Fast Fourier Transform: Theory and Algorithms

Lecture 8

Vladimir Stojanović

6.973 Communication System Design – Spring 2006

Massachusetts Institute of Technology

Discrete Fourier Transform – A review

- Definition $X_k = \sum_{i=0}^{N-1} x_i W_N^{ik}, \quad k = 0, \ldots, N-1, \quad W_N = e^{-j2\pi/N}$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix}.$$

- $\{X_k\}$ is periodic
  - Since $\{X_k\}$ is sampled, $\{x_n\}$ must also be periodic
- From a physical point of view, both are repeated with period N
- Requires $O(N^2)$ operations

Fast Fourier Transform History

- Twiddle factor FFTs (non-coprime sub-lengths)
  - 1805 Gauss
    - Predates even Fourier's work on transforms!
  - 1903 Runge
  - 1965 Cooley-Tukey
  - 1984 Duhamel-Vetterli (split-radix FFT)

Gauss, C. F., "Nachlass: Theoria interpolationis methodo nova tractata," pp. 265–303, in *Carl Friedrich Gauss, Werke, Band 3*, Göttingen: Königlichen Gesellschaft der Wissenschaften, 1866.

- FFTs w/o twiddle factors (coprime sub-lengths)
  - 1960 Good's mapping
    - application of Chinese Remainder Theorem ~100 A.D.
  - 1976 Rader – prime length FFT
  - 1976 Winograd's Fourier Transform (WFTA)
  - 1977 Kolba and Parks (Prime Factor Algorithm – PFA)

Divide and conquer

$$X_k = \sum_{i=0}^{N-1} x_i W_N^{ik}, \quad k = 0, \ldots, N-1, \quad W_N = e^{-j2\pi/N}$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i}. \quad z = W_N^{-k}:$$

$$X_k = X(z)_{z=W_N^{-k}}.$$

- Divide and conquer always has less computations

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i} = \sum_{l=0}^{r-1} \sum_{i \in I_l} x_i z^{-i},$$

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

Suppose all I_l sets have same number of elements N_1 so, $N=N_1*N_2$, $r=N_2$

Each inner-most sum takes N_1^2 multiplications

The outer sum will need N_2 multiplications per output point N_2*N for the whole sum (for all output points)

- Hence, total number of multiplications

$$N_2 \cdot N + N_2 \cdot N_1^2 = N_1 \cdot N_2(N_1 + N_2) < N_1^2 \cdot N_2^2 \quad \text{if } N1, N2 > 2$$

Generalizations

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

- The inner-most sum has to represent a DFT
  - Only possible if the subset (possibly permuted)
    - Has the same periodicity as the initial sequence
  - All main classes of FFTs can be cast in the above form
- Both sums have same periodicity (Good's mapping)
  - No permutations (i.e. twiddle factors)
  - All the subsets have same number of elements m=N/r
    - (m,r)=1 – i.e. m and r are coprime
- If not, then inner sum is one stap of radix-r FFT
- If r=3, subsets with N/2, N/4 and N/4 elements
  - Split-radix algorithm

The cost of mapping

- The goal for divide and conquer

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

$$\sum \text{cost(subproblems)} + \text{cost(mapping)}$$
$$< \text{cost(original problem)}.$$

- Different types balance mapping with subproblem cost
- E.g. in radix-2
  - subproblems are trivial (only sum and differences)
  - Mapping requires twiddle factors (large number of multiplies)
- E.g. in prime-factor algorithm
  - Subproblems are DFTs with coprime lengths (costly)
  - Mapping trivial (no arithmetic operations)

FFTs with twiddle factors

- Reintroduced by Cooley-Tukey '65

$$X_k = \sum_{i=0}^{N-1} x_i W_N^{ik}, \quad k=0,\ldots,N-1, \quad W_N = e^{-j2\pi/N}$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i}. \quad z = W_N^{-k}:$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i} = \sum_{l=0}^{r-1} \sum_{i \in I_l} x_i z^{-i},$$

Start from general divide and conquer

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

Keep periodicity compatible with periodicity of the input sequence

Use decimation

$$I_{n_1} = \{n_2 N_1 + n_1\},$$

$$n_1 = 0,\ldots,N_1-1, \quad n_2 = 0,\ldots,N_2-1,$$

$$N = N_1 \cdot N_2. \quad \{x_i \mid i=0,\ldots,N-1\} \quad \{x_i \mid i \in I_l\}$$

$$X(z) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_2N_1+n_1} z^{-(n_2N_1+n_1)}$$

$$X(z) = \sum_{n_1=0}^{N_1-1} z^{-n_1} \sum_{n_2=0}^{N_2-1} x_{n_2N_1+n_1} z^{-n_2N_1},$$

$$X_k = X(z)|_{z=W_N^{-k}}$$

$$= \sum_{n_1=0}^{N_1-1} W_N^{n_1k} \sum_{n_2=0}^{N_2-1} x_{n_2N_1+n_1} W_N^{n_2N_1k}.$$

$$W_N^{iN_1} = e^{-j2\pi N_1 i/N} = e^{-j2\pi i/N_2} = W_{N_2}^i$$

$$X_k = \sum_{n_1=0}^{N_1-1} W_N^{n_1k} \sum_{n_2=0}^{N_2-1} x_{n_2N_1+n_1} W_{N_2}^{n_2k}.$$

almost N_1 DFTs of size N_2

Cooley-Tukey FFT contd.

