

(12) 发明专利

(10) 授权公告号 CN 101131686 B

(45) 授权公告日 2010.06.02

(21) 申请号 200710147940.3

(22) 申请日 2007.08.24

(30) 优先权数据

11/510,218 2006.08.25 US

(73) 专利权人 辉达公司

地址 美国加利福尼亚州

(72) 发明人 拉维·布卢苏 帕塔·斯里拉姆

(74) 专利代理机构 北京市磐华律师事务所

11336

代理人 董巍 顾珊

(51) Int. Cl.

G06F 17/14 (2006.01)

H04N 7/26 (2006.01)

(56) 对比文件

US 20050033788 A1, 2005.02.10, 全文.

US 5883823 A, 1999.03.16, 全文.

CN 1132114 C, 2003.12.24, 全文.

CN 1449114 A, 2003.10.15, 全文.

审查员 富瑶

权利要求书 7 页 说明书 20 页 附图 4 页

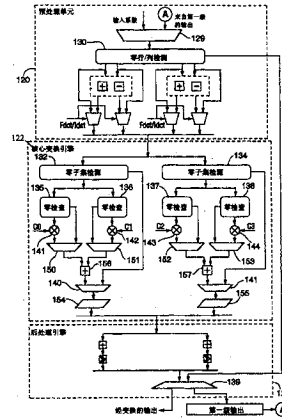
(54) 发明名称

二维变换的方法及系统、视频处理系统和变换引擎电路

(57) 摘要

在某些实施例中,提供一种方法及一种系统,所述方法用于对数据块序列中的每一块执行 2D 变换(例如离散余弦反变换)且所述系统经配置以执行所述 2D 变换,其中所述 2D 变换包括行变换及列变换。为对数据行或列执行行变换或列变换,这些实施例判定构成所述行(列)的分区的数据值的每一不同子集是否包括至少一个零值、所述分区的第一子集的每一不同子集是否包括至少一个零值、及所述分区的至少一个其他子集中的每一不同子集是否包括至少一个零值。当对包括至少一个零值及至少一个非零值的每一行或列执行行(列)变换时,均绕过或以降低功率的方式执行对至少一个零值的至少一个变换运算,其中假如所述零值为非零值,则原本将以消耗全功率的方式执行此种变换运算。

CN 101131686 B



1. 一种经配置以对输入数据块序列中的每一输入数据块执行 2D 变换的系统,其中每一所述输入数据块均包含输入数据值行及列,且所述 2D 变换包含行变换及列变换,所述系统包括:

行变换电路,其经配置以响应于每一所述输入数据块通过对所述输入数据块的每一行执行所述行变换而产生经局部变换数据块;及

列变换电路,其经配置以对所述经局部变换数据块的每一列执行所述列变换,

其中所述行变换电路经配置以在对每一所述行执行所述行变换时判定构成所述行的分区的所述数据值的每一不同子集是否包括至少一个零值、所述行的分区的子集中的第一子集的每一不同子集是否包括至少一个零值、及所述行的分区的子集中的第二子集的每一不同子集是否包括至少一个零值,及

所述行变换电路经配置以绕过或以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

2. 如权利要求 1 所述的系统,其中所述行变换电路经配置以绕过对所述数据值中被判定为零值的所述至少一者的所述至少一个变换运算。

3. 如权利要求 1 所述的系统,其中所述行变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的所述至少一者执行所述至少一个变换运算。

4. 如权利要求 3 所述的系统,其中所述行变换电路包括至少一个具有输入端集合的电路,且所述行变换电路经配置以通过如下作业以所述降低功率的方式执行所述至少一个变换运算:使所述具有输入端集合的电路运行,而在对所述数据值中被判定为零值的所述一者的所述变换运算期间不更新所述输入端集合中的至少一个输入端。

5. 如权利要求 1 所述的系统,其中所述 2D 变换是离散余弦反变换。

6. 如权利要求 1 所述的系统,其还包括:

缓冲器,其耦接至所述行变换电路及所述列变换电路,其中所述行变换电路经配置以将所述经局部变换数据块存储于所述缓冲器中,且所述列变换电路经配置以从所述缓冲器中检索所述经局部变换数据块的列。

7. 如权利要求 6 所述的系统,其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的第一分区的子集中的第一子集的每一不同子集是否包括至少一个零值、及所述列的第一分区的子集中的第二子集的每一不同子集是否包括至少一个零值,及

所述列变换电路经配置以绕过对所述数据值中被判定为零值的至少一者的至少一个变换运算。

8. 如权利要求 6 所述的系统,其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的第一分区的子集中的第一子集的每一不同子集是否包括至少一个零值、及所述列的第一分区的子集中的第二子集的每一不同子集是否包括至少一个零值,及

所述列变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

9. 如权利要求 8 所述的系统,其中所述列变换电路包括至少一个具有输入端集合的电路,且所述列变换电路经配置以通过如下作业以所述降低功率的方式执行所述至少一个变

换运算；使所述具有输入端集合的电路运行，而在对所述数据值中被判定为零值的所述一者的所述变换运算期间不更新所述输入端集合中的至少一个输入端。

10. 如权利要求 1 所述的系统，其中每一所述行包括值 x_i ，其中 i 是处于 $0 \leq i \leq N-1$ 范围内的整数，且 N 为偶数整数，所述行的分区包括第一子集与不同于所述第一子集的第二子集，所述第一子集的分区包括第三子集与不同于所述第三子集的第四子集，且所述第二子集的分区包括第五子集及不同于所述第五子集的第六子集，及

其中所述行变换电路经配置以在对每一所述行执行所述行变换时判定所述第一子集是否完全由零值组成且所述第二子集是否完全由零值组成，且在判定所述第一子集完全由零值组成时绕过对所述第一子集中所述数据值的至少一个所述变换运算，且在判定所述第二子集完全由零值组成时绕过对所述第二子集中所述数据值的至少一个所述变换运算。

