



(12) 发明专利

(10) 授权公告号 CN 101088237 B

(45) 授权公告日 2013. 02. 06

(21) 申请号 200580034598. 5

(22) 申请日 2005. 08. 16

(30) 优先权数据

60/602, 169 2004. 08. 17 US

(85) PCT申请进入国家阶段日

2007. 04. 10

(86) PCT申请的申请数据

PCT/US2005/029108 2005. 08. 16

(87) PCT申请的公布数据

W02006/023490 EN 2006. 03. 02

(73) 专利权人 塔特公司

地址 美国马萨诸塞州

(72) 发明人 马修·巴恩西尔

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 林锦辉

(51) Int. Cl.

H04R 5/04 (2006. 01)

H04S 1/00 (2006. 01)

(56) 对比文件

US 20030161477 A1, 2003. 08. 28, 说明书第 42 至 44 段, 摘要, 附图 2.

US 6588867 B1, 2003. 07. 08, 全文.

US 6259482 B1, 2001. 07. 10, 全文.

US 6118879 A, 2000. 09. 12, 说明书第 9 栏第 48 行至 65 行, 附图 3.

包本刚等. 数字滤波器的 IIR 和 FIR 设计. 《零陵学院学报》. 2003, 第 24 卷 (第 2 期), 第 67 — 68 页.

审查员 田琳琳

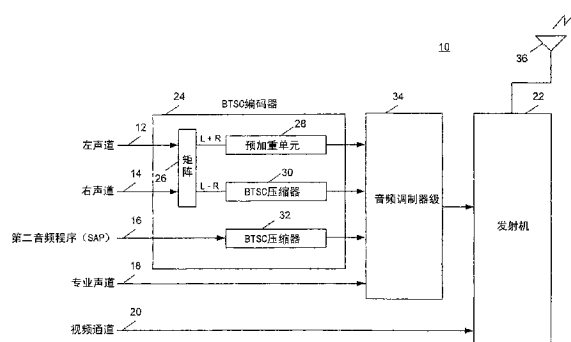
权利要求书 2 页 说明书 36 页 附图 8 页

(54) 发明名称

用于处理电视音频信号的可配置的递归数字滤波器

(57) 摘要

一种电视音频信号编码器, 包括将左声道音频信号和右声道音频信号相加以产生和信号的设备。该设备还将左和右音频信号中的一个从另一个中减去以产生差值信号。该编码器还包括可配置的无限脉冲响应数字滤波器, 其选择性地使用一组或多组滤波器系数对差值信号进行滤波。该组滤波器系数以递归方式通过单个乘法器应用到差值信号, 以准备差值信号用于发送。



1. 一种电视音频信号编码器,包括:

求和设备,配置为将左声道音频信号和右声道音频信号相加,以生成和信号,并且将所述左和右音频信号中的一个信号从所述左和右信号中的另一个信号中减去,以生成差值信号;以及

可配置的无限脉冲响应数字滤波器,配置为以比所述求和设备的时钟速度快的选择的时钟速度运行且选择性地多次使用至少一组滤波器系数来对所述差值信号进行滤波,其中,该至少一组滤波器系数在一递归循环内以递归方式由单个乘法器应用到所述差值信号,以便准备所述差值信号用于发送,其中,所述可配置的无限脉冲响应数字滤波器包括反馈通路且被配置为将该至少一组滤波器系数以递归方式应用到所述差值信号,且其中该至少一组滤波器系数在该递归循环的迭代之间可被改变。

2. 如权利要求1所述的电视音频信号编码器,其中,所述反馈通路包括移位寄存器,用于延迟与所述差值信号关联的数字信号。

3. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器配置为对与所述差值信号关联的信号做乘法,并且提供该乘法的输出。

4. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器包括选择器,配置为选择数字输入信号。

5. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器包括选择器,配置为选择所述滤波器系数中的一个。

6. 如权利要求4所述的电视音频信号编码器,其中,所述选择器包括多路复用器。

7. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器配置为提供低通滤波器。

8. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器包括单个加法器,用于以递归方式将所述滤波器系数应用到所述差值信号。

9. 如权利要求1所述的电视音频信号编码器,其中,所述电视音频信号遵循广播电视系统委员会(BTSC)标准。

10. 如权利要求1所述的电视音频信号编码器,其中,所述电视音频信号遵循准瞬时压缩音频复用(NICAM)标准。

11. 如权利要求1所述的电视音频信号编码器,其中,所述电视音频信号遵循A2/Zweiton标准。

12. 如权利要求1所述的电视音频信号编码器,其中,所述电视音频信号遵循EIA-J标准。

13. 如权利要求1所述的电视音频信号编码器,其中,所述可配置的无限脉冲响应数字滤波器在集成电路中实现。

14. 一种电视音频信号解码器,包括:

分离设备,配置为以一时钟速度运行且从差值信号以及和信号中分离左声道音频信号和右声道音频信号,其中,所述和信号包括所述左声道音频信号和右声道音频信号之和;及

可配置的无限脉冲响应信号数字滤波器,配置为以比所述分离设备的时钟速度快的选择的时钟速度运行且选择性地多次使用至少一组滤波器系数来对差值信号进行滤波,其中,所述差值信号是通过将左声道和右声道音频信号中的一个信号从所述左声道和右声道

音频信号中另一个信号中减去而得到的,其中,该至少一组滤波器系数在一递归循环中以递归方式由单个乘法器应用到所述差值信号,以便准备所述差值信号用于分离所述左声道和右声道音频信号,其中所述可配置的无限脉冲响应数字滤波器包括反馈通路且被配置为以递归方式将该至少一组滤波器系数应用到所述差值信号,且其中该至少一组滤波器系数在该递归循环的迭代之间可被改变。

15. 如权利要求 14 所述的电视音频信号解码器,其中,所述反馈通路包括移位寄存器,用于延迟与所述差值信号关联的数字信号。

16. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器配置为对与所述差值信号关联的信号做乘法,并且提供该乘法的输出。

17. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器包括选择器,配置为选择数字输入信号。

18. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器包括选择器,配置为选择所述滤波器系数中的一个。

19. 如权利要求 17 所述的电视音频信号解码器,其中,所述选择器包括多路复用器。

20. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器配置为提供低通滤波器。

21. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器包括单个加法器,用于以递归方式将所述滤波器系数应用到所述差值信号。

22. 如权利要求 14 所述的电视音频信号解码器,其中,所述电视音频信号遵循广播电视系统委员会 (BTSC) 标准。

23. 如权利要求 14 所述的电视音频信号解码器,其中,所述电视音频信号遵循准瞬时压扩音频复用 (NICAM) 标准。

24. 如权利要求 14 所述的电视音频信号解码器,其中,所述电视音频信号遵循 A2/Zweiton 标准。

25. 如权利要求 14 所述的电视音频信号解码器,其中,所述电视音频信号遵循 EIA-J 标准。

26. 如权利要求 14 所述的电视音频信号解码器,其中,所述可配置的无限脉冲响应数字滤波器在集成电路中实现。

用于处理电视音频信号的可配置的递归数字滤波器

[0001] 相关申请

[0002] 本申请涉及下述的共同受让人、要求其优先权并且其内容整体结合在此作为参考的美国申请：2004年8月17日递交的申请号为60/602,169的“Digital Architecture for a BTSC Encoder/Decoder with SAP”的美国临时专利申请。

[0003] 技术领域

[0004] 本公开涉及处理电视音频信号，并且尤其是涉及一种用于与编码和解码电视音频信号一起使用可配置结构。

[0005] 背景技术

[0006] 在1984年，美国在联邦通信委员会的支持下采用了对于电视的立体声音频的发送和接收的标准。该标准被整理在FCC的公报OET-60中，并且该标准在广播电视系统委员会提议之后通常被称为BTSC系统，或者被称为MTS（多声道电视伴音）系统。

[0007] 在BTSC系统之前，广播电视音频是单声道的，由单个“声道”或单个音频内容组成。立体声音频通常需要对两个独立音频声道的发送，并且需要接收机能够检测并恢复两个声道。为了满足FCC的新发送标准应与现有单声道电视机“兼容”的要求（即单声道接收机能够从新型立体声广播中再现适当的音频信号），广播电视系统委员会采用了与FM无线电系统类似的方法：将立体声左和右音频信号合并以形成两个新信号：和（Sum）信号以及差值（Difference）信号。

[0008] 单声道电视接收机仅检测并解调由左和右立体声信号之和组成的所述和信号。立体声接收机接收和信号和差值信号，重组这些信号，提取出原始立体声左和右信号。

[0009] 为了发送，和信号直接调制伴音FM载波，正象是单声道音频信号一样。但是，差值声道首先被调制到高于上述伴音载波中心频率的位于31.768/kHz的AM副载波上。FM调制的特性是背景噪声每个八度音增加3分贝（dB），并且因此，由于新的副载波位于距伴音载波中心频率比和信号或单声道信号更远处，附加噪声被引入到差值声道中，并因此被引入到所恢复的立体声信号中。在很多环境中，事实上这种增大的噪声特性为立体声信号带来了太大的噪声从而不能满足FCC所提出的需求，并且因此BTSC系统在差值声道信号通路中放置了噪声降低系统。

[0010] 该系统有时被称为dbx噪声降低（在公司开发了这种技术之后），是压缩扩展类型的，包括编码器和解码器。该编码器在发送之前自适应地对差值信号进行滤波，从而使得在解码时振幅和频率内容隐藏了（“掩盖”）了在发送过程期间产生的噪声。所述解码器通过将差值信号恢复成原始形式并且由此确保噪声被信号内容在听觉上掩盖，从而完成处理。

[0011] dbx噪声减少系统还用于编码和解码第二音频程序（SAP）信号，其在BTSC标准中定义为附加信息声道，并且通常用于例如以可替换语言运载程序、读取用于遮蔽（blind）的服务或其他服务。

[0012] 当然，成本对于电视制造商来说是最主要的关注点。作为激烈竞争和消费者期望的结果，消费电子产品的利润率，特别是电视产品，可能非常小。因为解码器位于电视接收机中，制造商对于解码器的成本是敏感的，降低解码器的成本是必须的并且是值得力的

目标。虽然编码器不位于电视接收机中并且对于利润立足点来说并不敏感,任何将降低编码器的制造成本的开发也提供好处。

[0013] 发明内容

[0014] 根据本公开的一个方面,提供一种电视音频信号编码器,包括:求和设备,配置为将左声道音频信号和右声道音频信号相加,以生成和信号,并且将所述左和右音频信号中的一个信号从所述左和右信号中的另一个信号中减去,以生成差值信号;以及可配置的无限脉冲响应数字滤波器,配置为以比所述求和设备的时钟速度快的选择的时钟速度运行且选择性地多次使用至少一组滤波器系数来对所述差值信号进行滤波,其中,该至少一组滤波器系数在一递归循环内以递归方式由单个乘法器应用到所述差值信号,以便准备所述差值信号用于发送,其中,所述可配置的无限脉冲响应数字滤波器包括反馈通路且被配置为将该至少一组滤波器系数以递归方式应用到所述差值信号,且其中该至少一组滤波器系数在该递归循环的迭代之间可被改变。

[0015] 在一个实施例中,该反馈通路可以包括移位寄存器,用于延迟与差值信号关联的数字信号。该可配置的无限脉冲响应数字滤波器可以乘以与差值信号关联的信号,且提供该乘法的输出。该可配置的无限脉冲响应数字滤波器可以包括选择器,用于选择数字输入信号或选择其中一个滤波器系数。在一些设置中,该选择器可以包括多路复用器。该无限脉冲响应数字滤波器可以配置为提供不同的滤波功能,诸如低通滤波器。该可配置的无限脉冲响应数字滤波器还可以包括单个加法器,以递归的方式将滤波器系数应用到差值信号。该电视音频信号可以遵守广播电视系统委员会 (BTSC) 标准、准瞬时压扩音频复用 (NICAM) 标准、A2/Zweiton 标准、EIA-J 标准或者其他相似的音频标准。该可配置的无限脉冲响应数字滤波器可以实现在集成电路中。

[0016] 根据本公开的另一方面,提供一种电视音频信号解码器,包括:分离设备,配置为以一时钟速度运行且从差值信号以及和信号中分离左声道和右声道音频信号,其中,所述和信号包括所述左声道音频信号和右声道音频信号之和;及可配置的无限脉冲响应信号数字滤波器,配置为以比所述分离设备的时钟速度快的选择的时钟速度运行且选择性地多次使用至少一组滤波器系数来对差值信号进行滤波,其中,所述差值信号是通过将左声道和右声道音频信号中的一个信号从所述左声道和右声道音频信号中另一个信号中减去而得到的,其中,该至少一组滤波器系数在一递归循环中以递归方式由单个乘法器应用到所述差值信号,以便准备所述差值信号用于分离所述左声道和右声道音频信号,其中所述可配置的无限脉冲响应数字滤波器包括反馈通路且被配置为以递归方式将该至少一组滤波器系数应用到所述差值信号,且其中该至少一组滤波器系数在该递归循环的迭代之间可被改变。

[0017] 在另一个实施例中,该反馈通路可以包括移位寄存器,用于延迟与差值信号关联的数字信号。该可配置的无限脉冲响应数字滤波器可以乘以与差值信号关联的信号并且提供该乘法的输出。该可配置的无限脉冲响应数字滤波器可以包括选择器,用于选择数字输入信号或选择其中一个滤波器系数。在一些设置中,该选择器可以包括多路复用器。该无限脉冲响应数字滤波器可以配置为提供不同的滤波器功能,诸如低通滤波器。该可配置的无限脉冲响应数字滤波器也可以包括单个加法器,用于以递归的方式将该滤波器系数应用到差值信号。该电视音频信号可以遵守广播电视系统委员会 (BTSC) 标准、准瞬时压扩音频

复用 (NICAM) 标准、A2/Zweiton 标准、EIA-J 标准或其它相似音频标准。该可配置的无限脉冲响应数字滤波器可以实现在集成电路中。

[0018] 从下面的细节描述中,本公开的附加优点以及各个方面对于本领域技术人员将更加明显,其中简单地通过阐述预期用于实现本发明的最佳模式,来示出和描述本发明的实施例。如下所述,本公开可以是其他不同的实施例,并且其若干细节容许以各种显而易见的方式改动,所有都不脱离本公开的精神。因此,附图和说明在本质上应该看作是描述性的,而不是限制性的。

[0019] 附图说明

[0020] 图 1 是示出电视信号发送系统的框图,该系统配置为遵循 BTSC 电视音频信号系统;

[0021] 图 2 是示出包括在图 1 所示的电视信号发送系统中的 BTSC 编码器的一部分的框图;

[0022] 图 3 是示出配置为用于接收和解码由图 1 所示的电视信号发送系统所发送的 BTSC 电视音频信号的电视接收机系统的框图;

[0023] 图 4 是示出包括在图 3 所示的电视信号接收机系统中的 BTSC 解码器的一部分的框图;

[0024] 图 5 是用于执行图 2 和图 4 所示的编码器和解码器的操作的可配置的无限脉冲响应滤波器的简图;

[0025] 图 6 是二阶无限脉冲响应滤波器的传递函数的图形表示,该二阶无限脉冲响应滤波器可以由图 5 所示的无限脉冲响应滤波器实现;