$$X_k = \sum_{n_1=0}^{N_1-1} W_N^{n_1 k} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} W_{N_2}^{n_2 k}.$$

can be taken mod N_2

$$W_{N_2}^{k} = W_{N_2}^{N_2+k'} = W_{N_2}^{N_2} \cdot W_{N_2}^{k'} = W_{N_2}^{k'}.$$

$$Y_{n_1,k} = \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} W_{N_2}^{n_2 k}.$$

$$Y_{n_1,k} = Y_{n_1,k_2}$$

1. N_1 DFTs of length N_2

$$X_k = \sum_{n_1=0}^{N_1-1} Y_{n_1,k} W_N^{n_1 k}.$$

$$k = k_1 N_2 + k_2$$

$$k_1 = 0, \ldots, N_1 - 1, \quad k_2 = 0, \ldots, N_2 - 1.$$

$$X_{k_1 N_2 + k_2} = \sum_{n_1=0}^{N_1-1} Y_{n_1,k_2} W_N^{n_1(k_1 N_2 + k_2)},$$

$$X_{k_1 N_2 + k_2} = \sum_{n_1=0}^{N_1-1} Y_{n_1,k_2} W_N^{n_1 k_2} W_{N_1}^{n_1 k_1}.$$

$$Y'_{n_1,k_2} = Y_{n_1,k_2} W_N^{n_1 k_2}.$$

2. N multplications with twiddle factors

$$X_{k_1 N_2 + k_2} = \sum_{n_1=0}^{N_1-1} Y'_{n_1,k_2} W_{N_1}^{n_1 k_1}.$$

3. N_2 DFTs of length N_1

- Step 1: Evaluate N_1 DFTs of length N_2
- Step 2: N multiplications with twiddle factors
- Step 3: Evaluate N_2 DFTs of length N_1

- Vector x_i mapped to matrix $x_{n1,n2}$ (N_1xN_2)
- Compute N_1 DFTs of length N_2 on each row
- Point-to-point multiply with twiddle factors
- Compute N2 DFTs of length N1 on the columns

2-D view of Cooley-Tukey mapping

- N=15 (N_1=3, N_2=5)

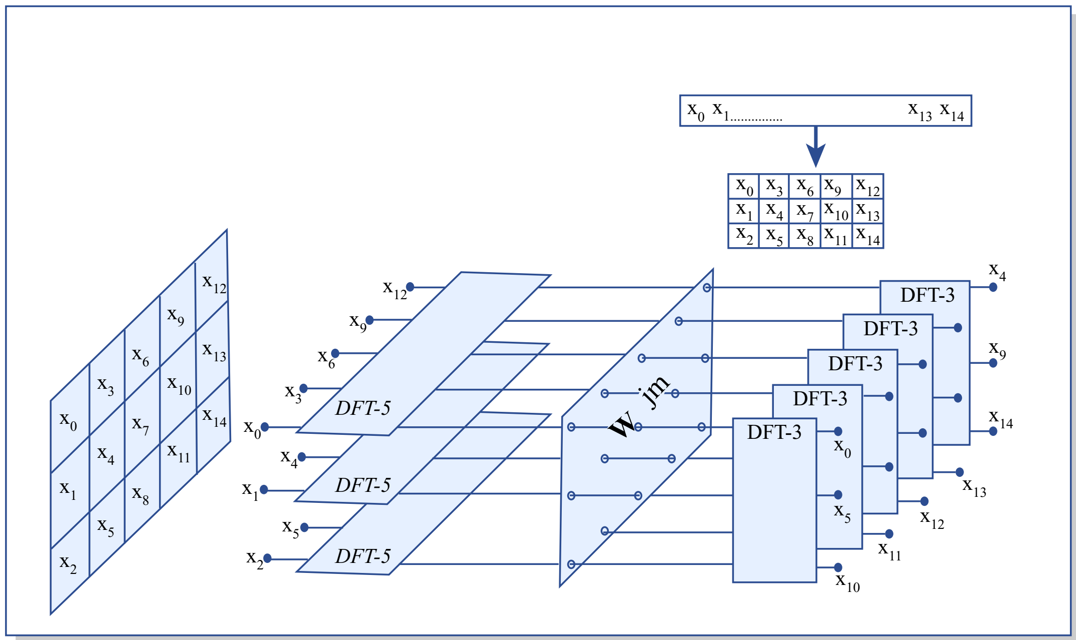


Figure by MIT OpenCourseWare.

- Cannot exchange the order of DFTs
  - Because of twiddle multiply
  - Different mapping for N1=5, N2=3
    - Not just transpose

x_0	x_5	x_{10}
x_1	x_6	x_{11}
x_2	x_7	x_{12}
x_3	x_8	x_{13}
x_4	x_9	x_{14}

Figure by MIT OpenCourseWare.

Radix 2 and radix 4 algorithms

- Lengths as powers of 2 or 4 are most popular
- Assume $N=2^n$
  - $N_1=2$, $N_2=2^{n-1}$ (divides input sequence into even and odd samples – decimation in time – DIT)

$$X_{k_2} = \sum_{n_2=0}^{N/2-1} x_{2n_2} W_{N/2}^{n_2k_2} + W_N^{k_2} \sum_{n_2=0}^{N/2-1} x_{2n_2+1} W_{N/2}^{n_2k_2},$$

$$X_{N/2+k_2} = \sum_{n_2=0}^{N/2-1} x_{2n_2} W_{N/2}^{n_2k_2} - W_N^{k_2} \sum_{n_2=0}^{N/2-1} x_{2n_2+1} W_{N/2}^{n_2k_2}.$$

$$Y_{n_1,k} = \sum_{n_2=0}^{N_2-1} x_{n_2N_1+n_1} W_{N_2}^{n_2k}. \qquad Y'_{n_1,k_2} = Y_{n_1,k_2} W_N^{n_1k_2}.$$

$$X_{k_1N_2+k_2} = \sum_{n_1=0}^{N_1-1} Y'_{n_1,k_2} W_{N_1}^{n_1k_1}.$$

"Butterfly"
(sum or difference followed or preceeded by a twiddle factor multiply)

  - X_m and $X_{N/2+m}$ outputs of N/2 2-pt DFTs on outputs of 2, N/2-pt DFTs weighted with twiddle factors

DIT radix-2 implementations

- Several different ways
  - Reorder the input data
    - Input samples for inner DFTs in subsequent locations
    - Results in bit-reversed input, in-order output DIT
  - Selectively compute DFTs on evens and odds
    - Perform in-place computation
    - Output in bit-reversed order (X3 in position six (011->110))