11. 如权利要求 10 所述的系统，其中所述行变换电路经配置以在对每一所述行执行所述行变换时判定所述第一子集与所述第二子集中的每一者均包括至少一个非零值时，判定所述第三子集、所述第四子集、所述第五子集、及所述第六子集中的每一者是否完全由零值组成。

12. 如权利要求 11 所述的系统，其中所述第三子集由所述第一子集的偶数值组成，所述第四子集由所述第一子集的奇数值组成，所述第五子集由所述第二子集的偶数值组成，且所述第六子集由所述第二子集的奇数值组成。

13. 如权利要求 1 所述的系统，其中所述行变换电路经配置以在对每一所述行执行所述行变换时执行重新排序作业，其中所述重新排序作业通过对所述行的数据值进行重新排序来产生经重新排序的数据值，且所述行变换电路经配置以判定所述经重新排序的数据的第一子集是否包括至少一个零值及所述经重新排序的数据的第二子集是否包括至少一个零值，其中所述经重新排序的数据的第二子集不同于所述经重新排序的数据的第一子集。

14. 一种经配置以对输入数据块序列中的每一输入数据块执行 2D 变换的系统，其中每一所述输入数据块均包含输入数据值行及列，且所述 2D 变换包含行变换及列变换，所述系统包括：

列变换电路，其经配置以响应于每一所述输入数据块通过对所述输入数据块的每一列执行所述列变换而产生经局部变换数据块；及

行变换电路，其经配置以对所述经局部变换数据块的每一行执行所述行变换，

其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的分区的数据值的每一不同子集是否包括至少一个零值、所述列的分区中的第一子集的每一不同子集是否包括至少一个零值、及所述列的分区中的第二子集的每一不同子集是否包括至少一个零值，及

所述列变换电路经配置以绕过或以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

15. 如权利要求 14 所述的系统，其中所述列变换电路经配置以绕过对所述数据值中被判定为零值的所述至少一者的所述至少一个变换运算。

16. 如权利要求 14 所述的系统，其中所述列变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的所述至少一者执行所述至少一个变换运算。

17. 如权利要求 16 所述的系统，其中所述列变换电路包括至少一个具有输入端集合的

电路,且所述列变换电路经配置以通过如下作业以所述降低功率的方式执行所述至少一个变换运算:使所述具有输入端集合的电路运行,而在对所述数据值中被判定为零值的所述一者的所述变换运算期间不更新所述输入端集合中的至少一个输入端。

18. 如权利要求 14 所述的系统,其中所述 2D 变换是离散余弦反变换。

19. 如权利要求 14 所述的系统,其还包括:

缓冲器,其耦接至所述行变换电路及所述列变换电路,其中所述列变换电路经配置以将所述经局部变换数据块存储于所述缓冲器中,且所述行变换电路经配置以从所述缓冲器中检索所述经局部变换数据块的行。

20. 如权利要求 14 所述的系统,其中每一所述列包括值 x_i ,其中 i 是处于 $0 \leq i \leq N-1$ 范围内的整数,且 N 为偶数整数,所述列的分区包括第一子集与不同于所述第一子集的第二子集,所述第一子集的分区包括第三子集与不同于所述第三子集的第四子集,且所述第二子集的分区包括第五子集及不同于所述第五子集的第六子集,及

其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定所述第一子集是否完全由零值组成且所述第二子集是否完全由零值组成,且在判定所述第一子集完全由零值组成时绕过对所述第一子集中所述数据值的至少一个所述变换运算,且在判定所述第二子集完全由零值组成时绕过对所述第二子集中所述数据值的至少一个所述变换运算。

21. 如权利要求 20 所述的系统,其中所述列变换电路经配置以在当对每一所述列执行所述列变换时判定所述第一子集与所述第二子集中的每一者均包括至少一个非零值时,判定所述第三子集、所述第四子集、所述第五子集、及所述第六子集中的每一者是否完全由零值组成。

22. 如权利要求 21 所述的系统,其中所述第三子集由所述第一子集的偶数值组成,所述第四子集由所述第一子集的奇数值组成,所述第五子集由所述第二子集的偶数值组成,且所述第六子集由所述第二子集的奇数值组成。

23. 如权利要求 14 所述的系统,其中所述列变换电路经配置以在对每一所述列执行所述列变换时执行重新排序作业,其中所述重新排序作业通过对所述列的数据值进行重新排序来产生经重新排序的数据值,且所述列变换电路经配置以判定所述经重新排序的数据的第一子集是否包括至少一个零值及所述经重新排序的数据的第二子集是否包括至少一个零值,其中所述经重新排序的数据的第二子集不同于所述经重新排序的数据的第一子集。

24. 一种用于对输入数据块序列执行 2D 变换的方法,其中所述块中的每一者均包括输入数据值行及列,且所述 2D 变换包括行变换及列变换,所述方法包括如下步骤:

(a) 对所述输入数据块中一者的每一行执行所述行变换,包括通过对每一所述行的输入数据值执行变换运算,以产生经局部变换数据块;及

(b) 对所述经局部变换数据块的每一列执行所述列变换,包括通过对每一所述列的数据值执行其他变换运算,其中步骤 (a) 包括如下步骤:

(c) 判定构成每一所述行的分区的所述数据值的每一不同子集是否包括至少一个零值,判定所述分区的第一子集的每一不同子集是否包括至少一个零值,及判定所述分区的至少一个其他子集的每一不同子集是否包括至少一个零值;及

(d) 当对被判定包含至少一个零值及至少一个非零值的每一所述行执行所述行变换时,绕过或以降低功率的方式对所述输入数据值中被判定为零值的至少一者执行所述变换

运算中的至少一者,其中假如所述零值为非零值,则原本将以消耗全功率的方式执行所述变换运算中的所述一者。

25. 如权利要求 24 所述的方法,其中步骤 (a) 包括如下步骤:操作具有第一输入端及第二输入端的乘法电路以执行乘法运算,其中将确定至所述第一输入端的第一值乘以确定至所述第二输入端的第二值,且其中步骤 (d) 包括如下步骤:通过阻止所述第一输入端与所述第二输入端中至少一者的双态转换来以降低功率的方式对所述输入数据值中被判定为零值的所述至少一者执行所述乘法运算。