[0026] 图 7 是 BTSC 编码器的一部分的框图,该 BTSC 编码器着重于可以由图 5 所示的可配置的无限脉冲响应滤波器执行的操作;以及

[0027] 图 8 是 BTSC 解码器的一部分的框图,该 BTSC 编码器着重于可以由图 5 所示的可配置的无限脉冲响应滤波器执行。

[0028] 具体实施方式

[0029] 参考图 1,兼容 BTSC 的电视信号发射机 10 的功能性框图包括 5 条线路(例如导线、电缆等等),提供用于发送的信号。特别地,左和右音频声道在各个线路 12 和 14 上提供。SAP 信号由线路 16 提供,其中该信号具有提供附加声道信息的内容(例如替换语言等等)。第四线路 18 提供专业声道,通常被广播电视和有线电视公司所使用。视频信号由线路 20 提供给发射机 22。左、右和 SAP 声道提供到 BTSC 编码器 24,该编码器 24 准备用于发送的音频信号。具体地,左和右音频声道提供到矩阵 26,该矩阵 26 从音频信号中计算出和信号(例如 L+R)以及差值信号(例如 L-R)。典型地,矩阵 26 的运算是利用数字信号处理器(DSP)或相似硬件或软件来执行的-基于对电视音频和视频信号处理领域技术人员来说公知的技术。一旦生成和信号和差值信号(即 L+R 和 L-R),就将其编码用于发送。具体地,和信号(即 L+R)提供到预加重(pre-emphasis)单元 28,该预加重单元 28 改变和信号中所选择频率分量相对于其他频率分量的振幅。该改变可以是负向意义的,其中对所选频率分量的振幅进行抑制,或者该改变可以是正向意义的,其中对所选频率分量的振幅进行加强。

[0030] 差值信号(即 L-R)提供到 BTSC 压缩器 30,该压缩器 30 在发送之前对该信号进行自适应滤波,从而使得当解码时,信号振幅和频率内容抑制了发送期间引入的噪声。与差

值信号类似, SAP 信号提供到 BTSC 压缩器 32。音频调制器级 34 接收所处理的和信号、差值信号以及 SAP 信号。另外, 来自专业声道的信号提供到音频调制器级 34。由音频调制器级 34 调制这四个信号并且将其提供到发射机 22。将这四个音频信号与视频通道所提供的视频信号一起调节为达到用于发送的条件, 并且将其提供到天线 36 (或天线系统)。各种对于电视系统和电信领域的技术人员来说公知的信号发送技术可以由发射机 22 和天线 36 实现。例如, 发射机 22 结合到有线电视系统、广播电视系统或其他相似的电视系统。

[0031] 参考图 2, 显示了表示由 BTSC 压缩器 30 的一部分所执行的操作的框图。通常, 由 BTSC 压缩器 30 执行的对于差值声道 (即 L-R) 的处理与预加重单元 28 执行的对于和声道 (即 L+R) 的处理相比复杂很多。由差值声道处理 BTSC 压缩器 30 所提供的附加处理, 与接收 BTSC 信号的解码器 (未显示) 所提供的反向 (complementary) 处理组合, 将差值声道的信噪比维持在可接受的级别上, 即使是存在与差值声道的发送和接收有关的更高噪声下限 (floor) 时也是如此。BTSC 压缩器 30 本质上通过动态地压缩或减小差值信号的动态范围来产生编码后的差值信号, 从而使得编码后信号可以通过有限动态范围的发送通路进行发送, 并且使得接收该编码后信号的解码器可以通过以反向方式扩展压缩的差值信号来恢复在原始差值信号中的基本上全部动态范围。在一些设置中, BTSC 压缩器 30 实施在此作为参考结合的美国专利 No. 4, 539, 526 中所描述的自适应信号加权系统的信号的特定形式, 并且该形式是公知的有利于通过具有相对较窄、频率相关的动态范围而发送具有相对较大动态范围的信号。

[0032] BTSC 标准严格地定义 BTSC 编码器 24 和 BTSC 解码器 30 和 32 的预期操作。具体地, BTSC 标准提供传递函数和 / 或对于包括在例如 BTSC 压缩器 30 中的每个组件的操作的指导, 并且传递函数用理想的模拟滤波器的数学表述方式来描述。一旦从矩阵 26 接收到差值信号 (即 L-R), 可以将该信号提供到内插与固定预加重 (interpolation and fixed pre-emphasis) 级 38。在一些数字 BTSC 编码器中, 内插设置为是采样速率的两倍并且内插可以通过线形内插、抛物插值法或 n 阶滤波器 (例如有限脉冲响应 (FIR) 滤波器、无限脉冲响应 (IIR) 滤波器等) 而完成。该内插与固定预加重级 38 还提供预加重。在内插和预加重之后, 将差值信号提供到分频器 40, 该分频器 40 通过从差值信号确定的量将差值信号分频, 并且在下面对其进行了详细描述。

[0033] 将分频器 40 的输出提供到频谱压缩单元 42, 频谱压缩单元 42 对差值信号进行加重滤波。通常, 频谱压缩单元 42 通过放大具有相对较低振幅的信号并削弱具有相对较大振幅的信号来“压缩”或减小差值信号的动态范围。在一些设置中, 频谱压缩单元 42 从差值信号中产生内部控制信号, 该内部控制信号控制所应用的预加重 / 去加重。典型地, 频谱压缩单元 42 动态压缩差值信号的高频部分, 压缩的量由所编码的差值信号的高频部分中的能量级所确定。频谱压缩单元 42 因此提供向着差值信号的更高频部分的附加信号压缩。这是因为差值信号在频谱的高频部分往往是具有更大噪音。当利用解码器中的频谱扩展器采用与编码器的频谱压缩单元相反的方式对编码后的差值信号进行解码时, L-R 信号的信噪比基本上能够保持。

[0034] 一旦由频谱压缩单元 42 进行了处理, 差值信号就提供给过调制 (over-modulation) 保护单元 44 以及限带单元 46。与其他元件相似的是, BTSC 标准对过调制保护单元 44 和限带单元 46 的操作提供了所建议的指导。通常, 限带单元 46 以及过调

制保护单元 44 的一部分可以实现为低通滤波器。过调制保护单元 44 也实现为用于将编码的差值信号限制到全调制的门限设备,全调制是对于调制电视信号中的伴音副载波的最大可允许背离程度。

[0035] 在 BTSC 压缩器 30 中包括两个反馈通路 48 和 50。反馈通路 50 包括频谱控制带通滤波器 52,该频谱控制带通滤波器 52 典型地具有相对较窄的通带,对所述通带向着更高音频频率进行加权,以便提供用于频谱压缩单元 42 的控制信号。为了使由频谱控制带通滤波器 52 产生的控制信号满足条件,反馈通路 50 还可以包括乘法器 54(配置为对频谱控制带通滤波器 52 所提供的信号进行平方)、积分器 56 以及用于提供控制信号给频谱压缩单元 42 的平方根设备。反馈通路 48 也包括带通滤波器(即,增益控制带通滤波器 60),该带通滤波器对来自限带单元 46 的输出信号进行滤波,以便设置经由分频器 40 施加到内插与固定预加重级 38 的输出信号上的增益。与反馈通路 50 相似,反馈通路 48 还包括乘法器 62、积分器 62 以及平方根设备 66,以便使提供给分频器 40 的信号达到条件。

[0036] 参考图 3,显示了表示电视接收机系统 68 的框图,该系统 68 包括天线 70(或天线的系统),以便从电视发送系统 10(如图 1 所示)接收兼容 BTSC 的广播信号。由天线 70 接收的信号提供给接收机 72,该接收机 72 能够检测并隔离电视发送信号。但是,在一些设置中,接收机 72 可以根据对于电视信号广播领域的技术人员公知的另一电视信号发送技术,接收兼容 BTSC 的信号。例如,该电视信号可以通过有线电视系统或卫星电视网络提供给接收机 72。

[0037] 一旦接收到该电视信号,接收机 72 使信号达到条件(例如放大、滤波、频率调整等)并且从发送信号中分离出视频信号和音频信号。视频内容被提供给视频处理系统 74,该视频处理系统 74 准备视频信号中包含的视频含量,以便在与电视接收机系统 68 关联的屏幕(例如阴极射线管等)上进行呈现。包含分离的音频内容的信号被提供给解调器级 76,该解调器级 76 例如移去在电视发送系统 10 上应用到音频信号上的调制。该解调的音频信号(例如 SAP 信号、专业信号、和信号、差值信号)被提供给 BTSC 解码器 78,该 BTSC 解码器 78 正确地解码每个信号。SAP 声道被提供 SAP 声道解码器 80,而专业声道被提供给专业声道解码器 82。在分离 SAP 声道和专业声道之后,解调后的和信号(即 L+R 信号)被提供给去加重单元 84,该去加重单元 84 采用与预加重单元 28(如图 1 所示)基本相反的方式处理该和信号。在对该和信号的频谱内容进行去加重之后,该信号被提供给矩阵 88,用于分离左和右声道音频信号。

[0038] 差值信号(即 L-R)也由解调制级 76 进行解调,并且被提供给包括在 BTSC 解码器 78 中的 BTSC 扩展器 86。BTSC 扩展器 86 遵循 BTSC 标准,并且如下面的细节所描述地,使差值信号达到条件。矩阵 88 从 BTSC 扩展器 86 接收差值信号,并且连同和信号将左和右音频声道分离成独立信号(在图 3 中标示为“L”和“R”)。通过分离所述信号,单个左和右声道音频信号可以达到条件并且提供给分离的扬声器。在这个示例中,左和右音频声道被提供给放大器级 90,该放大器级 90 在提供各个信号到用于广播左声道音频内容的扬声器 92 和用于广播右声道音频内容的扬声器 94 之前,将相同(或不同)增益应用到每个声道。

[0039] 参考图 4,框图标识了一些由 BTSC 扩展器 86 所执行的用于使差值信号达到条件的一些操作。一般地,BTSC 扩展器 86 执行与由 BTSC 压缩器 32(在图 2 中所示)执行的操作相反的操作。具体地,压缩的差值信号被提供给信号通路 96,用于解压缩该信号,并且提供

给两个通路 98 和 100,这两个通路产生相应的控制和增益信号以帮助对差值信号的处理。为了初始化该处理,压缩的差值信号被提供给限带单元 102,该限带单元 102 对该压缩的差值信号进行滤波。该限带单元 102 将一个信号提供给通路 98 以产生控制信号并且提供给通路 100 以产生增益信号。通路 100 包括增益控制带通滤波器 104、乘法器 106(将增益控制带通滤波器的输出进行平方)、积分器 108 以及平方根设备 110。信号通路 98 还从限带单元 102 接收信号并且利用频谱控制带通滤波器 112、平方设备 114、积分器 116 和平方根设备 118 处理该信号。通路 98 然后提供一个控制信号给频谱扩展单元 120,该频谱扩展单元 120 执行与由图 2 中所示的频谱压缩单元 42 所执行的操作相反的操作。由通路 100 所产生的增益信号被提供给乘法器 122,该乘法器 122 接收来自频谱扩展单元 120 的输出信号。乘法器 122 将频谱上扩展的差值信号提供给固定去加重单元 124,该固定去加重单元 124 采用与由 BTSC 压缩器 30 执行的滤波相反的方式对信号进行滤波。通常,术语“去加重”意味着所解码信号的所选择频率分量的改变,所述改变是采用与对原始信号的编码相反的方式在负向或正向进行的。

[0040] BTSC 编码器 24 和 BTSC 解码器 78 都包括多个滤波器,所述滤波器将音频信号的振幅作为频率的函数进行调节。在一些现有技术中,电视发送系统和接收系统,每个滤波器都采用分立的模拟元件实现。但是,由于数字信号处理中的优点,一些 BTSC 编码器和 BTSC 解码器可以在数字域中采用一个或多个集成电路(IC)实现。另外,多个数字 BTSC 编码器和/或解码器可以在单个 IC 上实现。例如,编码器和解码器可以被合并到作为超大规模集成电路(VLSI)系统的一部分单个 IC 中。

[0041] IC 的成本的一个重要部分是与芯片的物理尺寸直接成比例的,特别是其“死区”的尺寸或者芯片中活动的非封装部分。在一些设置中,在数字 BTSC 编码器和解码器中执行的滤波操作可以使用通用目的数字信号处理器执行,通用目的数字信号处理器设计为执行一系列的 DSP 功能和操作。这些 DSP 引擎倾向于具有相对较大的死区,并且因此用于较昂贵地实现 BTSC 编码器和解码器。此外,DSP 可以专用于执行其他功能和操作。通过共享资源,由 DSP 所执行的处理可以过载并且干扰 BTSC 编码器和解码器功能和操作的处理。

[0042] 在一些设置中,BTSC 编码器和解码器可以合并基本元件组以降低成本。例如,乘法器、加法器以及多路复用器的组可以进行合并,以便产生 BTSC 编码器和解码器的功能。但是,虽然基本相同的元件的组可以简单地进行制造,但是这些元件呈现出明显的死区并且增加了 IC 的总成本。因此,需要减少用于实现数字 BTSC 编码器和/或解码器的完全相同的电路元件的数量。

[0043] 参考图 5,显示了可配置的无限脉冲响应(IIR)滤波器 126 的框图,该滤波器可以执行多种用于数字 BTSC 编码器和/或解码器的操作。具体地,可配置的 IIR 滤波器 126 包括能够执行各种滤波、乘法以及延迟操作的数字结构。关于滤波操作,通过提供可选择的滤波系数,可配置的 IIR 滤波器 126 可以配置为用于各种类型的滤波器和不同的滤波操作。例如,可以选择滤波系数以提供低通滤波器、高通滤波器、带通滤波器或其他类型的滤波器设计领域公知的滤波器。因此,可配置 IIR 滤波器 126 的一个或相对较小数量的实现,可以用于提供 BTSC 编码器和解码器的大部分或全部的滤波需要。通过减少解码器和编码器滤波器的数量,IC 芯片的实现区域可以减小,同时 BTSC 编码器和解码器的生产称本也减少。可配置 IIR 滤波器 126 的其他实施方式在于 2005 年 3 月 24 日提交的美国专利申请号

No.11/089,385,“Configurable Filter for Processing Television Audio Signals”中进行了描述,其结合在此作为参考。

[0044] 与对用于选择滤波器系数的元件的使用的同时,通过使用递归的数字结构,可以进一步减少元件的数量。在这个示例性设计中,可配置的 IIR 滤波器 126 包括反馈通路 128,该反馈通路 128 将数字信号从该结构的输出部分传送到用于进一步处理的元件。通过经由反馈通路 128 传送所处理的数字信号,可以由可配置的 IIR 滤波器 126 提供各种类型的递归处理。例如,通过经由反馈通路 128 传送信号,可以实现更高阶的滤波器(例如二阶或更高)。

[0045] 在这个实现中,各种数字输入信号提供给充当选择器的多路复用器 130 的输入端。例如,信号可以从诸如 BTSC 压缩器 30(如图 2 所示)的压缩器的各种部分输入。内插与固定预加重级 38、增益控制带通滤波器 60 和频谱控制带通滤波器 52 可以提供数字信号至多路复用器 130。根据适当的调度,多路复用器 130 选择一个输入用于处理适当的输入信号。所选择的信号被提供给输入寄存器 132,然后在适当的时间提供给多路复用器 134。多路复用器 134 将来自输入寄存器 132 的数据(例如新的输入数据)或者来自单个乘法器 138 的先前计算的乘积(经由乘积寄存器 140)提供给单个加法器 136。加法器 136 还接收来自多路复用器 142 的输入数据,该数据可以是来自和寄存器 144(优选地连接到加法器 136 的输出端)的先前积累的数据或者是来自乘法器 138 的乘积数据(优选地经由乘积寄存器 140 和寄存器 146 提供)。

