(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 106970895 B
(45)授权公告日 2023.10.03

(21)申请号 201610024296.X

(22)申请日 2016.01.14

(65)同一申请的已公布的文献号
申请公布号 CN 106970895 A

(43)申请公布日 2017.07.21

(73)专利权人 普天信息技术有限公司
地址 100080 北京市海淀区海淀北二街6号普天大厦

(72)发明人 王纪宁

(74)专利代理机构 北京路浩知识产权代理有限公司 11002
专利代理师 李相雨

(51)Int.Cl.
G06F 17/14(2006.01)
G06F 9/355(2006.01)

(56)对比文件
CN 205486097 U,2016.08.17
CN 102063411 A,2011.05.18
CN 102342071 A,2012.02.01
CN 102611667 A,2012.07.25
CN 103226543 A,2013.07.31
CN 103493039 A,2014.01.01
US 8612505 B1,2013.12.17

审查员 杜克奎

权利要求书2页 说明书10页 附图2页

(54)发明名称

基于FPGA的FFT装置及方法

(57)摘要

本发明涉及一种基于FPGA的FFT装置及方法，该装置包括：缓存模块、控制模块和基-4蝶形运算器；所述控制模块分别与缓存模块和基-4蝶形运算器相连，用于控制数据的输入、输出，用于控制数据以乒乓缓存的方式缓存至缓存模块中，用于控制数据以循环寻址的方式在基-4蝶形运算器中完成FFT运算；所述缓存模块用于初始输入前3/4的数据，输出后3/4的运算结果，并用于保存中间结果；所述基-4蝶形运算器用于初始输入后1/4的数据，输出前1/4的运算结果。本发明提高了运算速度，在DSP乘法器使用数量与应用基-2蝶形运算单元的FFT装置相当的情况下减少了存储数据的RAM的总深度。

CN 106970895 B

1.一种基于FPGA的FFT装置的FFT方法,其特征在于,包括:

顺序输入第一帧数据,完成第一帧数据的1级蝶形运算后,采用乒乓缓存顺序输入第二帧数据,并完成第一帧数据的M级蝶形运算;

完成第一帧数据的蝶形运算结果的顺序输出,同时进行第二帧数据的缓存及蝶形运算;

完成第二帧数据的M级蝶形运算,同时采用乒乓缓存进行第三帧数据的缓存,并开始进行第三帧数据的1级蝶形运算;

不断重复数据的缓存、蝶形运算和结果输出过程,完成多帧数据的蝶形运算;

其中,M为蝶形运算的级数,N为FFT运算的点数,$N=4^M$;数据读取和存储采用循环寻址方式;

所述基于FPGA的FFT装置包括:

缓存模块、控制模块和基-4蝶形运算器;

所述控制模块分别与缓存模块和基-4蝶形运算器相连,用于控制数据的输入、输出,用于控制数据以乒乓缓存的方式缓存至缓存模块中,用于控制数据以循环寻址的方式在基-4蝶形运算器中完成FFT运算;

所述缓存模块用于初始输入前3/4的数据,输出后3/4的运算结果,并用于保存中间数据;

所述基-4蝶形运算器用于初始输入后1/4的数据,输出前1/4的运算结果。

2.根据权利要求1所述的FFT方法,其特征在于,所述顺序输入第一帧数据,完成第一帧数据的1级蝶形运算后,采用乒乓缓存顺序输入第二帧数据,并完成第一帧数据的M级蝶形运算;完成第一帧数据的蝶形运算结果的顺序输出,同时进行第二帧数据的缓存及蝶形运算包括:

顺序输入前3/4的第一帧数据至缓存模块的第一部分,当后1/4的第一帧数据到达基-4蝶形运算器时,根据蝶形运算图直接与缓存模块中的数据进行蝶形运算,并将1级蝶形运算的结果保存至缓存模块的第一部分;

完成第一帧数据的M级蝶形运算,基-4蝶形运算器顺序输出第一帧数据的蝶形运算结果的前1/4,后3/4的运算结果保存至缓存模块的第一部分;采用乒乓缓存顺序输入前3/4的第二帧数据至缓存模块的第二部分,当后1/4的第二帧数据到达基-4蝶形运算器时,根据蝶形运算图直接与缓存模块中的数据进行蝶形运算;

缓存模块的第一部分顺序输出第一帧数据的蝶形运算结果的后3/4;

相应地,所述缓存模块的数据读取和存储采用循环寻址方式。

3.根据权利要求1所述的FFT方法,其特征在于,所述循环寻址方式包括:

进行1级蝶形运算,将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中;

进行中间级的蝶形运算,按照循环寻址方式读取缓存模块中的数据,将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中;

进行最后一级的蝶形运算,按照循环寻址方式将蝶形运算结果保存至缓存模块中,依次读取缓存模块中的数据并顺序输出蝶形运算结果。

4.根据权利要求3所述的FFT方法,其特征在于,所述进行1级蝶形运算,将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中包括:

进行1级蝶形运算，将1级蝶形运算结果分成16组，将所述16组蝶形运算结果的第0-3组数据依次存入第一RAM、第二RAM、第三RAM和第四RAM；将所述16组蝶形运算结果的第4-7组数据依次存入第二RAM、第三RAM、第四RAM和第一RAM；将所述16组蝶形运算结果的第8-11组数据依次存入第三RAM、第四RAM、第一RAM和第二RAM；将所述16组蝶形运算结果的第12-15组数据依次存入第四RAM、第一RAM、第二RAM和第三RAM。

5.根据权利要求4所述的FFT方法，其特征在于，所述进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据，将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中；进行最后一级的蝶形运算，按照循环寻址方式将蝶形运算结果保存至缓存模块中，依次读取缓存模块中的数据并顺序输出蝶形运算结果包括：

进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第一端口、第二端口、第三端口和第四端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第二端口、第三端口、第四端口和第一端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第三端口、第四端口、第一端口和第二端口；按照循环寻址方式读取缓存模块的数据输入到基-4蝶形运算器的第四端口、第一端口、第二端口和第三端口；