26. 如权利要求 24 所述的方法,其中步骤 (d) 包括如下步骤:当对被判定为包括至少一个零值及至少一个非零值的每一所述行执行所述行变换时,绕过对所述输入数据值中被判定为零值的所述至少一者的所述变换运算中的所述至少一者。

27. 如权利要求 24 所述的方法,其中步骤 (b) 包括如下步骤:

判定构成每一所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值,判定所述第一分区的第一子集的每一不同子集是否包括至少一个零值,及判定所述第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值;及

当对被判定为包括至少一个零值及至少一个非零值的每一所述列执行所述列变换时,绕过对所述数据值中被判定为零值的至少一者的所述其他变换运算中的至少一者,其中假如所述零值为非零值,则原本将以消耗全功率的方式执行所述其他变换运算中的所述一者。

28. 如权利要求 24 所述的方法,其中步骤 (b) 包括如下步骤:

判定构成每一所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值,判定所述第一分区的第一子集的每一不同子集是否包括至少一个零值,及判定所述第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值;及

当对被判定包含至少一个零值及至少一个非零值的每一所述列执行所述列变换时,以降低功率的方式对所述数据值中被判定为零值的至少一者执行所述其他变换运算中的至少一者,其中假如所述零值为非零值,则原本将以消耗全功率的方式执行所述其他变换运算中的所述一者。

29. 如权利要求 24 所述的方法,其中每一所述块均为 DCT 系数块,所述 DCT 系数已通过视频数据执行 DCT 而产生,且所述 2D 变换是离散余弦反变换。

30. 如权利要求 24 所述的方法,其中所述序列中至少所有块为稀疏数据块。

31. 一种用于对输入数据块序列执行 2D 变换的方法,其中所述块中的每一者均包括输入数据值行及列,且所述 2D 变换包括行变换及列变换,所述方法包括如下步骤:

(a) 对所述输入数据块中一者的每一行执行所述行变换,包括通过对每一所述行的输入数据值执行变换运算,以产生经局部变换数据块;及

(b) 对所述经局部变换数据块的每一列执行所述列变换,包括通过对每一所述列的数据值执行其他变换运算,其中步骤 (b) 包括如下步骤:

(c) 判定构成每一所述列的分区的所述数据值的每一不同子集是否包括至少一个零值,判定所述分区的第一子集的每一不同子集是否包括至少一个零值,及判定所述分区的至少一个其他子集的每一不同子集是否包括至少一个零值;及

(d) 当对被判定包含至少一个零值及至少一个非零值的每一所述列执行所述列变换

时,绕过或以降低功率的方式对所述数据值中被判定为零值的至少一者执行所述其他变换运算中的至少一者,其中假如所述零值为非零值,则原本将以消耗全功率的方式执行所述其他变换运算中的所述一者。

32. 如权利要求 31 所述的方法,其中步骤 (b) 包括如下步骤:操作具有第一输入端及第二输入端的乘法电路以执行乘法运算,其中将确定至所述第一输入端的第一值乘以确定至所述第二输入端的第二值,且其中步骤 (d) 包括如下步骤:通过阻止所述第一输入端与所述第二输入端中至少一者的双态转换,以降低功率的方式对所述输入数据值中被判定为零值的所述至少一者执行所述乘法运算。

33. 如权利要求 31 所述的方法,其中步骤 (d) 包括如下步骤:当对被判定为包括至少一个零值及至少一个非零值的每一所述列执行所述列变换时,绕过对所述数据值中被判定为零值的所述至少一者的所述其他变换运算中的所述至少一者。

34. 如权利要求 31 所述的方法,其中每一所述块均为 DCT 系数块,所述 DCT 系数已通过视频数据执行 DCT 而产生,且所述 2D 变换是离散余弦反变换。

35. 如权利要求 31 所述的方法,其中所述序列中至少所有块为稀疏数据块。

36. 一种视频处理系统,其包括经配置以对输入数据块序列中的每一输入数据块执行 2D 变换的变换引擎电路,其中每一所述输入数据块均包含输入数据值行及列,且所述 2D 变换包含行变换及列变换,所述变换引擎电路包括:

行变换电路,其经配置以响应于每一所述输入数据块通过对所述输入数据块的每一行执行所述行变换而产生经局部变换数据块;及

列变换电路,其经配置以对所述经局部变换数据块的每一列执行所述列变换,

其中所述行变换电路经配置以在对每一所述行执行所述行变换时判定构成所述行的分区的所述数据值的每一不同子集是否包括至少一个零值、所述行的分区的第一子集的每一不同子集是否包括至少一个零值、及所述行的分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及

所述行变换电路经配置以绕过或以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

37. 如权利要求 36 所述的系统,其中所述行变换电路经配置以绕过对所述数据值中被判定为零值的所述至少一者的所述至少一个变换运算。

38. 如权利要求 36 所述的系统,其中所述行变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的所述至少一者执行所述至少一个变换运算。

39. 如权利要求 38 所述的系统,其中所述行变换电路包括至少一个具有输入端集合的电路,且所述行变换电路经配置以通过如下作业以所述降低功率的方式执行所述至少一个变换运算:使所述具有输入端集合的电路运行,而在对所述数据值中被判定为零值的所述一者的所述变换运算期间不更新所述输入端集合中的至少一个输入端。

40. 如权利要求 36 所述的系统,其中所述 2D 变换是离散余弦反变换。

41. 如权利要求 36 所述的系统,其中所述变换引擎电路还包括:

缓冲器,其耦接至所述行变换电路及所述列变换电路,其中所述行变换电路经配置以将所述经局部变换数据块存储于所述缓冲器中,且所述列变换电路经配置以从所述缓冲器中检索所述经局部变换数据块的列。

