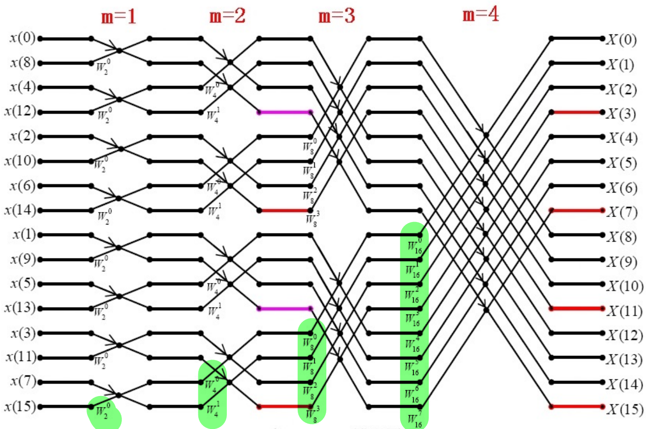
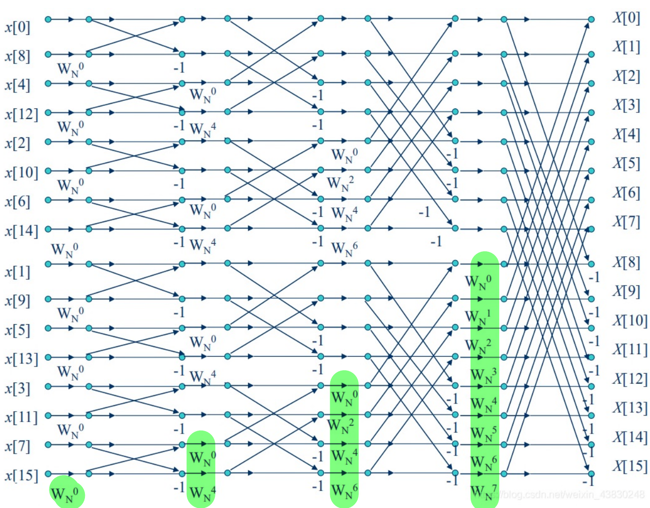
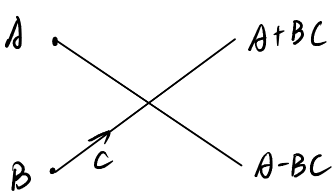
5.1 分别画出 N=4 点的按时间抽取和按频率抽取基 2-FFT 算法的完整流图。



16 点 DIT-FFT 蝶形图



找好规律!



顺箭头为+

逆箭头为-

5.2 设一个 DSP 芯片的运行时钟频率为 100MHz，运行一次复数乘法和复数加法的时间各需要一个时钟周期，计算一个 N=1024 点的基 2-DIT-FFT 算法的总运算时间大约等于多少？若采用 DFT 直接计算，总运算时间是多少？FFT 算法将 DFT 速度提高了多少倍？

	复乘	复加
DFT	N^2	$N(N-1)$
FFT	$\frac{N}{2}\log_2 N$	$N\log_2 N$

	实乘	实加
复乘	4	2
复加	0	2

5.3 设 N=256，一个基 2-DIT-FFT 算法的运算流图总共包含了多少级？每级包含几个蝶形运算单元？设输入存储器容量为 256，地址编号为 0~255，按整序要求，$x(139)$ 的序列值应该存放在第几号地址的存储器单元？

N点 FFT 包含 $\log_2 N$ 级

每级 $\frac{N}{2}$ 个蝶形单元

每个单元一次复乘，两次复加

此外就是涉及码位倒序了

5.4 简要解释基 2-FFT 算法的同址运算特点。

答：同址意为每次运算结果存入原输入数据占用的存储单元，这种利用同一存储单元存储蝶形计算输入输出数据的方法称为同址运算。$N=2^M$ 点的 FFT 共进行 M 级运算，每级由 $N/2$ 个蝶形运算组成，同级中每个蝶形输入数据只对本蝶形有用，每次运算结果存入原输入数据占用的存储单元，整个 N 点的 FFT 运算只需要 N 个存储单元。同址运算使变量寻址更简单，节省内存单元，提高效率，降低成本。

5.5 已知 $x(n)$ 是一个 2N 点有限长实数序列，它的 2N 点 DFT 为 $X(k)$，设计一个用一次 N 点 FFT 完成计算 $X(k)$ 的算法流程，并与直接完成 2N 点 FFT 算法相比，估算计算量节省了多少？

总结得到利用一次N点FFT计算$2N$点实序列的主要步骤如下：

$$x(n)\begin{cases} x_1(n) = x(2n) \\ x_2(n) = x(2n+1) \end{cases} \rightarrow \begin{array}{c} y(n) = x_1(n) + j \cdot x_2(n) \\ \Updownarrow \quad\quad \Updownarrow \quad\quad \Updownarrow \\ Y(k) = X_1(k) + j \cdot X_2(k) \end{array} \quad \text{一次}N\text{点FFT}$$

根据对称性质：

$$\begin{array}{lllll} Re[y(n)] & \Leftrightarrow & Y_{ep}(k) & \rightarrow & X_1(k) = Y_{ep}(k) \\ j \cdot Im[y(n)] & \Leftrightarrow & Y_{op}(k) & \rightarrow & X_2(k) = \dfrac{1}{j} \cdot Y_{op}(k) \end{array}$$

再利用DIT的蝶形图：

$$\begin{cases} X(k) = X_1(k) + W_{2N}^k X_2(k) & k = 0,1,2,...,N-1 \\ X(k+N) = X_1(k) - W_{2N}^k X_2(k) & k = 0,1,2,...,N-1 \end{cases}$$

得到$2N$点FFT变换后的结果。