[0046] 为了提供用于对先前处理的信号进行处理和递归处理的数字输入信号,反馈通路 128 将加法器 136 的输出提供给乘法器 138。更具体地,加法器 136 的输出被提供给多路复用器 148,该多路复用器 148 提供输出信号给移位寄存器 150。在加法器 136 的输出信号和(从移位寄存器 150 输出的)一个信号的延迟形式中的一个被提供给移位寄存器 150 的输入端。通过在反馈通路 128 中加入移位寄存器 150,可以在在乘法器 138 的处理之前将一个时间延迟应用到数字信号。为了滤波应用,由移位寄存器 150 引入的时间延迟可以用于实现更高阶滤波器(例如二阶滤波器)。

[0047] 移位寄存器 150 的输出被提供给多路复用器 148 的输入端(如上所述)。反馈通路 128 通过多路复用器 152 将数据提供给乘法器 138。更具体地,数字信号可以经由导线 154 直接地从加法器 136 的输出端反馈。信号也可以是作为移位寄存器 150 的输出端所提供的反馈或者移位寄存器 150 的输出的延迟形式(经由寄存器 156)所提供的反馈。外部被乘数也可以被提供到多路复用器的多个输入端 158。如图所示,外部数据可以被提供到多路复用器 152 的一个或多个输入线路 158。寄存器 160 被提供给来自多路复用器 152 的输出信号,用于准备乘法器 138 的乘法。

[0048] 诸如滤波器系数(具有固定的或变化的值)之类的数据可以由多路复用器 162 提供给可配置的 IIR 滤波器 126。具体地,表示滤波器系数的数据可以从输入线路 164 提供给多路复用器 162。外部被乘数也可以由输入线路 164 提供。随着从外部提供所述系数或被乘数,所述系数或被乘数可以由寄存器 166 提供给多路复用器 162。与多路复用器 152 相似,多路复用器 162 提供数据到寄存器 168,用于准备向乘法器 138 提供数据。

[0049] 由于反馈通路 128 包括在可配置的 IIR 滤波器 126 中,因此可以引入单个乘法器(即乘法器 138),以便在其中提供乘法功能用于实现滤波器。通过实现该单个乘法器方案,

可以保存集成电路的实时状态并将其用于提供其他功能性。例如,可以引入一系列的输出寄存器,以便直接提供乘积寄存器 140 的输出给外部设备和元件。另外,由于反馈通路 128,单个加法器(即加法器 136)提供附加功能以便实现各种类型的 IIR 滤波器。此外,通过使用单个元件,在加法器 136 的情况下,保存了附加芯片的实时状态以将其用于其他元件。例如,可以引入一系列输出寄存器 172 以便引导加法器 136 的输出(经由和寄存器 144)到位于相同集成电路或外部设备的外部元件或模块。

[0050] 除了提供乘法功能(利用由输出寄存器 170 所提供的输出)以及滤波功能(利用由输出寄存器 172 所提供的输出)之外,可配置的 IIR 滤波器 126 还可以提供时间延迟功能。例如,移位寄存器 150 的输出和/或寄存器 156 的输出可以用于提供被提供给寄存器的一个或多个数字信号的时间延迟形式。

[0051] 为了允许可配置的 IIR 滤波器 126 执行多种类型的滤波操作,多路复用器 130 控制由哪个输入端提供输入信号。简要参考图 2,可以连接多路复用器 130 的一些输入端,以便对于在 BTSC 压缩器 30 中执行的每个滤波操作提供输入信号。例如,至增益控制带通滤波器 60 的输入可以连接到多路复用器 130 的输入端。相似地,至频谱控制带通滤波器 52 的输入可以被连接到多路复用器 130 的另一输入端。然后,多路复用器 130 可以控制可配置的 IIR 滤波器 126 要执行哪个特定滤波操作。例如,在一个时间周期中,可以选择适当输入端并且可配置的 IIR 滤波器 126 可以配置为提供增益控制带通滤波器 60 的滤波器功能。然后,在另一时间周期中,多路复用器 130 可以用于选择另一个输入端以便执行不同的滤波操作。在选择另一输入的同时,可配置的 IIR 滤波器 126 可以相应地配置为提供不同类型的滤波功能,诸如由频谱控制带通滤波器 52 提供的滤波。

[0052] 为了执行多个滤波操作,例如为了 BTSC 压缩器或 BTSC 扩展器的滤波操作,可配置的 IIR 滤波器 126 以基本比数字压缩器或扩展器的其他部分更快的时钟速度进行工作。通过以更快的时钟速度工作,可配置的 IIR 滤波器 126 可以执行一种类型的滤波而不引起数字压缩器或扩展器其他操作被延迟。例如,通过以基本上很快时钟速度操作可配置的 IIR 滤波器 126,该结构可以首先配置为执行用于增益控制带通滤波器 60 的滤波,而基本上不会对下一个滤波器配置的执行(例如用于频谱控制带通滤波器 52 的滤波器操作)产生延迟。

[0053] 在一个设置中,可配置的 IIR 滤波器 126 可以实现为二阶 IIR 滤波器。参考图 6,对于典型的二阶 IIR 滤波器显现出 z 域信号流图 174。输入节点 176 接收标识为 $X(z)$ 的输入信号。该输入信号被提供给加法器 178,该加法器 178 将该信号与下面描述的处理后的信号相加。加法器 178 的输出被提供到增益级 180,该增益级 180 将滤波器系数 a_0 应用到输入信号。在一些应用中,滤波器系数 a_0 具有单一值。相似地,在增益级 182 处将滤波器系数 b_0 应用到输入信号。在延迟级 184 处,当输入信号进入到滤波器的第一阶部分时应用一个时间延迟(即在 z 域中表示为 z^{-1}),并且在各个增益级 186 和 188 应用滤波器系数 a_1 和 b_1 。在延迟级 190 处应用第二延迟(即 z^{-1}),用于生成滤波器 174 的第二阶部分,并且在各个增益级 192 和 194 处应用滤波器系数 a_2 和 b_2 。各个加法器 196、198 和 200 将来自多个增益级的信号相加,并且将滤波后信号提供给输出节点 202,从而使得输出信号 $Y(z)$ 可以从二阶滤波器 174 的传递函数 $H(z)$ 中确定,如下等式 (1) 所示:

$$[0054] \quad H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

[0055] 可以对包含在传递函数中的每个系数（即 b_0 、 a_0 、 b_1 、 a_1 、 b_2 和 a_2 ）分配特定值，以生成期望类型的滤波器。例如，可以分配特定值给各个系数以生成低通滤波器、高通滤波器或带通滤波器等等。因此，通过为每个系数提供适当值，二阶滤波器的类型和特性（例如通带、转降（roll-off）等等）可以配置并且重新配置成具有不同的系数组的另一种类型的滤波器（根据应用）。虽然该实施例描述了二阶滤波器，但是在其他设置中可以实现 n 阶滤波器。例如，可以实现更高阶（例如三阶、四阶等）滤波器或更低阶（例如一阶）滤波器。另外，对于一些应用来说，可配置的 IIR 滤波器 126 的递归数字结构可以级联，以生成 n 阶滤波器。

[0056] 回来参考图 5，在使用多路复用器 130 为可配置的 IIR 滤波器 126 选择特定输入的同时，选择了该滤波器使用的系数，以实现不同类型的滤波器并且提供特定滤波器特性。例如，可以选择系数以实现用于编码或解码 BTSC 音频信号的低通滤波器、高通滤波器、带通滤波器或其他相似类型的滤波器。由于反馈通路 128 所提供的递归处理，不同的系数或系数组可以由多路复用器 152 和 / 或多路复用器 162 进行选择。通过为不同的递归迭代选择不同的系数，可以实现各种滤波器。例如，可以控制多路复用器 162 选择与二阶滤波器关联的滤波器系数（例如 a_0 、 b_0 、 a_1 、 b_1 等）。然后，为了下一个迭代，多路复用器 162 可以选择另一滤波器系数。通过提供这些可选择的系数值，可配置的 IIR 滤波器 126 可以配置为提供用于编码和解码操作的滤波器。一旦以递归方式完成用于一个应用（例如增益控制带通滤波器 60）的滤波，多路复用器 130 就可以设置在用于提供输入信号给另一应用（例如频谱控制带通滤波器 52）的位置。通过选择该输入，可以由多路复用器 162 和 / 或多路复用器 152 选择新的滤波器系数，以便提供执行下一个应用的滤波所需要的特定滤波器类型和滤波器特性。

[0057] 在图 6 中显示的示例中，可配置的 IIR 滤波器 126 配置为二阶滤波器，但是一些编码和 / 或解码滤波应用可以要求更高阶的滤波器。为了提供更高阶的滤波器，可以通过反馈通路 128 来执行附加的递归迭代。通过使用该反馈通路，信号可以使用相同（或不同）的滤波器系数多次通过 IIR 滤波器。因此，可以采用单个乘法器（即乘法器 138）和单个加法器（即加法器 136）执行用于各种类型的滤波器以及各种阶的滤波器实现的滤波操作。为了说明由可配置的 IIR 滤波器 126 执行的迭代，示出了数字指示符（即 1、2、3、4、5）以表示其中执行每个功能的各个时钟周期。在这种附图说明中，以序列：1、2、3、4、5 来执行这些功能。因此需要五个时钟周期来计算二阶滤波器的输出。另外，这个执行功能的序列可以以循环的方式重复（例如 1、2、3、4、5、1、2、3、4、5 等等）。

[0058] 对于电子和滤波器设计领域的技术人员来说公知的各种技术和元件可以用于实现所述多路复用器（例如多路复用器 130、152、162 等）。例如多路复用器 130 可以采用一个或多个多路复用器来实现以便在多个输入中进行选择。除了多路复用器，其他类型的数字选择设备也可以采用以便选择合适的滤波器系数。可以选择各种系数值来配置 IIR 滤波器，例如 IIR 滤波器 174。例如，在 Hanna 的美国专利 5,796,842 中所描述的系数可以由可配置的 IIR 滤波器 126 所使用，本专利在此引用作为参考。在一些设置中，滤波器系数存储在与 BTSC 编码器或解码器关联的存储器（未显示）中，并且由适当的多路复用器在适当的

时间取回。例如,这些系数可以存储在与 BTSC 编码器和解码器关联的存储器芯片(例如随机访问存储器(RAM)、只读存储器(ROM)等)或者其他类型的存储设备(例如硬盘驱动器、CD-ROM等)中。这些系数还可以存储在各种软件结构中,例如查找表或其他类似的结构中。

[0059] 可配置的 IIR 滤波器 126 包括单个加法器 136 以及单个乘法器 138。对于电子电路设计和数字设计领域的技术人员来说公知的各种技术和 / 或元件可以用于实现包括在可配置的 IIR 滤波器 126 中的加法器 136 和乘法器 138。例如,逻辑门诸如一个或多个“与”门,可以实现为每个乘法器。为了引入时间延迟,对于电子电路设计和数字设计领域的技术人员来说公知的各种技术和 / 或元件可以被采用以产生移位寄存器 150(在图 5 中所示)并且通过将数字化的输入信号值存储并保持适当数量的时钟周期,来提供延迟。

[0060] 在这个实施例中,可配置的 IIR 滤波器 126 利用硬件元件实现,但是在一些设置中,该结构的一个或多个操作部分可以用软件实现。在附件 A 中列出了用于执行可配置的 IIR 滤波器 126 的操作的代码的一个示例性列表。该示例性代码是用 Verilog 提供的,Verilog 一般是在制造之前由电子设计者使用来描述和设计芯片和系统的硬件描述性语言。该代码可以存储在存储设备(例如 RAM、ROM、硬盘驱动器、CD-ROM 等)并且可以从该存储设备中取回,并且在一个或多个通用目的处理器和 / 或专用处理器(诸如专用 DSP)上执行。

[0061] 参考图 7,提供 BTSC 压缩器 30 的框图,其中,图中高亮的部分示出可以由可配置的 IIR 滤波器 126 的单个(或多个)实现所执行的功能。具体地,由内插与固定预加重级 38 执行的滤波可以由可配置的 IIR 滤波器 126 所执行。例如,多路复用器 130 的输入端可以连接到内插与固定加重级 38 中的适当的滤波器的输入端。相应地,当多路复用器 130 的该输入被选择时,可以从存储器中取回滤波器系数并用于生成适当的滤波器类型和滤波器特性。相似地,增益控制带通滤波器 60 可以分配到数字可配置的 IIR 滤波器 126 中的多路复用器 130 的另一输入端,并且频谱控制带通滤波器 52 可以被分配到多路复用器 130 的又一输入端。对于这些可选择的输入端的每一个来说,存储了相应的滤波器系数(例如在存储器中)并且可以由可配置的 IIR 滤波器 126 的多路复用器 152 和 / 或多路复用器 162 取回所述系数。在这个实施例中,与 BTSC 压缩器 30 的这四部分关联的滤波可以由可配置的 IIR 滤波器 126 选择性地执行,但是在其他设置中,压缩器的更多或更少的滤波操作可以由可配置的 IIR 滤波器所执行。另外,可配置的 IIR 滤波器 126 还经由乘法器 138 和输出寄存器 170(在图 5 中所示)提供乘法功能。因此,乘法器 54 和 62 的操作都可以由可配置的 IIR 滤波器 126 提供。

[0062] 参考图 8,BTSC 扩展器 86 中的高亮部分标识可以由一个或多个可配置的 IIR 滤波器所执行的滤波操作,所述 IIR 滤波器可以利用可配置的 IIR 滤波器 126 实现。例如,与限带单元 102 关联的滤波可以由可配置的 IIR 滤波器 126 执行。具体地,多路复用器 130 的一个输入端可以分配给限带单元 102,从而使得当选择该输入时,取回适当的滤波系数并且由可配置的 IIR 滤波器 126 使用这些系数。相似地,与增益控制带通滤波器 104(分配给多路复用器 130 的另一输入端)、频谱控制带通滤波器 112(分配给多路复用器 30 的另一输入端)以及固定去加重单元 124(分配给选择器 130 的又一个输入端)关联的滤波可以合并到可配置的 IIR 滤波器 126 中。另外,由于可配置的 IIR 滤波器 126 的乘法功能,可配置的 IIR 滤波器 126 可以为一个或多个乘法器 106、114 和 122 提供乘法功能。

[0063] 虽然前面所述的实施例使用可配置的 IIR 滤波器 126 用于 BTSC 编码器和 BTSC 解码器,遵循电视音频标准的编码器和解码器可以实现可配置 IIR 滤波器。例如,与在欧洲使用的准瞬时压扩音频复用 (NICAM) 关联的编码器和 / 或解码器可以结合一个或多个可配置的 IIR 滤波器,诸如 IIR 滤波器 126。相似地,实施 A2/Zweiton 电视音频标准 (当前在欧洲和亚洲部分使用的) 或日本电子工业联盟 (EIA-J) 标准的编码器和解码器可以结合一个或多个可配置的 IIR 滤波器。

[0064] 虽然前面所述的实施例采用可配置的 IIR 滤波器 126 编码和解码从右和左音频声道产生的差值信号,该可配置的 IIR 滤波器也可以用于编码和解码其他音频信号。例如,可配置 IIR 滤波器 126 可以用于编码和 / 或解码 SAP 声道、专业声道、和声道或一个或多个其他单个或合并类型的电视音频声道。

[0065] 已经描述了多个实施例。但是,应该理解可以进行各种修改。因此,其他实现也在后面的权利要求的范围中。

[0066] 附录 A

[0067] Appendix A

[0068] /*****

[0069] This module is the cascaded direct-form II implementation of

[0070] one or more discrete-time filters.It is actually a single

[0071] second-order section that can be ' recycled' .