42. 如权利要求 41 所述的系统,其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的第一分区的第一子集的每一不同子集是否包括至少一个零值、及所述列的第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及

所述列变换电路经配置以绕过对所述数据值中被判定为零值的至少一者的至少一个变换运算。

43. 如权利要求 42 所述的系统,其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的第一分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的第一分区的第一子集的每一不同子集是否包括至少一个零值、及所述列的第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及

所述列变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

44. 一种视频处理系统,其包括经配置以对输入数据块序列中的每一输入数据块执行 2D 变换的变换引擎电路,其中每一所述输入数据块均包含输入数据值行及列,且所述 2D 变换包含行变换及列变换,所述变换引擎电路包括:

列变换电路,其经配置以响应于每一所述输入数据块通过对所述输入数据块的每一列执行所述列变换而产生经局部变换数据块;及

行变换电路,其经配置以对所述经局部变换数据块的每一行执行所述行变换,

其中所述列变换电路经配置以在对每一所述列执行所述列变换时判定构成所述列的分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的分区的每一不同子集是否包括至少一个零值、及所述列的分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及

所述列变换电路经配置以绕过或以降低功率的方式对所述数据值中被判定为零值的至少一者执行至少一个变换运算。

45. 如权利要求 44 所述的系统,其中所述列变换电路经配置以绕过对所述数据值中被判定为零值的所述至少一者的所述至少一个变换运算。

46. 如权利要求 44 所述的系统,其中所述列变换电路经配置而以降低功率的方式对所述数据值中被判定为零值的所述至少一者执行所述至少一个变换运算。

47. 如权利要求 46 所述的系统,其中所述列变换电路包括至少一个具有输入端集合的电路,且所述列变换电路经配置以通过如下作业以所述降低功率的方式执行所述至少一个变换运算:使所述具有输入端集合的电路运行,而在对所述数据值中被判定为零值的所述一者的所述变换运算期间不更新所述输入端集合中的至少一个输入端。

48. 如权利要求 44 所述的系统,其中所述 2D 变换是离散余弦反变换。

49. 如权利要求 44 所述的系统,其还包括:

缓冲器,其耦接至所述行变换电路及所述列变换电路,其中所述列变换电路经配置以将所述经局部变换数据块存储于所述缓冲器中,且所述行变换电路经配置以从所述缓冲器中检索所述经局部变换数据块的行。

50. 一种变换引擎电路,其经配置以对输入数据块序列的每一输入数据块执行 2D 变换的行变换,其中每一所述输入数据块均包括输入数据值行及列,且所述 2D 变换包含所述行

变换及列变换,所述变换引擎电路包括:

第一电路,其经耦接以接收每一所述输入数据块的每一行,且经配置以判定构成每一所述行的分区的所述数据值的每一不同子集是否包括至少一个零值、所述行的分区的子集中的第一子集的每一不同子集是否包括至少一个零值、及所述行的分区的子集中的至少一个其他子集的每一不同子集是否包括至少一个零值;及

变换电路,其耦接至所述第一电路,且经配置以对每一所述行的数据值执行所述行变换,其中所述变换电路经配置以在所述第一电路控制下运行,以绕过或以降低功率的方式对所述数据值中被所述第一电路确定为零值的至少一者执行至少一个变换运算。

51. 如权利要求 50 所述的变换引擎电路,其中所述变换电路经配置以绕过对所述数据值中被所述第一电路确定为零值的所述至少一者的所述至少一个变换运算。

52. 如权利要求 50 所述的变换引擎电路,其中所述变换电路经配置而以降低功率的方式对所述数据值中被所述第一电路确定为零值的所述至少一者执行所述至少一个变换运算。

53. 如权利要求 50 所述的变换引擎电路,其中所述 2D 变换是离散余弦反变换。

54. 一种变换引擎电路,其经配置以对输入数据块序列的每一输入数据块执行 2D 变换的列变换,其中每一所述输入数据块均包括输入数据值行及列,且所述 2D 变换包含所述列变换及行变换,所述变换引擎电路包括:

第一电路,其经耦接以接收每一所述输入数据块的每一列,且经配置以判定构成每一所述列的分区的所述数据值的每一不同子集是否包括至少一个零值、所述列的分区的子集中的第一子集的每一不同子集是否包括至少一个零值、及所述列的分区的子集中的至少一个其他子集的每一不同子集是否包括至少一个零值;及

变换电路,其耦接至所述第一电路,且经配置以对每一所述列的数据值执行所述列变换,其中所述变换电路经配置以在所述第一电路控制下运行,以绕过或以降低功率的方式对所述数据值中被所述第一电路确定为零值的至少一者执行至少一个变换运算。

55. 如权利要求 54 所述的变换引擎电路,其中所述变换电路经配置以绕过对所述数据值中被所述第一电路确定为零值的所述至少一者的所述至少一个变换运算。

56. 如权利要求 54 所述的变换引擎电路,其中所述变换电路经配置而以降低功率的方式对所述数据值中被所述第一电路确定为零值的所述至少一者执行所述至少一个变换运算。

57. 如权利要求 54 所述的变换引擎电路,其中所述 2D 变换是离散余弦反变换。

二维变换的方法及系统、视频处理系统和变换引擎电路

技术领域

[0001] 本发明涉及用于对二维 (2D) 数据值阵列执行 2D 变换 (例如离散余弦反变换或哈达玛 <Hadamard> 反变换) 的方法及系统。在典型实施例中, 本发明涉及通过避免进行不为变换零值所需的功率消耗运算而以会降低功率消耗的方式对 2D 图像数据值阵列执行 2D 正交变换的反变换的方法及电路。

背景技术

[0002] 在本发明通篇中, 使用术语视频数据“块”来表示数据子集, 所述数据子集包含视频数据帧, 在所述帧的矩形区域内具有空间位置。视频数据块可以但并非必需由压缩 (或以其他方式编码) 的视频数据组成。视频数据块的实例是以传统方式定义的经 MPEG 编码的视频帧宏块。