其中，每次转换输入端口的长度为$1/4^{M}\times N$；

将每个中间级的蝶形运算结果分为16组，按照循环寻址的方式保存在缓存模块中；

进行最后一级蝶形运算，将基-4蝶形运算器中第一端口的数据保存至第一RAM，将基-4蝶形运算器中第二端口的数据保存至第三RAM，将基-4蝶形运算器中第三端口的数据保存至第二RAM，将基-4蝶形运算器中第四端口的数据保存至第四RAM；

其中，所述缓存模块进行了多级数据划分，直到每组数据的个数为1；

依次读取缓存模块中的数据并顺序输出蝶形运算结果。

6.根据权利要求1所述的FFT方法，其特征在于，所述缓存模块为多个双口RAM或多个单口RAM。

7.根据权利要求6所述的FFT方法，其特征在于，所述双口RAM的个数为7个或8个，由FFT运算的点数决定。

8.根据权利要求6所述的FFT方法，其特征在于，

所述多个双口RAM的总深度小于等于FFT运算点数的两倍。

9.根据权利要求1所述的FFT方法，其特征在于，所述基-4蝶形运算器的个数为1个或2个，由FFT运算的点数决定。

基于FPGA的FFT装置及方法

技术领域

[0001] 本发明涉及数字信号领域,尤其涉及一种基于FPGA的FFT装置及方法。

背景技术

[0002] 在无线通信系统中,经常使用快速傅里叶变换FFT对输入时域信号进行变换分析,观察频域波形,以获取信号的频域特征。OFDM利用离散傅立叶反变换和离散傅里叶变换(IDFT/DFT)代替多载波调制和解调的实现,即在发射端对待调制数据进行IFFT运算来实现调制,接收端对接收到的数据进行FFT运算实现解调,从而大大降低了系统实现的复杂度。

[0003] FPGA可以很好地解决并行性和速度问题,并且有配置灵活、易于升级等特点,是常用的实现快速傅里叶变换FFT的方法。例如,Xilinx的Virtex6系列芯片在FPGA内部,不仅提供了多个称为DSP Slices的计算单元,还提供了可读写的LUT单元、双端口RAM单元。

[0004] 目前Xilinx的Virtex6系列芯片内部的FFT算法IP软核分为四种模式,分别为:流水的数据流I/O(Pipelined,Streaming I/O)、基-4突发I/O(Radix-4,Burst I/O)、基-2突发I/O(Radix-2,Burst I/O)、基-2Lite突发I/O(Radix-2Lite,Burst I/O)。按结构可分为Pipelined和Burst两种,下面简单介绍两种结构的实现方法,如下:

[0005] (1)流水的数据流I/O。

[0006] 流水的数据流I/O结构通过一组基-2蝶形单元处理引擎的流水来实现连续数据处理。每个处理引擎都有存储器块来存储输入数据和中间数据。

[0007] (2)基-4突发I/O。

[0008] 对于基-4突发I/O结构,FFT IP核用一个基-4蝶形单元处理引擎实现。

[0009] 对于流水的数据流I/O结构,IP核在处理当前帧数据变换计算的同时,可以加载下一帧输入数据并输出前一帧的变换结果数据,可以连续输入数据并在一定的计算延时后获得连续的计算结果输出。输入数据是顺序的,输出数据可以是倒序的或者顺序的。下面以8点为例说明基-2蝶形流水线式的FFT(如图1所示)。

[0010] 基-2DIF以两点为单位进行蝶形运算,在进入运算前先进行数据缓存,使输入数据的上半部分与下半部分相结合。基本结构如下:

[0011] 设一个时钟周期缓存一个数据,即第一个时钟缓存0,第二个时钟缓存1...输入数据缓存RAM空间为4,即当第五个数据“4”到来时,缓存的数据“0”与“4”直接进行蝶形运算,而不用存储数据。最终输出的频域数据遵循倒序排列,8点的基-2蝶形流水线式的FFT的输入输出表如表1所示:

[0012] 表1 基-2蝶形流水线式的FFT的输入输出表

[0013]

输入(正序)	十进制	输出(倒序)	十进制
000	0	000	0
001	1	100	4
010	2	010	2
011	3	110	6

100	4	001	1
101	5	101	5
110	6	011	3
111	7	111	7

[0014] 根据蝶形图,8点基-2FFT分为3级,运算前缓存数据需要空间为4,中间数需要的空间分别为4、2,最后顺序输出数据时需要先进行所有点的缓存,再寻址输出,缓存空间为8。共计使用RAM空间为18,每级使用一个蝶形运算,共计使用3个蝶形运算单元,假设1一个蝶形运算单元使用3个DSP乘法器,则共计使用9个DSP乘法器。

[0015] 应用基-2蝶形运算单元的FFT装置利用每级放置蝶形单元及存储中间数据,让数据可以连续进行固定点FFT,随着FFT运算的点数增多,占用的资源也随着增长,并且由于每级运算只使用一个基-2蝶形单元,算数的先后是固定的,所以在最后一级要求顺序输出时,需要额外增加RAM,表2统计了应用基-2蝶形运算单元的FFT装置采用scale缩放模式进行处理时,存储数据占用的RAM的总深度和进行运算占用的DSP乘法器的数量。

[0016] 表2 应用基-2蝶形运算单元的FFT装置占用的资源量

[0017]

FFT 点数	基-2 级数	流水结构级数	多个 RAM 的总深度	占用的 DSP 乘法器
256	8	8	638	9
1024	10	10	2558	12
4096	12	12	10238	15
16384	14	14	40958	18
65536	16	16	163838	21

发明内容

[0018] 本发明所要解决的技术问题是:现有的FFT装置在数据顺序输出对RAM的利用率低、需要较多的FPGA资源的问题。

[0019] 为解决上述技术问题,本发明一方面提出了一种基于FPGA的FFT装置。该基于FPGA的FFT装置包括:

[0020] 缓存模块、控制模块和基-4蝶形运算器;