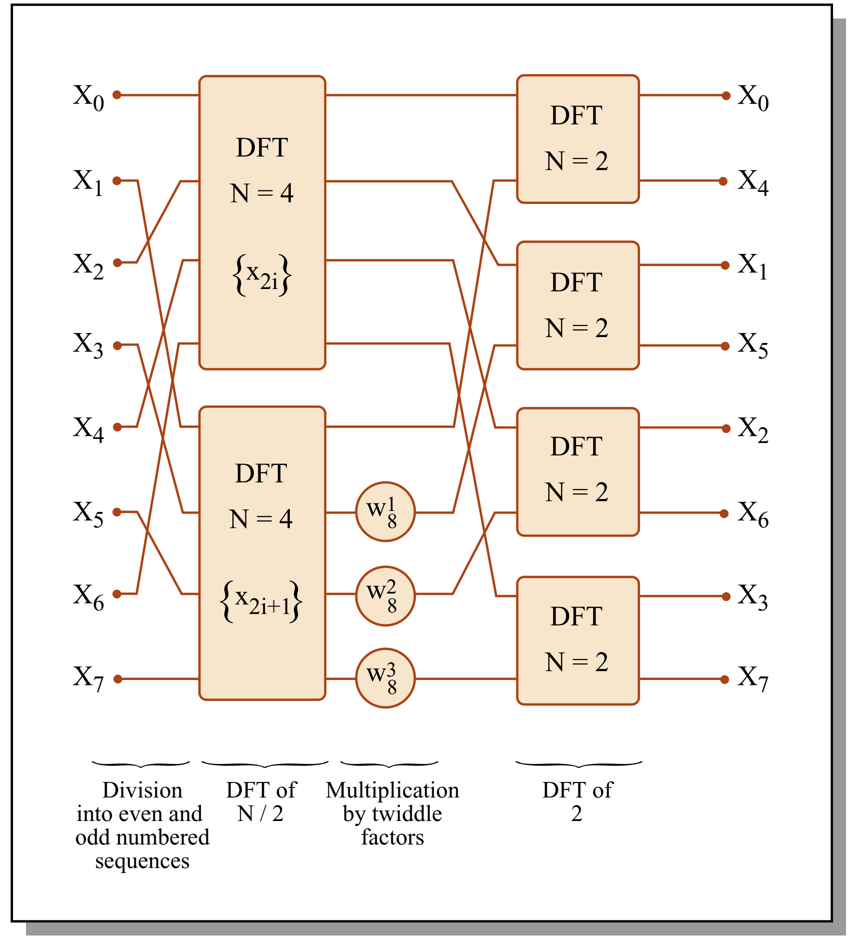


Figure by MIT OpenCourseWare.

Which type is this implementation?

Decimation in frequency (DIF) radix-2 implementation

- If reverse the role of N_1 and N_2, get DIF
  - $N_1=N/2$, $N_2=2$

$$X_{2k_1} = \sum_{n_1=0}^{N/2-1} W_{N/2}^{n_1 k_1}(x_{n_1}+x_{N/2+n_1}),$$

$$X_{2k_1+1} = \sum_{n_1=0}^{N/2-1} W_{N/2}^{n_1 k_1} W_N^{n_1}(x_{n_1}-x_{N/2+n_1}).$$

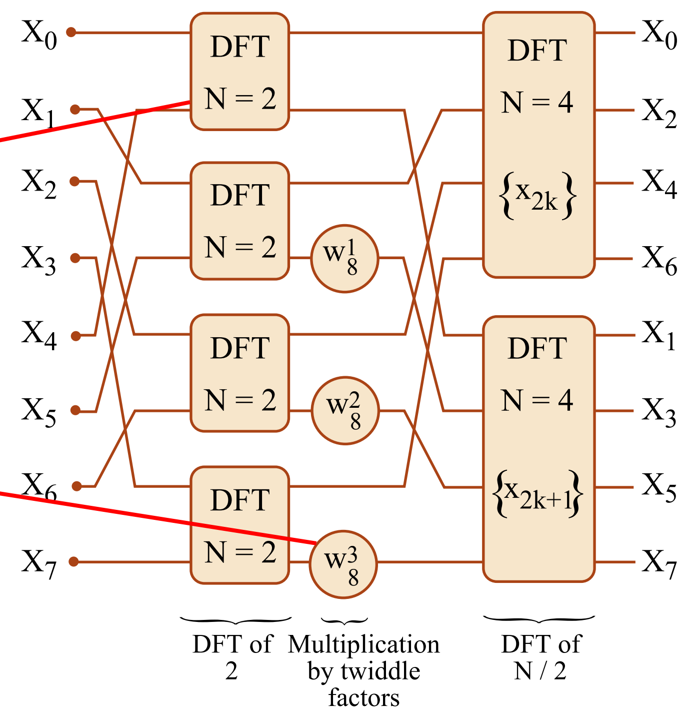


Figure by MIT OpenCourseWare.