[0003] 在许多传统应用中, 图像数据 (例如视频数据) 或其他数据会经历二维 (“2D”) 变换, 且经变换的数据会在此后进行反变换, 以恢复原始数据。这些变换的实例包括 2D 离散余弦变换 (二维 “DCT”)、2D 哈达玛变换、及 2D 傅立叶变换。

[0004] 在本发明通篇中, “绕过”一运算这一表达方式 (其原本将产生运算输出值) 表示产生或确定一替代输出值 (代替运算输出值) 而不实际执行该运算。“绕过”一如下运算将会使节点接地 (从而将其迫至地电位) 而不在乘法电路及加法电路中实际执行乘法及加法步骤: 对乘法电路的输入端确定一零值 “z1” 及一非零常数 “c” 以使该电路在其输出端上确定一电流 “cz1”, 对第二乘法电路的输入端确定另一零值 “z2” 及不同的非零常数 “d”, 以使该电路在其输出端上确定一电流 “dz2”, 并响应于电流 “cz1” 及 “dz2” 来操作加法电路, 以在一节点处确定输出电压 “cz1+dz2” (等于高于地电位零伏)。

[0005] 本发明涉及用于对 2D 数据值阵列 (即由数据值行及列组成的阵列) 执行 2D 变换的方法及系统, 其中每一值均具有较大的为零值的机率。在典型实施例中, 本发明涉及一种用于对 2D 数据值阵列执行 2D 正交变换的反变换 (例如 2D 离散余弦反变换或哈达玛反变换) 的改良的方法及系统。在一类较佳实施例中, 本发明涉及一种用于对 DCT 系数执行二维 IDCT (2D 离散余弦反变换) 的改良的方法及系统。所述 DCT 系数是通过对视频数据阵列 (或其他图像数据) 执行 2D 离散余弦变换而产生, 且分别具有较大的为零值的机率。

[0006] 在本发明通篇中, “零值” (或“零数据值”) 这一表达方式表示指示为零值的数据。类似地, “零输入数据值” 这一表达方式表示指示为零值的输入数据。例如, 零输入值可为具有零值的输入数据字 (例如 DCT 系数, 或者颜色分量或视频数据像素)。

[0007] 在本发明通篇中, “稀疏” 数据这一表达方式 (例如要经历反变换的稀疏数据块) 表示指示有可能为零值的值的数据。例如, 指示相对多的零值及相对少的非零值的输入数据块 (例如 DCT 系数块) 是稀疏数据块。

[0008] 反变换实施方案通常是任何要遵从任何视频压缩及解压缩标准的系统的实施方案的主要部分。其为一在计算上繁杂的过程, 且极大地影响处理循环及功率消耗要求。执行视频压缩及解压缩标准的移动装置 (例如便携式媒体播放器) 具有特别严格的处理循环

及功率消耗要求；其需要满足由该应用所设立的严格的性能要求并消耗极低的功率，以使电池寿命最大化；且变换引擎通常必须能够支持多种压缩标准及与这些标准相伴随的不同要求。

[0009] 对数据块进行 2D 变换（包括 2D 反变换）的典型传统实施方案是使用下述技术的不同组合来提高性能或减小功率：

[0010] 1. 不对被外部构件识别为未编码块的块进行变换（其中提供至变换引擎的每一输入块均由该外部构件识别为编码块或未编码块）。然而，该技术存在缺点，包括其可导致执行不必要的变换运算（例如对被识别为编码块但仅由零 DC 系数组成的块进行变换）。

[0011] 2. 识别每一输入数据块中完全由零值组成的全部行或列（“零行”或“零列”），并绕过原本将对每一此种行或列执行的正常变换运算（例如通过为每一零行或零列输出预定值—通常为“零”）。所述零行及零列可由外部装置加以规定或者由变换引擎在内部进行识别。然而，在许多其中行（或列）不为零行（或零列）、但为仅包含极少数量非零值的稀疏的行（或列）的常见情形中，此种传统技术并不会提供性能或减小功率；

[0012] 3. 识别（根据输入数据）指示应使用相同系数（先前所确定的用于乘以输入数据行或列中的数据值的系数）乘以下一输入数据行或列中的数据值的状态，且不对这些系数进行更新—原本在确定用于乘以所述下一输入数据行或列中的数据值的新系数时将会执行更新；以及

[0013] 4. 执行分布式算术变换（2D 变换的基于查找表的实施方案）。典型的基于查找表的实施方案通过减小对块进行变换时所必须执行的乘法运算的数量来减小开销。然而，设计此种实施方案通常非常复杂，因为通常需要使用非常大的 ROM 表以及多端口 ROM，且设计约束条件通常会限制可实现的功率消耗改进量。

[0014] 在第 2005/0033788 号美国专利申请公开案以及相关的第 6,799,192 号美国专利中所述的另一种传统 2D 变换中，确定数据块每一列中的最末非零项（当执行 DDCT 的列变换阶段时），且变换系统随后跳转至八个不同“专门 IDCT”程序例程中恰当的一者来以软件形式执行 IDCT 运算，以对每一列进行反变换。显然，可采用较简单的变换运算（所需的乘法及加法运算较少）来处理具有相对多的零（如具有处于较高位置的最末非零值所指示）的列，并可使用较复杂的变换运算（所需乘法及加法运算较多）来处理具有较少的零（如具有处于较低位置的最末非零值所指示）的列。所述参考文献还教导，当执行 IDCT 的行变换阶段时（在列变换阶段之后），确定块中每一行的最末非零项，且变换系统随后跳转至八个不同“专门 IDCT”程序例程中恰当的一者来以软件形式执行 IDCT 运算，以对每一行进行反变换。