[0072] *****/

[0073] // Generated by ParallelSOSFilterGenerator.m on 03-Jun-2004 08:59:29

[0074] //

[0075] // Delay Register Width = 35

[0076] // Filter Coefficients are in Q15

[0077] // ***** Filter 1 Sum Interpolation & Precmphasis *****

[0078] // This filter is in Q15 format.(Q15 is required.)

[0079] // The ' b' coefficients in the last stage are in Q18 format.

[0080] //

[0081] // -----(192kHz Sample Rate)-----

[0082] // Max Delay Register Value = 127.936----->99.976% headroom

[0083] // Max Output Value = 23.781----->99.964% headroom

[0084] //

[0085] //

[0086] // b0 b1 b2 a0 a1 a2

[0087] // -----

[0088] // -6.6033712e-012 3.2117006e-002 1.3600689e+000
1.0000000e+000-1.5758348e+000 6.2789060e-001

[0089] // 2.0305836e-002 9.3963698e-002 6.5186177e-002

```

1.0000000e+000-1.5671917e+000 6.7399133e-001
[0090] // 1.1605752e-001 2.4121539e-002 3.7646657e-004
1.0000000e+000-1.5759558e+000 7.6826398e-001
[0091] // 4.8703374e-001 -4.5435961e-001 0.0000000e+000
1.0000000e+000-1.6429813e+000 9.1158385e-001
[0092] //
[0093] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0094] // Section Del.max Outmax
[0095] // -----
[0096] // 1 9.9273109932813 13.4026919779221
[0097] // 2 127.9360639360639 22.7268252592786
[0098] // 3 127.9360639360638 17.9496934617571
[0099] // 4 127.9360639360638 23.7933114009230
[0100] //
[0101] // ***** Filter 2 Sum Lowpass
*****
[0102] // This filter is in Q15 format(Q15 is required.)
[0103] // The ' b' coefficients in the last stage are in Q17 format.
[0104] //
[0105] // -----(192kHz Sample Rate)-----
[0106] // Max Delay Register Value = 268.666 ----->99.949% headroom
[0107] // Max Output Value = 4.086 ----->99.997% headroom
[0108] //
[0109] //
[0110] // b0 b1 b2 a0 a1 a2
[0111] // -----
--

[0112] // 8.4345402e-001 3.3105654e-001 8.4345402e-001 1.0000000e+000
-1.5131160e+000 5.8400921e-001
[0113] // 1.7907148e-001 -2.3574553e-001 1.7907148e-001 1.0000000e+000
-1.5854796e+000 7.0937033e-001
[0114] // 3.0401514e-001 -4.9151080e-001 3.0401514e-001 1.0000000e+000
-1.6621327e+000 8.4198754e-001
[0115] // 3.5735700e-001 -6.0794041e-001 3.5735700e-001 1.0000000e+000
-1.71074Z9e+000 9.2558392e-001
[0116] // 2.6136244e-001 -4.5235123e-001 2.6136243e-001 1.0000000e+000
-1.7366273e+000 9.6885152e-001
[0117] // 3.3644153e-001 -5.8564210e-001 3.3644153e-001 1.0000000e+000

```

-1.7519173e+000 9.9157326e-001

[0118] //

[0119] // Magnitudes are relative to the system input, not necessarily the filter input.

[0120] // Secuon Del.max Outmax

[0121] // -----

[0122] // 1 14.8950457230439 30.0525971031239

[0123] // 2 268.6657342657337 32.1765299950301

[0124] // 3 268.6657342657345 26.2168014264289

[0125] // 4 268.6657342657350 17.3374523088056

[0126] // 5 268.6657342657346 7.0533046891134

[0127] // 6 268.6657342657326 4.0876878442666

[0128] //

[0129] // ***** Filter 3 Diff Interpolation & Preemphasis *****

[0130] // This filter is in Q15 format(Q15 is required.)

[0131] // The ' b' coefficients in the last stage are in Q12 format.

[0132] //

[0133] // -----(192kHz Sample Rate)-----

[0134] // Max Delay Register Value = 255.872----->99.951% headroom

[0135] // Max Outout Value = 259.440----->99.994% headroom

[0136] //

[0137] //

[0138] // b0 b1 b2 a0 a1 a2

[0139] // -----

[0140] // -1.2723363e-010 4.8662511e-002 3.5810704e+000 1.0000000e+000
-9.7203440e-001 1.0899437e-001

[0141] // 7.2682533e-003 4.9327184e-002 5.0728513e-002 1.0000000e+000
-1.5758348e+000 6.2789060e-001

[0142] // 1.3165964e-001 5.6960961e-002 3.6425943e-003 1.0000000e+000
-1.5671917e+000 6.7399133e-001

[0143] // 2.4084812e-001 1.3833431e-003 0.0000000e+000 1.0000000e+000
-1.5759558e+000 7.6826398e-001

[0144] // 9.4429177e+000 -1.8107524e+001 8.6732681e+000 1.0000000e+000
-1.6429813e+000 9.1158385e-001

[0145] //

[0146] // Magnitudes are relative to the system input, not necessarily the filter input.

```

[0147] // Section      Del.max      Outmax
[0148] // -----      -----      -----
[0149] // 1      4.4056120909651      12.2436103717271
[0150] // 2      255.8721278721281      27.3647542209154
[0151] // 3      255.8721278721284      49.1968802729879
[0152] // 4      255.8721278721282      61.9802894946887
[0153] // 5      255.8721278721280      259.5699347572192
[0154] //
[0155] // ***** Filter 4 Diff Gain Ctrl Bandpass
*****
[0156] // This filter is in Q17 format(Q17 is reqmred)
[0157] // The ' b' coefficients in the last stage are in Q21 format.
[0158] //
[0159] // -----(192kHz Sample Rate)-----
[0160] // Max Delay Register Value = 13051.645----->90.042% headroom
[0161] // Max Output Value      = 1.915----->99.907% headroom
[0162] //
[0163] //
[0164] //      b0      b1      b2      a0      a1      a2
[0165] // -----
-
[0166] // 3.3052890e-002 0.0000000e+000 -3.3052890e-002 1.0000000e+000
-1.9327087e+000 9.3278529e-001
[0167] //
[0168] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0169] // Section      Del.max      Outmax
[0170] // -----      -----      -----
[0171] // 1      13051.6450743394910      1.9154790026336
[0172] //
[0173] // ***** Filter 5 Diff Gain Ctrl Integrator
*****
[0174] // This filter is in Q13 format.(Q13 is required.)
[0175] // The ' b' coefficients in the last stage are in Q28 format
[0176] //
[0177] // -----(192kHz Sample Rate)-----
[0178] // Max Delay Register Value = 1499902.076----->28.479% headroom
[0179] // Max Output Value      = 225.000----->12.109% headroom
[0180] //

```

```
[0181]  //
[0182]  //      b0          b1          b2          a0          a1          a2
[0183]  //-----
[0184]  //  7.5042399e-005  7.5042399e-005  0.0000000e+000  1.0000000e+000
-9.9984992e-001  0.0000000e+000
[0185]  //
[0186]  // Magnitudes are relative to the system input, not necessarily the
filter input.
[0187]  // Section      Del.max          Outmax
[0188]  // -----      -----          -----
[0189]  //  1          1499902.0762455594000          225.1124999993376
[0190]  //
[0191]  // ***** Filter 6 Diff Spec Ctrl Bandpass
*****
[0192]  // This filter is in Q17 format(Q17 is required.)
[0193]  // The ' b' coefficients in the last stage are in Q18 format.
[0194]  //
[0195]  // -----(192kHz Sample Rate)-----
[0196]  // Max Delay Register Value = 39.333----->99.970% headroom
[0197]  // Max Output Value          = 3.683----->99.978% headroom
[0198]  //
[0199]  //
[0200]  //      b0          b1          b2          a0          a1          a2
[0201]  // -----
---
[0202]  //  8.6691012e-001 -1.7351060e+000  8.6819671e-001  1.0000000e+000
-1.7132232e+000  7.6815543e-001
[0203]  //  3.3300661e-001  1.8508507e-004 -3.3343983e-001  1.0000000e+000
-1.2652200e+000  3.3953192e-001
[0204]  //
[0205]  // Magnitudes are relative to the system input, not necessarily the
filter input.
[0206]  // Section      Del.max          Outmax
[0207]  // -----      -----          -----
[0208]  //  1          39.3329526574003          3.7578420694186
[0209]  //  2          14.8039499335385          3.6850497626675
[0210]  //
[0211]  // ***** Filter 7 Diff Spec Ctrl Integrator
*****
```

```
[0212] // This filter is in Q22 format. (Q22 is required)
[0213] // The ' b' coefficients in the last stage are in Q27 format.
[0214] //
[0215] // -----(192kHz Sample Rate)-----
[0216] // Max Delay Register Value = 1949.451----->52.406% headroom
[0217] // Max Output Value          = 0.684----->31.633% headroom
[0218] //
[0219] //
[0220] //      b0          b1          b2          a0          a1          a2
[0221] // -----
--
[0222] // 1.7543809e-004 1.7543809e-004 0.0000000e+000 1.0000000e+000
-9.9954323e-001 0.0000000e+000
[0223] //
[0224] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0225] // Section      Del.max          Outmax
[0226] // -----      -----          -----
[0227] // 1          1949.4512384999191          0.6840160013750
[0228] //
[0229] // ***** Filter 8 Diff Spectral Compression
*****
[0230] // This filter is in Q15 format. (Q15 is required.)
[0231] // The ' b' coefficients in the last stage are in Q11 format.
[0232] //
[0233] // -----(192kHz Sample Rate)-----
[0234] // Max Delay Register Value = 4819.452----->99.081% headroom
[0235] // Max Output Value          = 4880.066----->99.942% headroom
[0236] //
[0237] //
[0238] //      b0          b1          b2          a0          a1          a2
[0239] // -----
-
[0240] // 3.8625515e+001 -3.8130535e+001 0.0000000e+000 1.0000000e+000
-9.8671384e-001 0.0000000e+000
[0241] //
[0242] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0243] // Section      Del.max          Outmax
```

```

[0244] // -----
[0245] // 1      4819.4523948899350      4882.5058955489840
[0246] //
[0247] // ***** Filter 9 Diff Lowpass
*****
[0248] // This filter is in Q15 format. (Q15 is required.)
[0249] // The ' b' coefficients in the last stage are in Q17 format.
[0250] //
[0251] // -----(192kHz Sample Rate)-----
[0252] // Max Delay Register Value = 255.872----->99.951% headroom
[0253] // Max Output Value      = 3.891----->99.997% headroom
[0254] //
[0255] //
[0256] //      b0      b1      b2      a0      a1      a2
[0257] // -----
-----

[0258] // 8.4345402e-001  3.3105654e-001  8.4345402e-001  1.0000000e+000
-1.5131160e+000  5.8400921e-001
[0259] // 1.7907148e-001 -2.3574553e-001  1.7907148e-001  1.0000000e+000
-1.5854796e+000  7.0937033e-001
[0260] // 3.0401514e-001 -4.9151080e-001  3.0401514e-001  1.0000000e+000
-1.6621327e+000  8.4198754e-001
[0261] // 3.5735700e-001 -6.0794041e-001  3.5735700e-001  1.0000000e+000
-1.7107429e+000  9.2558392e-001
[0262] // 2.6136244e-001 -4.5235123e-001  2.6136243e-001  1.0000000e+000
-1.7366273e+000  9.6885152e-001
[0263] // 3.3644153e-001 -5.8564210e-001  3.3644153e-001  1.0000000e+000
-1.7519173e+000  9.9157326e-001
[0264] //
[0265] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0266] // Section      Del. max      Outmax
[0267] // -----
[0268] // 1      14.1857578314703      28.6215210505942
[0269] // 2      255.8721278721279      30.6443142809810
[0270] // 3      255.8721278721285      24.9683823108847
[0271] // 4      255.8721278721286      16.5118593417196
[0272] // 5      255.8721278721287      6.7174330372509
[0273] // 6      255.8721278721255      3.8930360421587

```

```

[0274] //
[0275] // ***** Filter 10 SAP Interpolation &
Precmphasis *****
[0276] // This filter is in Q15 format.(Q15 is requited.)
[0277] // The ' b' coefficients in the last stage are in Q12 format.
[0278] //
[0279] // -----(192kHz Sample Rate)-----
[0280] // Max Delay Register Value = 127.936----->99.976% headroom
[0281] // Max Output Value      = 79.936----->99.998% headroom
[0282] //
[0283] //
[0284] //      b0      b1      b2      a0      a1      a2
[0285] // -----
---
[0286] // -1.8231226e-009 2.1884032e-002 1.7219441e+000 1.0000000e+000
-9.7203440e-001 1.0899437e-001
[0287] //  3.1584799e-003 2.2857938e-002 2.4903862e-002 1.0000000e+000
-1.7083043e+000 7.3325661e-001
[0288] //  6.0720051e-002 2.7399739e-002 1.8184569e-003 1.0000000e+000
-1.7179059e+000 7.6872240e-001
[0289] //  9.5595731e-002 5.6899673e-004 0.0000000e+000 1.0000000e+000
-1.7487378e+000 8.3882941e-001
[0290] //  1.0900737e+001 -2.0903005e+001 1.0012267e+001 1.0000000e+000
-1.8173075e+000 9.4015130e-001
[0291] //
[0292] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0293] // Section      Del.max      Outmax
[0294] // -----      -----      -----
[0295] // 1      2.2028060429820      2.94148557345591
[0296] // 2      127.9360639360640      6.5017997596439
[0297] // 3      127.9360639360640      11.5065707750231
[0298] // 4      127.9360639360641      12.3029392557771
[0299] // 5      127.9360639360646      79.9758599046644
[0300] //
[0301] // ***** Filter 11 SAP Gain Ctrl Bandpass
*****
[0302] // This filter is in Q17 format(Q17 is required.)
[0303] // The ' b' coefficients in the last stage are in Q21 format.