Duality DIT<->DIF

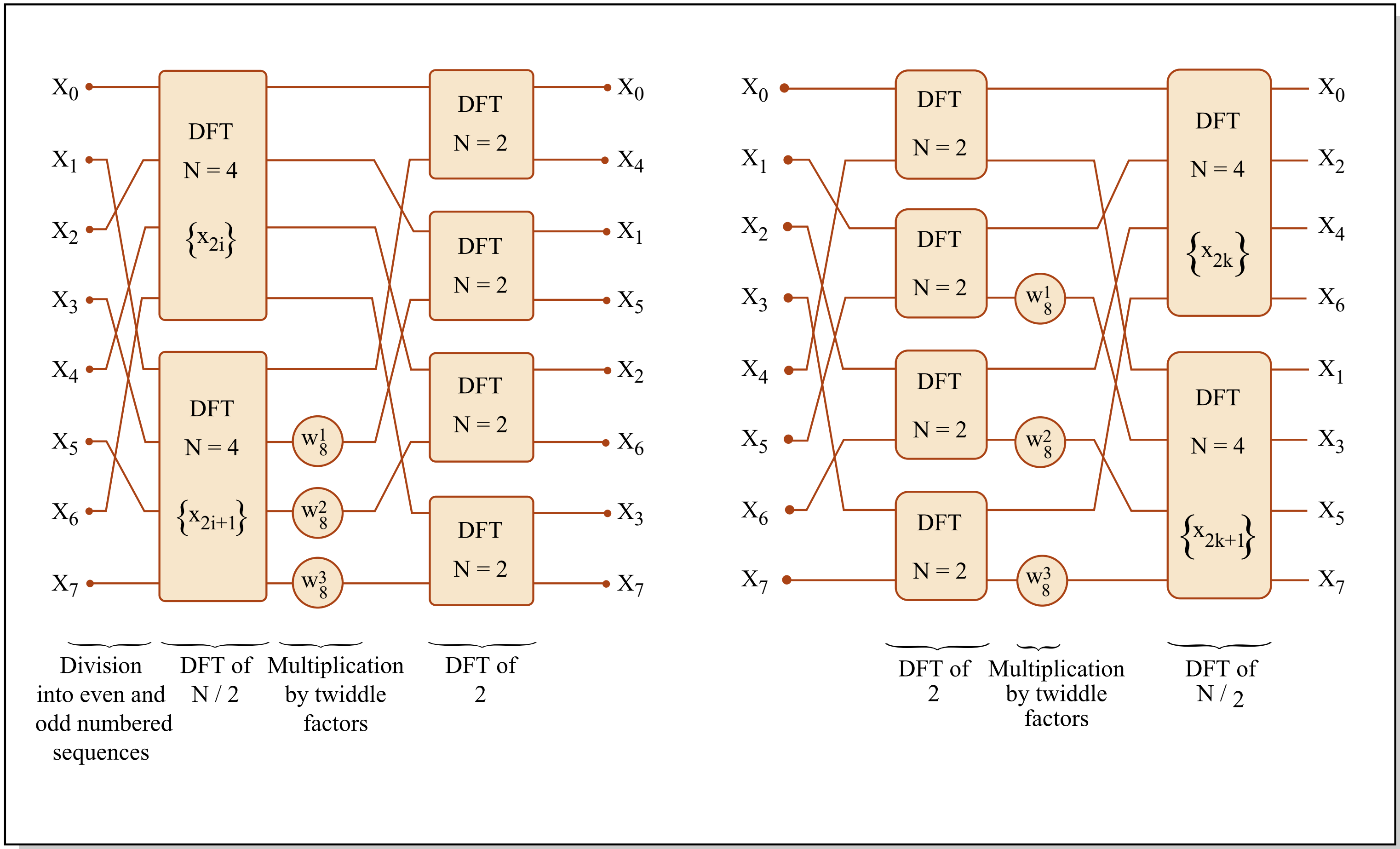


Figure by MIT OpenCourseWare.

- Which one is DIT (DIF)?
- How can we get one from another?

Complexity of radix-2 FFTs

- DFT of length N replaced by two length-N/2
  - At the cost of N complex multiplications (twiddle)
    - And N complex additions (2pt DFTs)
- Iterate the scheme $\log_2 N$-1 times
  - Obtain trivial transforms (length 2) of the length-N/2 DFTs

$$O_M[DFT_{radix\text{-}2}] \approx N/2(\log_2 N - 1)$$

$$O_A[DFT_{radix\text{-}2}] \approx N(\log_2 N - 1)$$

- Twiddle multiplies (W_N^i)
  - Complex multiply – 3 real mult + 3 real add
  - If i is multiple of N/4, no arithmetic operation required (why?)

$$M[DFT_{radix\text{-}2}] = 3N/2 \log_2 N - 5N + 8,$$

$$A[DFT_{radix\text{-}2}] = 7N/2 \log_2 N - 5N + 8.$$

4 butterflies (one general, 3 special cases)

Radix-4

- $N=4^n$, $N_1=4$, $N_2=N/4$
  - 4 DFTs of length N/4
  - 3N/4 twiddle multiplies
  - N/4 DFTs of length 4
- Cost of length-4 DFT
  - No multiplication
  - Only 16 real additions
- Reduces the number of stages to $\log_4 N$

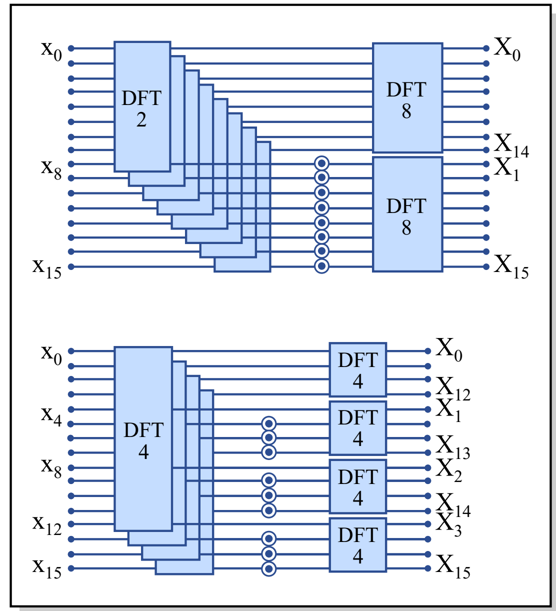


Figure by MIT OpenCourseWare.

$$O_M[\mathrm{DFT}_{\text{radix-4}}] \approx 3N/4(\log_4 N - 1).$$

$$M[\mathrm{DFT}_{\text{radix-4}}] = 9N/8 \log_2 N - 43N/12 + 16/3,$$

$$A[\mathrm{DFT}_{\text{radix-4}}] = 25N/8 \log_2 N - 43N/12 + 16/3.$$

$$O_M[\mathrm{DFT}_{\text{radix-2}}] \approx N/2(\log_2 N - 1)$$

$$M[\mathrm{DFT}_{\text{radix-2}}] = 3N/2 \log_2 N - 5N + 8,$$

$$A[\mathrm{DFT}_{\text{radix-2}}] = 7N/2 \log_2 N - 5N + 8.$$

  - Radix-8 can reduce number of operations even more

Mixed-radix and Split-radix

- Mixed-radix
  - Diferent radices in different stages
- Split-radix
  - Different radices in the same stage
    - Simultaneously on different parts of the transform
  - Can achieve lowest number of adds and multiplies for length 2^n inputs

