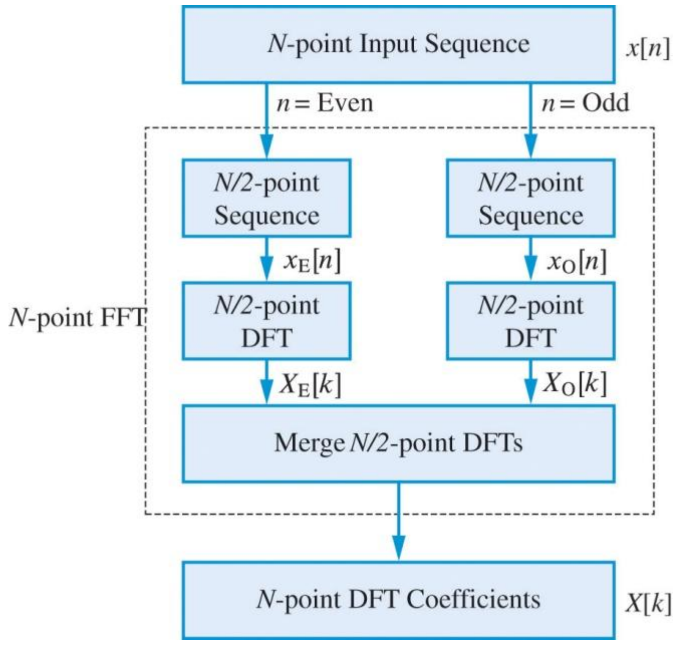
Matlab Code Examples

Direct computation of the DFT

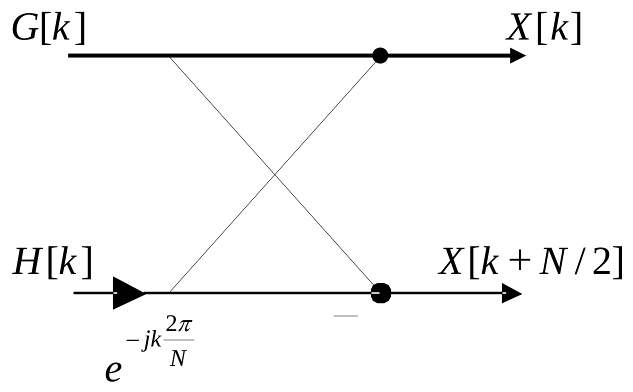
```
function Xdft=dftdirect(x)
% Direct computation of the DFT
N=length(x); Q=2*pi/N;
for k=1:N
      S=0;
      for n=1:N
           W(k,n)=exp(-j*Q*(k-1)*(n-1));
           S=S+W(k,n)*x(n);
      end
      Xdft(k)=S;
end
```

Matlab Code Examples

Recursive computation of the DFT using Decimation & Merge

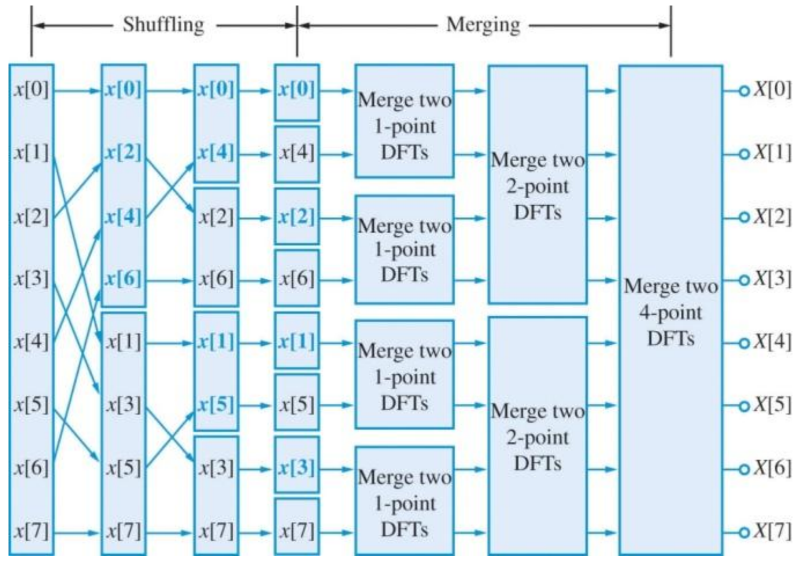


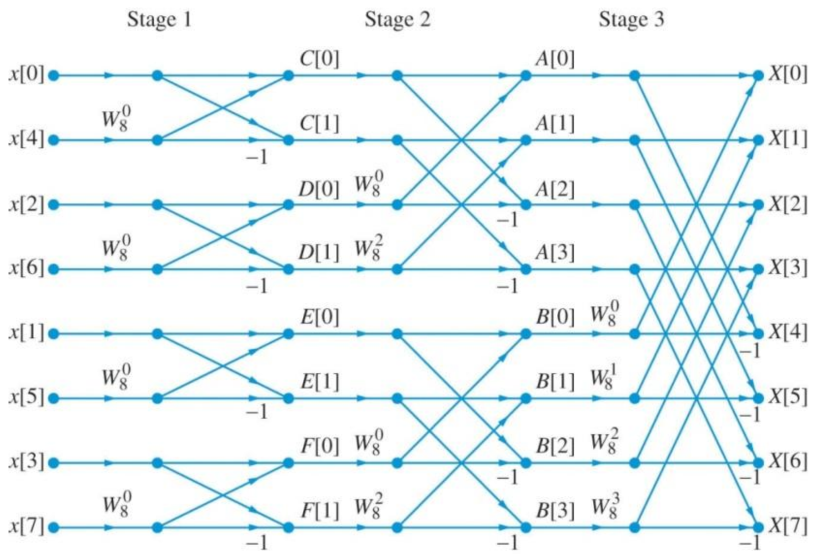
```
function Xdft = fftrecur(x)
% Recursive computation of the DFT using divide & conquer
% N should be a power of 2
N = length(x);
if N ==1
   Xdft = x;
else
        m = N/2;
        XE = fftrecur(x(1:2:N));
        XO = fftrecur(x(2:2:N));
        W = exp(-2*pi*sqrt(-1)/N).^(0:m-1)';
        temp = W.*XO;
        Xdft = [ XE+temp ; XO-temp ];
end
```



Matlab Code Examples

DIT FFT





```
function x=bitrev(x)
% Bit reversal algorithm based on Gold and Rader (1969)
N=length(x); r=0;
for n=0:N-2;
    if n<r % swap samples only for the first half of array
        temp=x(n+1);
        x(n+1)=x(r+1);
        x(r+1)=temp;
    end
    k=N/2; % even n: adds to the previous r;
           %  odd n: subtract from the previous r
    while k <= r
        r=r-k; % keep subtracting reverse carry
        k=k/2; % generate next reverse carry
    end
    r=r+k; % even n: add N/2; odd n: add the last carry.
end
```

```
function x=fftditr2(x)
% DIT Radix-2 FFT Algorithm
N=length(x); nu=log2(N);
x = bitrevorder(x);
for m=1:nu;
    L=2^m;
    L2=L/2;
    for ir=1:L2;
        W=exp(-1i*2*pi*(ir-1)/L);
        for it=ir:L:N;
            ib=it+L2;
            temp=x(ib)*W;
            x(it)=x(it)+temp;
            x(ib)=x(it)-temp;
        end
    end
end
```