[0015] 在第 2005/0033788 号美国专利申请公开案及第 6,799,192 号美国专利中所述的技术存在诸多问题及局限性，包括当处理具有典型的零值及非零值图案的许多列及行时，因此种技术不会提高性能或减小功率消耗，因而此种技术效率偏低。例如，当所要变换的列或行包含零（尤其是许多零）、但最末项为非零时，此种技术将选择一会消耗很大功率的复杂（例如最复杂）的“专门 IDCT”例程来变换该列或行。相比之下，本发明的较佳实施例则通过不对行及列中由零值组成的部分（例如对由零值组成的每半行或半列、或者每四分之一行或四分之一列）执行变换运算或以功率减小方式执行这些变换运算来提高性能及减小功率消耗。本发明的某些较佳实施例通过不对所要变换的行或列中的每一单独零值执行

变换运算来提高性能及减小功率消耗（或者以功率减小的方式对行或列中的每一单独零值执行变换运算）。

[0016] 在第 2005/003378S 号美国专利申请公开案或第 6,799,192 号美国专利中未表明可藉由独立地处理所要变换的每一行或列的各子集来增强其中所揭示技术可实现的性能改良及功率消耗降低,且未表明如何这些进行或是否可这些进行。相比之下,本发明的较佳实施例可按顺序对所要变换的每一行或列的不同子集执行相同的运算（例如反变换）,其中每一行或列的各子集决定对所述行或列的划分,且仅通过减小决定每一此种划分的子集的大小便可增强这些实施例所能实现的性能改良及功率消耗降低。例如,本发明的某些较佳实施例对所要变换的每一 $8N$ 位的行或列的各 $2N$ 位的子集按顺序执行各组运算（每一行或列四组运算）,以实现优异的性能改良及功率消耗降低优点,且本发明的其他较佳实施例对所要变换的每一 $8N$ 位的行或列的 N 位的子集按顺序执行各组运算（每一行或列八组运算）,以实现甚至更好的性能改良及功率消耗降低优点。

[0017] 另一种传统的 2D 变换由 Rohini Krishnan 等人阐述于标题为“Design of a 2DDCT/IDCT Application Specific VLIW Processor Supporting Scaled and Sub-sampled Blocks”（第 16 届 VLSI 设计国际会议,六页（2003））的论文中。该论文教导对 IDCT 电路确定完整数据块的缩减规模形式（例如通过舍弃一 8×8 块的偶数行而产生的 8×4 块）,且操作所述 IDCT 电路以对所述缩减规模形式的块进行反变换包括绕过某些原本用于对所述完整块进行反变换的 IDCT 电路。该方法可避免计算将最终被舍弃的输出值,但不检测并跳过将根本无助于最终结果的运算。

[0018] 另一种传统 2D 变换阐述于第 5,883,823 号美国专利中。此种变换识别所要变换的输入块的各区域,并以不同方式处理每一区域（例如对某些区域的所有元素执行 IDCT,且仅对其他区域的非零元素执行 IDCT）。例如,第 5,883,823 号美国专利清楚地揭示（在第 10 栏第 53 行 - 第 11 栏第 26 行中）一种 IDCT 计算,其中对一 8×8 块的一个象限（即对应于最低频率范围的 4×4 象限）的所有元素（无论是零还是非零）执行一“区域性”IDCT 计算,并对该 8×8 块的其他三个 4×4 象限（即对应于较高频率范围的三个 4×4 象限）中每一者的仅非零元素执行另一 IDCT 计算。然而,第 5,883,823 号美国专利并未揭示或表明如何识别每一要仅对非零元素执行 IDCT 计算的区域的非零元素（或者如何有效地识别这些非零系数）、或者如何仅对块中区域的非零元素执行 IDCT 计算、或者如何有效地（且以消耗降低的功率的方式）仅对这些非零元素执行此种 IDCT 计算。

发明内容

[0019] 在一类实施例中,本发明是一种经配置以对一输入数据块序列中的每一块执行 2D 变换（例如离散余弦反变换）的系统,其中每一块均包含输入数据值行及列,且所述 2D 变换包含一行变换及一列变换。在这些实施例中,所述系统经配置以通过如下方式执行所述 2D 变换:对每一块的所有行执行行变换以产生经局部变换的数据块、并然后对所述经局部变换的数据块的每一列执行列变换,或者通过对每一块的所有列执行所述列变换以产生一经局部变换的数据块、并然后对所述经局部变换的数据块的每一行执行所述行变换。为简化说明,我们应说明经配置以通过如下方式执行所述 2D 变换的类别的实施例:对每一块的所有行执行行变换以产生经局部变换的数据块,并然后对所述经局部变换的数据块的每一

列执行列变换。应了解,本文中每一处所提及的“行”及“列”均可分别以提及“列”及“行”来取代,以描述该类别中的其他实施例。

[0020] 在本文中,数据值集合(例如行或列)的“子集”这一术语是使用其广义,并可表示一行(或列)数据值、一行(或列)数据值中的偶数元素、一行(或列)数据值中的奇数元素、一行(或列)数据值中的每第N个数据值、位反转次序的一行或列数据值中的偶数元素(适用于FFT蝶形运算)、或另一种数据值子集。

[0021] 为执行行变换,所述类别的系统包括经配置以对输入数据块的每一行的输入数据值执行变换运算(通常包含乘法及加法)以产生一经局部变换的数据块的电路。所述系统通常包括一缓冲器,所述经局部变换的数据即存储于所述缓冲器中。为执行所述列变换,所述系统包括经配置以对所述经局部变换的数据块中每一列的数据值执行变换运算(通常包含乘法及加法)的电路。所述类别的每一实施例均经配置以使得当对每一行输入数据执行所述行变换时,其均判定构成所述行的第一分区的数据值的每一不同子集是否包括至少一个零值(例如由零值组成)、判定所述行的第一分区的第一子集的每一不同子集是否包括至少一个零值、及判定所述行的第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及当对包含至少一个零输入数据值及至少一个非零输入数据值的每一所述行执行所述行变换时,绕过或以降低功率的方式执行对至少一个(且较佳对每一个)所述零输入数据值的至少一个变换运算,其中假如所述零值为非零值,此种变换运算原本将以消耗全功率的方式执行(例如,绕过至少一个原本将使用乘法及加法电路执行的乘法或加法)。当对每一包含至少一个零输入数据值及至少一个非零输入数据值的行执行所述行变换时,某些此种实施例中的用于对所述零输入值执行所述变换运算的电路均允许,而不更新其输入中的至少一个,以避免消耗在使每一此种输入双态转换时原本将消耗的功率。

