



# [12] 发明专利说明书

[21] ZL 专利号 98813921.9

[43] 授权公告日 2003 年 5 月 28 日

[11] 授权公告号 CN 1109991C

[22] 申请日 1998.12.18 [21] 申请号 98813921.9

[30] 优先权

[32] 1998.1.21 [33] US [31] 09/010130

[86] 国际申请 PCT/SE98/02393 1998.12.18

[87] 国际公布 WO99/38089 英 1999.7.29

[85] 进入国家阶段日期 2000.9.21

[71] 专利权人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72] 发明人 R·B·赫尔贝里

审查员 张静海

[74] 专利代理机构 中国专利代理(香港)有限公司

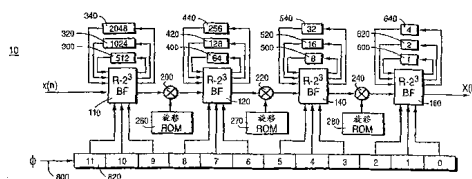
代理人 吴立明 张志醒

权利要求书 6 页 说明书 16 页 附图 10 页

[54] 发明名称 流水线式快速傅里叶变换处理器

[57] 摘要

用离散傅里叶变换(DFT)蝶模块来形成快速傅里叶变换(FFT)处理器,所述蝶模块在最佳示例性实施例均具有大于4的长度。在一第一示例性实施例中,所述FFT处理器使用了长度—8的蝶。在一第二示例性实施例中,所述FFT处理器使用了长度—16的蝶。此外,用低能耗固定系数乘法器来执行各蝶模块中的非简单旋转因子乘法。通过分解简单和非简单旋转因子并利用旋转因子在复数平面中的对称性和/或旋转因子的分解,可减少不同非简单旋转因子乘法的数量。依照上述和其它因子,本发明可用最小的能耗和IC芯片表面面积来构成FFT处理器。



1、一种用最小集成电路芯片面积形成的快速傅里叶变换 (FFT) 处理器，它用于有效地执行低能耗快速傅里叶变换操作，所述处理器  
5 器包括：

一个或多个离散傅里叶变换 (DFT) 模块，它们具有大于 4 的长度，每个 DFT 模块均包括一固定系数乘法器电路，它用于执行旋转因子乘法；

其中，由各 DFT 模块来处理输入数据流，以生成变换后的输出  
10 数据序列。

2、如权利要求 1 的 FFT 处理器，其特征在于，各 DFT 模块的长度为 8。

3、如权利要求 1 的 FFT 处理器，其特征在于，各 DFT 模块的长度为 16。

4、如权利要求 1 的 FFT 处理器，其特征在于，各 DFT 模块中的  
15 旋转因子乘法包括非简单乘法和简单乘法，并且，用固定系数乘法器来执行非简单乘法。

5、如权利要求 4 的 FFT 处理器，其特征在于，在没有执行无操作、改变符号或变换实和虚分量的乘法器的情况下执行所述简单乘法。  
20 法。

6、如权利要求 4 的 FFT 处理器，其特征在于，用相同的固定系数乘法器电路来执行不同的非简单旋转因子值的乘法。

7、如权利要求 4 的 FFT 处理器，其特征在于，将至少一个非简单旋转因子分解成两个分量旋转因子，以减少不同非简单旋转因子  
25 值的数量。

8、如权利要求 4 的 FFT 处理器，其特征在于，用至少两个非简单旋转因子之间的对称关系来减少不同非简单旋转因子值的数量。

9、如权利要求 2 的 FFT 处理器，其特征在于，仅用一个固定系数乘法电路来实现各 8-点 DFT 模块中的所有非简单旋转因子乘法。

10、如权利要求 9 的 FFT 处理器，其特征在于，所述一个固定系数乘法电路包括实和虚分量乘法器，在第一加法器中将实分量乘法器的输出加至虚分量乘法器的输出，在第二加法器中从虚乘法器  
30

输出中减去实乘法器的输出。

11、如权利要求 3 的 FFT 处理器，其特征在于，每个 16-点 DFT 模块均用两个固定系数乘法器电路来执行所有的非简单旋转因子乘法。

5           12、如权利要求 11 的 FFT 处理器，其特征在于，每个 16-点 DFT 模块中的两个固定系数乘法器电路均包括：

          第一固定系数乘法器电路，它具有实和虚分量乘法器，在第一加法器中将实分量乘法器的输出加至虚分量乘法器的输出，在第二加法器中从虚乘法器输出中减去实乘法器的输出；以及

10           第二固定系数乘法器电路，它具有实和虚分量变换器，这些变换器对一个旋转因子乘法来说转换至第一状态，对另一个旋转因子乘法来说转换至第二状态。

          13、一种用 FFT 处理器来计算快速傅里叶变换 (FFT) 的方法，所述处理器是用最小集成电路芯片面积形成的，以使用减少的能耗有效地执行快速傅里叶变换操作，所述方法包括下列步骤：

15           用多个处理模块来形成上述 FFT 处理器，每个处理模块均具有大于 4 的长度，每个处理模块均包括一固定系数乘法器，它用于执行旋转因子乘法；以及

          在各个处理模块中处理输入数据流，以生成变换后的输出数据序列。

20           14、如权利要求 13 的方法，其特征在于，各处理模块的长度为 8。

          15、如权利要求 13 的方法，其特征在于，各处理模块的长度为 16。

25           16、如权利要求 13 的方法，其特征在于，该方法还包括：

          将各处理器模块中的旋转因子乘法分成非简单乘法和简单乘法；以及

          用一个或多个固定系数乘法器来执行非简单乘法。

30           17、如权利要求 16 的方法，其特征在于，在没有执行无操作、改变符号或转换实与虚分量的乘法器的情况下执行简单旋转因子乘法。

          18、如权利要求 16 的方法，其特征在于，该方法还包括：

在各处理模块中用与各个不同非简单旋转因子值乘法相对应的固定系数乘法器电路来执行不同的非简单旋转因子值的乘法。

19、如权利要求 16 的方法，其特征在于，该方法还包括：

5 在各处理模块中用相同的固定系数乘法器电路来执行不同的非简单旋转因子值的乘法。

20、如权利要求 16 的方法，其特征在于，该方法还包括：

将至少一个非简单旋转因子分解成两个分量旋转因子，以减少不同非简单旋转因子值的数量。

21、如权利要求 16 的方法，其特征在于，该方法还包括：

10 用至少两个非简单旋转因子之间的对称关系来减少不同非简单旋转因子值的数量。

22、如权利要求 14 的方法，其特征在于，该方法还包括：

15 仅用一个固定系数乘法电路来实现各 8-点处理模块中的所有非简单旋转因子乘法，以便执行各 8-点处理模块中的所有非简单旋转因子乘法。

23、如权利要求 15 的方法，其特征在于，该方法还包括：

20 仅用两个固定系数乘法电路来实现各 16-点处理模块中的所有非简单旋转因子乘法，以便执行各 16-点处理模块中的所有非简单旋转因子乘法。

24、一种用最小集成电路芯片面积形成的快速傅里叶变换 (FFT) 处理器，它用于有效地执行低能耗快速傅里叶变换操作，所述处理器包括：

一个或多个离散傅里叶变换 (DFT) 模块，每个 DFT 模块均具有三个连在一起的 2-点蝶单元；以及

25 用连接两个 2-点蝶单元的一个固定系数乘法器电路实现的一个或多个旋转因子乘法器；

其中，由上述三个蝶单元来处理输入数据流，以生成变换后的输出数据序列。

25、如权利要求 24 的 FFT 处理器，其特征在于，所述 FFT 处理器是流水线式的，带有流水线寄存器，它位于各蝶单元之间，并且，各个 2-点蝶单元均包括一位于输出与输入之间的反馈通路。

26、如权利要求 25 的 FFT 处理器，其特征在于，所述反馈通路

是一单延迟反馈通路，并且，每个 DFT 模块包括三个反馈寄存器，每个寄存器均包含有不同长度的反馈序列。

27、如权利要求 26 的 FFT 处理器，其特征在于，DFT 模块通过一动态乘法器与旋转因子存储器相连，因此，使来自蝶模块之一的输出在于相邻 DFT 模块中被处理之前乘以从动态乘法器内的旋转因子存储器中检索出来的旋转因子。

28、如权利要求 24 的 FFT 处理器，其特征在于，在不用乘法器电路的情况下执行简单旋转因子乘法。

29、如权利要求 28 的 FFT 处理器，其特征在于，用无操作、用改变符号操作或实与虚分量交换操作来执行简单旋转因子乘法。

30、如权利要求 24 的 FFT 处理器，其特征在于，所述固定系数乘法电路包括实和虚分量乘法器，在第一加法器中将实分量乘法器的输出加至虚分量乘法器的输出，在第二加法器中从虚输出中减去实输出。

31、如权利要求 24 的 FFT 处理器，其特征在于，用相同的固定系数乘法器电路来执行要在各个 DFT 模块中加以执行的所有非简单旋转因子乘法。

32、一种用最小集成电路芯片面积形成的快速傅里叶变换 (FFT) 处理器，它用于有效地执行低能耗快速傅里叶变换操作，所述处理器包括：

一个或多个离散傅里叶变换 (DFT) 模块，每个 DFT 模块均具有四个连在一起的 2-点蝶单元；以及

用一固定系数乘法器电路实现的一个或多个旋转因子乘法器，它们将上述 2-点蝶单元中的两个连接起来；

其中，由上述三个蝶单元来处理输入数据流，以生成变换后的输出数据序列。

33、如权利要求 32 的 FFT 处理器，其特征在于，所述 FFT 处理器是流水线式的，带有流水线寄存器，它位于各蝶单元之间，并且，各个 2-点蝶单元均包括一位于输出与输入之间的反馈通路。

34、如权利要求 33 的 FFT 处理器，其特征在于，所述反馈通路是一单延迟反馈通路，并且，每个 DFT 模块包括四个反馈寄存器，每个寄存器均包含有不同长度的反馈序列。

35、如权利要求 32 的 FFT 处理器，其特征在于，在不用乘法器电路的情况下执行简单旋转因子乘法。

36、如权利要求 35 的 FFT 处理器，其特征在于，用无操作、用改变符号操作或实与虚分量交换操作来执行简单旋转因子乘法。

5 37、如权利要求 32 的 FFT 处理器，其特征在于，第一固定系数乘法电路包括第一实和虚分量乘法器，在第一加法器中将上述第一实分量乘法器的输出加至第一虚分量乘法器的输出，在第二加法器中从虚乘法器输出中减去实乘法器输出。

10 38、如权利要求 37 的 FFT 处理器，其特征在于，第二固定系数乘法电路包括实和虚分量变换器，这些变换器对一个旋转因子乘法来说转换至第一状态，对另一个旋转因子乘法来说转换至第二状态。

39、如权利要求 38 的 FFT 处理器，其特征在于，用所述第一和第二固定系数乘法器电路来执行要在各个 DFT 模块中加以执行的所有非简单旋转因子乘法。

15 40、一种用 FFT 处理器来计算快速傅里叶变换 (FFT) 的方法，所述处理器是用最小集成电路芯片面积形成的，以使用减少的能耗有效地执行快速傅里叶变换操作，所述方法包括下列步骤：

20 用一个或多个处理模块来形成上述 FFT 处理器，每个处理模块均实现按流水线的方式连在一起的三个 2-点蝶单元以及通过一固定系数乘法器电路连接的至少两个蝶单元；