[0021] 所述控制模块分别与缓存模块和基-4蝶形运算器相连,用于控制数据的输入、输出,用于控制数据以乒乓缓存的方式缓存至缓存模块中,用于控制数据以循环寻址的方式在基-4蝶形运算器中完成FFT运算;

[0022] 所述缓存模块用于初始输入前3/4的数据,输出后3/4的运算结果;并用于保存中间数据;

[0023] 所述基-4蝶形运算器用于初始输入后1/4的数据,输出前1/4的运算结果。

[0024] 可选地,所述缓存模块为多个双口RAM或多个单口RAM。

[0025] 可选地，所述双口RAM的个数为7个或8个，由FFT运算的点数决定。

[0026] 可选地，所述多个RAM的总深度小于等于FFT运算点数的两倍。

[0027] 可选地，所述基-4蝶形运算器的个数为1个或2个，由FFT运算的点数决定。

[0028] 本发明另一方面提出了一种采用如上所述的基于FPGA的FFT装置的FFT方法，包括：

[0029] 顺序输入第一帧数据，完成第一帧数据的1级蝶形运算后，采用乒乓缓存顺序输入第二帧数据，并完成第一帧数据的M级蝶形运算；

[0030] 完成第一帧数据的蝶形运算结果的顺序输出，同时进行第二帧数据的缓存及蝶形运算；

[0031] 完成第二帧数据的M级蝶形运算，同时采用乒乓缓存进行第三帧数据的缓存，并开始进行第三帧数据的1级蝶形运算；

[0032] 不断重复数据的缓存、蝶形运算和结果输出过程，完成多帧数据的蝶形运算；

[0033] 其中，M为蝶形运算的级数，N为FFT运算的点数，$N=4^M$；数据读取和存储采用循环寻址方式。

[0034] 可选地，所述顺序输入第一帧数据，完成第一帧数据的1级蝶形运算后，采用乒乓缓存顺序输入第二帧数据，并完成第一帧数据的M级蝶形运算；完成第一帧数据的蝶形运算结果的顺序输出，同时进行第二帧数据的缓存及蝶形运算包括：

[0035] 顺序输入前3/4的第一帧数据至缓存模块的第一部分，当后1/4的第一帧数据到达基-4蝶形运算器时，根据蝶形运算图直接与缓存模块中的数据进行蝶形运算，并将1级蝶形运算的结果保存至缓存模块的第一部分；

[0036] 完成第一帧数据的M级蝶形运算，基-4蝶形运算器顺序输出第一帧数据的蝶形运算结果的前1/4，后3/4的运算结果保存至缓存模块的第一部分；采用乒乓缓存顺序输入前3/4的第二帧数据至缓存模块的第二部分，当后1/4的第二帧数据到达基-4蝶形运算器时，根据蝶形运算图直接与缓存模块中的数据进行蝶形运算；

[0037] 缓存模块的第一部分顺序输出第一帧数据的蝶形运算结果的后3/4；

[0038] 相应地，所述缓存模块的数据读取和存储采用循环寻址方式。

[0039] 可选地，所述循环寻址方式包括：

[0040] 进行1级蝶形运算，将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中；

[0041] 进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据，将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中；

[0042] 进行最后一级的蝶形运算，按照循环寻址方式将蝶形运算结果保存至缓存模块中，依次读取缓存模块中的数据并顺序输出蝶形运算结果。

[0043] 可选地，所述进行1级蝶形运算，将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中包括：

[0044] 进行1级蝶形运算，将1级蝶形运算结果分成16组，将所述16组蝶形运算结果的第0-3组数据依次存入第一RAM、第二RAM、第三RAM和第四RAM；将所述16组蝶形运算结果的第4-7组数据依次存入第二RAM、第三RAM、第四RAM和第一RAM；将所述16组蝶形运算结果的第8-11组数据依次存入第三RAM、第四RAM、第一RAM和第二RAM；将所述16组蝶形运算结果的第12-15组数据依次存入第四RAM、第一RAM、第二RAM和第三RAM。

[0045] 可选的，所述进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据，将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中包括：

[0046] 进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第一端口、第二端口、第三端口和第四端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第二端口、第三端口、第四端口和第一端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第三端口、第四端口、第一端口和第二端口；按照循环寻址方式读取缓存模块的数据输入到基-4蝶形运算器的第四端口、第一端口、第二端口和第三端口；

[0047] 其中，每次转换输入端口的长度为$1/4^{M}\times N$；

[0048] 将每个中间级的蝶形运算结果分为16组，按照循环寻址的方式保存在缓存模块中。

[0049] 可选地，所述进行最后一级的蝶形运算，按照循环寻址方式将蝶形运算结果保存至缓存模块中，依次读取缓存模块中的数据并顺序输出蝶形运算结果包括：

[0050] 进行最后一级蝶形运算，将基-4蝶形运算器中第一端口的数据保存至第一RAM，将基-4蝶形运算器中第二端口的数据保存至第三RAM，将基-4蝶形运算器中第三端口的数据保存至第二RAM，将基-4蝶形运算器中第四端口的数据保存至第四RAM；

[0051] 其中，所述缓存模块进行了多级数据划分，直到每组数据的个数为1；

[0052] 依次读取缓存模块中的数据并顺序输出蝶形运算结果。

[0053] 本发明提出的基于FPGA的FFT装置及方法，采用基4蝶形运算器，提高了运算速度，采用循环寻址的方式在存储中间数据时不需要额外的RAM，在数据顺序输出时不需要额外的RAM，在DSP乘法器使用数量与应用基-2蝶形运算单元的FFT装置相当的情况下减少了存储数据的RAM的总深度，提高了对RAM的利用率，节省了FPGA的资源。