[0022] 较佳地,所述类别的一实施例还经配置以使得当对每一列经局部变换的数据执行所述列变换时,其均判定构成所述列的第一分区的数据值的每一不同子集是否包括至少一个零值(例如由零值组成)、判定所述列的第一分区的第一子集的每一不同子集是否包括至少一个零值、及判定所述列的第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值,及当对包含所述经局部变换的数据的至少一个零值及所述经局部变换的数据的至少一个非零值的每一所述列执行所述列变换时,其均绕过(或以降低功率的方式执行)所述变换运算中的至少一者,然而假如所述零值是非零值,则其原本将以消耗全功率的方式对所述经局部变换的至少一个(且较佳对每一个)所述数据值执行所述变换运算中的至少一者(例如,至少一个原本将使用乘法及加法电路执行的乘法或加法)。例如,当对每一包含至少一个零数据值及至少一个非零数据值的列执行所述列变换时,一个此种实施例经配置以绕过对所述列的零值的变换运算,而假如所述零值为非零值时,原本将执行对所述列的零值的变换运算(例如,该实施例绕过用于执行所述变换运算的电路)。

[0023] 本发明的较佳实施例判定所要变换的每一数据块是否完全由零值组成。在判断出数据块完全由零值组成时,绕过或以降低功率的方式执行对所述块的值的变换运算(行及列变换运算二者)。这些较佳实施例还按顺序(例如以迭代方式)判定所要变换的数据块中每一行或列的诸多不同子集中的每一者是否包含至少一个零值。此一实施例的一实施例将涉及由值 x_i (其中 i 是处于 $0 \leq i \leq N-1$ 范围内的整数,且 N 为偶数整数)组成的数据行(或列),将所述行(或列)划分成第一数据值子集与不同于所述第一子集的第二数据值子

集,将所述第一子集划分成第三数据值子集与不同于所述第三子集的第四子集,并将所述第二子集划分成第五数据值子集与不同于所述第五子集的第六子集。所述实例性实施例判定所述第一子集是否完全由零值组成及所述第二子集是否完全由零值组成。通常,当所述行(或列)由余弦变换系数(通过对视频数据帧执行 DCT 而产生)组成时,所述第一子集由低频系数(值 x_i , 其中 i 是处于 $0 \leq i \leq (N/2)-1$ 范围内的整数)组成,所述第二子集由高频系数(值 x_i , 其中 i 是处于 $N/2 \leq i \leq N-1$ 范围内的整数)组成,且所述第二子集具有很高的仅由零值组成的机率(且具有比所述第一子集高得多的仅由零值组成的机率)。

[0024] 该实例性实施例在确定出第一子集完全由零值组成之后,绕过对第一子集中的值的变换运算或者以降低功率的方式执行变换运算(例如绕过用于执行这些运算的电路,或者操作所述电路,使其至少一个输入不被更新,以避免消耗功率-原本将会消耗功率来使每一此种输入双态转换)。在确定出第二子集完全由零值组成时,绕过或以降低功率的方式执行对第二子集中的值的变换运算。

[0025] 在确定出第一子集与第二子集中每一者均包括至少一个非零值时,该实例性实施例判定是否第三子集、第四子集、第五子集、及第六子集中的每一者均完全由零值组成。在一典型实施方案中,第一子集由低频系数(值 x_i , 其中 i 是处于 $0 \leq i \leq (N/2)-1$ 范围内的整数)组成,第二子集由高频系数(值 x_i , 其中 i 是处于 $N/2 \leq i \leq N-1$ 范围内的整数)组成,第三子集由第一子集的偶数值(值 x_i , 其中 i 是处于 $0 \leq i \leq (N/2)-1$ 范围内的偶数整数)组成,第四子集由第一子集的奇数值(值 x_i , 其中 i 是处于 $0 \leq i \leq (N/2)-1$ 范围内的奇数整数)组成、第五子集由第二子集的偶数值组成,且第六子集由第二子集的奇数值组成。对于被确定完全由零值组成的第三子集、第四子集、第五子集、及第六子集中的每一者,均绕过或以降低功率的方式执行对该子集的值变换运算(例如绕过用于执行这些运算的电路,或者操作所述电路而不更新其至少一个输入,以避免消耗功率-原本将会消耗功率来使每一此种输入双态转换)。而对于被确定包含至少一个非零值的第三子集、第四子集、第五子集、及第六子集中的每一者,均以消耗全功率的方式对该子集的值执行变换运算。

[0026] 在上述实例性实施例的变化形式(及本发明的其他实施例)中,构成所要变换的数据块的每一行或列(或一行或列的子集)的数据值是在判定每一包含该行或列(或其子集)的一分区的不同子集是否完全由零值组成(或为零值)之前或期间重新排序。例如,如果一行由值 x_i (其中 i 是 $0 \leq i \leq N-1$ 范围内的整数,其中 N 是偶数整数)组成,所述分区由所述行的不同的第一及第二子集组成,第一子集由所述行的不同的第三及第四子集组成,且第二子集由所述行的不同的第五及第六子集组成,则所述第一子集由值 x_i (其中 i 是 $0 \leq i \leq (N/2)-1$ 范围内的整数)组成,所述第二子集由高频系数(值 x_i , 其中 i 是 $N/2 \leq i \leq N-1$ 范围内的整数)组成,

[0027] 所述第三子集由值 x_i (其中 i 处于 $0 \leq i \leq (N/4)-1$ 范围内)组成,这些值 x_i 为第一子集的重重新排序版本的偶数值,