```

```
[0304] //
[0305] // -----(192kHz Sample Rate)-----
[0306] // Max Delay Register Value = 13051.591----->90.042% headroom
[0307] // Max Output Value      = 1.984----->99.903% headroom
[0308] //
[0309] //
[0310] //      b0      b1      b2      a0      a1      a2
[0311] // -----
--

[0312] // 3.3052890e-002 0.0000000e+000 -3.3052890e-002 1.0000000e+000
-1.9327087e+000 9.3278529e-001
[0313] //
[0314] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0315] // Section      Del.max      Outmax
[0316] // -----      -----      -----
[0317] // 1      13051.5913591675250      1.9854692831948
[0318] //
[0319] // ***** Filter 12 SAP Gain Ctrl Integrator
*****
[0320] // This filter is in Q13 format. (Q13 is required.)
[0321] // The ' b' coefficients in the last stage are in Q28 format.
[0322] //
[0323] // -----(192kHz Sample Rate)-----
[0324] // Max Delay Register Value = 1499902.076----->28.479% headroom
[0325] // Max Output Value      = 225.000----->12.109% headroom
[0326] //
[0327] //
[0328] //      b0      b1      b2      a0      a1      a2
[0329] // -----
--

[0330] // 7.5042399e-005 7.5042399e-005 0.0000000e+000 1.0000000e+000
-9.9984992e-001 0.0000000e+000
[0331] //
[0332] // Magnitudes are relative to the system input, not necessarily the
filter input
[0333] // Section      Del.max      Outmax
[0334] // -----      -----      -----
[0335] // 1      1499902.0762455594000      225.1124999993376
```

```

[0336] //
[0337] // ***** Filter 13 SAP Spec Ctrl Bandpass
*****
[0338] // This filter is in Q17 format. (Q17 is required.)
[0339] // The ' b' coefficients in the last stage ate in Q18 format.
[0340] //
[0341] // -----(192kHz Sample Rate)-----
[0342] // Max Delay Register Value = 60.285----->99.954% headroom
[0343] // Max Output Value      = 3340----->99980% headroom
[0344] //
[0345] //
[0346] //      b0      b1      b2      a0      a1      a2
[0347] // -----
-----

[0348] // 8.6691012e-001 -1.7351060e+000  8.6819671e-001 1.0000000e+000
-1.7132232e+000 7.6815543e-001
[0349] // 3.3300661e-001  1.8508507e-004 -3.3343983e-001 1.0000000e+000
-1.2652200e+000 3.3953192e-001
[0350] //
[0351] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0352] // Section      Del.max      Outmax
[0353] // -----      -----      -----
[0354] // 1      60.2854167958472      3.3663545412242
[0355] // 2      18.1337922344007      3.3413388460805
[0356] //
[0357] // ***** Filter 14 SAP Spec Ctrl Integrator
*****
[0358] // This filter is in Q22 format. (Q22 is required.)
[0359] // The ' b' coefficients in the last stage are in Q27 format.
[0360] //
[0361] // -----(192kHz Sample Rate)-----
[0362] // Max Delay Register Value = 1949.451----->52.406% headroom
[0363] // Max Output Value      = 0.684----->31.633% headroom
[0364] //
[0365] //
[0366] //      b0      b1      b2      a0      a1      a2
[0367] // -----
-----

```

```

[0368] // 1.7543809e-004 1.7543809e-004 0.0000000e+000 1.0000000e+000
-9.9954323e-001 0.0000000e+000
[0369] //
[0370] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0371] // Section      Del.max      Outmax
[0372] // -----      -----      -----
[0373] // 1      11949.4512384999191      0.6840160013750
[0374] //
[0375] // ***** Filter 15 SAP Spectral
Cormpression *****
[0376] // This filter is in Q15 format. (Q15 is required.)
[0377] // The ' b' coefficients in the last stage are in Q11 format.
[0378] //
[0379] // -----(192kHz Sample Rate)-----
[0380] // Max Delay Register Value = 4819.452----->99.081% headroom
[0381] // Max Output Value      = 4880.066----->99.942% headroom
[0382] //
[0383] //
[0384] //      b0      b1      b2      a0      a1      a2
[0385] // -----
-----

[0386] // 3.8625515e+001 -3.8130535e+001 0.0000000e+000 1.0000000e+000
-9.8671384e-001 0.0000000e+000
[0387] //
[0388] // Magnitudes are relative to the system input, not necessarily the
filter input.
[0389] // Section      Del.max      Outmax
[0390] // -----      -----      -----
[0391] // 1      4819.4523948899350      4882.5058955489840
[0392] //
[0393] // ***** Filter 16 SAP Lowpass
*****
[0394] // This filter is in Q15 format. (Q15 is required.)
[0395] // The ' b' coefficients in the last stage are in Q15 format.
[0396] //
[0397] // -----(192kHz Sample Rate)-----
[0398] // Max Delay Register Value = 127.936----->99.976% headroom
[0399] // Max Output Value      = 3.873----->99.999% headroom

```

```

[0400]  //
[0401]  //
[0402]  //   b0           b1           b2           a0           a1           a2
[0403]  //  -----
--

[0404]  //  1.4236972e-001 -6.1114088e-002  1.4236972e-001  1.0000000e+000
-1.6681586e+000  7.0166935e-001
[0405]  //  1.7291208e-001 -2.8898617e-001  1.7291208e-001  1.0000000e+000
-1.7369812e+000  7.9467313e-001
[0406]  //  3.0167936e-001 -5.5024915e-001  3.0167936e-001  1.0000000e+000
-1.8076638e+000  8.9011358e-001
[0407]  //  3.5737308e-001 -6.6631149e-001  3.5737309e-001  1.0000000e+000
-1.8511403e+000  9.4867213e-001
[0408]  //  2.6223393e-001 -4.9257885e-001  2.6223391e-001  1.0000000e+000
-1.8738553e+000  9.7876206e-001
[0409]  //  1.4418675e+000 -2.7151244e+000  1.4418675e+000  1.0000000e+000
-1.8862768e+000  9.9419075e-001
[0410]  //
[0411]  // Magnitudes are relatrve to the system input, not necessarily the
filter input.
[0412]  // Section      Del.max           Outmax
[0413]  //  -----      -----           -----
[0414]  //  1           30.0017134252655       6.7084273462983
[0415]  //  2           127.9360639360641       7.1340558588011
[0416]  //  3           127.9360639360641       5.7714816868819
[0417]  //  4           127.9360639360642       3.8011918024620
[0418]  //  5           127.9360639360640       1.5533507892725
[0419]  //  6           127.9360639360671       3.8749910462902
[0420]  //
[0421]  module SOSIVCom(i1Clk,
[0422]                  i1Start,
[0423]                  i1DRegClear,
[0424]                  i17DataIn1,
[0425]                  or21DataOut1,
[0426]                  i17DataIn2,
[0427]                  or19DataOut2,
[0428]                  i17DataIn3,
[0429]                  or25DataOut3,
[0430]                  i20DataIn4,

```

```
[0431]          or19DataOut4,
[0432]          i22DataIn5,
[0433]          or22DataOut5,
[0434]          i20DataIn6,
[0435]          or20DataOut6,
[0436]          i23DataIn7,
[0437]          or23DataOut7,
[0438]          i22DataIn8,
[0439]          or29DataOut8,
[0440]          i18B0_08,
[0441]          i18B1_08,
[0442]          i18A1_08,
[0443]          i16DataIn9,
[0444]          or18DataOut9,
[0445]          i16DataIn10,
[0446]          or23DataOut10,
[0447]          i20DataIn11,
[0448]          or19DataOut11,
[0449]          i22DataIn12,
[0450]          or22DataOut12,
[0451]          i20DataIn13,
[0452]          or20DataOut13,
[0453]          i23DataIn14,
[0454]          or23DataOut14,
[0455]          i22DataIn15,
[0456]          or29DataOut15,
[0457]          i18B0_15,
[0458]          i18B1_15,
[0459]          i18A1_15,
[0460]          i16DataIn16,
[0461]          or18DataOut16) ;
[0462] input      i1Clk ;
[0463] input      i1Start ;
[0464] input      i1DRegClear ;
[0465] input [16:0] i17DataIn1 ;
[0466] output [20:0] or21DataOut1 ;
[0467] input [16:0] i17DataIn2 ;
[0468] output [18:0] or19DataOut2 ;
[0469] input [16:0] i17DataIn3 ;
```

[0470] output [24:0] or25DataOut3;
[0471] input [19:0] i20DataIn4;
[0472] output [18:0] or19DataOut4;
[0473] input [21:0] i22DataIn5;
[0474] output [21:0] or22DataOut5;
[0475] input [19:0] i20DataIn6;
[0476] output [19:0] or20DataOut6;
[0477] input [22:0] i23DataIn7;
[0478] output [22:0] or23DataOut7;
[0479] input [21:0] i22DataIn8;
[0480] output [28:0] or29DataOut8;
[0481] input [17:0] i18B0_08;
[0482] input [17:0] i18B1_08;
[0483] input [17:0] i18A1_08;
[0484] input [15:0] i16DataIn9;
[0485] output [17:0] or18DataOut9;
[0486] input [15:0] i16DataIn10;
[0487] output [22:0] or23DataOut10;
[0488] input [19:0] i20DataIn11;
[0489] output [18:0] or19DataOut11;
[0490] input [21:0] i22DataIn12;
[0491] output [21:0] or22DataOut12;
[0492] input [19:0] i20DataIn13;
[0493] output [19:0] or20DataOut13;
[0494] input [22:0] i23DataIn14;
[0495] output [22:0] or23DataOut14;
[0496] input [21:0] i22DataIn15;
[0497] output [28:0] or29DataOut15;
[0498] input [17:0] i18B0_15;
[0499] input [17:0] i18B1_15;
[0500] input [17:0] i18A1_15;
[0501] input [15:0] i16DataIn16;
[0502] output [17:0] or18DataOut16;
[0503] reg signed [20:0] or21DataOut1;
[0504] reg signed [18:0] or19DataOut2;
[0505] reg signed [24:0] or25DataOut3;
[0506] reg signed [18:0] or19DataOut4;
[0507] reg signed [21:0] or22DataOut5;
[0508] reg signed [19:0] or20DataOut6;

[0509] reg signed [22:0] or23DataOut7 ;
[0510] reg signed [28:0] or29DataOut8 ;
[0511] reg signed [17:0] or18DataOut9 ;
[0512] reg signed [22:0] or23DataOut10 ;
[0513] reg signed [18:0] or19DataOut11 ;
[0514] reg signed [21:0] or22DataOut12 ;
[0515] reg signed [19:0] or20DataOut13 ;
[0516] reg signed [22:0] or23DataOut14 ;
[0517] reg signed [28:0] or29DataOut15 ;
[0518] reg signed [17:0] or18DataOut16 ;
[0519] reg signed [37:0] r38InputReg ;
[0520] reg signed [49:0] r50Addend1 ;
[0521] reg signed [49:0] r50Addend2 ;
[0522] wire signed [49:0] w50Sum ;
[0523] reg signed [49:0] r50Sum ,
[0524] reg signed [34:0] r35MultInput1 ;
[0525] reg signed [17:0] r18MultInput2 ;
[0526] wire signed [49:0] w50Product ;
[0527] reg signed [49:0] r50Product ;
[0528] reg signed [49:0] r50ProductReg ;
[0529] reg r1ShiftRegEnable ;
[0530] reg [34:0] r35ShiftRegInput ;
[0531] wire signed [34:0] w35D1 ;
[0532] reg signed [34:0] r35D2 ;
[0533] reg [86:0] r87DelayShiftReg34 ;
[0534] reg [86:0] r87DelayShiftReg33 ;
[0535] reg [86:0] r87DelayShiftReg32 ;
[0536] reg [86:0] r87DelayShiftReg31 ;
[0537] reg [86:0] r87DelayShiftReg30 ;
[0538] reg [86:0] r87DelayShiftReg29 ,
[0539] reg [86:0] r87DelayShiftReg28 ;
[0540] reg [86:0] r87DelayShiftReg27 ;
[0541] reg [86:0] r87DelayShiftReg26 ;
[0542] reg [86:0] r87DelayShiftReg25 ;
[0543] reg [86:0] r87DelayShiftReg24 ;
[0544] reg [86:0] r87DelayShiftReg23 ;
[0545] reg [86:0] r87DelayShiftReg22 ;
[0546] reg [86:0] r87DelayShiftReg21 ;
[0547] reg [86:0] r87DelayShiftReg20 ;

```

[0548] reg      [86:0] r87DelayShiftReg19 ;
[0549] reg      [86:0] r87DelayShiftReg18 ;
[0550] reg      [86:0] r87DelayShiftReg17 ;
[0551] reg      [86:0] r87DelayShiftReg16 ;
[0552] reg      [86:0] r87DelayShiftReg15 ;
[0553] reg      [86:0] r87DelayShiftReg14 ;
[0554] reg      [86:0] r87DelayShiftReg13 ;
[0555] reg      [86:0] r87DelayShiftReg12 ;
[0556] reg      [86:0] r87DelayShiftReg11 ;
[0557] reg      [86:0] r87DelayShiftReg10 ;
[0558] reg      [86:0] r87DelayShiftReg09 ;
[0559] reg      [86:0] r87DelayShiftReg08 ;
[0560] reg      [86:0] r87DelayShiftReg07 ;
[0561] reg      [86:0] r87DelayShiftReg06 ;
[0562] reg      [86:0] r87DelayShiftReg05 ;
[0563] reg      [86:0] r87DelayShiftReg04 ;
[0564] reg      [86:0] r87DelayShiftReg03 ;
[0565] reg      [86:0] r87DelayShiftReg02 ;
[0566] reg      [86:0] r87DelayShiftReg01 ;
[0567] reg      [86:0] r87DelayShiftReg00 ;
[0568] wire      w1CoeffieientClkEnable ;
[0569] reg      r1CoeffieientClkEnable ;
[0570] wire signed [17:0] w18Coefficient ;
[0571] reg      [3:0] r4SectionState ;
[0572] reg      [5:0] r6SectionNumber,
[0573] parameter
[0574]     FILTER01_LAST_SECTION = 6' d3
[0575]     FILTER02_LAST_SECTION = 6' d9
[0576]     FILTER03_LAST_SECTION = 6' d14
[0577]     FILTER04_LAST_SECTION = 6' d15
[0578]     FILTER05_LAST_SECTION = 6' d16
[0579]     FILTER06_LAST_SECTION = 6' d18
[0580]     FILTER07_LAST_SECTION = 6' d19,
[0581]     FILTER08_LAST_SECTION = 6' d20,
[0582]     FILTER09_LAST_SECTION = 6' d26,
[0583]     FILTER10_LAST_SECTION = 6' d31,
[0584]     FILTER11_LAST_SECTION = 6' d32,
[0585]     FILTER12_LAST_SECTION = 6' d33,
[0586]     FILTER13_LAST_SECTION = 6' d35,

```

```

[0587]     FILTER14_LAST_SECTION = 6' d36,
[0588]     FILTER15_LAST_SECTION = 6' d37,
[0589]     FILTER16_LAST_SECTION = 6' d43,
[0590]     LAST_SECTION          = FILTER16_LAST_SECTION;
[0591] //State Machine States
[0592] parameter
[0593]     IDLE      = 4' d0,
[0594]     WAIT1    = 4' d6,
[0595]     STATEIA   = 4' d7
[0596]     STATE1    = 4' d1
[0597]     STATE2    = 4' d2
[0598]     STATE3    = 4' d3
[0599]     STATE4    = 4' d4
[0600]     STATE5    = 4' d5
[0601]     LAST      = 4' d8;
[0602] //synthesis translate_off
[0603] initial
[0604] begin
[0605]     r4SectionState = IDLE;
[0606] end
[0607] //synthesis translate_on
[0608] assign w1 CoefficientClkEnable = r1CoefficientClkEnable | i1 Start;
[0609] CoefSelectSOSIVCom
[0610]     CoefSelect1 (.i1 Clk      (i1 Clk),
[0611]                 .i1ClkEnable (w1 CoefficientClkEnable),
[0612]                 .i18B0_08    (i18B0_08),
[0613]                 .i18B1_08    (i18B1_08),
[0614]                 .i18A1_08    (i18A1_08),
[0615]                 .i18B0_15    (i18B0_15),
[0616]                 .i18B1_15    (i18B1_15),
[0617]                 .i18A1_15    (i18A1_15),
[0618]                 .or18Data     (w18Coefficient));
[0619] MultSOSIVCom
[0620]     Multiplier1(.i35MultInput1 (r35MultInput1),
[0621]                 .i18MultInput2 (r18MultInput2),
[0622]                 .ow50Product    (w50Product));
[0623] AdderSOSIVCom
[0624]     Adder1 (.i50Addend1 (r50Addend1),
[0625]             .i50Addend2 (r50Addend2),