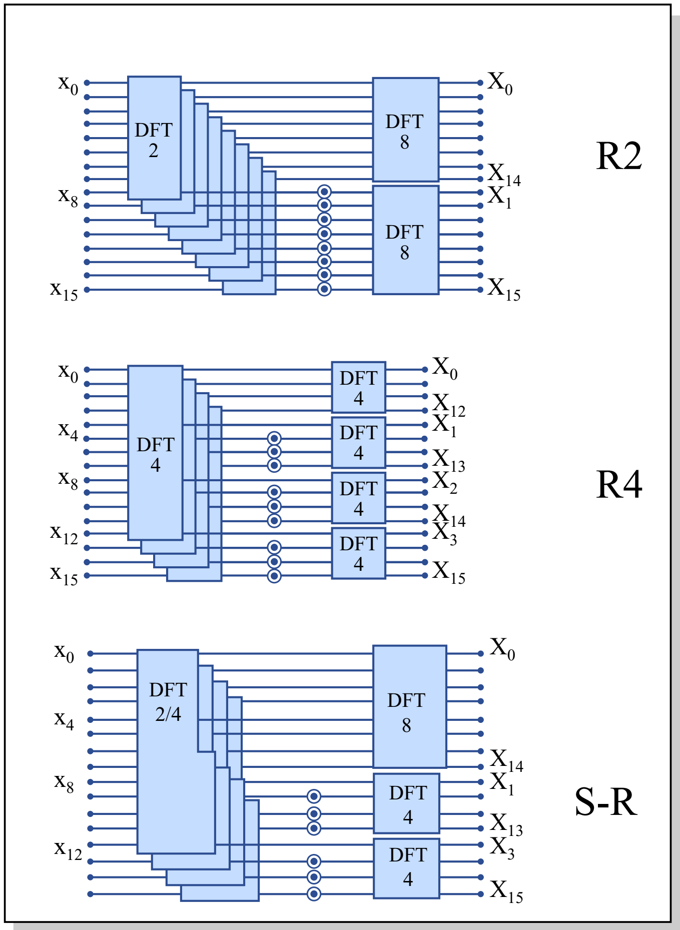


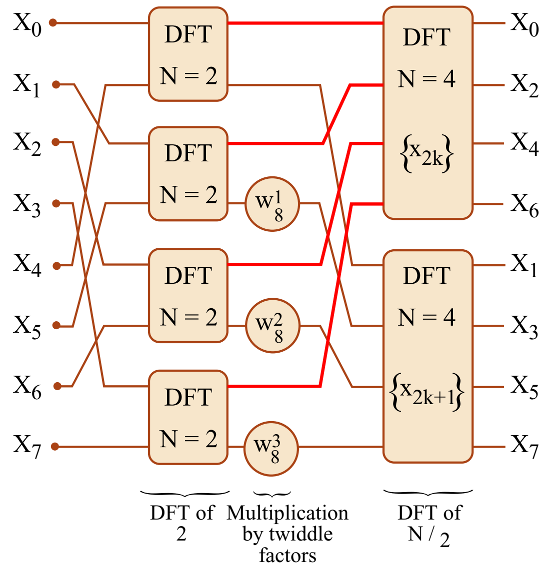
Figure by MIT OpenCourseWare.

Split-radix (DIF SRFFT)

- Look at DIF radix-2
  - X_{2k1} don't have twiddles

$$X_{2k_1} = \sum_{n_1=0}^{N/2-1} W_{N/2}^{n_1 k_1}(x_{n_1} + x_{N/2+n_1}),$$

$$X_{2k_1+1} = \sum_{n_1=0}^{N/2-1} W_{N/2}^{n_1 k_1} W_N^{n_1}(x_{n_1} - x_{N/2+n_1}).$$



$$X_{2k_1} = \sum_{n_1=0}^{N/2-1} W_{N/2}^{n_1 k_1}(x_{n_1} + x_{N/2+n_1}),$$

$$X_{4k_1+1} = \sum_{n_1=0}^{N/4-1} W_{N/4}^{n_1 k_1} W_N^{n_1} \mathrm{X}[(x_{n_1} - x_{N/2+n_1}) + j(x_{n_1+N/4} - x_{n_1+3N/4})],$$

$$X_{4k_1+3} = \sum_{n_1=0}^{N/4-1} W_{N/4}^{n_1 k_1} W_N^{3n_1} \mathrm{X}[(x_{n_1} + x_{N/2+n_1}) - j(x_{n_1+N/4} - x_{n_1+3N/4})].$$

Figure by MIT OpenCourseWare.

- Even samples X_{2k} in DIF should be computed separately from other samples
  - With same algorithm (recursively) as the original sequence
- No general rule for odd samples
  - Radix-4 is more efficient than radix-2
  - Higher radices are inefficient

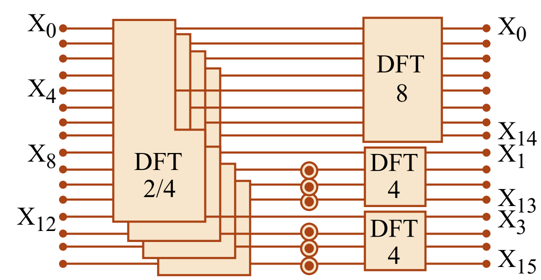


Figure by MIT OpenCourseWare.

