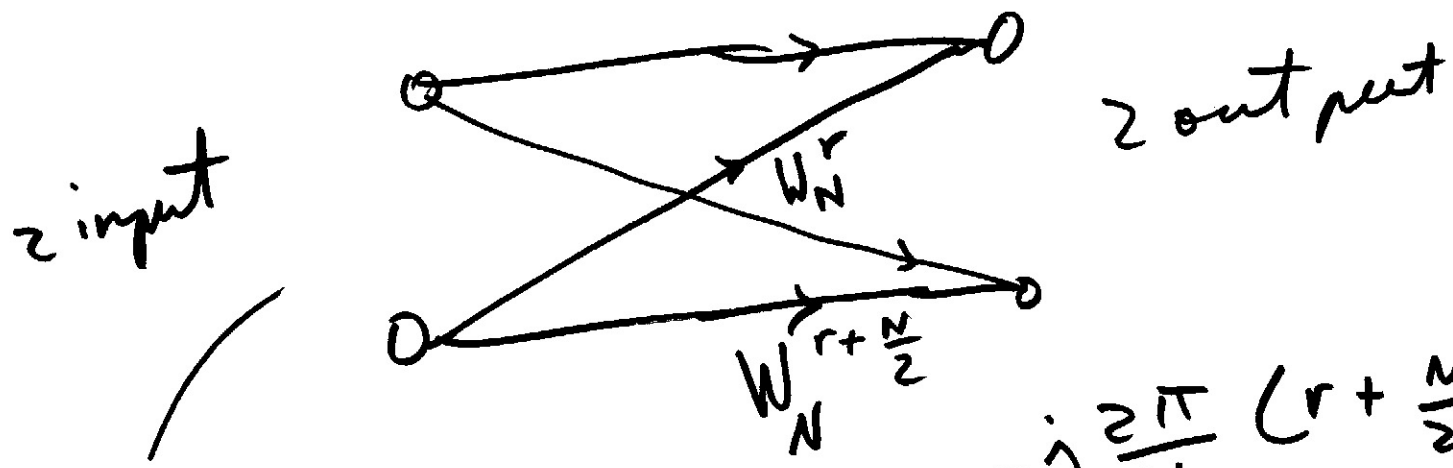
Use Twiddle factor to redraw my basic flow graph for Dec. in Time.

Fig 9.3 of O&S.

$\downarrow$

9.7 of O&S.

Discovered: FFT flowgraph is made of butterflies of the following form:



observe: $W_N^{r+N/2} \triangleq e^{-j\frac{2\pi}{N}\left(r+\frac{N}{2}\right)}$

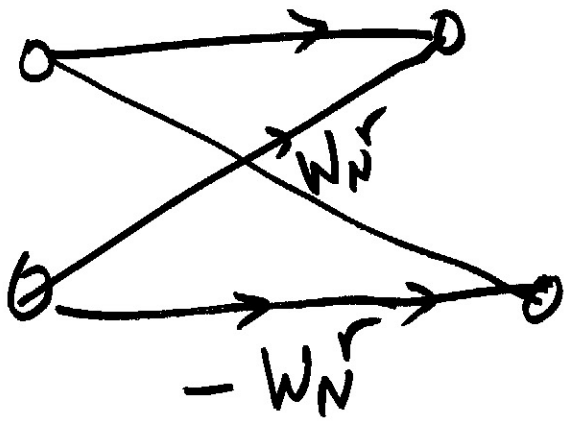
$$= e^{-j\frac{2\pi r}{N}} e^{-j\frac{2\pi N}{2N}}$$

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

$$= -W_N^r$$

$$W_N^{r+N/2} = -W_N^r$$

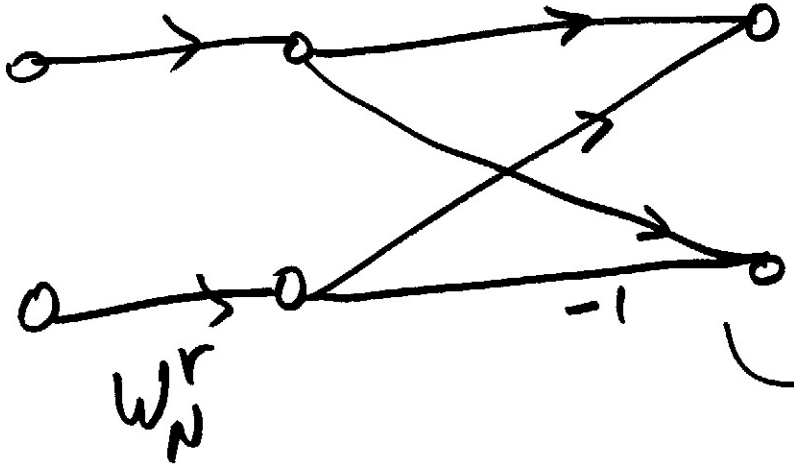
2 inputs



2 outputs.

2 mult
2 adds.

2 input



2 outputs

1 mult
2 adds.

new improved
more efficient
butterfly.

show Fig 9.10

From 9.10, observations

1. In place computation.
   start with an array of N #s
   and keep ~~re~~ re-using it,
   over-writing it.

2. Embarrasingly parallel.

3. $\log_2 N$ stage.

4. Each stage has $N/2$ butterflies
   with 2 inputs and 2 outputs

5. each processor do a one of the $N/2$
   butterflies in each stage.
   parallelize it to $N/2$ processor.

(6) Each butterfly 2 adds
1 mult.

Total: Each stage.

$N/2$ butterflies
each butter 1 mult
2 add

$\rightarrow$ $N/2$ mult
N adds

$\log_2 N$ stages

$\Rightarrow$ $\boxed{\begin{array}{ll} N/2 \log_2 N & \text{mult.} \\ N \log_2 N & \text{adds} \end{array}}$ $\leftarrow$ Total.

much better N^2

Ex $N = 64 \times 10^6$

$N^2 = 64 \times 64 \times 10^{12}$

$N \log_2 = 64 \times 10^6 \times (6 + 20) = 64 \times 26 \times 10^6$

$$\frac{N^2}{N \log_2 N} = \frac{\cancel{64} \times 64 \times 10^{12}}{\cancel{64} \times 26 \times 10^6} \simeq 2 \times 10^6$$

$\frac{2000000}{4600}$ = 500 hours = 3 weeks

1 sec

$N \log N$

Fig 9.10 → input order is messed up.

Bit Reversing

x(0)	0	000	000	0
x(4)	1	001	100	4
x(2)	2	010	010	2
x(6)	3	011	110	6
x(1)	4	100	001	1
x(5)	5	101	101	5
x(3)	6	110	011	3
x(7)	7	111	111	7