使输入数据流作用于上述三个蝶单元中第一个单元的输入并对该输入数据流进行处理；

在上述三个蝶单元中第二个单元内处理前述第一蝶单元的输出；以及

25 在上述三个蝶单元中第三个单元内处理前述第二蝶单元的输出，以生成输出数据序列。

41、如权利要求 40 的方法，其特征在于，该方法还包括：

用相同的固定系数乘法器来执行不同的非简单旋转乘法。

30 42、如权利要求 41 的方法，其特征在于，用上述固定系数乘法器来执行用于各处理模块的所有非简单旋转因子乘法。

43、一种用 FFT 处理器来计算快速傅里叶变换 (FFT) 的方法，所述处理器是用最小集成电路芯片面积形成的，以使用减少的能量

有效地执行快速傅里叶变换操作，所述方法包括下列步骤：

用一个或多个处理模块来形成上述 FFT 处理器，每个处理模块均具有按流水线的方式连在一起的四个 2-点蝶单元以及通过一固定系数乘法器电路连接的至少两个蝶单元；

5           使输入数据流作用于上述四个蝶单元中第一个单元的输入并对该输入数据流进行处理；

          在上述四个蝶单元中第二个单元内处理前述第一蝶单元的输出；

          在上述四个蝶单元中第三个单元内处理前述第二蝶单元的输出；以及

10

          在上述四个蝶单元中第四个单元内处理前述第二蝶单元的输出，以生成输出数据序列。

44、如权利要求 43 的方法，其特征在于，该方法还包括：

          用相同的固定系数乘法器来执行不同的非简单乘法。

15           45、如权利要求 43 的方法，其特征在于，用两个固定系数乘法器来执行用于各处理模块的所有非简单旋转因子乘法。

## 流水线式快速傅里叶变换处理器

5 本发明涉及一种快速傅里叶变换 (FFT) 处理器, 具体地说, 本发明涉及一种 FFT 流水线式处理器结构, 它优化为最大限度地减少能耗和集成电路面积的要求。

傅里叶变换是一种周知的技术, 它可用于分析随时间变化的信号。简单地说, 傅里叶变换可将信号从随时间变化的格式转换成随  
10 频率变化的格式。逆傅里叶变换进行相反的转变。在用一系列按规则时间周期采样的连续信号样本以离散的形式表示一信号时, 相应的傅里叶变换称为离散傅里叶变换 (DFT)。

从较高层次上说, DFT 是一种简单的算法。该算法包括步骤为输入函数的数字化数据点、用正弦和余弦函数乘各个数据点、在相应的  
15 累加器中对最终的积即用于正弦分量的积和由于余弦分量的积求和。在按这种方式处理了各个数据点时, 用所处理的数据点的数量除正弦和余弦累加器。最终的数值是用于目前正测量的频率的正弦和余弦分量的均值。对所有的整数个频率重复这一过程, 直至频率等于奈奎斯特频率的两倍。

20 更正式地说, 可将 DFT 和逆 DFT 定义成如下:

$$F(f) = \sum_{T=0}^{N-1} f(T) W_N^{fT} \quad (1)$$

$$f(T) = 1/N \sum_{f=0}^{N-1} F(f) W_N^{-fT} \quad (2)$$

其中:

F(f) = 频率分量或变换值

30 f(T) = 时基数据点或逆变换

N=数据点的数量

T=离散的时间



$f$ =离散的频率

$$W_N = e^{-j2\pi/n} = \cos(2\pi/N) - j\sin(2\pi/N) = \text{“旋转因子”}$$

因此，旋转因子是一复数，在一般情况下，频率域和时间域函数均为复数，两个复数值的积会产生以下的项：

$$(A+jB)(C+jD) = AC+jAD+jBC-BD = (AC-BD) + j(AD+BC) \quad (3)$$

项  $(A+jB)$  可例如看作是时间域函数，项  $(C+jD)$  可看作是  $W_N$  即  $W_N = \cos(2\pi/N) + j\sin(2\pi/N)$ 。

DFT 的实际问题是要花很长时间进行计算。事实上，就  $N$  个数据点来说，执行 DFT 需要进行约  $N^2$  次复杂操作。一次复杂操作包括计算出正弦和余弦函数、乘以数据点并将积加起来。这一问题在可能有数十万个数据点要加以“实时”变换的应用中特别麻烦。另一方面，如果减少数据点的数量，则可成平方地减少操作的次数。因此，将数据序列分解成两个相等的部分并分别对各个部分进行处理可以节省一半的计算操作。这是用于推导出快速过傅里叶变换 (FFT) 的方法。将输入数据阵列分成越来越小阵列，以减少计算量，然后，用特征交叉模型来重新组合变换结果，所述特征交叉模型称为“蝶”，它实际是一种小型 FFT。FFT 中的蝶的长度称为 FFT 的“基数” ( $R$ )。因此，如果用多个小的 DFT 例如长度为 2 或 4 的蝶来取代大的 DFT，则可显著地减少复杂操作的次数。即使操作的次数随 DFT 的长度的减小而降低，DFT 长度的减少也要“花费”约  $N$  次操作，因此，对 FFT 来说有常见的  $N \log N$  次计算复杂性。

图 1A 示出了  $N=8$  个数据点阵列  $D_0-D_7$  的实例性 FFT 蝶信号流的图。将所述阵列的偶数分量输入给前 4-点 DFT ( $8$ -点 DFT 的长度的 一半)，将奇数数据点  $D_1, D_2, D_5$  和  $D_7$  输入给后 4-点 DFT。将这两个四点阵列的输出结合起来，以便通过第二次重复由四个频率分量构成的各个集合，然后将偶数和奇数集加到一起，而生成与八点 DFT 相对应的八点序列。但是，在求和之前，奇数 DFT 频率分量必须是异相的，因为，时间域中奇数项会移动一个数据点。可用多个数据块并用增量为  $\pi/4$  的零到  $2\pi$  的弧度范围表示上述相移。

这种分解—求解的方法可扩展成如图 1B 所示，在图 1B 中，将

各个四-点 DFF 分解成两个 2-点 DFT。当然，必须在以后将四个 2-点 DFT 组合成为两个 4-点 DFT，它们如上所述那样被组合成一个 8-点 DFT。这可以再减少整个处理时间几乎一半。

5 因此，可将 8-点 FFT 输入数据分成仅由两个或四个数据点构成的子集，可对这些子集进行二或四点离散傅里叶变换。使变换后的输出乘以适当的“旋转因子”，然后使其经历另一次的二或四点傅里叶变换。

就许多信号处理系统和应用而言，对高速数字信号作实时 FFT 计算是很重要的。异步数字用户专用线路 (ADSL)、数字音频广播 (DAB)、数字视频广播 (DVB)、正交分频多路调制 (OFDM) 是其中一种的多载体调制 (MCM) 方案、声纳、雷达、基于数据块的过滤和快速卷积、分样过滤器库、用于磁性存储器的均衡器、回声消除器以及多路均衡均为高速 FFT 应用的实例。FFT 处理器还例如在数字移动蜂窝式无线电系统中有应用，在这种系统中，要最大限度地减少  
10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000

15 减少能耗和 IC 芯片的大小。能耗越大，就会产生越大的热量。对芯片大小来说，有一个上限，对特定的封装 IC 可使用多少能量来说也有一个上限。减少能耗可使用更便宜的封装 IC 芯片。在这些最重要的因素中，是考虑形成一种诸如 FFT 处理器之类的芯片处理器。

20 尽管例如就实时应用而言处理速度当然很重要，但是，能耗还会随所执行的乘法、加法和寄存器操作的次数而增加。IC 芯片的面积会随象所使用的乘法器、加法器和寄存器之类的硬件组件的数量而增加。本发明的目的是最大限度地减少组件的数量和所执行的操作量，以便最大限度地减少 IC 芯片的面积和能耗。

25 存在有多种不同的方法去提高速度和 / 或最大限度地减少能耗和 IC 芯片面积需求。最成功的方法之一是使所说的处理过程成流水线。流水线式的处理器将计算负荷分成连续的处理级，以便进行并行处理。从本质上说，流水线操作可使得从处理器的前一处理级中获得的局部结果能在没有延迟的情况下被立即地用于下一个处理级。实时的流水线式处理器的处理速度必须与输入数据速率即用于  
30 连续操作的数据采集速度相匹配。这就意味着 FFT 流水线式处理器必须在 N 个时钟周期内计算 N 长度的 DFT，因为，数据采集速度为每个周期一个样本。

以 Shousheng He 名义于 1997 年 5 月 29 日公布的 W097/19412 公开了一种所提出的用于甚大规模积分 (VLSI) 的流水线式 FFT 结构。所提供的这种流水线式 FFT 结构是一种单通路、延迟-反馈 (SDF)、基数-2 的 FFT, 其中, 分解旋转因子以形成基数-4 结构, 基数-2<sup>2</sup> 与基数-4 算法有同样的乘法复杂性, 但保持有基数-2 的蝶结构。

W097/19412 中说明了 He 如何将所有的乘法器分解成简单或非简单乘法器。从结构上说, 图 2A 示出了与 HE 所述相类似的实时流水线式 FFT, 它用于 256 个数据点即  $N = 256$ 。具体地说, 将输入数据序列传给第一对蝶单元 9 和 10。128 个字的反馈寄存器 1 将蝶 9 的输出与其输入连起来。第二蝶单元 10 具有一 64 个字的反馈寄存器 2。乘法器 17 将处理器包括蝶单元 9 和 10 的第一级与处理器包括蝶单元 11 和 12 的第二级连起来并用旋转因子  $W_1(N)$  乘数据流。蝶单元 9、11、13 和 15 的结构分别不同于图 2B 和 2C 所述的蝶单元 10、12、14 和 16。蝶单元 11 和 12 配备有反馈寄存器 3 和 4, 它们分别具有 32 个字和 16 个字的容量。位于处理器第二与第三级之间的乘法器 17 用旋转因子  $W_2(N)$  乘数据流。处理器的第三级包括蝶 13 和 14、8 个字的反馈寄存器 5 以及 4 个字的反馈寄存器 6。位于处理器的第三与第四级之间的乘法器用旋转因子  $W_3(N)$  乘数据流。处理器的第四级包括蝶 15 和 16、两个字的反馈寄存器 7 以及一个字的反馈寄存器 8。输出序和  $X(K)$  源于处理器的第四级的输出。用时钟信号 19 来同步的二进制计数器 18 用作同步控制器和地址计数器, 它可用于在处理器的各级之间所使用的旋转因子。图 2B 所示的 BF2I 型蝶包括两个加法器 21、两个减法器 22 以及四个多路调制器 23。所述多路调制器的操作受控于控制信号 27。图 2C 所示的 BF2II 型蝶在结构上与 BF2I 型蝶相类似, 但包括一  $2 \times 2$  的转换器 26 和逻辑门 24 即带有逆输入的与门。控制信号 25 作用于与门 24 的逆输入。作用于多路调制器 23 的控制信号 27 还作用于与门 24 的非逆输入。来自与门 24 的输出驱动转换器 26。

图 2A 中的基数-2<sup>2</sup> 的单延迟反馈 FFT 处理器的操作如下。在第一个  $N/2$  周期中, 第一蝶模块中 2 至 1 多路调制器 23 转换至位置 “0”, 而蝶是空闲的。来自左面的输入数据被传至反馈位移寄存器, 直至它们被填满。在下一个  $N/2$  周期中, 多路调制器 23 转到

位置“1”，蝶单元用输入的数据和存储在位移寄存器中的数据计算 2-点 DFT。

$$Z1(n) = x(n) + x(n+N/1) \quad (4)$$

$$5 \quad 0 \leq n < N/2$$

$$Z1(n+N/2) = x(n) - x(n+N/2) \quad (5)$$

10 在加载时间序列的下一帧的头一半时，将蝶输出  $Z1(n)$  被传送以施加旋转因子，并将  $Z1(n+N/2)$  传回至要在下一个  $N/2$  周期中加以“多路调制”的位移寄存器。