```

```
[0626]         .ow50Sum      (w50Sum)) ;
[0627]   assign w35D1 = {r87DelayShiftReg34[0],
[0628]                   r87DelayShiftReg33[0],
[0629]                   r87DelayShiftReg32[0],
[0630]                   r87DelayShiftReg31[0],
[0631]                   r87DelayShiftReg30[0],
[0632]                   r87DelayShiftReg29[0],
[0633]                   r87DelayShiftReg28[0],
[0634]                   r87DelayShiftReg27[0],
[0635]                   r87DelayShiftReg26[0],
[0636]                   r87DelayShiftReg25[0],
[0637]                   r87DelayShiftReg24[0],
[0638]                   r87DelayShiftReg23[0],
[0639]                   r87DelayShiftReg22[0],
[0640]                   r87DelayShiftReg21[0],
[0641]                   r87DelayShiftReg20[0],
[0642]                   r87DelayShiftReg19[0],
[0643]                   r87DelayShiftReg18[0],
[0644]                   r87DelayShiftReg17[0],
[0645]                   r87DelayShiftReg16[0],
[0646]                   r87DelayShiftReg15[0],
[0647]                   r87DelayShiftReg14[0],
[0648]                   r87DelayShiftReg13[0],
[0649]                   r87DelayShiftReg12[0],
[0650]                   r87DelayShiftReg11[0],
[0651]                   r87DelayShiftReg10[0],
[0652]                   r87DelayShiftReg09[0],
[0653]                   r87DelayShiftReg08[0],
[0654]                   r87DelayShiftReg07[0],
[0655]                   r87DelayShiftReg06[0],
[0656]                   r87DelayShiftReg05[0],
[0657]                   r87DelayShiftReg04[0],
[0658]                   r87DelayShiftReg03[0],
[0659]                   r87DelayShiftReg02[0],
[0660]                   r87DelayShiftReg01[0],
[0661]                   r87DelayShiftReg00[0]} ;
[0662]   always @(r4SectionState or i1 DRegClear or w35D1 or w50Sum)
[0663]   begin
[0664]     if(i1 DRegClear)begin
```

```

[0665]     r1ShiftRegEnable<= 1' b1;
[0666]     r35ShiftRegInput<= 35' b0;
[0667] end else begin
[0668]     case(r4SectionState)
[0669]         IDLE:
[0670]             begin
[0671]                 r1ShiftRegEnable<= 1' b0;
[0672]                 r35ShiftRegInput<= 35'bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
[0673]             end
[0674]         STATE1A, STATE1:
[0675]             begin
[0676]                 r1ShiftRegEnable<= 1' b1;
[0677]                 r35ShiftRegInput<= w50Sum[49:15];
[0678]             end
[0679]         STATE5:
[0680]             begin
[0681]                 r1ShiftRegEnable<= 1' b1;
[0682]                 r35ShiftRegInput<= w35D1;
[0683]             end
[0684]         default;
[0685]             begin
[0686]                 r1ShiftRegEnable<= 1' b0;
[0687]                 r35ShiftRegInput<= 35' bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x;
[0688]             end
[0689]         endcase
[0690]     end
[0691] end
[0692] always @(r4SectionState or r50Sum or r38InputReg or r50Product or
r50ProductReg)
[0693]     case(r4SectionState)
[0694]         STATE1A, STATE1:begin
[0695]             r50Addend1 <= r50Sum;
[0696]             r50Addend2 <= {{12{r38InputReg[37]}}}, r28InputReg;
[0697]         end
[0698]         STATE2:begin
[0699]             r50Addend1 <= r50ProductReg;
[0700]             r50Addend2 <= r50Product;
[0701]         end

```

```

[0702]     STATE3:begin
[0703]         r50Addend1 <= r50Sum;
[0704]         r50Addend2 <= r50Product;
[0705]     end
[0706]     STATE5:begin
[0707]         r50Addend1 <= r50ProductReg;
[0708]         r50Addend2 <= r50Product;
[0709]     end
[0710]     default:begin
[0711]         r50Addend1 <= 50' bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxx;
[0712]         r50Addend2 <= 50' bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxx;
[0713]     end
[0714] endcase
[0715] always @(posedge ilClk)
[0716] begin
[0717]     r50Product <= w50Product;
[0718]     r18MultInput2 <= w18Coefficient;
[0719]     if(r1ShiftRegEnable)begin
[0720]         r87DelayShiftReg34<= {r5ShiftRegInput[34],
r87DelayShiftReg34[86:1]};
[0721]         r87DelayShiftReg33<= {r35ShiftRegInput[33],
r87DelayShiftReg33[86:1]};
[0722]         r87DelayShiftReg32<= {r35ShiftRegInput[32],
r87DelayShiftReg32[86:1]};
[0723]         r87DelayShiftReg31<= {r35ShiftRegInput[31],
r87DelayShiftReg31[86:1]};
[0724]         r87DelayShiftReg30<= {r35ShiftRegInput[30],
r87DelayShiftReg30[86:1]};
[0725]         r87DelayShiftReg29<= {r35ShiftRegInput[29],
r87DelayShiftReg29[86:1]};
[0726]         r87DelayShiftReg28<= {r35ShiftRegInput[28],
r87DelayShiftReg28[86:1]};
[0727]         r87DelayShiftReg27<= {r35ShiftRegInput[27],
r87DelayShiftReg27[86:1]};
[0728]         r87DelayShiftReg26<= {r35ShiftRegInput[26],
r87DelayShiftReg26[86:1]};
[0729]         r87DelayShiftReg25<= {r35ShiftRegInput[25],

```

r87DelayShiftReg25[86:1]} ;
[0730] r87DelayShiftReg24< = {r35ShiftRegInput[24],
r87DelayShiftReg24[86:1]} ;
[0731] r87DelayShiftReg23< = {r35ShiftRegInput[23],
r87DelayShiftReg23[86:1]} ;
[0732] r87DelayShiftReg22< = {r35ShiftRegInput[22],
r87DelayShiftReg22[86:1]} ;
[0733] r87DelayShiftReg21< = {r35ShiftRegInput[21],
r87DelayShiftReg21[86:1]} ;
[0734] r87DelayShiftReg20< = {r35ShiftRegInput[20],
r87DelayShiftReg20[86:1]} ;
[0735] r87DelayShiftReg19< = {r35ShiftRegInput[19],
r87DelayShiftReg19[86:1]} ;
[0736] r87DelayShiftReg18< = {r35ShiftRegInput[18],
r87DelayShiftReg18[86:1]} ;
[0737] r87DelayShiftReg17< = {r35ShiftRegInput[17],
r87DelayShiftReg17[86:1]} ;
[0738] r87DelayShiftReg16< = {r35ShiftRegInput[16],
r87DelayShiftReg16[86:1]} ;
[0739] r87DelayShiftReg15< = {r35ShiftRegInput[15],
r87DelayShiftReg15[86:1]} ;
[0740] r87DelayShiftReg14< = {r35ShiftRegInput[14],
r87DelayShiftReg14[86:1]} ;
[0741] r87DelayShiftReg13< = {r35ShiftRegInput[13],
r87DelayShiftReg13[86:1]} ;
[0742] r87DelayShiftReg12< = {r35ShiftRegInput[12],
r87DelayShiftReg12[86:1]} ;
[0743] r87DelayShiftReg11< = {r35ShiftRegInput[11],
r87DelayShiftReg11[86:1]} ;
[0744] r87DelayShiftReg10< = {r35ShiftRegInput[10],
r87DelayShiftReg10[86:1]} ;
[0745] r87DelayShiftReg09< = {r35ShiftRegInput[9],
r87DelayShiftReg09[86:1]} ;
[0746] r87DelayShiftReg08< = {r35ShiftRegInput[8],
r87DelayShiftReg08[86:1]} ;
[0747] r87DelayShiftReg07< = {r35ShiftRegInput[7],
r87DelayShiftReg07[86:1]} ;
[0748] r87DelayShiftReg06< = {r35ShiftRegInput[6],
r87DelayShiftReg06[86:1]} ;

```

[0749]      r87DelayShiftReg05 <= {r35ShiftRegInput[5],
r87DelayShiftReg05[86:1]};
[0750]      r87DelayShiftReg04 <= {r35ShiftRegInput[4],
r87DelayShiftReg04[86:1]};
[0751]      r87DelayShiftReg03 <= {r35ShiftRegInput[3],
r87DelayShiftReg03[86:1]};
[0752]      r87DelayShiftReg02 <= {r35ShiftRegInput[2],
r87DelayShiftReg02[86:1]};
[0753]      r87DelayShiftReg01 <= {r35ShiftRegInput[1],
r87DelayShiftReg01[86:1]};
[0754]      r87DelayShiftReg00 <= {r35ShiftRegInput[0],
r87DelayShiftReg00[86:1]};
[0755]  end
[0756]  if(i1 DRegClear)begin
[0757]      r4SectionState <= IDLE;
[0758]      r1CoefficientClkEnable <= 1' b0;
[0759]      r35D2 <= 35' d0;
[0760]      r50ProductReg <= 50' d0;
[0761]      r50Sum <= 50' d0;
[0762]      or21DataOut1 <= 21' d0;
[0763]      or19DataOut2 <= 19' d0;
[0764]      or25DataOut3 <= 25' d0;
[0765]      or19DataOut4 <= 19' d0;
[0766]      or22DataOut5 <= 22' d0;
[0767]      or20DataOut6 <= 20' d0;
[0768]      or23DataOut7 <= 23' d0;
[0769]      or29DataOut8 <= 29' d0;
[0770]      or18DataOut9 <= 18' d0;
[0771]      or23DataOut10 <= 23' d0;
[0772]      or19DataOut11 <= 19' d0;
[0773]      or22DataOut12 <= 22' d0;
[0774]      or20DataOut13 <= 20' d0;
[0775]      or23DataOut14 <= 23' d0;
[0776]      or29DataOut15 <= 29' d0;
[0777]      or18DataOut16 <= 18' d0;
[0778]  end else begin
[0779]      case(r4SectionState)
[0780]      IDLE:
[0781]          begin

```

```
[0782]     if(i1Start)begin
[0783]         r4SectionState    <= WAIT1 ;
[0784]         r1CoefficientClkEnable <= 1' b1 ;
[0785]         r6SectionNumber    <= 6' d0 ;
[0786]         r38InputReg        <= {{6{i17DataIn1[16]}} , i17DataIn1, 15' b0} ;
[0787]     end
[0788] end
[0789] WAIT1 :
[0790] begin
[0791]     r4SectionState <= STATE1A ;
[0792]     r35MultInput1 <= r35D2 ;
[0793] end
[0794] STATE1A :
[0795] begin
[0796]     r4SectionState <= STATE2 ;
[0797]     r35D2          <= w35D1 ;
[0798]     r35MultInput1 <= w50Sum[49:15] ;
[0799]     r50Sum         <= w50Sum ;
[0800] end
[0801] STATE1 :
[0802] begin
[0803]     r4SectionState <= STATE2 ;
[0804]     r35D2          <= w35D1 ;
[0805]     r35MultInput1 <= w50Sum[49:15] ;
[0806]     r50Sum         <= w50Sum ;
[0807]     r50ProductReg <= r50Product ;
[0808] end
[0809] STATE2 :
[0810] begin
[0811]     r4SectionState <= STATE3 ;
[0812]     r35MultInput1 <= w35D1 ;
[0813]     r50Sum         <= w50Sum ;
[0814] end
[0815] STATE3 :
[0816] begin
[0817]     r4SectionState <= STATE4 ;
[0818]     r35MultInput1 <= r35D2 ;
[0819]     r50Sum         <= w50Sum ;
[0820] end
```

```

[0821]   STATE4:
[0822]     begin
[0823]       r4SectionState <= STATE5 ;
[0824]       case(r6SectionNumber)
[0825]         FILTER01_LAST_SECTION:or21DataOut1<= r50Sum[38:18] ;
[0826]         FILTER02_LAST_SECTION:or19DataOut2<= r50Sum[35:17] ;
[0827]         FILTER03_LAST_SECTION:or25DataOut3<= r50Sum[36:12] ;
[0828]         FILTER04_LAST_SECTION:or19DataOut4<= r50Sum[39:21] ;
[0829]         FILTER05_LAST_SECTION:or22DataOut5<= r50Sum[49:28] ;
[0830]         FILTER06_LAST_SECTION:or20DataOut6<= r50Sum[37:18] ;
[0831]         FILTER07_LAST_SECTION:or23DataOut7<= r50Sum[49:27] ;
[0832]         FILTER08_LAST_SECTION:or29DataOut8<= r50Sum[39:11] ;
[0833]         FILTER09_LAST_SECTION:or18DataOut9<= r50Sum[34:17] ;
[0834]         FILTER10_LAST_SECTION:or23DataOut10<= r50Sum[34:12] ;
[0835]         FILTER11_LAST_SECTION:or19DataOut11<= r50Sum[39:21] ;
[0836]         FILTER12_LAST_SECTION:or22DataOut12<= r50Sum[49:28] ;
[0837]         FILTER13_LAST_SECTION:or20DataOut13<= r50Sum[37:18] ;
[0838]         FILTER14_LAST_SECTION:or23DataOut14<= r50Sum[49:27] ;
[0839]         FILTER15_LAST_SECTION:or29DataOut15<= r50Sum[39:11] ;
[0840]         FILTER16_LAST_SECTION:or18DataOut16<= r50Sum[32:15] ;
[0841]       endcase
[0842]       r38InpurReg <= r50Sum[37:0] ;
[0843]       r50ProductReg <= r50Product ;
[0844]       r35MultInput1 <= w35D1 ;
[0845]     end
[0846]   STATE5:
[0847]     begin
[0848]       r6SeconNumber<= r6SectionNumber+1 ;
[0849]       if(r6SectionNumber == LAST_SECTION)
[0850]         begin
[0851]           r4SectionState <= LAST ;
[0852]           r1CoefficientClkEnable<= 1' b0 ;
[0853]         end
[0854]       else begin
[0855]         r4SectionState <= STATE1 ;
[0856]         r35MultInput1 <= r35D2 ;
[0857]       end
[0858]       case(r6SectionNumber)
[0859]         FILTER01_LAST_SECTION:r38InputReg<= {{6{i17DataIn2[16]}}}, i17DataIn2,

```

```

15' b0} ;
[0860] FILTER02_LAST_SECTION:r38InputReg< = {{6{i17DataIn3[16]}}}, i17DataIn3,
15' b0} ,
[0861] FILTER03_LAST_SECTION:r38InputReg< = {{3{i20DataIn4[19]}}}, i20DataIn4,
15' b0} ;
[0862] FILTER04_LAST_SECTION:r38InputReg< = {i22DataInS[21], i22DataIn5,
15' b0} ;
[0863] FILTER05_LAST_SECTION:r38InputReg< = {{3{i20DataIn6[19]}}}, i20DataIn6,
15' b0} ;
[0864] FILTER06_LAST_SECTION:r38InputReg< = {i23DataIn7, 15' b0} ;
[0865] FILTER07_LAST_SECTION:r38InputReg< = {i22DataIn8[21], i22DataIn8,
15' b0} ;
[0866] FILTER08_LAST_SECTION:r38InputReg< = {{7{i16DataIn9[15]}}}, i16DataIn9,
15' b0} ;
[0867] FILTER09_LAST_SECTION:r38InputReg< = {{7{i16DataIn10[15]}}},
i16DataIn10, 15' b0} ;
[0868] FILTER10_LAST_SECTION:r38InputReg< = {{3{i20DataIn11[19]}}},
i20DataIn11, 15' b0} ;
[0869] FILTER11_LAST_SECTION:r38InputReg< = {i22DataIn 12[21], i22DataIn 12,
15' b0} ;
[0870] FILTER12_LAST_SECTION;r38InputReg< = {{3{i20DataIn13[19]}}},
i20DataIn13, 15' b0} ;
[0871] FILTER13_LAST_SECTION:r38InputReg< = {i23DataIn 14, 15' b0} ;
[0872] FILTER14_LAST_SECTION:r38InputReg< = {i22DataIn15[21], i22DataIn15,
15' b0} ;
[0873] FILTER15_LAST_SECTION:r38InputReg< = {{7{i16DataIn16[15]}}},
i16DataIn16, 15' b0} ;
[0874]          endcase
[0875]          r50Sum      <= w50Sum ;
[0876]          end
[0877]          LAST:
[0878]          begin
[0879]              r4SecdonState <= IDLE ;
[0880]              r50ProductReg <= r50Product ;
[0881]          end
[0882]          default:
[0883]          begin
[0884]              r4SectionState <= IDLE ;
[0885]          end