Split-radix (DIT SRFFT)

- Dual to DIF SRFFT
  - Considers separately subsets $\{x_{2i}\}$, $\{x_{4i+1}\}$ and $\{x_{4i+3}\}$

$$I_0=\{2i\},\quad I_1=\{4i+1\},\quad I_2=\{4i+3\}$$

$$X(z)=\sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i\in I_l} x_i z^{-i+i_{0l}}.$$

$$X_k=\sum_{I_0} x_{2i} W_N^{k(2i)} + W_N^k \sum_{I_1} x_{4i+1} W_N^{k(4i+1)-k} + W_N^{3k} \sum_{I_2} x_{4i+3} W_N^{k(4i+3)-3k},$$

  - Redundancy in X_k, $X_{k+N/4}$, $X_{k+N/2}$, $X_{k+3N/4}$ computation

$$X_k=\sum_{i=0}^{N/2-1} x_{2i} W_{N/2}^{ik} + W_N^k \sum_{i=0}^{N/4-1} x_{4i+1} W_{N/4}^{ik} + W_N^{3k} \sum_{i=0}^{N/4-1} x_{4i+3} W_{N/4}^{ik},$$

$$X_{k+N/4}=\sum_{i=0}^{N/2-1} x_{2i} W_{N/2}^{ik} + jW_N^k \sum_{i=0}^{N/4-1} x_{4i+1} W_{N/4}^{ik} - jW_N^{3k} \sum_{i=0}^{N/4-1} x_{4i+3} W_{N/4}^{ik},$$

$$M[\mathrm{DFT}_{\text{split-radix}}]=N\log_2 N-3N+4,$$

$$A[\mathrm{DFT}_{\text{split-radix}}]=3N\log_2 N-3N+4.$$

$$X_{k+N/2}=\sum_{i=0}^{N/2-1} x_{2i} W_{N/2}^{ik} - W_N^k \sum_{i=0}^{N/4-1} x_{4i+1} W_{N/4}^{ik} - W_N^{3k} \sum_{i=0}^{N/4-1} x_{4i+3} W_{N/4}^{ik},$$

$$X_{k+3N/4}=\sum_{i=0}^{N/2-1} x_{2i} W_{N/2}^{ik} - jW_N^k \sum_{i=0}^{N/4-1} x_{4i+1} W_{N/4}^{ik} + jW_N^{3k} \sum_{i=0}^{N/4-1} x_{4i+3} W_{N/4}^{ik}.$$

FFTs without twiddle factors

- Divide and conquer requirements
  - N-long DFT computed from DFTs with lengths that are factors of N (allows the inner sum to be a DFT)
    - Provided that subsets I_l guarantee periodic x_i

$$X(z) = \sum_{n_1=0}^{N_1-1} z^{-n_1} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} z^{-n_2 N_1},$$

$$X_k = X(z)\big|_{z=W_N^{-k}}$$

$$= \sum_{n_1=0}^{N_1-1} W_N^{n_1 k} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} W_N^{n_2 N_1 k}.$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i} = \sum_{l=0}^{r-1} \sum_{i \in I_l} x_i z^{-i},$$

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

  - When N factors into co-prime factors $N=N_1*N_2$
    - Starting from any x_i form subset with compatible periodicity (the periodicity of the subset divides the periodicity of the set)

$$\{x_{i+N_1 n_2} \mid n_2 = 1,\ldots,N_2-1\} \text{ or } \{x_{i+N_2 n_1} \mid n_1 = 1,\ldots,N_1-1\}$$

    - Both subsets have only one common point x_i
    - Allows a rearrangement of the input (periodic) vector into a matrix with a periodicity in both dimensions (rows and columns), both periodicities being comatible with the initial one

Good’s mapping

- FFTs without twiddle factors all based on the same mapping
  - Turns original transform into a set of small DFTs with coprime lengths

$$\{x_{i+N_1 n_2} \mid n_2 = 1,\ldots,N_2-1\} \;\; or \;\; \{x_{i+N_2 n_1} \mid n_1 = 1,\ldots,N_1-1\}$$

equivalent to

$$i = \langle n_1 \cdot N_2 + n_2 \cdot N_1 \rangle_N,$$
$$n_1 = 1,\ldots,N_1-1, \quad n_2 = 1,\ldots,N_2-1$$
$$N = N_1 N_2,$$

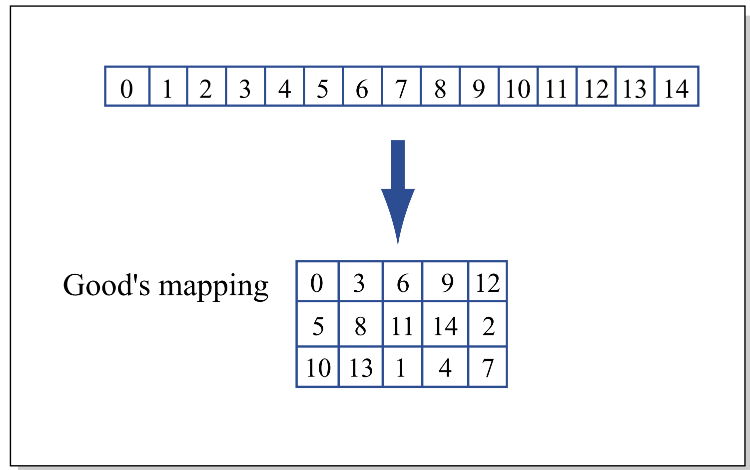


Figure by MIT OpenCourseWare.

- This mapping is one-to-one if N1 and N2 are coprime
- All congruences modulo N1 obtained
  - For a given congruence modulo N2 and vice versa

Just another arrangement of CRT

- Chinese Remainder Theorem (CRT)
  - If we know the residue of some number k modulo two coprime numbers N_1 and N_2 $\langle k \rangle_{N_1}$ $\langle k \rangle_{N_2}$
  - It is possible to reconstruct $\langle k \rangle_{N_1 N_2}$
  - Let $\langle k \rangle_{N_1} = k_1$ $\langle k \rangle_{N_2} = k_2$
  - Then $\langle k \rangle_{N_1 N_2} = \langle N_1 t_1 k_2 + N_2 t_2 k_1 \rangle_N$

$$\langle t_1 N_1 \rangle_{N_2} = 1 \ \ and \ \ \langle t_2 N_2 \rangle_{N_1} = 1$$

t_1 multiplicative inverse of N_1 mod N_2
t_2 multiplicative inverse of N_2 mod N_1
t_1, t_2 always exist since N_1, N_2 coprime $(N_1,N_2)=1$

What are t_1, t_2 for $N_1=3$, $N_2=5$?