15 除蝶输入序列的“距离”刚好为  $N/4$  且通过借助转换器 26 和受控加/减操作的实-虚交换来实现简单旋转因子乘法以外，第二蝶的操作与第一蝶的操作相类似。这就需要有来自同步计数器 18 的两位控制信号 25 和 27。然后，数据经过一按 75% 利用率工作的全复数乘法器 17，以逐字地形成基数 4FFT 的第一级结果。下一步的处理过程按在每个连续蝶处理级减半的输入距离重复这一方式。在  $N-1$  个时钟周期之后，按位反序的顺序输出整个的 DFT 变换结果  $X(K)$ 。然后，在不暂停的情况下对变换的下一帧进行处理，因为在处理器的各级处有流水线式的处理。

20 HE 的 W097/19412 申请书认为，这种基数-2<sup>2</sup>SDF FFT 处理器结构对流水线式 FFT 计算来说是最佳的。但是，用本发明可使得 FFT 处理器的 IC 面积和能耗有更大的减少。

25 使用多个小型 DFT（即 FFT，分解—求解的原理）的大型 DFT 的计算是一种可用迭代或流水线式结构来实现的多级处理过程。尽管这种分解—求解策略节省了计算，但是，存在有在较少的处理级之间执行的量有所增加的复数旋转因子乘法。从在各蝶处执行的旋转因子乘法是简单的角度来看，希望有两点或四点 DFT/蝶，因为，乘法器的系数仅为  $\pm 1$  或  $J$ 。换句话说，对二或四点 DFT 中的各个变换来说，不需要复数乘法电路，而只需要“简单”乘法。在不用乘法器的情况下仅通过传递数据而无操作或通过改变符号或通过转换实和虚分量来执行简单乘法。换句话说，可避免非简单、计算费力的乘法。但是，对 FFT 中 2-点与 4-点 DFT 蝶级之间的旋转因子乘法

30

来说，需要多次非简单复数乘法。

通常的想法是，当 FFT 的基数（即基本 DFT 计算单元的长度）增加至 8、16 或更大（即在分解求解法中将变换分成 8、16 或更多的分支）时，从每个高基数蝶（高基数 FFT 中的蝶）中所必须执行的非简单乘法的角度看，计算“成本”会显著地增加，从而会抵消通过蝶之间的数量有所减少的旋转因子乘法器所获得的收益。本发  
5 明人发现，情况并不一定是这样。与通常的想法相反，本发明用较少数量的低功率固定系数的乘法器在基数大于 4 的 FFT 中提供了一种非常有效且 IC 面积效率高的 FFT 处理器。利用某些旋转因子关系  
10 （以下将详细加以说明）可以仅使用较少量的固定系数乘法器。

因此，本发明涉及较高基数的快速傅里叶变换（FFT）处理器，同时能在有最小能耗的情况下仅使用最小的集成电部芯片面积去有效地执行快速傅里叶变换操作。最佳的是，本发明可用于任何基数大于 4 的 FFT 结构。示例性实施例使用了蝶模块，就实时的、流水  
15 线式 FFT 处理器结构而言，所述蝶模块具有 8 或 16 的长度。就基数-8 的实现形式而言，用基数- $2^3$  蝶处理器模块来形成上述 FFT 处理器。就基数-16 的实现形式而言，用基数- $2^4$  蝶处理器模块来形成上述 FFT 处理器。

在按基数- $2^3$  所实现的基数-8 示例性实施例中，用三个按流水线方式连在一起的 2-点蝶单元来实现各个蝶模块。输入数据序列作用于这三个蝶单元之一的输入并经过这三个流水线式蝶单元的处理，以生成经过傅里叶变换的数据序列。每个蝶单元均包括一个  
20 的延迟反馈寄存器。在基数-8 蝶模块所需的这三种非简单旋转因子乘法（在蝶中这三种非简单旋转因子乘法均被执行多次）中，本发明仅用一个固定系数乘法器电路实现了上述三种乘法。在按基数- $2^4$  所实现的基数-16 示例性实施例中，用四个按流水线方式连在一起的 2-点蝶单元来实现各个蝶模块。在基数-16 蝶模块所需的这九种非简单旋转因子乘法（在蝶中这九种非简单旋转因子乘法均被执行多次）中，本发明仅用两个固定系数乘法器电路实现了上述九  
25 种乘法。  
30

以下连同附图说明本发明的上述特点和目的（以及其它特点和

目的), 附图中:

图 1A 是一快速傅里叶变换 (FFT) 蝶的流程图, 其中, 用两个 4-点 DFT 来执行 8-点 DFT 操作;

图 1B 是一双蝶 FFT 流程图, 它用于实现 8-点 DFT;

5 图 2A 是就  $N = 256$  的基数- $2^2$  单个延迟反馈 (SDF) 流水线式 FFT 处理器结构;

图 2B 是图 2A 所示的基数- $2^2$  SDF 流水线式 FFT 处理器中使用的第一个 2 点蝶结构的图;

10 图 2C 是图 2A 所示的基数- $2^2$  SDF 流水线式 FFT 处理器中使用的第二个蝶结构的图;

图 3 是依照本发明一个示例性实施例的基数- $2^3$  单通路延迟反馈实时流水线式 FFT 处理器的图;

图 4 是示出了在实现图 3 所示的基数- $2^3$  FFT 时使用的基数- $2^3$  蝶;

15 图 5 是示出了复数平面中用于基数-8 和基数-16 DFT 蝶模块的旋转因子的向量图;

图 6 是用固定系数乘法器所实现的图 4 所示基数- $2^3$  蝶的图;

图 7 示出了用于实现  $W_8^1$  旋转因子乘法的一个示例性固定系数乘法器;

20 图 8 是依照本发明另一个示例性最佳实施例的基数- $2^4$  单通路延迟反馈实时流水线式 FFT 处理器的图;

图 9A 是与图 8 中基数- $2^4$  流水线式 FFT 处理器的各个处理级中所使用的 16-点 FFT 相对应的基数- $2^4$  蝶模块的图;

25 图 9B 是示出了不用乘法器实现的图 9A 中所示的简单旋转因子乘法的图;

图 9C 是示出了由多个非简单的基数-16 旋转因子乘法构成的固定系数乘法器的实现形式, 它可用于图 9A 所示的基数- $2^4$  蝶;

图 10 示出了一示例性固定系数乘法器, 它实现了用于多种基数-16 旋转因子的乘法;

30 图 11 是一曲线图, 它示出了用不同基数 FFT 处理器实现的 4096-

点 FFT 的 FFT 能耗;

图 12 是一曲线图, 它示出了用不同基数 FFT 处理器实现的 4076FFT 的 FFT 集成电路 (IC) 的面积消耗。

5 在以下的说明中, 为说明起见而不是为了进行限制, 说明了特定的细节, 诸如特定的实施例、技术、硬电路、装置等, 以便能对本发明加以全面的理解。例如, 本发明的最佳示例性实施例是用于基数 $-2^3$ 和基数 $-2^4$ 流水线式 FFT 处理器的。但是, 这些实施例和结构仅是示例性的, 本技术的专家可以看出, 可用不脱离这些特定细节的其它实施例来实现本发明。在其它的情况下, 略去了对周知 DFT / FFT 方法、装置和技术的详细说明, 以避免因不必要的细节妨碍了对本发明的说明。

10 如上所述, 对 DFT / FFT 计算的通常想法是, 用基数-8 或基数-16FFT 处理器什么都不能得到, 因为, 计算量不显著地降低。FFT 领域中的某些专家还认为, 分解-基数 FFT 是最有效的, 因为, 就计算长度为  $2^n$  的 DFT 而且, 它使用了周知的最少次数的非简单操作。与通常的想法及这些论点相反, 本发明人发现, 用较高基数 FFT 处理器具体说是基数-8 和基数-16 的 FFT 处理器、用固定系数乘法器并用旋转因子之间的某些优势关系去减少固定系数乘法器的实际数量, 以实现必要的旋转因子乘法, 可在降低能耗和集成电路 (IC) 芯片表面面积方面获得显著的收益。的确, 尽管就 IC 芯片面积或能量要求而言对较高基数的简单外推不一定导致更高的效率, 但本发明用复数平面中的旋转因子对称、旋转因子的分解以及固定系数乘法器去实现 IC 芯片和能耗方面的显著高的效率。

25

#### 示例性基数-8FFT 处理器

30 用诸如图 3 所示的实例之类的用于  $N=4096$  数据点阵列的基数 $-2^3$ 单通路延迟反馈 (SDF) 流水线式 FFT 处理器结构来最佳地实现本发明基数-8FFT 处理器的第一个示例性实施例。图 3 中的四个 8-点蝶中的每个蝶均由三个流水线式 2-点蝶构成。将输入数据序列  $x(n)$

输入给第一基数 $-2^3$ 蝶模块 110。来自各个基数 $-2^3$ (8)BF 模块的输出是输入给上述模块的 8-点 FFT, 其中, 频率分量 0-7 按位颠倒的次序 0、4、2、6、1、5、3、7 输出。该输出与乘法器 200 相连, 以便与第一旋转因子乘数  $W_{4096}^{n*k}$  相乘, 其中, 就各个  $n$  和从 0 至  $(4096 / 8 - 1) = 511$  的  $k$  值而言,  $n$  为按位颠倒次序 (见上) 的 0 至 7。因此, 将 4096 个旋转因子值存储在旋转 ROM260 中, 所述因子值的约 1/8 (512) 是简单的。

旋转积作用于第二基数 $-2^3$ 蝶模块 120, 在乘法器 220 中使输出序列与第二旋转因子乘数  $W_{512}^{n*k}$  相乘, 其中, 就各个  $n$  和从 0 至  $(4096 / 8^2 - 1) = 63$  的  $k$  值而言,  $n$  为按位颠倒次序的 0 至 7。将 4096 个旋转因子值存储在旋转 ROM270 中, 所述因子值的约 1/8 是简单的。

在第二基数 $-2^3$ 蝶模块 140 中处理旋转积。在乘法器 240 中使输出与第三旋转因子乘数  $W_{64}^{n*k}$  相乘, 其中, 就各个  $n$  和从 0 至  $(4096 / 8^3 - 1) = 7$  的  $k$  值而言,  $n$  为按位颠倒次序 (重复 64 次) 的 0 至 7, 将所述旋转因子值存储在旋转 ROM280 中。在最后一个基数 $-2^3$ 蝶模块 160 中处理旋转积, 以形成经傅里叶变换的输出序列  $X(K)$ 。

各蝶模块 110、120、140 和 160 均包括三个反馈通路。具体地说, 蝶模块 110 包括: 第一反馈通路, 它具有一 2048-字的反馈寄存器; 第二反馈通路, 它具有一 1020-字的反馈寄存器; 以及, 第三反馈通路, 它具有一 512-字的反馈寄存器。蝶模块 120 包括: 第一反馈通路, 它包括有一 256-字的寄存器 440; 第二反馈通路, 它包括有一 128-字的反馈寄存器 420; 以及, 第三反馈通路, 它包括有一 64-字的反馈寄存器 400。第三蝶模块 140 包括: 第一反馈通路, 它包括有一 32-字的寄存器 540; 第二反馈通路, 它包括有一 16-字的反馈寄存器 520; 以及, 第三反馈通路, 它包括有一 8-字的反馈寄存器 500。蝶模块 160 包括: 第一反馈通路, 它包括有一 4-字的寄存器 640; 第二反馈通路, 它包括有一 2-字的反馈寄存器 620; 以及, 第三反馈通路, 它包括有一 1-字的反馈寄存器 600。二进制计数器 820 由时钟信号 800 来同步。该二进制计数器用作一同步控制器和地址计数器, 它用于 FFT 处理器的各级之间所使用的全部因子。各反馈寄