```

```
[0886]         endcase
[0887]     end
[0888] end
[0889] endmodule
```

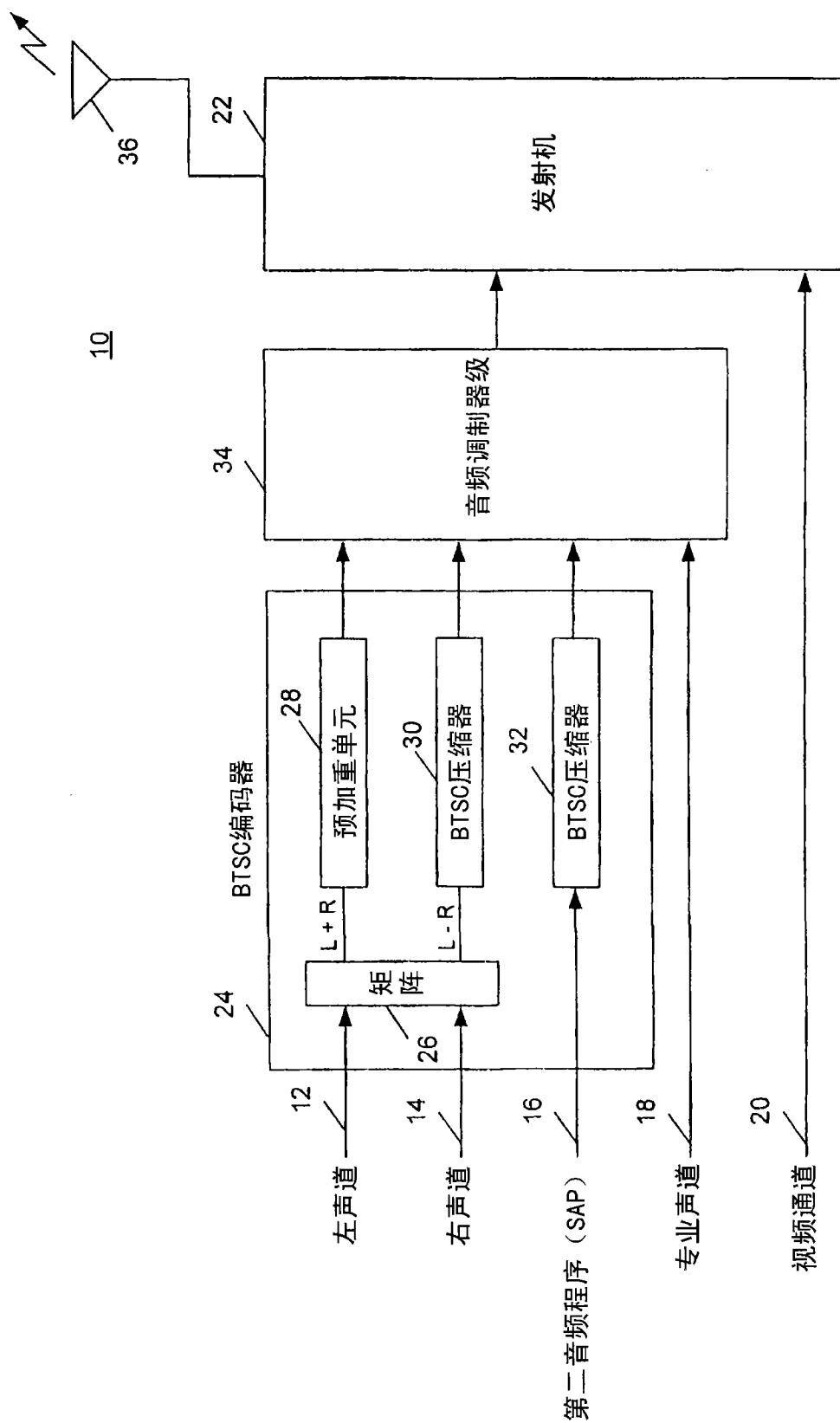


图1

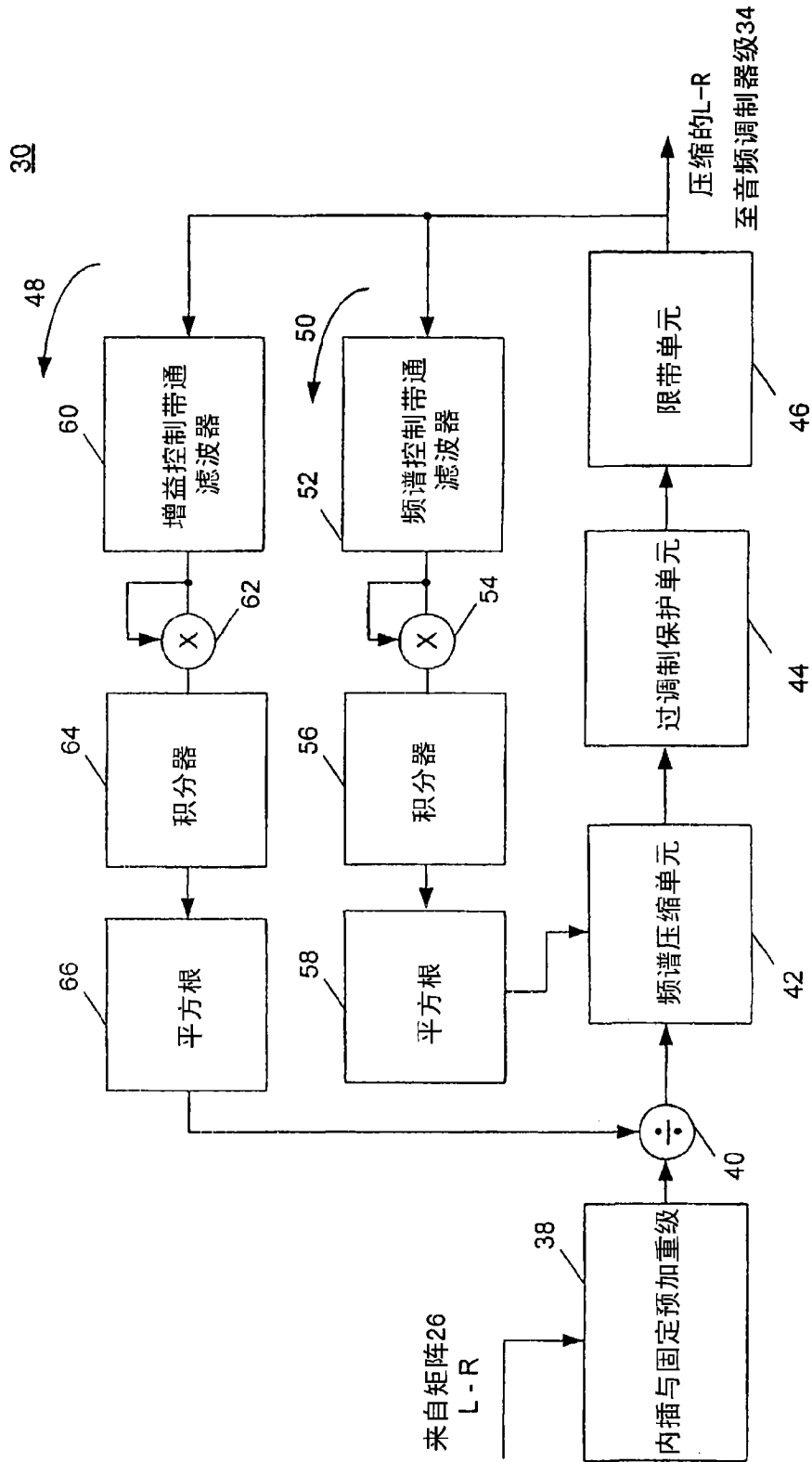


图2

68

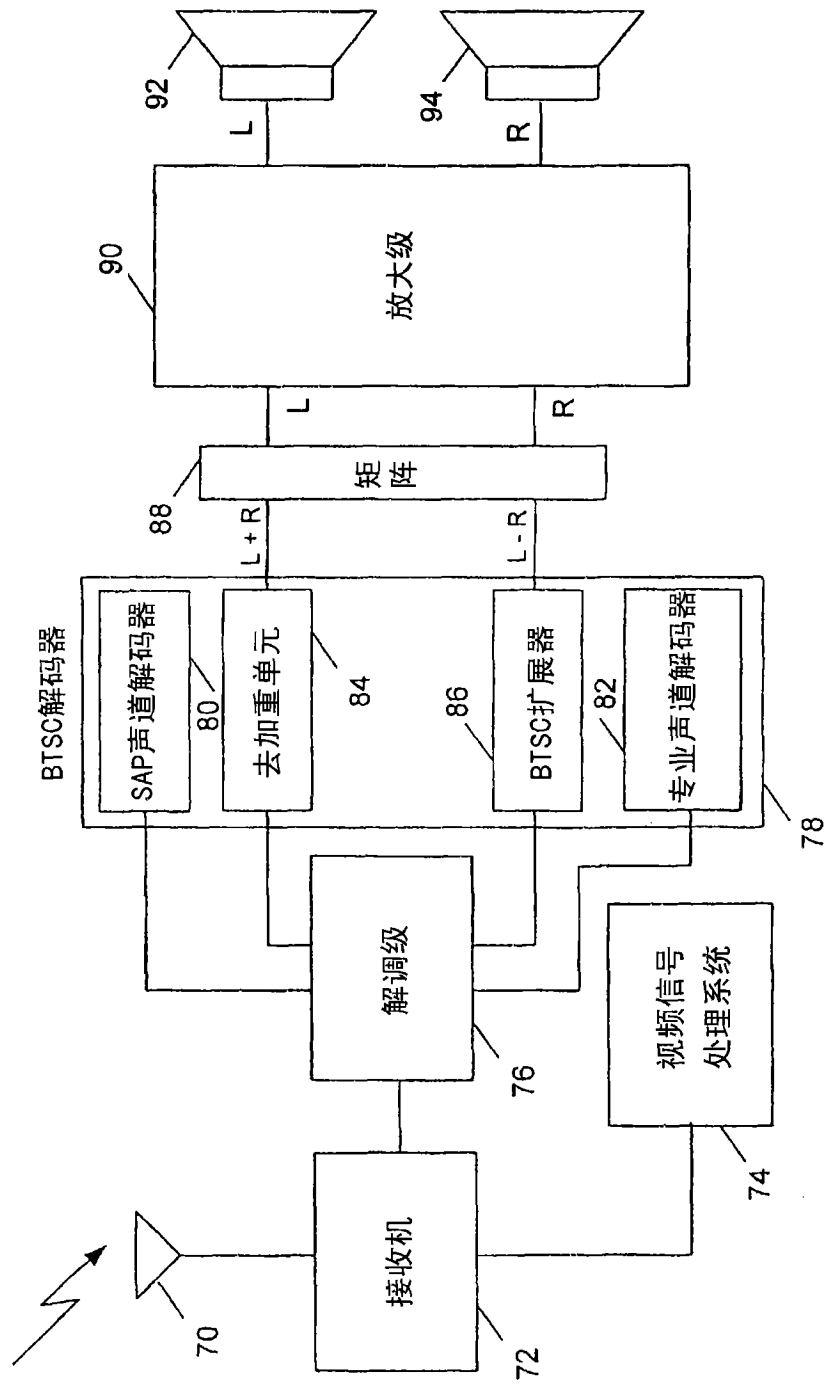


图3

86