附图说明

[0054] 通过参考附图会更加清楚的理解本发明的特征和优点，附图是示意性的而不应理解为对本发明进行任何限制，在附图中：

[0055] 图1为应用基-2蝶形运算单元的FFT装置的结构示意图；

[0056] 图2为本发明一个实施例的基于FPGA的FFT装置的结构示意图；

[0057] 图3为本发明一个实施例的基于FPGA的FFT装置的原理图；

[0058] 图4为发明一个实施例的基于FPGA的FFT的方法示意图。

具体实施方式

[0059] 下面将结合附图对本发明的实施例进行详细描述。

[0060] 图2示出了本发明一个实施例的基于FPGA的FFT装置的结构示意图。

[0061] 如图2所示，本实施例的基于FPGA的FFT装置包括：

[0062] 缓存模块1、控制模块2和基-4蝶形运算器3；

[0063] 控制模块2分别与缓存模块1和基-4蝶形运算器3相连，用于控制数据的输入、输出，用于控制数据以乒乓缓存的方式缓存至缓存模块1中，用于控制数据以循环寻址的方式在基-4蝶形运算器3中完成FFT运算；

[0064] 缓存模块1用于初始输入前3/4的数据，输出后3/4的运算结果，并用于保存中间数据；

[0065] 基-4蝶形运算器2用于初始输入后1/4的数据，输出前1/4的运算结果。

[0066] 本实施例的基于FPGA的FFT装置，采用基4蝶形运算器，提高了运算速度，采用循环寻址的方式在存储中间数据时不需要额外的RAM，在数据顺序输出时不需要额外的RAM，在DSP乘法器使用数量与应用基-2蝶形运算单元的FFT装置相当的情况下减少了存储数据的RAM的总深度，提高了对RAM的利用率，节省了FPGA的资源。

[0067] 在一种可选的实施方式中，所述缓存模块为多个双口RAM或多个单口RAM。在基于FPGA的FFT装置中，缓存模块为双口RAM可以达到使用RAM的个数较少的效果。

[0068] 所述双口RAM的个数为7个或8个，由FFT运算的点数决定。

[0069] 所述多个RAM的总深度小于等于FFT运算点数的两倍。

[0070] 所述基-4蝶形运算器的个数为1个或2个，由FFT运算的点数决定。

[0071] 图3为本发明一个实施例的基于FPGA的FFT装置的原理图。如图3所示，该FFT装置包括若干双口RAM和蝶形运算器及选择器，其中总RAM深度最多为FFT点数的2倍，宽度为数据宽度。蝶形运算最多设置2个基-4蝶形运算器，每8块RAM可在一周期内并行输出8个数据，可充分利用两个基-4蝶形运算单元，提高运算速度。

[0072] 图4为发明一个实施例的基于FPGA的FFT的方法示意图。如图4所示，采用如上所述的基于FPGA的FFT装置的FFT方法，包括：

[0073] S41：顺序输入第一帧数据，完成第一帧数据的1级蝶形运算后，采用乒乓缓存顺序输入第二帧数据，并完成第一帧数据的M级蝶形运算；

[0074] S42：完成第一帧数据的蝶形运算结果的顺序输出，同时进行第二帧数据的缓存及蝶形运算；

[0075] S43：完成第二帧数据的M级蝶形运算，同时采用乒乓缓存进行第三帧数据的缓存，并开始进行第三帧数据的1级蝶形运算；

[0076] S44：不断重复数据的缓存、蝶形运算和结果输出过程，完成多帧数据的蝶形运算；

[0077] 其中，M为蝶形运算的级数，N为FFT运算的点数，$N=4^M$；数据读取和存储采用循环寻址方式。

[0078] 进一步地，所述顺序输入第一帧数据，完成第一帧数据的1级蝶形运算后，采用乒乓缓存顺序输入第二帧数据，并完成第一帧数据的M级蝶形运算；完成第一帧数据的蝶形运算结果的顺序输出，同时进行第二帧数据的缓存及蝶形运算包括：

[0079] 顺序输入前3/4的第一帧数据至缓存模块的第一部分，当后1/4的第一帧数据到达基-4蝶形运算器时，根据蝶形运算图直接与缓存模块中的数据进行蝶形运算，并将1级蝶形运算的结果保存至缓存模块的第一部分；

[0080] 完成第一帧数据的M级蝶形运算，基-4蝶形运算器顺序输出第一帧数据的蝶形运算结果的前1/4，后3/4的运算结果保存至缓存模块的第一部分；采用乒乓缓存顺序输入前3/4的第二帧数据至缓存模块的第二部分，当后1/4的第二帧数据到达基-4蝶形运算器时，根据蝶形运算图直接与缓存模块中的数据进行蝶形运算；

[0081] 缓存模块的第一部分顺序输出第一帧数据的蝶形运算结果的后3/4；

[0082] 相应地，所述缓存模块的数据读取和存储采用循环寻址方式。

[0083] 下面以一个具体的例子说明该基于FPGA的FFT方法中的乒乓缓存过程。

[0084] 设一帧串行数据进行FFT运算的点数为4096点，采用基-4DIF运算，使用的RAM为图3中的RAM1-14(需要说明的是，图3中的RAM1-14为单口RAM，以下乒乓缓存的过程也是以单口RAM为例进行说明的；对于FFT的点数为4096点的运算，可以使用7个双口RAM，其过程和工作原理与使用单口RAM类似)，其过程如下：

[0085] (1)对输入串行数据帧0进行缓存，缓存空间设置为运算点数的3/4，即4096*0.75＝3072，即缓存到RAM6。

[0086] (2)当第3073个数据到来时，根据蝶形运算图，直接与之前缓存RAM中第1、1025、2049个数据进行基-4蝶形运算。并将计算结果存入缓存至RAM1～RAM8中。

[0087] (3)当第3074个数据到来时，根据蝶形运算图，直接与缓存中第2、1026、2050个数据进行基-4蝶形运算，并将计算结果存入缓存至RAM1～RAM8中。

[0088] (4)当第3075个数据到来时，根据蝶形运算图，直接与缓存中第3、1027、2051个数据进行基-4蝶形运算，并将计算结果存入缓存至RAM1～RAM8中。

[0089] 当第3076个数据到来时....