  - Reversing N_1 and N_2
    - Results in transposed mapping

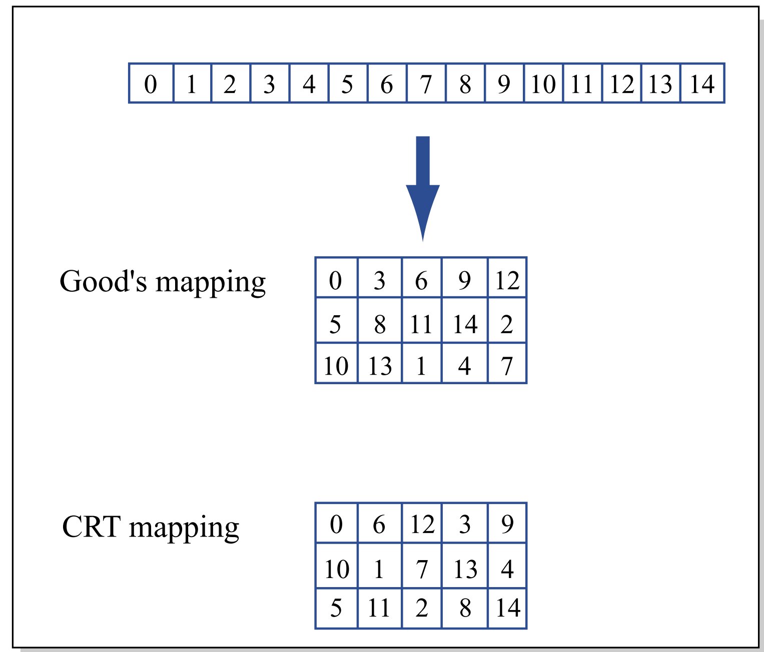


Figure by MIT OpenCourseWare.

Impact on DFT

- Formulating the true multi-dimensional transform

$$\langle k \rangle_{N_1 N_2} = \langle N_1 t_1 k_2 + N_2 t_2 k_1 \rangle_N$$

$$X_k = \sum_{i=0}^{N-1} x_i \, W_N^{ik}, \quad k = 0, \ldots, N-1,$$

$$X_{N_1 t_1 k_2 + N_2 r_2 k_1} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1 N_2 + n_2 N_1} \, W_N^{(n_1 N_2 + N_1 n_2)(N_1 t_1 k_2 + N_2 t_2 k_1)}$$

$$W_N^{N_2} = W_{N_1} \qquad W_{N_1}^{N_2 t_2} = W_{N_1}^{(N_2 t_2) N_1} = W_{N_1}$$

$$X_{N_1 t_1 k_2 + N_2 t_2 k_1} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1 N_2 + n_2 N_1} \, W_{N_1}^{n_1 k_2} \, W_{N_2}^{n_2 k_2},$$

$$X'_{k_1 k_2} = X_{N_1 t_1 k_2 + N_2 t_2 k_1} \qquad x'_{n_1, n_2} = x_{n_1 N_2 + n_2 N_1}$$

$$X'_{k_1 k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x'_{n_1 n_2} \, W_{N_1}^{n_1 k_1} \, W_{N_2}^{n_2 k_2}$$

Figure by MIT OCW.

True bidimensional transform!
(no extra twiddle factors)

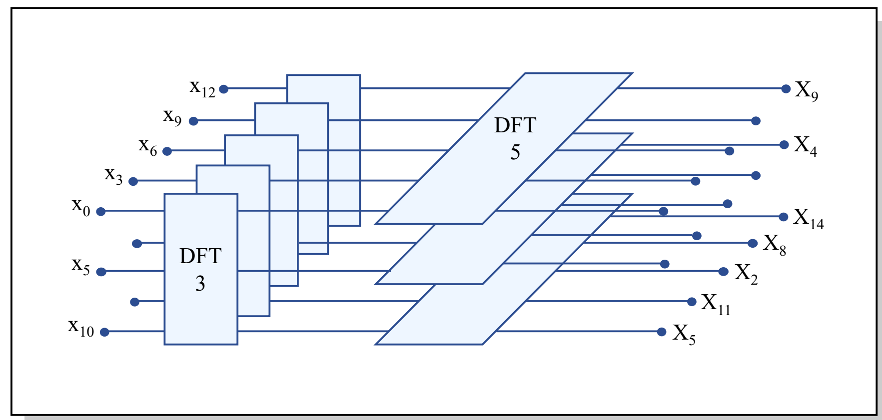


Figure by MIT OpenCourseWare.

Using convolution to compute DFTs

- All sub DFTs are prime length
  - Rader showed that prime-length DFTs can be computed as a result of cyclic convolution
  - E.g. length 5 DFT

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & W_5^1 & W_5^2 & W_5^3 & W_5^4 \\ 1 & W_5^2 & W_5^4 & W_5^1 & W_5^3 \\ 1 & W_5^3 & W_5^1 & W_5^4 & W_5^2 \\ 1 & W_5^4 & W_5^3 & W_5^2 & W_5^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

Permute last two rows and columns

$$\begin{bmatrix} X'_1 \\ X'_2 \\ X'_4 \\ X'_3 \end{bmatrix} = \begin{bmatrix} W_5^1 & W_5^2 & W_5^4 & W_5^3 \\ W_5^2 & W_5^4 & W_5^3 & W_5^1 \\ W_5^4 & W_5^3 & W_5^1 & W_5^2 \\ W_5^3 & W_5^1 & W_5^2 & W_5^4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix}$$

Cyclic correlation
(a convolution with a reversed sequence)

  - This is a general result

Example

- Results in smallest number of multiplies

$$(X'_0, X'_1, \ldots, X'_4)^T = C \cdot D \cdot B \cdot (x_0, x_1, \ldots, x_4)^T,$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & 0 \\ 1 & 1 & -1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & 0 & -1 \\ 1 & 1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

$$D = \mathrm{diag}[1, ((\cos u + \cos 2u)/2 - 1), (\cos u - \cos 2u)/2, -\mathrm{j}\sin u, -\mathrm{j}(\sin u + \sin 2u), \mathrm{j}(\sin u - \sin 2u)],$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Prime Factor Algorithm