存器均提供延迟的 X 样本，因此，可在不同时间对数据进行处理。

蝶模块 110、120、140 和 160 中的每一个均包括三个流水线式 2-  
点蝶单元 (BF2)。每个蝶模块均可实现基数-8 (即 8-点) FFT 算法，  
5 但该基数-8 蝶被“分解”成三个级联的基数-2 蝶单元。蝶模块 160  
例如包括一第一蝶单元 160A，它与第二蝶单元 160B 相连，第二蝶单  
元 160B 又与第三蝶单元 160C 相连。每个蝶单元 160A-160C 的内部  
结构均与图 2A 所示的相似。另一种蝶单元的实现形式可使用图 2B  
所示的结构，以使  $j$  和  $-1$  与实/虚位移及加/减位移相乘。但是，  
也可以用其它方法来实现简单乘法。

10 每个蝶单元均由同步控制器和地址计数器分别用时标脉冲 2、1  
和 0 来加以同步。每个蝶单元均包括一单通路反馈延时循环和寄存  
器。蝶单元 160A 包括一四字反馈寄存器 64。蝶单元 160B 包括一个  
2-字反馈寄存器 620，蝶单元 160C 包括一个 1-字反馈寄存器 600。

图 4 可推广用于使用长度为  $4X$ 、 $2X$  和  $X$  的反馈寄存器的任何基  
15 数- $2^3$  蝶单元，其中， $X=1、8、64、512$  等。序列或系数是在数据  
经过图 4 所示的乘法器时要乘以的值。所说的序列在变成下一个值  
之前被处理  $X$  次。 $X$  的值取决于将基数-8 蝶放置在流水线链路中的  
什么位置处。就最右侧的模块 160 而言， $X=1$ ；就最左侧的模块 110  
而言， $X=512$ ；就模块 120 而言， $X=64$ 。

20 参照图 1A 所示的蝶结构，多个蝶单元的输出仅乘以包括  $+1$ 、 $-1$ 、  
 $+j$ 、 $-j$  的简单乘数。但是，也可以用无操作、符号变化或实/虚分  
量“交换”来实现上述简单乘法 (无乘法器)。用蝶模块之间的动  
态乘法器 (如图 2A 中的动态乘法器) 来实现非简单旋转因子乘法 (如  
 $W1(n)$ 、 $W2(n)$ 、 $W3(n)$ )。

25 因此，通常的 FFT 处理器用动态乘法器来进行旋转因子乘法。  
动态乘法器从下述意义上来说是“动态的”即：如果该动态乘法器  
执行输入数据字“A”与系数字“B”之间的乘法，则字“B”的数字  
值可改变成任何实际值。虽然就输出序列乘以不同旋转因子系数值  
而言是灵活的，但是，尽管可用较小的 IC 面积来实现动态乘法器，  
30 但动态乘法器会消耗相当量的能量。在固定系数乘法器中，字“B”  
的数字值是恒定的。

实践中，通过使要相乘的多种形式的两个二进制字位移然后将

位移后的形式加在一起，从而实现二进制的乘法。由于对固定系数乘法器来说上述数据字之一是周知的，所以，乘法器不必执行任何其它乘法。结果，固定系数乘法器仅包括动态乘法器中所使用的硬件的约一半。此外，可将周知的数据字记录成非常有效的格式，因此，与动态乘法器相比，可显著地减少固定系数的能耗。固定系数乘法器的缺点在于，对各种不同系数而言，需要一独立的乘法器。因此，随不同的固定系数乘法器的数量的增加，会消耗 IC 芯片的表面面积。

随蝶模块的长度增加至 8 和 16，蝶模块之间的旋转因子乘法器的数量会减少。但是，经过乘法器的必须乘以旋转因子的数据的百分比会增加。还有，在各蝶模块中执行的旋转因子乘法的次数也会增加。就基数-4FFT 而言，经过各个四点蝶模块之间的 75% 的数据点必须乘以旋转因子。就基数-8FFT 而言，所述百分比会增加至 87.5%，就基数-16FFT 而言，所述百分比会增加至 93.75%。

蝶模块之间的动态乘法器所执行的旋转因子乘法的减少了的次数和各蝶模块中用于基数有所增加的 FFT 的旋转因子乘法次数的增加的净效应是所执行的乘法总数有较少的减少。可用本发明的方法来减少较高基数 FFT 的蝶模块的中使用的不同旋转因子值的个数。在给定了各蝶模块中执行的较少数量的不同值、非简单旋转因子乘法的情况下，本发明可用比动态乘法器消耗更少能量的固定系数乘法实现了这些乘法。随着各蝶模块中执行的不同值、非简单旋转因子乘法的次数的增加，从所消耗的有所增加的 IC 芯片面积角度看，固定系数乘法器在这种情况下会变得不太有吸引力。

本发明可通过(1)利用旋转因子在复数平面内的对称性以及(2)分解旋转因子来减少各蝶模块中执行的不同值、非简单旋转因子乘法的次数。以下连同图 5 所示的旋转因子向量图来说明上述两种技术。

在复数平面内可将旋转因子看作是具有数量“1”和特定相位角的向量。因此，将向量乘以旋转因子等价于将该向量旋转弧度为  $2\pi n/N$  的相位角。图 5 中可以看出，就基数-8 蝶模块而言，旋转因子  $W_8^0$  对应于简单的实值“1”。旋转因子  $W_8^2$  对应于简单的虚值“-j”。

旋转因子  $W_8^1$  对应于非简单的复数值  $1/\sqrt{2} (1-j)$ 。旋转因子  $W_8^3$  对应于非简单的复数值  $1/\sqrt{2} (-1-j)$ 。注意  $W_8^1$  与  $W_8^3$  之间的对称性。唯一的差别是，它们的实数分量的符号是不同的。利用上述  
5 对称关系，相同的乘法器电路例如  $W_8^1$  可用来执行用于  $W_8^1$  与  $W_8^3$  两者的非简单复数旋转因子乘法，对上述因子中的一个而言仅需要改变符号。

另外，可按如下方式分解旋转因子  $W_8^3$ ： $W_8^3 = W_8^1 \cdot W_8^2$ 。可用同  
10 样的  $W_8^1$  乘法器电路来实现旋转因子  $W_8^3$ ，随后是用符号改变和实与虚分量交换来实现的简单乘法即  $W_8^2 = -j$ ，通过利用旋转因子对称性或者通过将非简单旋转因子分解成共用因子，可减少专门的固定系数乘法器电路的数量，从而进一步将较高基数 FFT 处理器所需的 IC 芯片面积的数量减至最小。

因此，上述第一示例性实施例中的本发明使用了基数-8 蝶模块，  
15 该模块用固定系数乘法器来执行各基数-8 蝶模块中的非简单旋转因子。而且，利用旋转因子的对称或分解，可用最少的固定系数乘法器硬件来实现非简单旋转因子乘法。如前所述，每个基数-8 蝶均需要与旋转因子  $W_8^1$ 、 $W_8^2$  和  $W_8^3$  的旋转因子乘法。不用乘法器执行旋  
20 转因子  $W_8^3$  乘法，因为该乘法是简单的，也就是说，可用改变符号和实与虚分量交换来实现与  $-j$  的乘法。用相同的固定系数乘法电路来实现  $W_8^1$  和  $W_8^3$  旋转因子乘法。

图 6 示出了用简单乘法和单个固定系数乘法器电路实现的图 4  
25 的基数-2<sup>3</sup> 蝶模块 160。可根据所示的用于各旋转因子处理级 650 和 660 的旋转因子序列将适当的部件接入信号处理通路。处理级 650 包括一固定系数乘法器和简单旋转因子乘法。通过仅将信号处理通路连到所示的两个节点之间，可实现乘法“1”。当蝶单元 160A 的输出需要乘以非简旋转因子  $W_8^1$  时，就将固定系数乘法器 700 接入上述信号处理通路。这种接入操作可受控于同步控制器 820 通过对控  
30 制器号进行解码并因此而对开关进行控制而产生的时标信号。处理级 660 和 670 包括可开关的处理通路，它们可用于乘以“1”的简单

乘法和框 720 所示的乘以 $-j$ 的乘法。

图 7 示出了用于旋转因子  $W_8^1$  的固定系数乘法器的一种有效实现形式。记住  $W_8^1=1/\sqrt{2}(1-j)$  和  $W_8^3=1/\sqrt{2}(-1-j)$ 。尽管将图 6 中的信号处理通路显示为一信号线，但是，本技术的专家应该认识到，数据序列是包括实 (I) 和虚 (Q) 分量的复数。因此，图 7 示出了 I 和 Q 信号处理通路。因此，用两个比两个动态乘法器消耗更少能量的固定系数乘法器 800 和 820 来实现 I 和 Q 信号乘以  $1/\sqrt{2}$  的两种乘法。实与虚分量在加法器 840 中相加并输出，同时在用于  $W_8^1$  的减法器 860 中从虚分量中减去实分量。同样的乘法器电路可用于旋转因子  $W_8^3$  乘法，并对实分量改变符号。

#### 示例性基数 $-2^4$ FFT 处理器

图 8 示出了本发明的第二实例性最佳实现形式。图 8 中的实时流水线式 FFT 处理器 100 是除 FFT 操作以基数 $-16$ FFT 算法为基础而不是以基数 $-8$  算法为基础以外都与图 3 中基数 $-2^3$ SDF 结构相类似的基数 $-2^4$  单通路延迟反馈 (SDF) 处理器结构。将  $N=4096$  点的输入数据序列  $X(N)$  输入给第一基数 $-2^4$  蝶模块 1020。来自各个  $R-2^4(16)$  BF 模块的输出是输入给上述模块的 16-点 FFT，其中，频率分量按位颠倒的次序输出。该输出与乘法器 1100 相连，以便与第一旋转因子乘数  $W_{4096}^{n*k}$  相乘，其中，就各个  $n$  和从 0 至  $(4096/16-1)=255$  的  $k$  值而言， $n$  为按位颠倒次序 (见上) 的 0 至 15。因此，将 4096 个旋转因子值存储在旋转 ROM1120 中，所述因子值的约  $1/16(256)$  是简单的。

在第二基数 $-2^4$  蝶模块 1040 中处理旋转积，并在乘法器 1140 中将该旋转积乘以第二旋转因子乘数  $W_{256}^{n*k}$ ，其中，就各个  $n$  和从 0 至  $(4096/16^2-1)=15$  的  $k$  值而言， $n$  为按位颠倒次序的 0 至 15，它重复 6 次。将这些旋转因子值存储在旋转 ROM1160 中。在最后一个基数 $-2^4$  蝶模块 1060 中处理旋转积，以形成经变换的序列  $X(K)$ 。

基数 $-16$  蝶模块 1020 包括四个 2-点 FFT 蝶，每个蝶均具有一单

反馈通路和相应的反馈寄存器 1200、1220、1240 和 1260。蝶模块 1040 包括四个反馈寄存器 1300、1320、1340 和 1360。蝶模块 1060 包括四个反馈寄存器 1400、1420、1440 和 1460。FFT 处理器根据一同步和地址生成器 1520 所接收的时钟信号 1500 进行操作，所述同步和地址生成器 1520 对各蝶模块以及各模块中的各蝶单元的操作进行控制。

图 9A 说明了图 8 所示的基数 $-2^4(16)$  蝶单元中的一个。四个 2-点 FFT 蝶 1060a、1060b、1060c 和 1060d 按所示的流水线方式连在一起。蝶单元 1060A 包括一存储有  $8X$  字的单通路反馈寄存器 1460，其中，就蝶模块 1060 而言， $X=1$ ；就蝶模块 1040 而言， $X=16$ ；就蝶模块 1020 而言， $X=256$ 。蝶单元 1060B 包括一  $4X$  字反馈寄存器 1440，蝶单元 1060C 包括一  $2X$  字反馈寄存器 1420，蝶单元 1060D 包括一  $X$  字反馈寄存器 1400。除它们的反馈先进先出 (FIFO) 存储器有相应不同长度以外，上述四个 2-点蝶单元可以是相同的。

