LECTURE 17

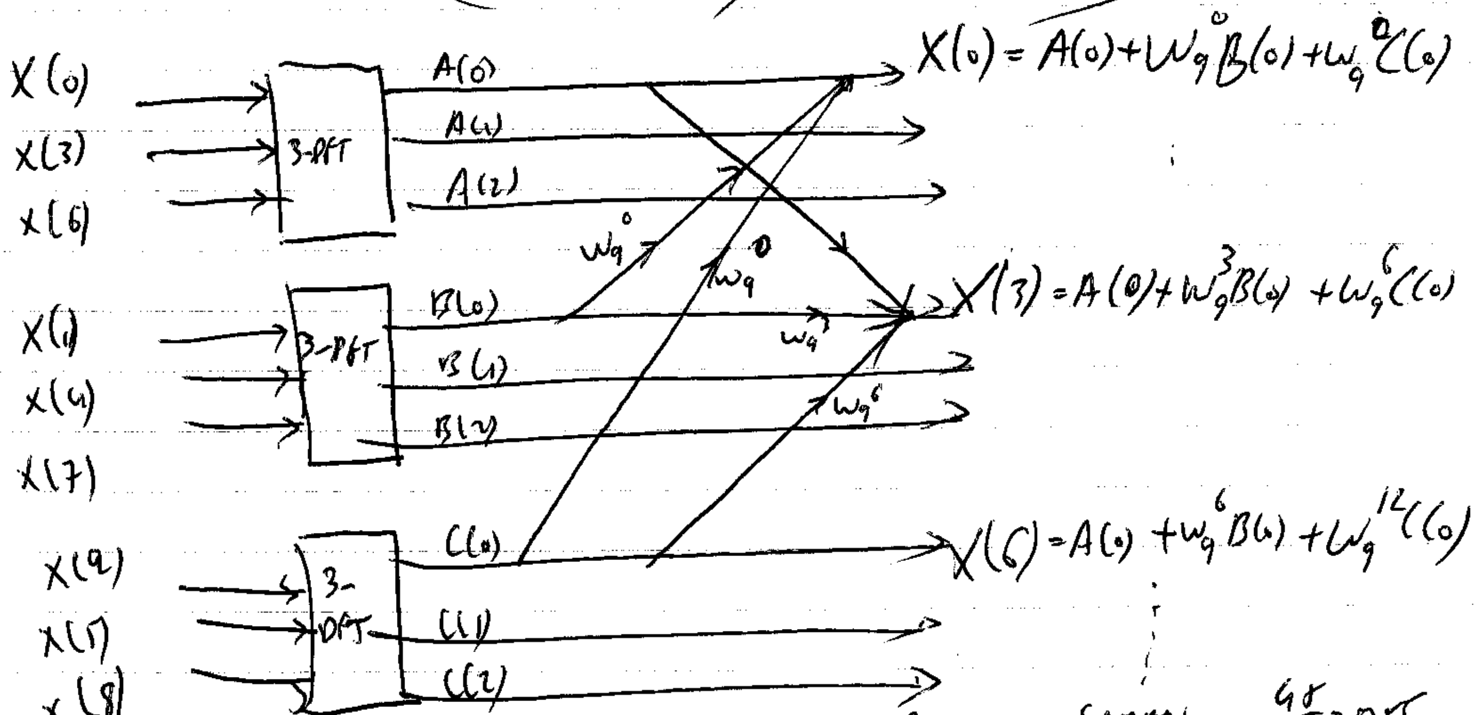
More examples on Decimation in time FFT algorithms

Let $N = 9 = 3^2$ This will be a radix 3 FFT implementation
In this case there is only one step of decomposition where we implement 3 3-DFT algorithms. Accordingly:

$$X(k) = \sum_{n=0}^{8} x[n] W_9^{nk} = \sum_{n=3m} x[n] W_9^{nk} + \sum_{n=3m+1} x[n] W_9^{nk} + \sum_{n=3m+2} x[n] W_9^{nk} =$$

$$= \sum_{m=0}^{2} x(3m) W_9^{3mk} + \sum_{m=0}^{2} x(3m+1) W_9^{(3m+1)k} + \sum_{m=0}^{2} x(3m+2) W_9^{(3m+2)k} =$$

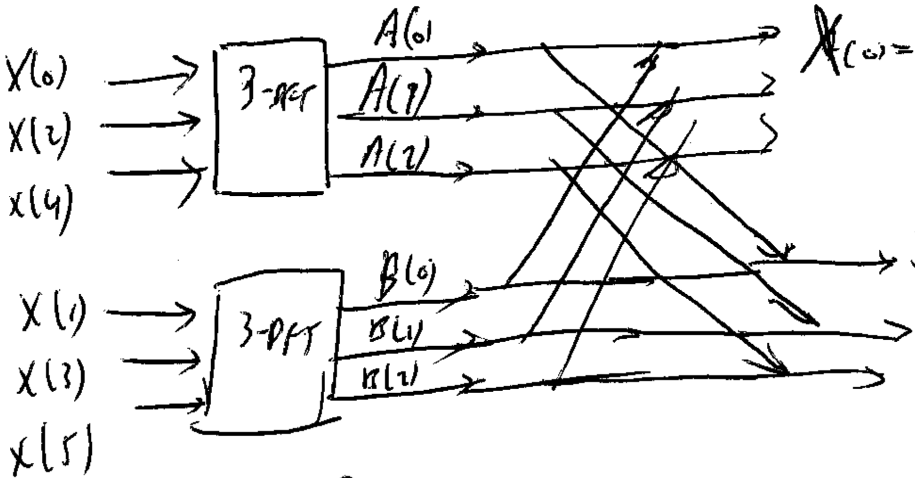
$$= \underbrace{\sum_{m=0}^{2} x(3m) W_3^{mk}}_{A(k),\ k=0,1,2} + W_9^{k} \underbrace{\sum_{m=0}^{2} x(3m+1) W_3^{mk}}_{B(k),\ k=0,1,2} + W_9^{2k} \underbrace{\sum_{m=0}^{2} x(3m+2) W_3^{mk}}_{C(k),\ k=0,1,2}$$