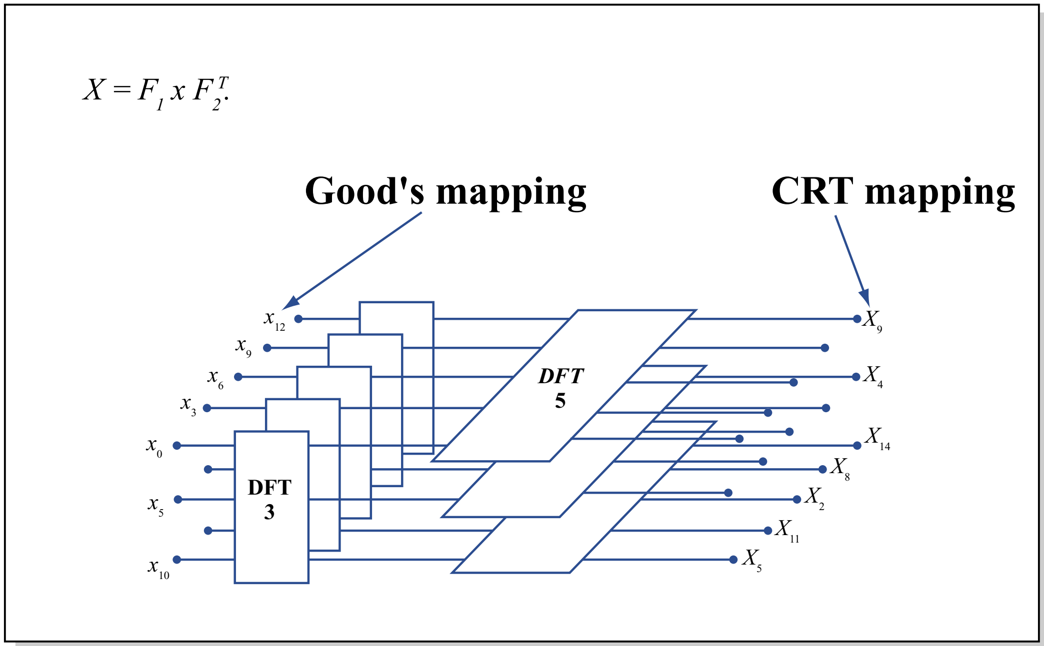


Figure by MIT OpenCourseWare.

- Efficient small DFTs are a key to the feasibility of this algorithm

$$M_{N_1 N_2} = N_1 M_2 + N_2 M_1,$$

$$A_{N_1 N_2} = N_1 A_2 + N_2 A_1,$$

$$m_{N_1 N_2 N_3 N_4} = m_{N_1} + m_{N_2} + m_{N_3} + m_{N_4},$$

$$a_{N_1 N_2 N_3 N_4} = a_{N_1} + a_{N_2} + a_{N_3} + a_{N_4}.$$

Winograd's Fourier Transform Algorithm

$$X = F_1 x F_2^T.$$

$$X = C_1 D_1 B_1 x B_2^T D_2 C_2^T.$$

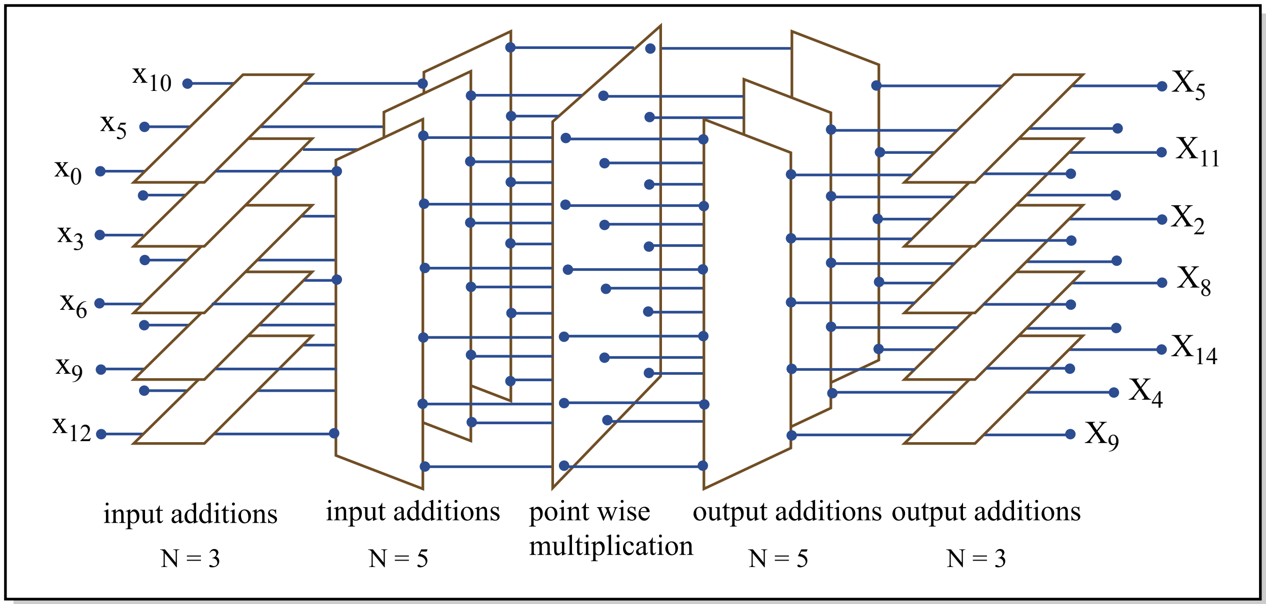


Figure by MIT OpenCourseWare.

- $B_1 x B_2$' only involves additions
- D – diagonal (so point multiply)
- Winograd transform has many more additions than twiddle FFTs