基数 $-16$  蝶需要下列旋转因子乘法： $W_{16}^1$ 、 $W_{16}^2$ 、 $W_{16}^3$ 、 $W_{16}^4$ 、 $W_{16}^5$ 、 $W_{16}^6$ 、 $W_{16}^7$ 、 $W_{16}^8$ 、 $W_{16}^9$ 。图 5 将这些旋转因子显示为复数平面内的向量。通过按下述方式分解这些基数 $-16$  旋转因子，可以简化旋转因子乘法：

$$W_{16}^3 = W_{16}^{-1} \cdot W_{16}^4, \text{ 其中, } W_{16}^4 = -j$$

$$W_{16}^9 = W_{16}^1 \cdot W_{16}^8, \text{ 其中, } W_{16}^4 = -1$$

$$W_{16}^2 = W_8^1, W_{16}^6 = W_8^3 = W_8^1 \cdot W_{16}^2.$$

因此，基数 $-16$  蝶中的某些旋转因子本身或它们的分解旋转因子分量仅为简单乘法从而不用乘法器来加以实现。此外，如图 5 所示， $W_{16}^1$  和  $W_{16}^{-1}$  相对实轴是对称的，因此，可通过改变虚分量的符号，从  $W_{16}^1$  中生成  $W_{16}^{-1}$ 。所以，可用相同的固定系数乘法器来实现  $W_{16}^1$  和  $W_{16}^{-1}$  旋转因子乘法。就  $W_{16}^2$  和  $W_{16}^6$  而言，所需的另一个专门固定系数乘法器是用于实现非简单旋转因子乘法的乘法器。如前所述，可用诸如图 7 所示的乘法器之类的单个  $W_8^1$  乘法器来实现上述两种旋转因子乘法。结果，可用刚好两个固定系数乘法器电路来实

现基数-16 蝶模块的所有旋转因子乘法，而其余的乘法 / 操作则是简单的例如是改变符号、乘以 1 以及实与虚分量交换。

5 参照图 9A-9C，蝶单元 1060a 和 1060b 以及 1060c 和 1060d 通过为 1 和  $-j$  的简单旋转因子乘法相连。图 9B 示出了标号 1600 和 1680 处的简单旋转因子乘法。蝶单元 1060b 与 1060c 通过两个非简单旋转因子乘法器 1620 和 1640 以及一简单复数旋转因子乘法器 1660 相连。图 9C 示出 3 如何用固定系数乘法器来实现上述非简单旋转因子乘法。不是用相应的固定系数乘法来实现各类型的基数-16 旋转因子乘法，而是用旋转因子的对称性和分解来将固定系数乘法器的数量减少至仅为两个。

10 图 10 说明了这种有效的实现形式，其中，固定系数乘法器以可切换的方式受控，以实现非简单旋转因子乘法。I 和 Q 通道被提供给 IQ 变换器 1720，该变换器受控于同步控制器生成的控制信号 1880。如果旋转因子乘法是  $W_{16}^1$ ，则 IQ 变换器 1720 和 IQ 变换器 1820（也受控于信号 1880）使 I 和 Q 数据直接经过，或者，如果旋转因子乘法是  $W_{16}^{-1}$ ，则 IQ 变换器 1720 和 IQ 变换器 1820 如图所示那样切换 I 和 Q 通路。在相应的固定系数乘法器 1740 和 1760 中处理来自 IQ 变换器 1720 的 I 和 Q 通路输出，以实现乘以两个不同固定数的乘法、 $A-B$  和  $A+B$ （A 是实分量，B 是虚分量）。根据控制信号 1880 在 IQ 变换器 1820 中处理来自部件 1740 和 1760 的输出，而 I 和 Q 的输出则提供给加法器 1840 和 1860。在减法器 1780 中减去原始输出的 I 和 Q 信号，并在固定系数乘法器 180 使上述 I 和 Q 信号乘以虚分量 B。根据控制信号 1880 的状态，加上来自部件 1800 的输出或者从输入的 I 和 Q 信号中减去上述输出。因此，可用图 10 所示的同样的三个乘法器 1740、1760 和 1800 来执行用于非简单旋转因子  $W_{16}^x$  的若干值的乘法。

30 图 11 和 12 所示的曲线图说明了本发明的优点。图 11 示出了 4096 点数据序列的 FFT 处理器的能耗与包括基数 2、4、8 和 16 在内的 FFT 基数长度的曲线图。使用大于 4 的基数、利用旋转因子的对称性和 / 或旋转因子的分解并使用固定系数乘法器，本发明能获得非常低的能耗，同时还能使 IC 面积消耗变低。

具体地说，如果全都使用动态乘法器，那么，通过将 FFT 处理器的基数增加到基数-4 以上，就基数-8 而言可以将能耗减少 12%，就基数-16 而言可以将能耗减少 5%。如果使用了专用的动态乘法器，则依照本发明第一示例性实施例，就基数-8FFT 处理器而言可获得最小的能耗。如果将专用的动态乘法器用于基数-16FFT 处理器，则就基数-8 而言能耗实际会增加而不是减少。但是，如果在 FFT 处理器中使用固定系数乘法器而不是动态乘法器，可以显著地节省 IC 面积和能耗。依照本发明，如果在所说的蝶中使用了固定系数乘法器，则基数-16FFT 处理器具有所有四种所说明的基数的最佳的最低能耗和非常低的面积消耗，（只有基数-8FFT 处理器有略低的面积消耗）。

尽管连同目前认为是最实用和最佳的实施例说明了本发明，但是，应该认识到，本发明并不局限于所公开的实施例，相反，本发明用于涵盖包括在后附权利要求的精神和范围内的各种改进形式和等价结构。例如，固定系数乘法器、旋转因子的对称性和 / 或旋转因子的分解可用于这样的 FFT 处理器，这种处理器具有除 8 或 16 以外长度的蝶单元。尽管示例性实施例使用了流水线式处理结构，但是，本发明也可应用于其它的 FFT 处理器结构例如迭代结构，所以，本发明并不局限于特定的 FFT 处理器结构。