[0090] 当第4096个数据到来时，根据蝶形运算图，直接与缓存中第1024、2048、3072个数据进行基-4蝶形运算，并将计算结果存入缓存RAM中。此时完成了第1级的所有蝶形运算。完成1级运算的数据存入的缓存RAM为RAM1～RAM8。

[0091] (5)对下一帧输入数据帧1进行缓存，缓存空间从RAM9开始，此1024个时钟周期内，可以利用2个基-4蝶形运算单元对1～6缓存RAM内的数据继续进行处理，此时1个时钟周期读出缓存器内8点数据进行蝶形运算，在1024周期内共完成1024*8＝8192点，即8192/4096＝2级蝶形运算。此时将3级运算完的数据仍存回RAM1～RAM8，实现原址运算。

[0092] (6)继续对帧1数据进行缓存，缓存空间为RAM11和RAM12，此1024个时钟周期内，可以利用2个基-4蝶形运算单元对1～6缓存RAM内的数据继续进行处理，此时1个时钟周期读出缓存器内8点数据进行蝶形运算，在1024周期内共完成1024*8＝8192点，即8192/4096＝2级蝶形运算。此时将5级运算完的数据仍存回RAM1～RAM8，实现原址运算。

[0093] (7)继续对帧1数据进行缓存，缓存空间为RAM13和RAM14，此1024个时钟周期内，可以利用1个基-4蝶形运算单元对7～14缓存RAM内的数据继续进行处理，此时1个时钟周期读出缓存器内4点数据进行蝶形运算，在1024周期内共完成1024*4＝4096点，即4096/4096＝1级蝶形运算。此时将6级运算完的数据仍存回RAM1～RAM8，实现原址运算。由于已经完成了最后一级的运算，在计算过程中，可以将6级运算后的结果直接输出，当所有计算完成时，结果输出1/4。

[0094] (8)对帧1进行1级运算，将运算结果存入RAM1～2、RAM9～14，同时RAM3输出上一帧帧0的运算结果。

[0095] (9)对RAM3、4进行缓存，缓存数据为下一帧帧2数据，同时RAM5输出上一帧的运算结果，此时帧1数据利用两个蝶形运算器完成3级蝶形运算。

[0096] (10)对RAM5、6进行缓存，同时RAM7开始帧0数据输出，帧1数据完成5级运算。

[0097] (11)对RAM7、8进行缓存，帧1数据完成6级运算并输出。

[0098] 进一步地，所述循环寻址方式包括：

[0099] 进行1级蝶形运算，将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中；

[0100] 进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据，将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中；

[0101] 进行最后一级的蝶形运算，按照循环寻址方式将蝶形运算结果保存至缓存模块中，依次读取缓存模块中的数据并顺序输出蝶形运算结果。

[0102] 具体地，所述进行1级蝶形运算，将1级蝶形运算结果按照循环寻址的方式保存在缓存模块中包括：

[0103] 进行1级蝶形运算，将1级蝶形运算结果分成16组，将所述16组蝶形运算结果的第0-3组数据依次存入第一RAM、第二RAM、第三RAM和第四RAM；将所述16组蝶形运算结果的第4-7组数据依次存入第二RAM、第三RAM、第四RAM和第一RAM；将所述16组蝶形运算结果的第8-11组数据依次存入第三RAM、第四RAM、第一RAM和第二RAM；将所述16组蝶形运算结果的第12-15组数据依次存入第四RAM、第一RAM、第二RAM和第三RAM。

[0104] 具体地，所述进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据，将中间级蝶形运算结果按照循环寻址的方式保存在缓存模块中包括：

[0105] 进行中间级的蝶形运算，按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第一端口、第二端口、第三端口和第四端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第二端口、第三端口、第四端口和第一端口；按照循环寻址方式读取缓存模块中的数据输入到基-4蝶形运算器的第三端口、第四端口、第一端口和第二端口；按照循环寻址方式读取缓存模块的数据输入到基-4蝶形运算器的第四端口、第一端口、第二端口和第三端口；

[0106] 其中，每次转换输入端口的长度为$1/4^{M}\times N$；

[0107] 将每个中间级的蝶形运算结果分为16组，按照循环寻址的方式保存在缓存模块中。

[0108] 具体地，所述进行最后一级的蝶形运算，按照循环寻址方式将蝶形运算结果保存至缓存模块中，依次读取缓存模块中的数据并顺序输出蝶形运算结果包括：

[0109] 进行最后一级蝶形运算，将基-4蝶形运算器中第一端口的数据保存至第一RAM，将基-4蝶形运算器中第二端口的数据保存至第三RAM，将基-4蝶形运算器中第三端口的数据保存至第二RAM，将基-4蝶形运算器中第四端口的数据保存至第四RAM；

[0110] 其中，所述缓存模块进行了多级数据划分，直到每组数据的个数为1；

[0111] 依次读取缓存模块中的数据并顺序输出蝶形运算结果。

[0112] 下面以一个具体的例子说明基于FPGA的FFT方法中的循环寻址的过程。(此次介绍的是使用一个蝶形运算器的方法，与两个蝶形运算器方法一致)

[0113] (1)将N点数据进行顺序输入到RAM中，直到3/4数据输入到RAM后，开始1级寻址计算。

[0114] (2)1级寻址：RAM1～3按照地址0～(1/4*N-1)依次读出数据并作为蝶形运算器的前3个输入，蝶形运算的第4个输入为直接过来的数据。经过计算后将蝶形运算器输出端口1～4的第0～(1/16*N-1)个数据依次存入RAM1、2、3、4，序号(1/16*N)～(1/8*N-1)依次存入RAM2、3、4、1，序号(1/8*N)～(3/16*N-1)依次存入RAM3、4、1、2，序号(3/16*N)～(1/4*N-1)依次存入RAM4、1、2、3。