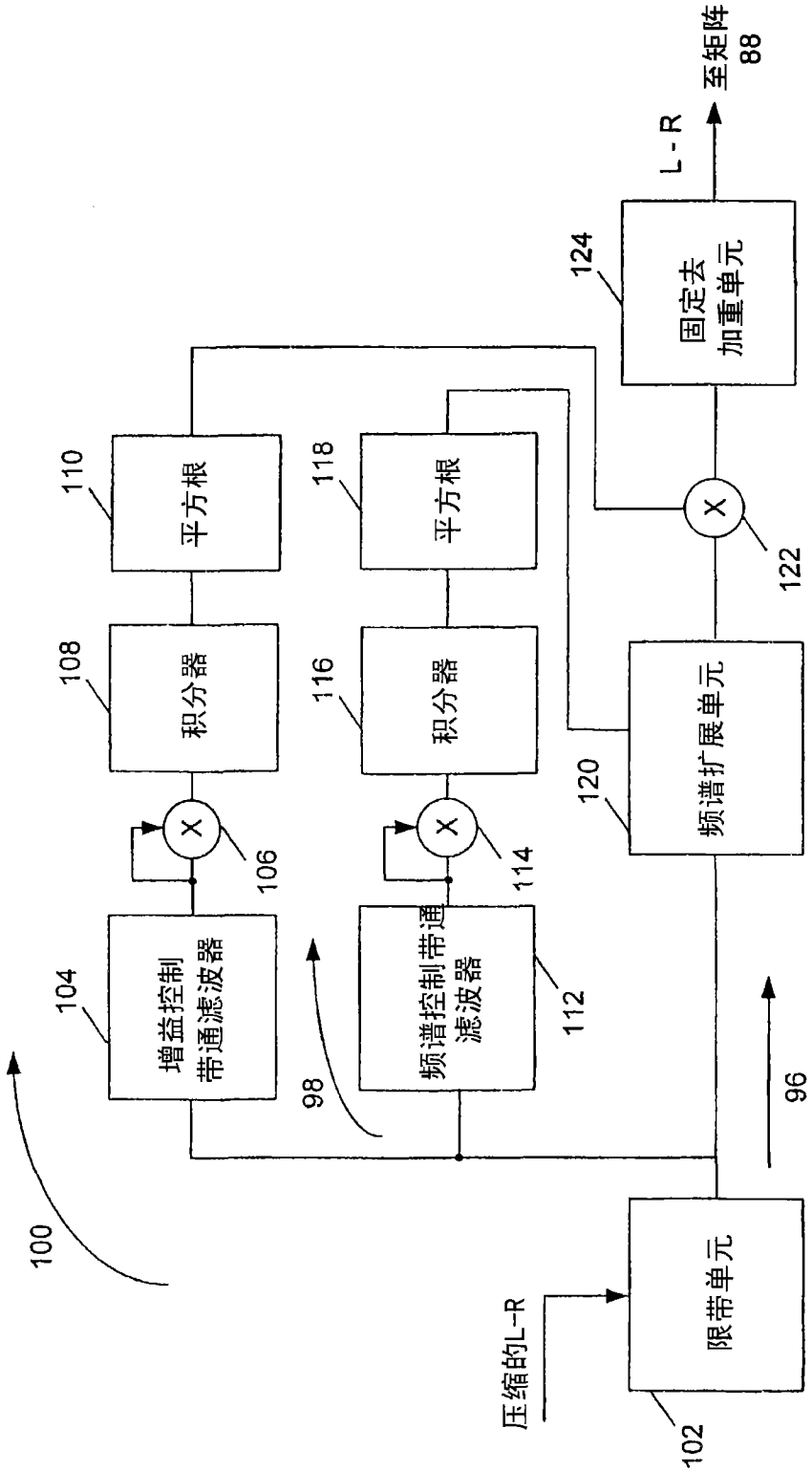


图4

126

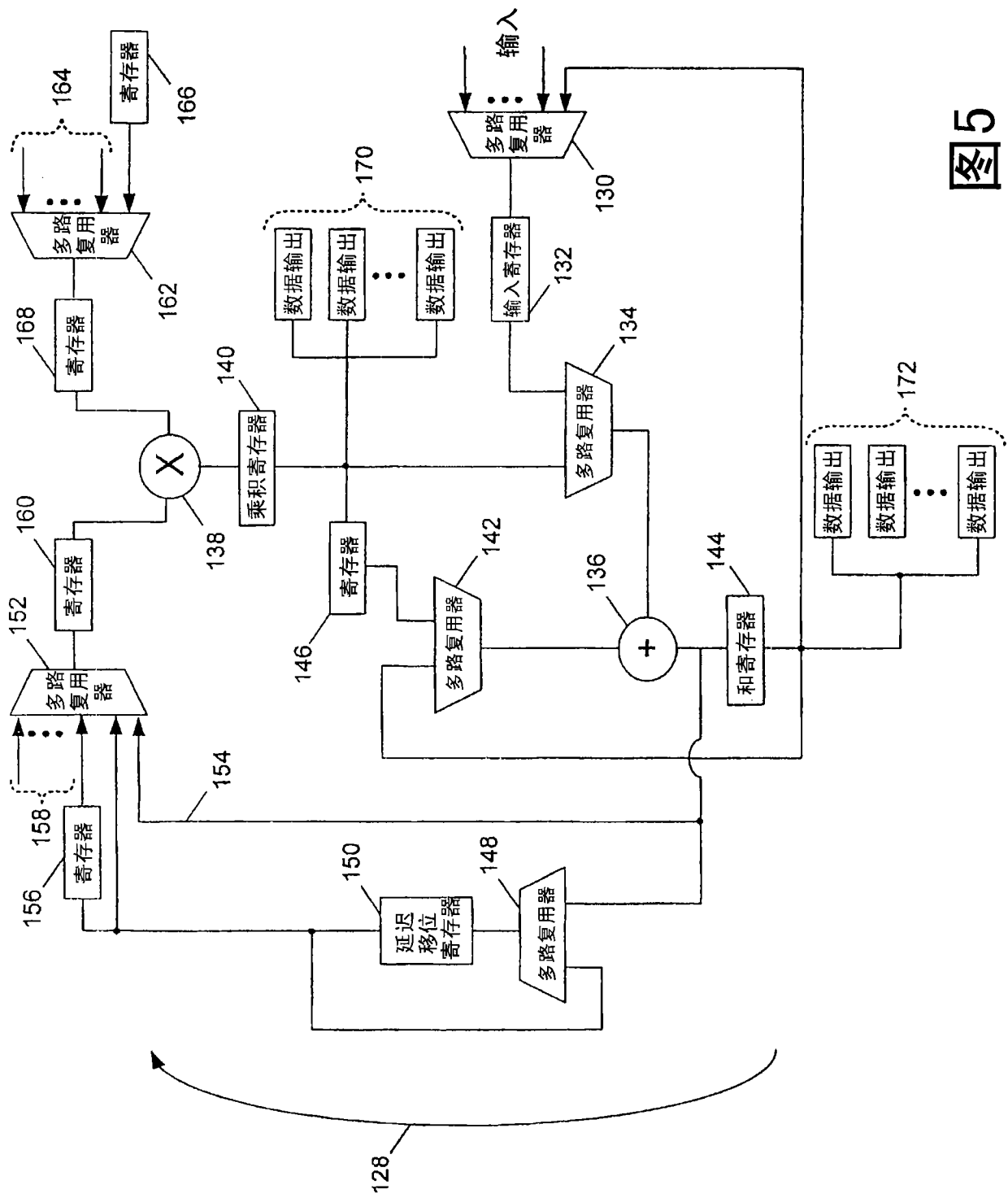
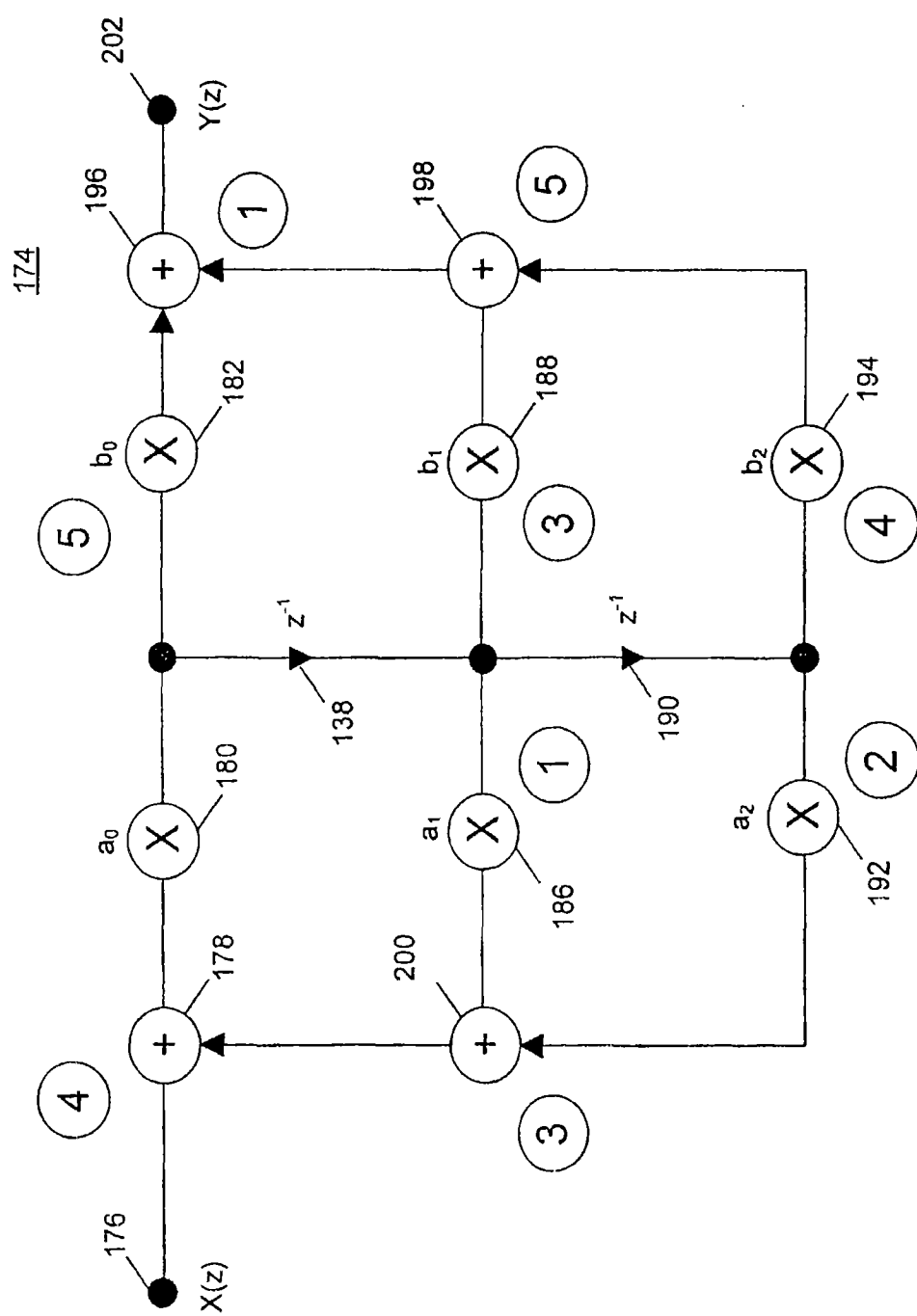


图5



9
圖

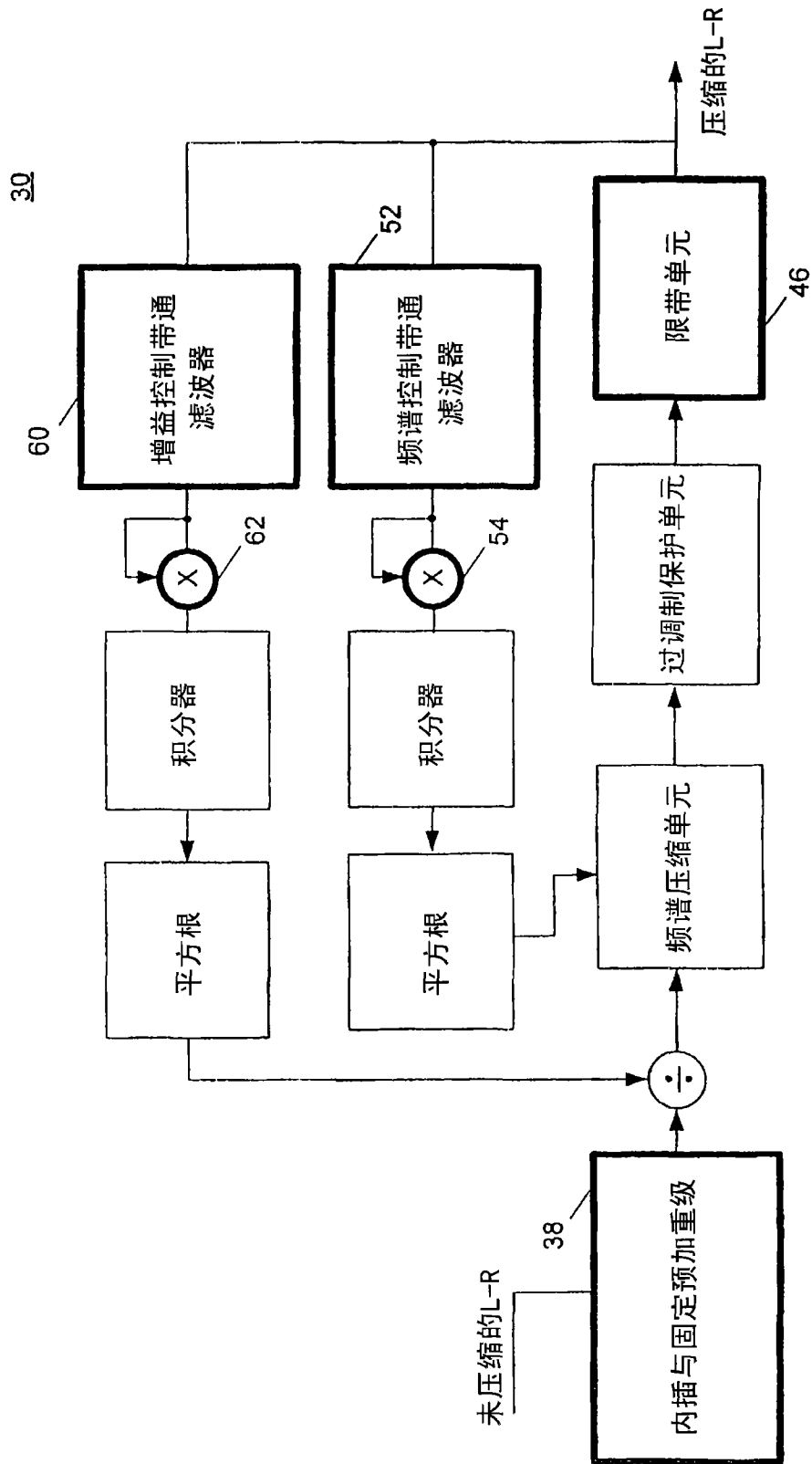


图7

86

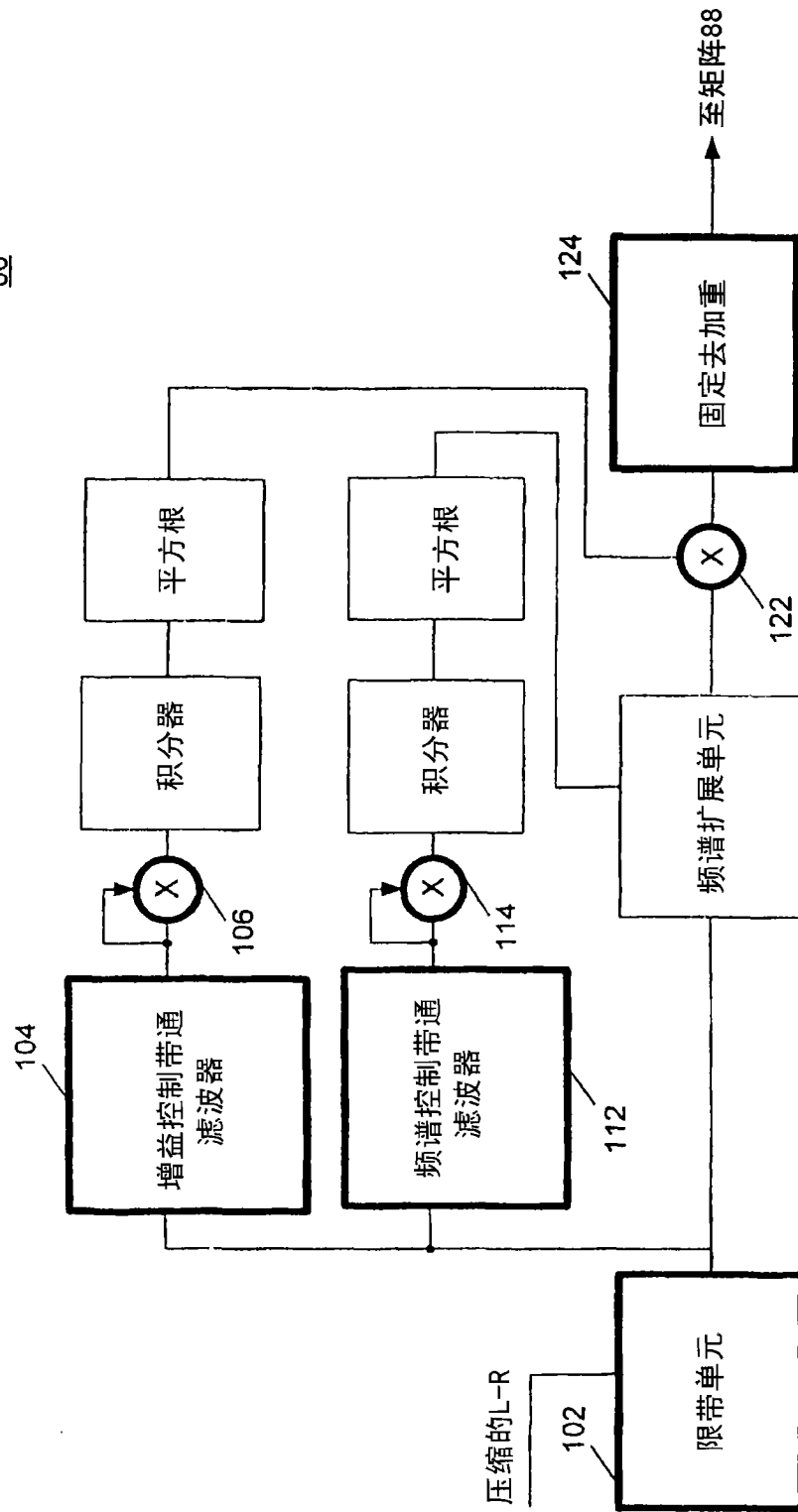


图8