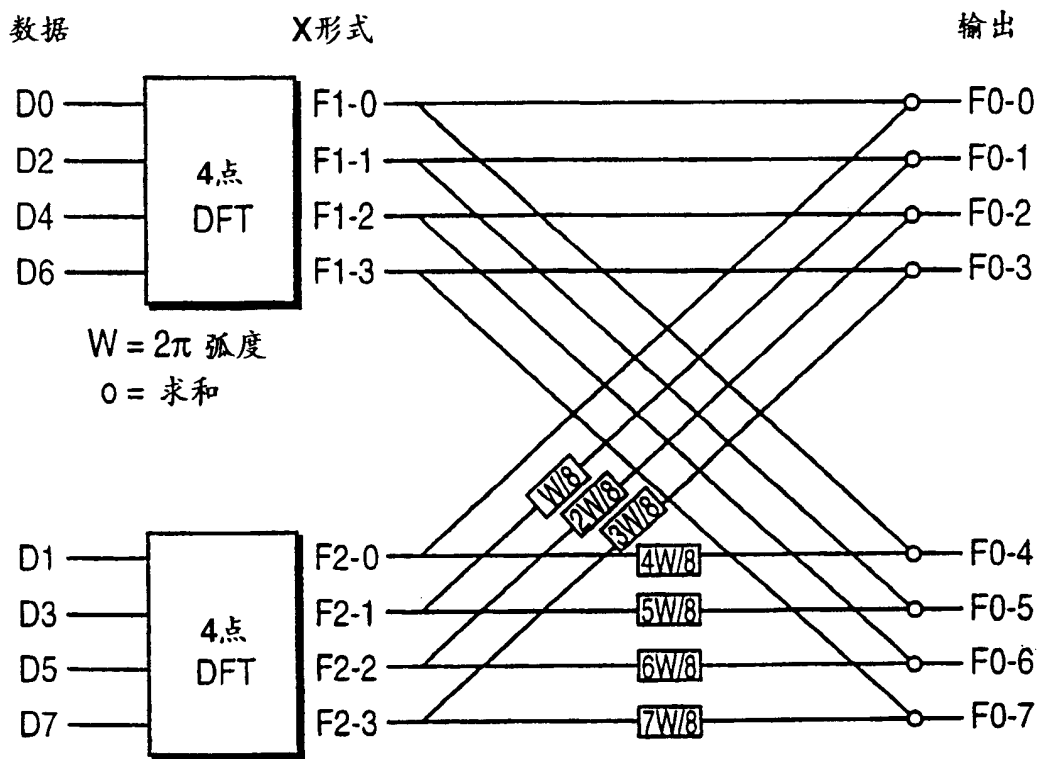


图 1A



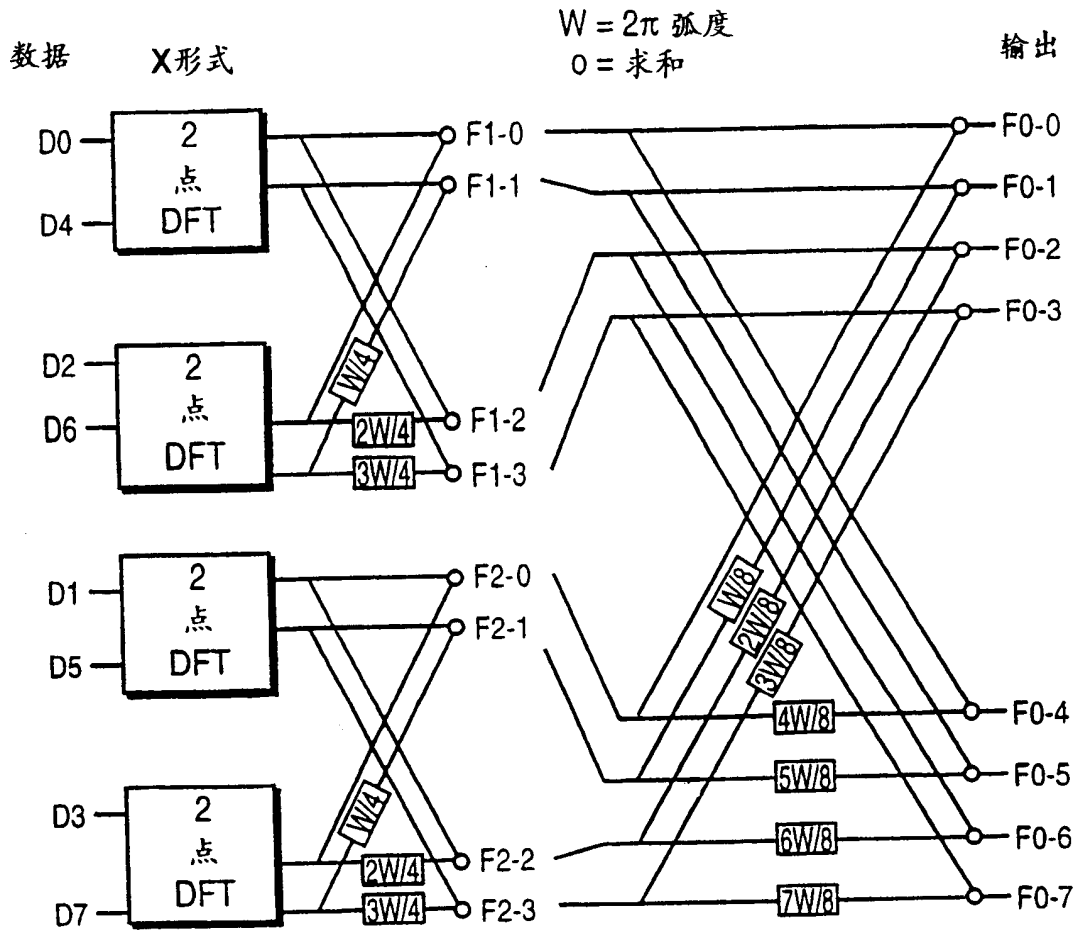


图 1B

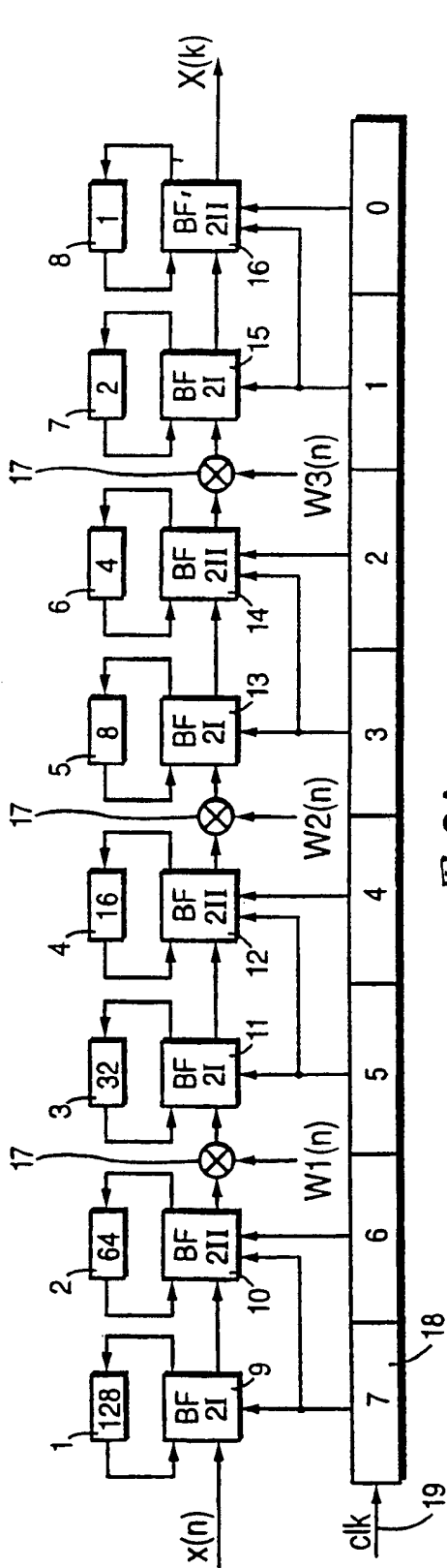


图 2A

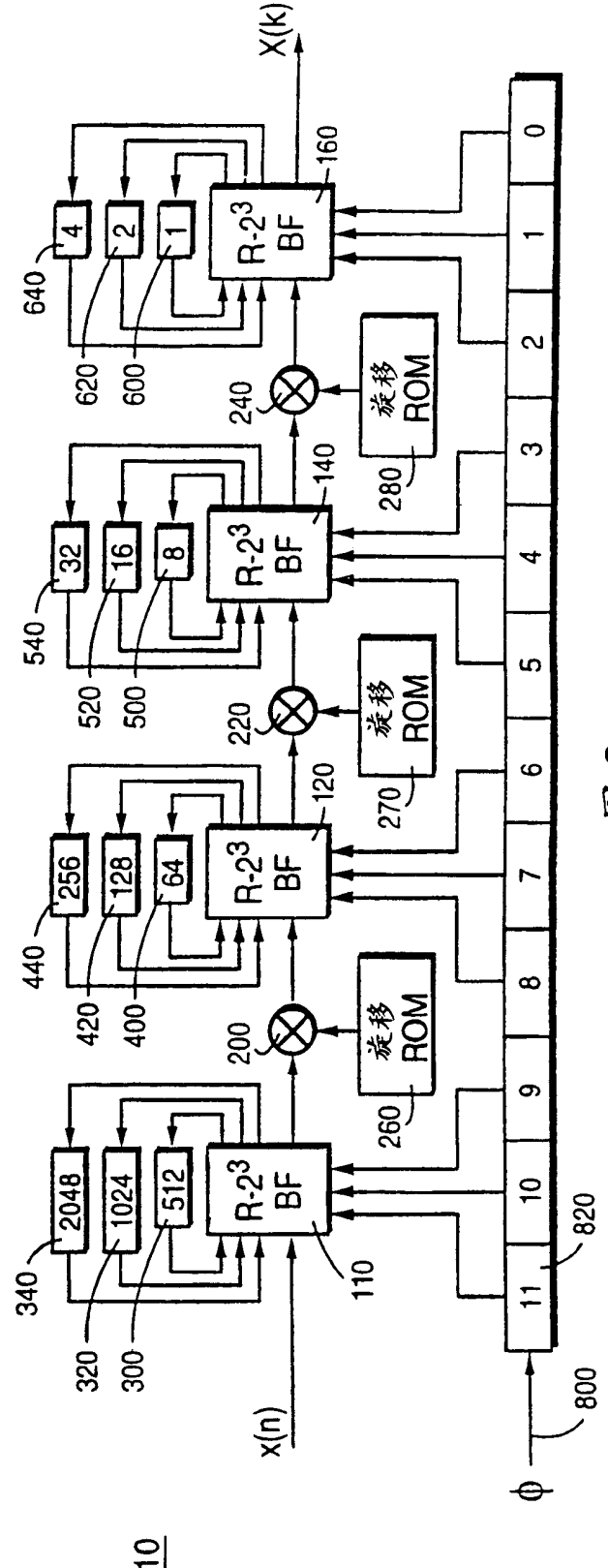
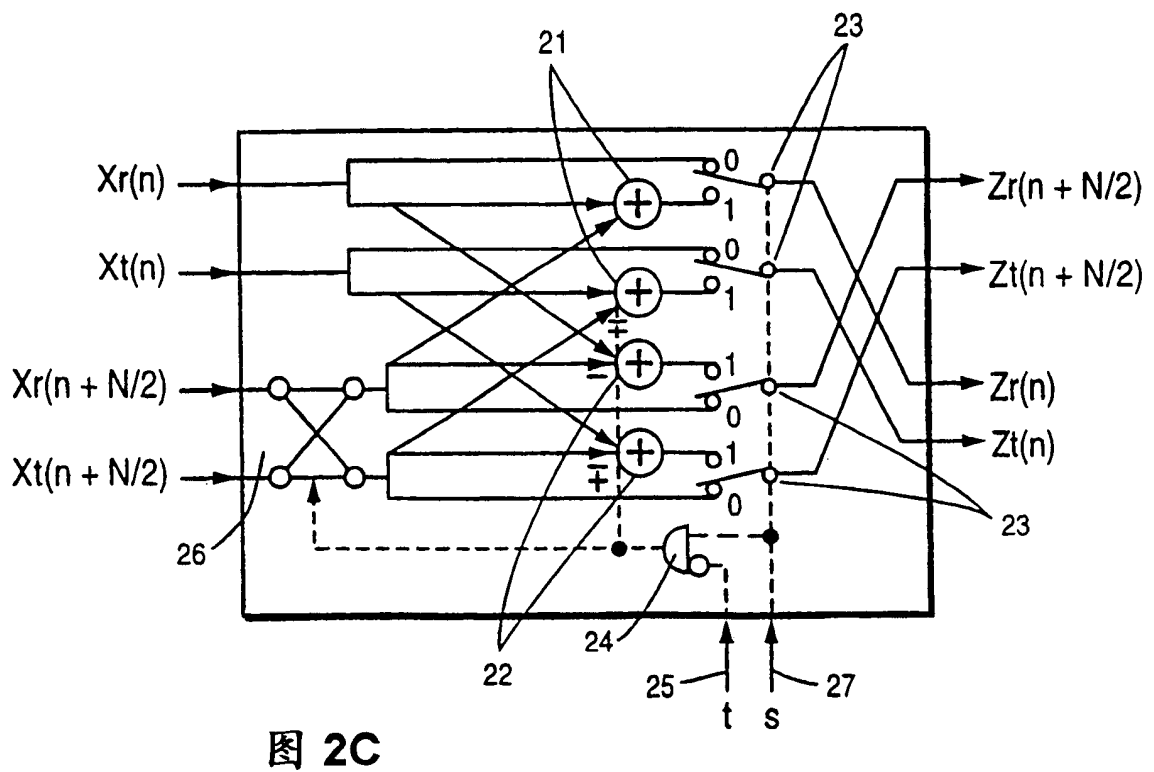
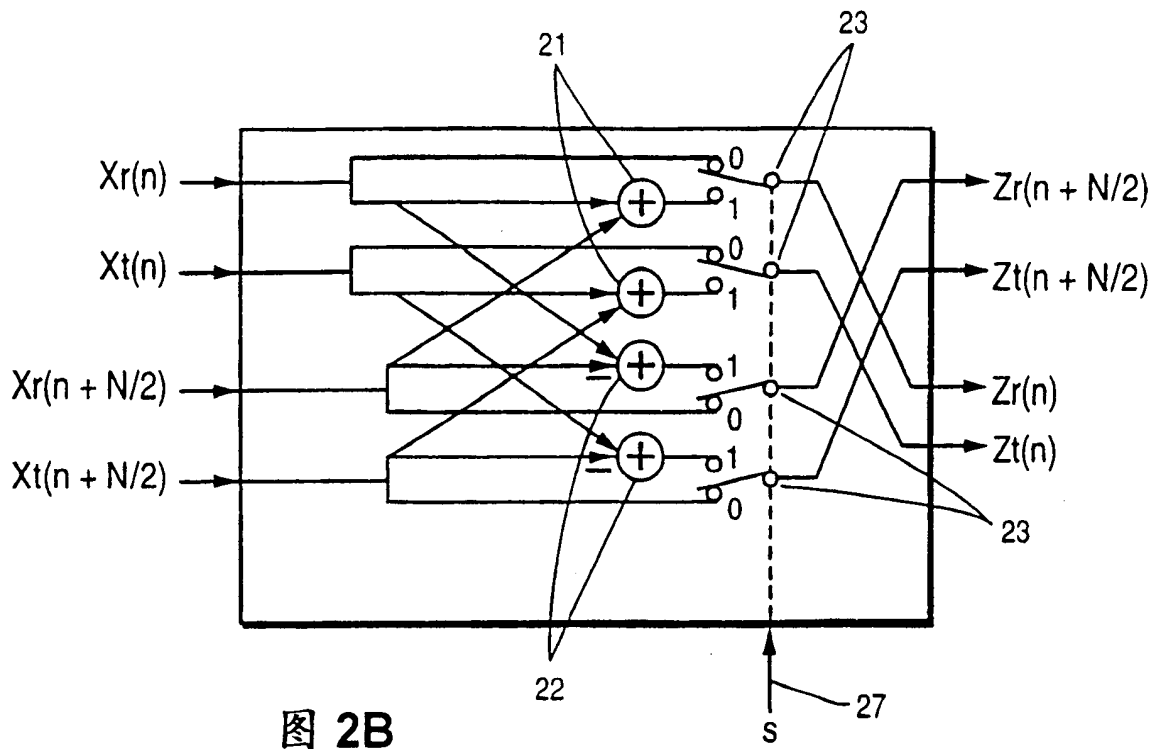