[0115] (3)2级寻址：RAM1读地址0～(1/16*N-1)、(1/16*N)～(1/8*N-1)、(1/8*N)～(3/

16*N-1)、(3/16*N)～(1/4*N-1)数据并分别作为蝶形运输器输入端口1、2、3、4的数据。同时RAM2读地址(1/16*N)～(1/8*N-1)、(1/8*N)～(3/16*N-1)、(3/16*N)～(1/4*N-1)、0～(1/16*N-1)的数据并作为蝶形运输器输入端口2、3、4、1的数据。RAM3读地址(1/8*N)～(3/16*N-1)、(3/16*N)～(1/4*N-1)、0～(1/16*N-1)、(1/16*N)～(1/8*N-1)数据并作为蝶形运算器输入端口3、4、1、2的数据。RAM4读地址(3/16*N)～(1/4*N-1)、0～(1/16*N-1)、(1/16*N)～(1/8*N-1)、(1/8*N)～(3/16*N-1)数据并作为蝶形运输器输入端口4、1、2、3的数据。经过计算后将蝶形运算器输出端口1～4的第0～(1/64*N-1)个数据依次存入RAM1、2、3、4,序号(1/64*N)～(1/32*N-1)依次存入RAM2、3、4、1,序号(1/32*N)～(3/64*N-1)依次存入RAM3、4、1、2,序号(3/64*N)～(1/16*N-1)依次存入RAM4、1、2、3。同样对剩下的数做相同操作,即序号(1/16*N)～(5/64*N-1)数据依次存入RAM1、2、3、4,序号(5/64*N)～(6/64*N-1)依次存入RAM2、3、4、1,序号(6/64*N)～(7/64*N-1)依次存入RAM3、4、1、2,序号(7/64*N)～(8/64*N-1)依次存入RAM4、1、2、3....

[0116] (4)3级寻址:RAM1读地址0～(1/64*N-1)、(1/64*N)～(2/64*N-1)、(2/64*N)～(3/64*N-1)、(3/64*N)～(4/64*N-1)数据并分别作为蝶形运输器输入端口1、2、3、4的数据。同时RAM2读地址(4/64*N)～(5/64*N-1)、(5/64*N)～(6/64*N-1)、(6/64*N)～(7/64*N-1)、(7/64*N)～(8/64*N-1)数据并分别作为蝶形运输器输入端口2、3、4、1的数据。RAM3读地址(8/64*N)～(9/64*N-1)、(9/64*N)～(10/64*N-1)、(10/64*N)～(11/64*N-1)、(11/64*N)～(12/64*N-1)数据并分别作为蝶形运输器输入端口3、4、1、2的数据。RAM4读地址(12/64*N)～(13/64*N-1)、(13/64*N)～(14/64*N-1)、(14/64*N)～(15/64*N-1)、(15/64*N)～(16/64*N-1)数据并分别作为蝶形运输器输入端口3、4、1、2的数据。同样对剩下的地址数据做同样操作。经过计算后将蝶形运算器输出端口1～4的第0～(1/256*N-1)个数据依次存入RAM1、2、3、4,序号(1/256*N)～(2/256*N-1)依次存入RAM2、3、4、1,序号(2/256*N)～(3/256*N-1)依次存入RAM3、4、1、2,序号(3/256*N)～(4/256*N-1)依次存入RAM4、1、2、3。同样对剩下的数做相同操作,即序号(4/256*N)～(5/256*N-1)数据依次存入RAM1、2、3、4,序号(5/256*N)～(6/256*N-1)依次存入RAM2、3、4、1,序号(6/256*N)～(7/256*N-1)依次存入RAM3、4、1、2,序号(7/256*N)～(8/256*N-1)依次存入RAM4、1、2、3....

[0117] (5)4、5、6、7级寻址....

[0118] (6)最后一级寻址:首先对RAM1依次读取地址0、2/16*N、3/16*N、1/16*N,输出的数据作为蝶形运算器的端口1输入,同时对RAM2依次读取地址(a+a1..)、(2/16*N+a+a1..)、(3/16*N+a+a1..)、(1/16*N+a+a1..),输出的数据作为蝶形运算器的端口2输入,对RAM3依次读取地址2*(a+a1...)、[2/16*N+2*(a+a1...)]、[3/16*N+2*(a+a1...)]、[1/16*N+2*(a+a1...)],输出的数据作为蝶形运算器的端口3输入,对RAM4依次读取地址3*(a+a1...)、[2/16*N+3*(a+a1...)]、[3/16*N+3*(a+a1...)]、[1/16*N+3*(a+a1...)],输出的数据作为蝶形运算器的端口4输入。将端口1的计算结果直接作为最终输出数据输出,将端口2的数据原址存入RAM3,将端口3的数据原址存入RAM2,将端口4的数据原址存入RAM4,接下来继续对RAM1进行读操作,读取地址为2/64*N、(2/16*N+2/64*N)、(3/16*N+2/64*N)、(1/16*N+2/64*N),对RAM2读取地址为(2/64*N+a+a1...)、[(2/16*N+2/64*N)+a+a1...]、[(3/16*N+2/64*N)+a+a1...]、[(1/16*N+2/64*N)+a+a1...],对RAM3读取地址为[2/64*N+2*(a+a1...)]、[(2/16*N+2/64*N)+2*(a+a1...)]、[(3/16*N+2/64*N)+2*(a+a1...)]、[(1/16*N+2/64*N)+

2*(a+a1...)],对RAM4读取地址为[2/64*N+3*(a+a1...)]、[(2/16*N+2/64*N)+3*(a+a1...)]、[(3/16*N+2/64*N)+3*(a+a1...)]、[(1/16*N+2/64*N)+3*(a+a1...)],同样输出的数据作为蝶形运算器的端口4输入。将端口1的计算结果直接作为最终输出数据输出,将端口2的数据原址存入RAM3,将端口3的数据原址存入RAM2,将端口4的数据原址存入RAM4...其中a,a1...代表级,如最后一级为3级运算,即64点,那么a=4,a1=1。如果最后一级为4级运算,即256点,那么a=16,a1=4,a2=1。如果最后一级为M级运算,即4^M点,那么a=4^M/16,a1=4^M/64...aM-1=1。

[0119] (7)当完成最后一级运算后,1/4*N的数据已经顺序输出完毕,接下来依次输出RAM2～4的数据即可。

[0120] 总结寻址不难发现:

[0121] 对第一级的数据可以从各个RAM依次读出并顺序输入到蝶形运算的4个端口进行运算。运算的输出数据分成16组(每个蝶形运算器同时输出4组数据,每个蝶形运算器输出端口产生4组数据),依次存入RAM1、2、3、4,RAM2、3、4、1,RAM3、4、1、2,RAM4、1、2、3中。

[0122] 对中间级的数据寻址时,RAM1始终从地址0开始读出,并且将读出的数据分别输入到蝶形运算器的输入端口1、2、3、4,依次循环。每次转换输入端口的数据长度依次为1级1/4*N,2级1/16*N..M级1/4^M*N,其中N=4^M。RAM2起始读地址为a1+a2...,若为1级计算,a1=1/4*N,a2,a3...=0,若为2级运算,a1=1/4*N,a2=1/16*N,a3...=0,若为M级计算,a1=1/4*N,a2=1/16*N...aM=1/4^M*N,之后顺序读出地址,地址在读到最大值时返回到地址0继续寻址,知道完成整个地址空间深度的一次循环。RAM3、4起始读地址分别为2(a1+a2...)和3(a1+a2...),其它同RAM2操作。当蝶形运算完毕开始写入时,与读地址位置一致,实现原址存储,需要注意的是,每个蝶形运算器输出端口的每4组数据需要放在不同的RAM中,每组的数据长度根据级数定义,1级运算输出为1/16*N,2级运算输出为1/64*N....端口1的数据存储位置依次为RAM1、2、3、4循环,端口2为RAM2、3、4、1循环,端口3为RAM3、4、1、2循环,端口4为RAM4、1、2、3循环。

[0123] 对最后1级的数据寻址时,根据基4蝶形运算图最终输出顺序的特点,寻址规律如下:第一步,将RAM内的数按照地址分成4组,称为1级组。每组的深度为1/16*N,编号为组1～4,然后RAM1按照0、2/16*N、3/16*N、1/16*N进行寻址,即顺序为组1、组3、组2、组4的首地址,其它RAM在此地址基础上加上a1+a2...即可。第二步,对每个1级组再分4组,称为2级组,每组的深度为1/64*N,编号为组1～4,然后RAM1按照组1、组3、组2、组4的地址寻址,其它RAM2在此基础上加上a1+a2...即可。(第一步中已经对二级组1寻址)第三步,对每个2级组再分4组....直到每组数据的个数为1停止分组。经过蝶形运算后,输出端口1的数据直接送至总模块的输出端口,端口2的数据存入RAM3中,端口3的数据存入RAM2中,端口4的数据存入RAM4中。当蝶形运算完毕后,1/4*N的数据输出完毕,此时再按顺序依次读取RAM2～4的数据输出。

[0124] 根据以上过程,本实施例的基于FPGA的FFT方法采用基4蝶形运算器,提高了运算速度,采用乒乓缓存、循环寻址的方式实现了数据的原址计算,在存储中间数据时不需要额外的RAM,在数据顺序输出时不需要额外的RAM,(对比表2和表3可知)基-2蝶形运算单元的FFT装置相当的情况下减少了存储数据的RAM的总深度,提高了对RAM的利用率,节省了FPGA的资源。表3统计了存储数据占用的RAM总深度和进行运算占用的DSP乘法器的数量,其中存

储RAM的位宽为数据位宽，如下：

[0125] 表3 应用基-4蝶形运算器的FFT装置占用的资源量

[0126]

FFT 点数	基-4 级数	多个 RAM 的总深度	DSP 乘法器	RAM 比（基2）	DSP 比（基2）
256	4^4	448	9	70%	100%
1024	4^5	2048	9	80%	75%
4096	4^6	7168	18	70%	120%
16384	4^7	28672	18	70%	100%
65536	4^8	131072	18	80%	86%

[0127] 本领域内的技术人员应明白，本发明的实施例可以为系统或计算机程序产品。因此，本发明的装置可采用完全硬件实施例的形式。本发明是参照根据本发明实施例的设备(系统)的流程图和/或方框图来描述的。尽管已描述了本发明的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例作出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本发明范围的所有变更和修改。

[0128] 本发明提出的基于FPGA的FFT装置及方法，采用基4蝶形运算器，提高了运算速度，采用循环寻址的方式在存储中间数据时不需要额外的RAM，在数据顺序输出时不需要额外的RAM，(对比表2和表3可知)基-2蝶形运算单元的FFT装置相当的情况下减少了存储数据的RAM的总深度，提高了对RAM的利用率，节省了FPGA的资源。

[0129] 虽然结合附图描述了本发明的实施方式，但是本领域技术人员可以在不脱离本发明的精神和范围的情况下做出各种修改和变型，这样的修改和变型均落入由所附权利要求所限定的范围之内。

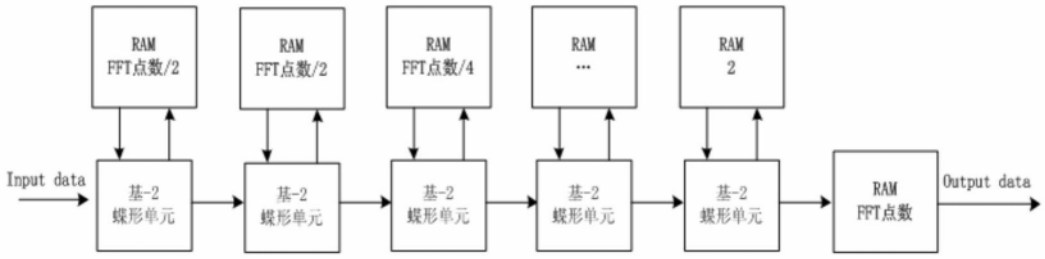
RAM
FFT点数/2
RAM
FFT点数/2
RAM
FFT点数/4
RAM
…
RAM
2
Input data
基-2
蝶形单元
基-2
蝶形单元
基-2
蝶形单元
基-2
蝶形单元
基-2
蝶形单元
RAM
FFT点数
Output data