FFT $\otimes^c = 3 \cdot 3^2 + 9 \cdot 2 = 45$ DFT $\otimes^c = 9^2 = 81$ Savings ratio : $\frac{45}{81} = 0.55$

Let $N = 6 = 2\times 3$. We can implement FFT as follows.
Either combining 2 3-DFTs or 3 2-DFTs ..

$$X(k) = \sum_{n=2m} x(n) W_6^{nk} + \sum_{n=2m+1} x(n) W_6^{nk} = \underbrace{\sum_{m=0}^{2} x(2m) W_3^{mk}}_{A(k)} + W_6^{k} \underbrace{\sum_{m=0}^{2} x(2m+1) W_3^{mk}}_{B(k)}$$



$X(0) = A(0) + W_6^0 B(0)$

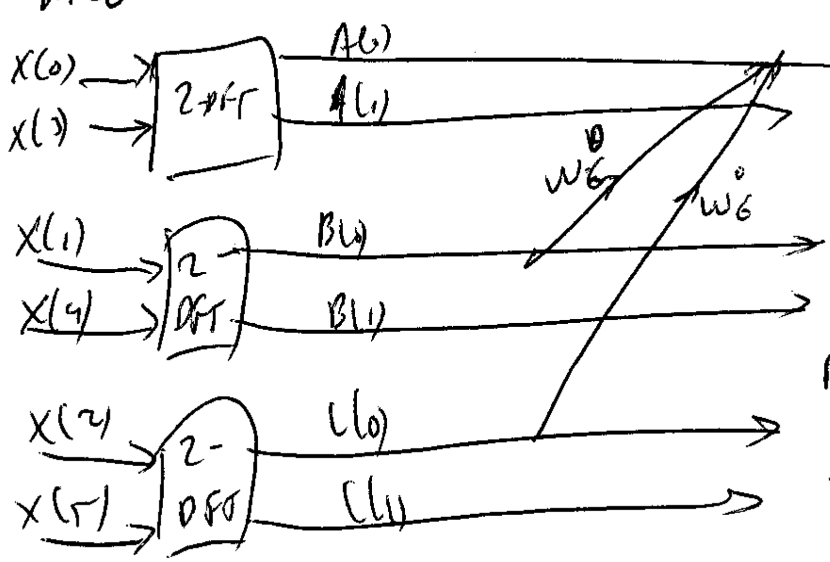
$X(3) = A(0) + W_6^3 B(0)$

FFT $\times$ C: $2 \cdot 3^2 + 6 = 18 + 6 = 24$ ratio : $\frac{24}{36}$

6-DFT $\times$ C : $6^2 = 36$

Otherwise.

$$X(k) = \sum_{m=0}^{1} x(3m) W_2^{mk} + W_6^{k} \sum_{m=0}^{1} x(3m+1) W_2^{mk} + W_6^{2k} \sum_{m=0}^{1} x(3m+2) W_2^{mk}$$



$X(0) = A(0) + B(0) + C(0)$

$X(1) = A(1) + W_2^1 B(1) + W_2^2 C(1)$

$X(2) = A(0) + W_6^2 B(0) + W_6^4 C(0)$

FFT $\times$ C: $3 \cdot 2^2 + 12 = 24$

DFT $\times$ C : 36

ratio $= \frac{24}{36}$

How can we derive an IFFT algorithm (radix 2, decimation in time)

Formula of N-IDFT : $x(n) = \sum_{k=0}^{N-1} \bar{X}(k) W_N^{-kn} = \frac{1}{2^q} \sum_{k=0}^{N-1} \bar{X}(k) W_N^{-kn}$

So:

1. Use the same diagram as in forward DFT (FFT)
2. Use W_N^{-r} instead of W_N^{r}
3. Multiply outputs of each butterfly by $\frac{1}{2}$
   (q stages. $\left(\frac{1}{2}\right)^q = \frac{1}{N}$.
4. Input $X(k)$ is in bit reversed order,
   output $x(n)$ is in normal order.

---

* Note: If N is a prime number there is __no__ FFT implementation. That is

$N = 7$ $\quad$ 7-DFT ≡ 7-FFT
$N = 13$ $\quad$ 13-DFT ≡ 13-FFT

$N = 12 = 2 \cdot 6 = 3 \cdot 4$ (There are 4 possible FFT implementations)