图 3



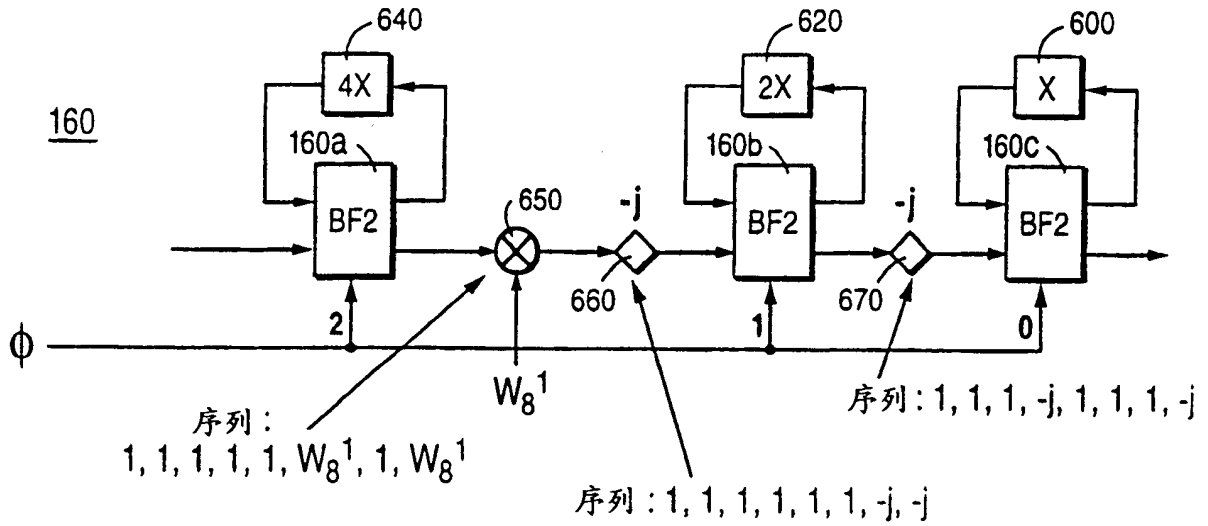


图 4

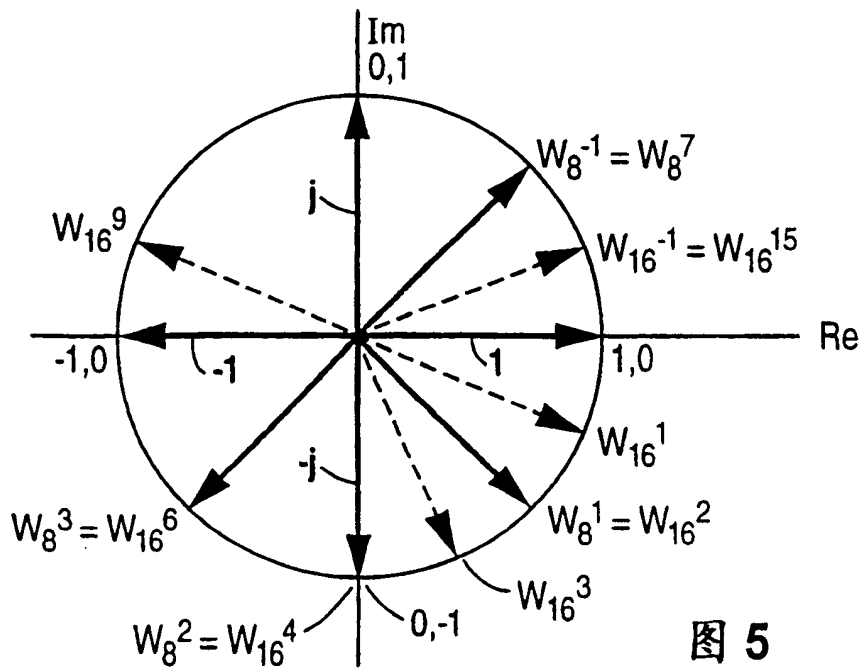
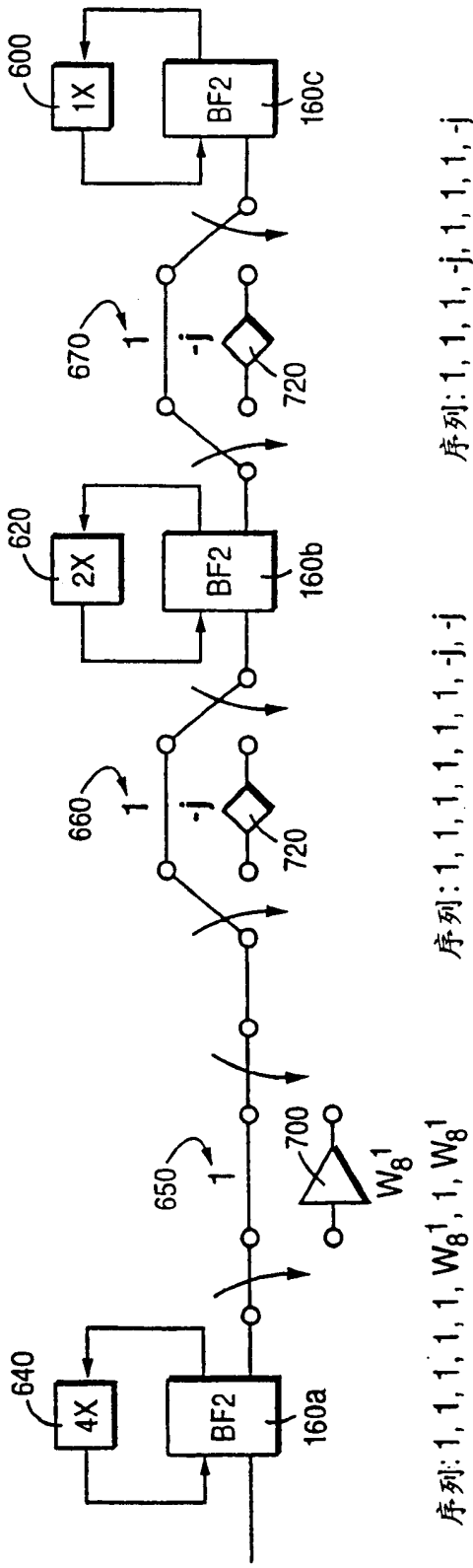