图1

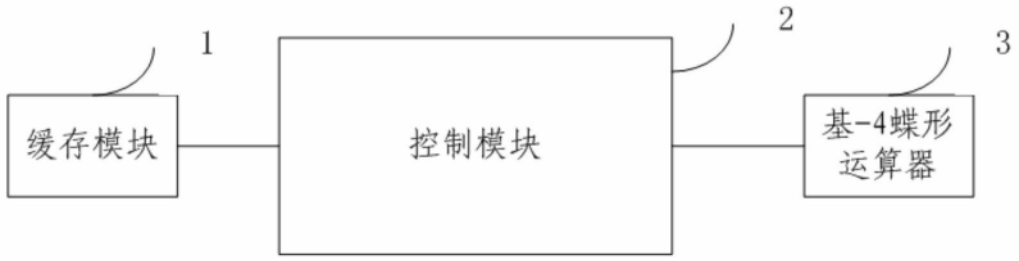
1
2
3
缓存模块
控制模块
基-4蝶形
运算器

图2

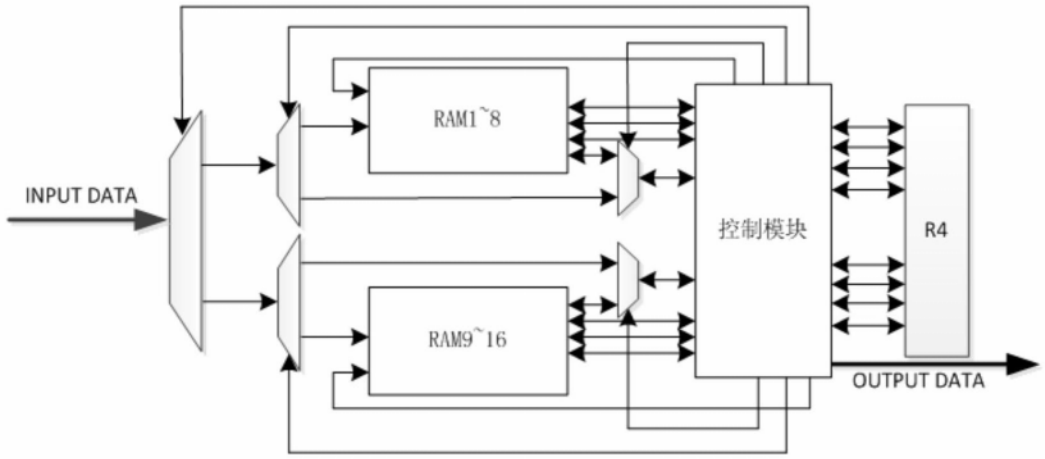
INPUT DATA
RAM1~8
RAM9~16
控制模块
R4
OUTPUT DATA

图3

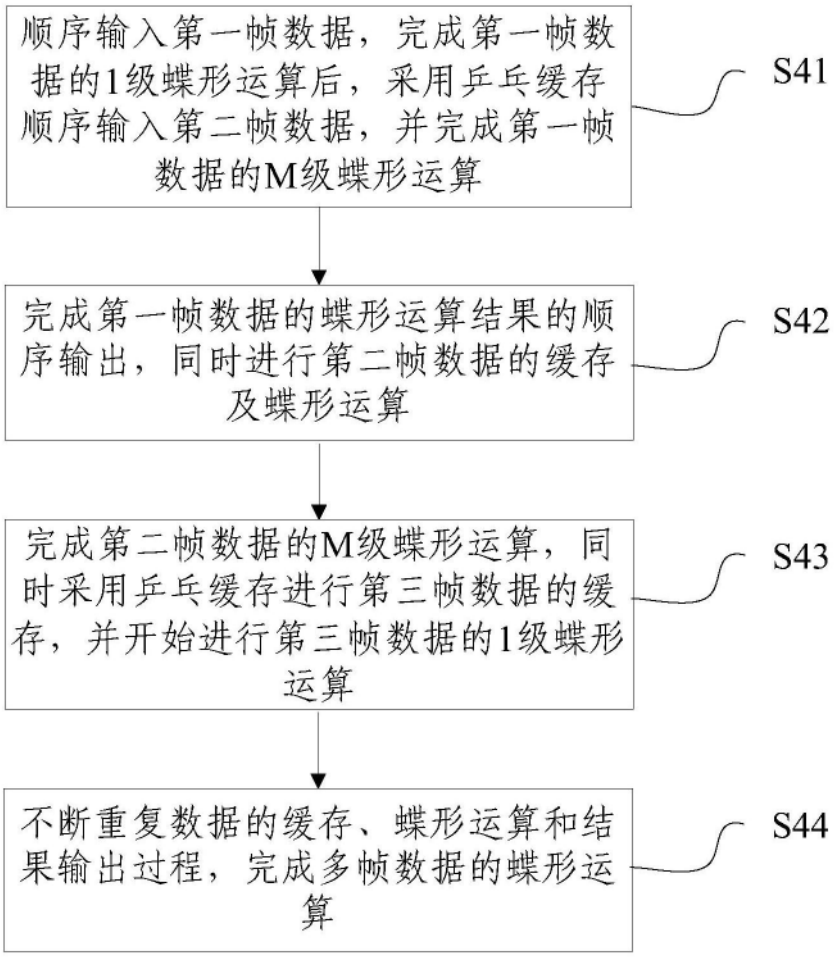
顺序输入第一帧数据，完成第一帧数据的1级蝶形运算后，采用乒乓缓存顺序输入第二帧数据，并完成第一帧数据的M级蝶形运算
S41
完成第一帧数据的蝶形运算结果的顺序输出，同时进行第二帧数据的缓存及蝶形运算
S42
完成第二帧数据的M级蝶形运算，同时采用乒乓缓存进行第三帧数据的缓存，并开始进行第三帧数据的1级蝶形运算
S43
不断重复数据的缓存、蝶形运算和结果输出过程，完成多帧数据的蝶形运算
S44

图4