图 5



基数 =  $2^3$   
N = 8

图 6

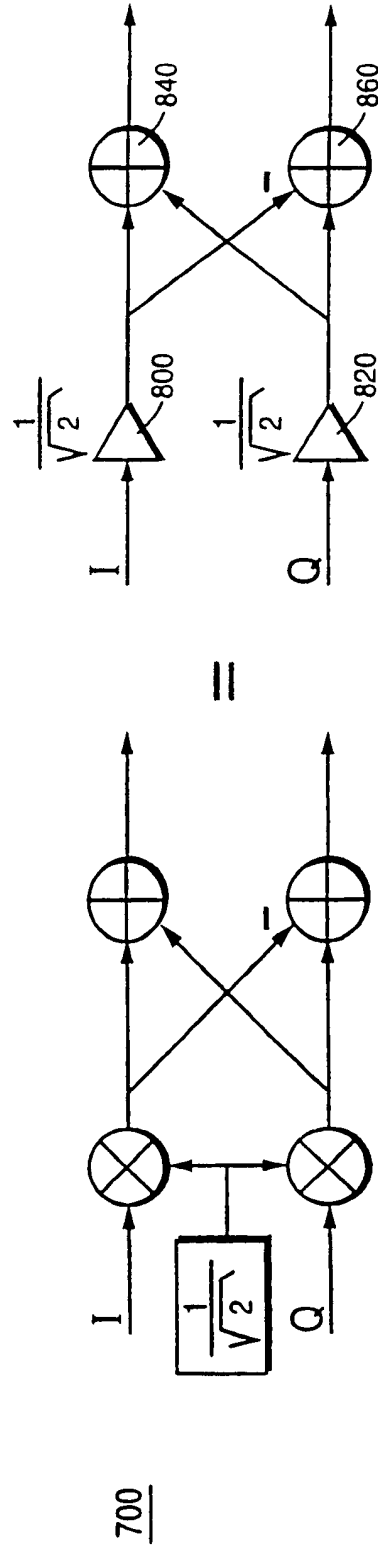


图 7

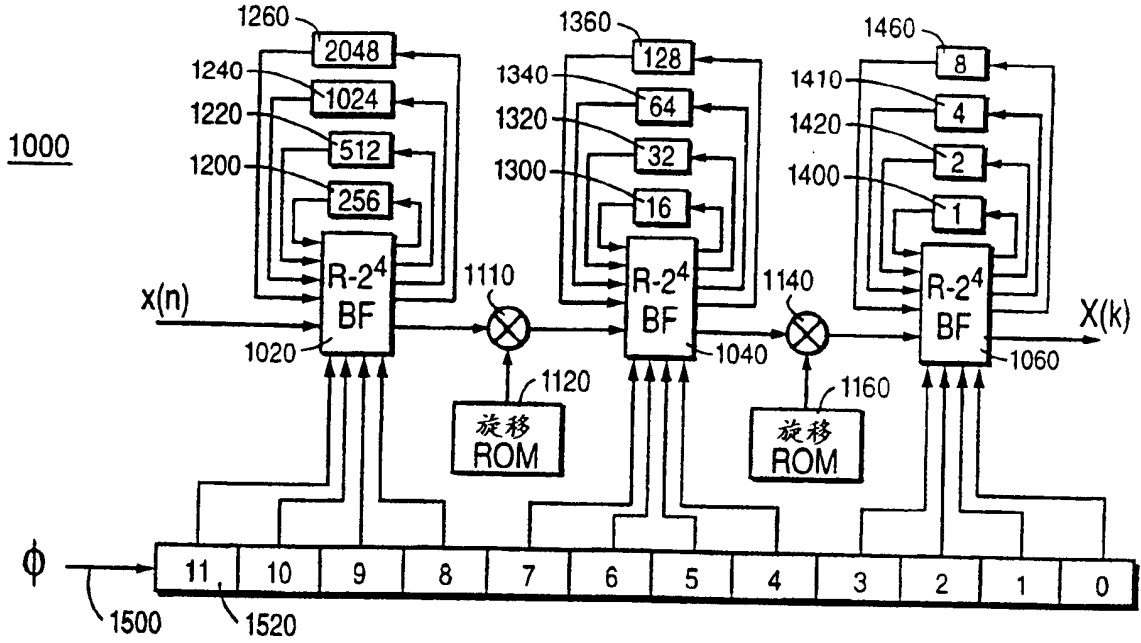


图 8

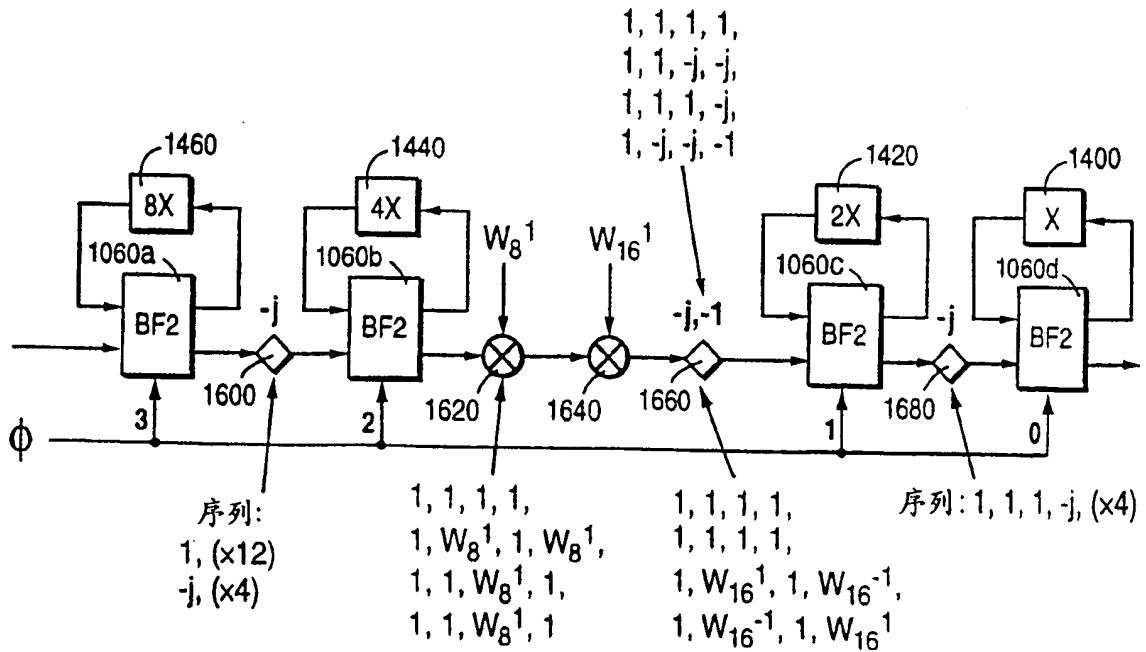


图 9A

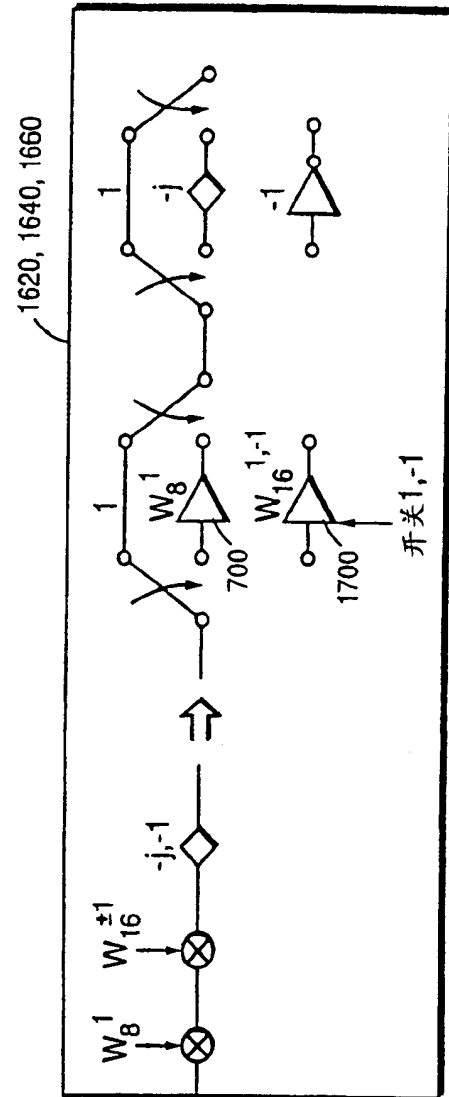


图 9C

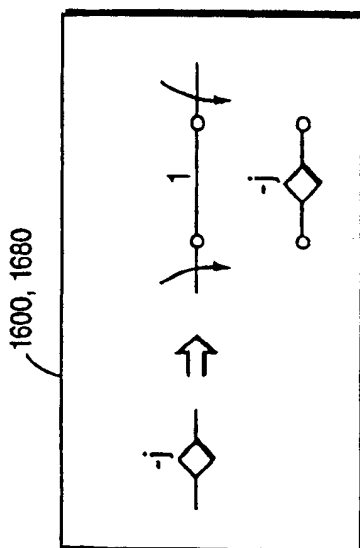


图 9B

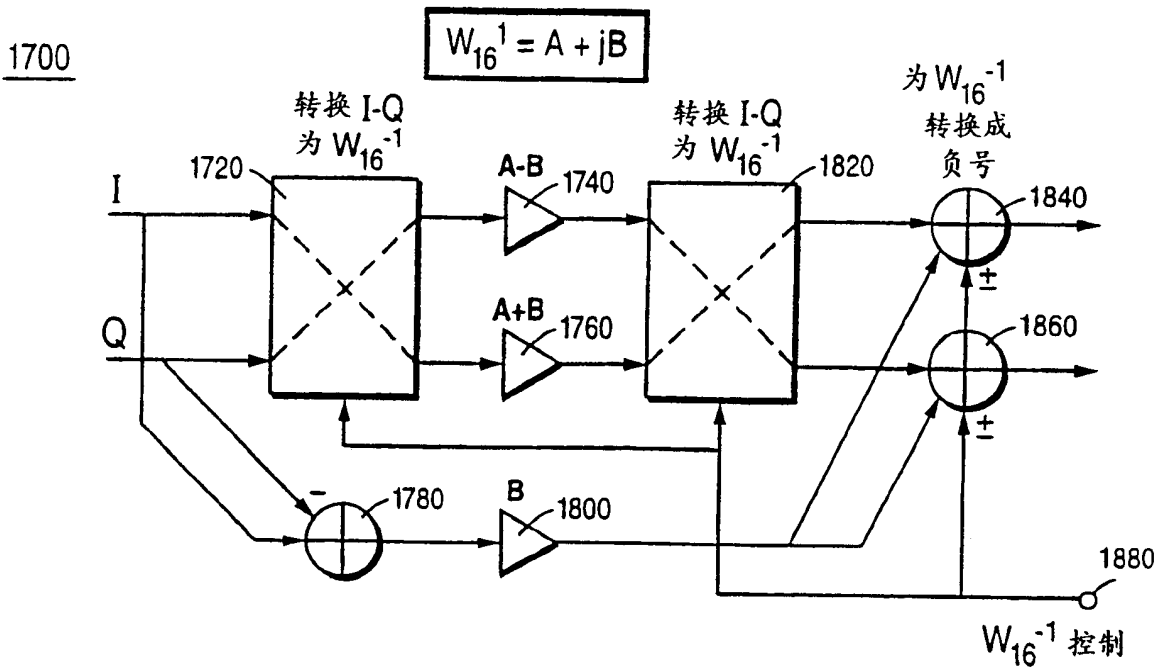


图 10

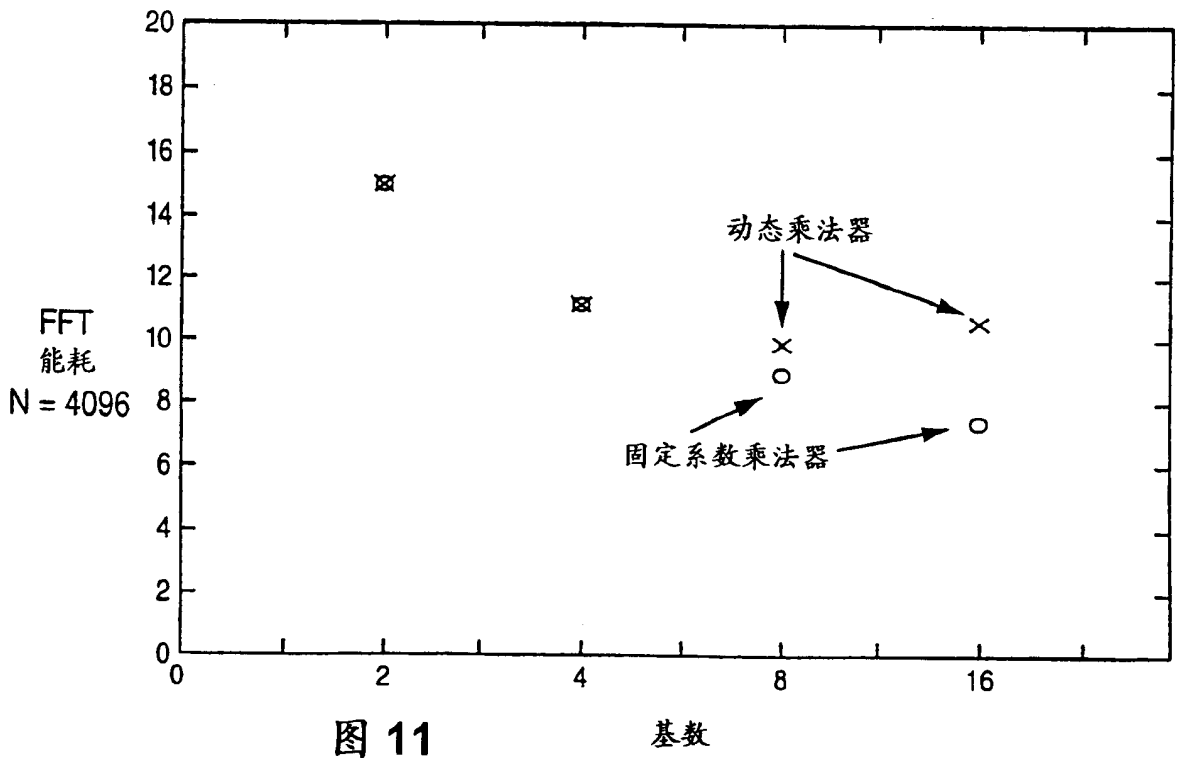


图 11



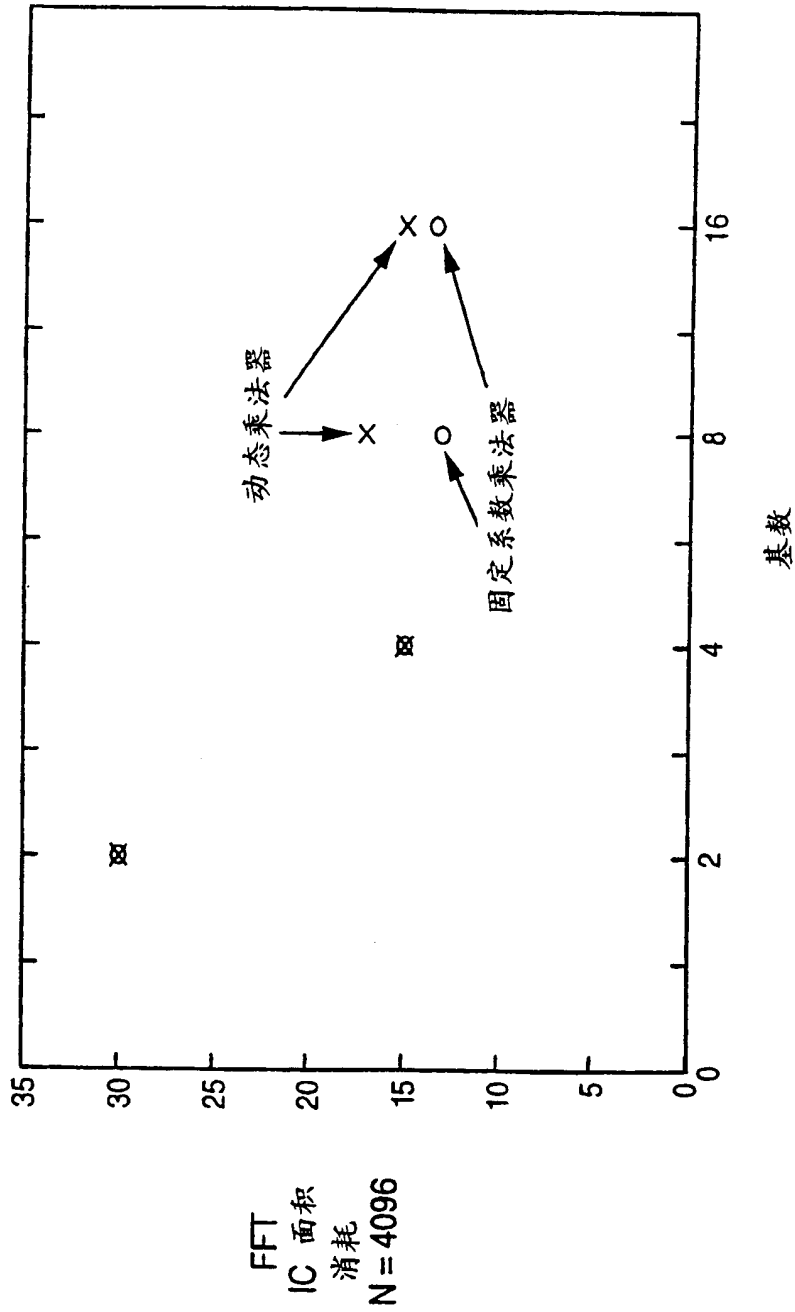


图12