[0028] 所述第四子集由值 x_i (其中 i 处于 $N/4 \leq i \leq (N/2)-1$ 范围内)组成,这些值 x_i 为第一子集的重重新排序版本的奇数值,

[0029] 所述第五子集可由值 x_i (其中 i 处于 $N/2 \leq i \leq (3N/4)-1$ 范围内)组成,这些值 x_i 为第二子集的重重新排序版本的偶数值,

[0030] 所述第六子集可由值 x_i (其中 i 处于 $3N/4 \leq i \leq N-1$ 范围内) 组成, 这些值 x_i 为第二子集的经重新排序版本的奇数值。

[0031] 在另一类实施例中, 本发明是一种用于对一输入数据块序列执行 2D 变换的方法, 所述块中的每一者均包括输入数据值行及列, 且所述 2D 变换包括行变换及列变换。在该类的典型实施例中, 每一块均为 DCT 系数块, 所述 DCT 系数是通过视频数据执行 DCT 而产生, 且所述 2D 变换是离散余弦反变换。在这些典型实施例中以及在该类的其他实施例中, 所述序列中的许多或所有块 (例如至少实质上所有块) 均为稀疏数据块。所述 2D 变换可包括如下步骤: 对每一输入数据块的所有行执行行变换以产生经局部变换的数据块, 并然后对所述经局部变换的数据块的每一列执行所述列变换; 或者包括如下步骤: 对每一输入数据块的所有列执行所述列变换以产生一经局部变换的数据块, 并然后对所述经局部变换的数据块的每一行执行所述行变换。为简化说明, 我们应说明该类中其中所述 2D 变换包括如下步骤的实施例: 对每一输入数据块的所有行执行所述行变换以产生经局部变换的数据块, 并然后对所述经局部变换的数据块的每一列执行所述列变换。应了解, 每一处所提及的“行”及“列”均可分别以提及“列”及“行”来取代, 以描述该类别的其他实施例。

[0032] 在所述类别中的某些实施例中, 所述方法包括如下步骤:

[0033] (a) 对所述输入数据块中一者的每一行执行所述行变换, 包括通过对每一所述行的输入数据值执行变换运算, 以产生一经局部变换的数据块; 以及

[0034] (b) 对所述经局部变换的数据块的每一列执行所述列变换, 包括通过对每一所述列的数据值执行其他变换运算, 其中步骤 (a) 包括如下步骤:

[0035] 判定构成每一所述行的第一分区的数据值的每一不同子集是否包括至少一个零值 (例如由零值组成), 判定所述第一分区的第一子集的每一不同子集是否包括至少一个零值, 及判定所述第一分区的至少一个其他子集的每一不同子集是否包括至少一个零值; 及

[0036] 当对包含至少一个零输入数据值及至少一个非零输入数据值的每一所述行执行所述行变换时, 绕过 (或以降低功率的方式执行) 所述变换运算中的至少一者, 而假如每一所述零输入值为非零值时, 原本将以消耗全功率的方式对至少一个 (且较佳对每一个) 所述零输入数据值执行所述变换运算 (例如, 绕过至少一个原本将使用乘法及加法电路对所述行的至少一个 (且较佳对每一个) 所述零输入数据值执行的乘法或加法)。

[0037] 例如, 当对每一包含至少一个零输入数据值及至少一个非零输入数据值的行执行所述行变换时, 步骤 (a) 包括如下步骤: 操作一具有第一输入端及第二输入端的乘法电路, 以执行乘法运算 (其中将对第一输入端确定的零输入值乘以对第二输入端确定的第二值), 而不更新对第二输入端确定的值, 以避免消耗在使第二输入端双态转换时原本将消耗的功率。

[0038] 较佳地, 步骤 (b) 包括如下步骤:

[0039] 判定构成每一所述列的分区的每一不同子集是否包括至少一个零值 (例如由零值组成), 判定所述分区的第一子集的每一不同子集是否包括至少一个零值, 及判定所述分区的至少一个其他子集的每一不同子集是否包括至少一个零值; 及

[0040] 当对包含至少一个零输入数据值及至少一个非零输入数据值的每一所述列执行所述列变换时, 绕过 (或以降低功率的方式执行) 所述其他变换运算中的至少一者, 而假

如每一所述零输入值为非零值时,原本将以消耗全功率的方式对至少一个(且较佳对每一个)所述零输入数据值执行所述额外变换运算(例如,绕过至少一个原本将使用乘法及加法电路对所述列的至少一个(且较佳对每一个)所述零输入数据值执行的乘法或加法)。

[0041] 根据本发明的典型实施例构建的变换电路的优点包括:

[0042] 因避免冗余的或不必要的计算或计算步骤而能提高(且较佳最佳化)计算效率(从而允许以较低频率运行)(例如,在变换输入数据值块方面,本发明变换引擎的典型实施例的性能与每一块的非零输入数据值的数量成正比)。此是在比传统变换引擎更精细的粒度水平-在变换输入数据值块方面,传统变换引擎的性能与每一块的包含至少一个非零值的行的数量成比例);

[0043] 切换活动减少而使功率消耗降低;

[0044] 提供智能的中间缓冲器存储器管理(在较佳实施例中);及

[0045] 通过避免冗余的或不必要的计算或计算步骤(例如避免不必要的乘法器输入双态转换)而使整数变换的性能加快。

[0046] 在某些实施例中,本发明的系统是一种视频处理系统(例如一种管线式视频解码系统),其包括一根据本发明构建的变换引擎。在某些此种实施例中,所述视频处理系统经配置以用作便携式媒体播放器的视频处理子系统。在其他实施例中,本发明的系统是一种便携式媒体播放器,其包括一视频处理子系统,该视频处理子系统包括根据本发明构建的变换引擎。

[0047] 本发明的其他方面是用于本发明系统任一实施例中的变换引擎及变换引擎电路、以及在本发明系统任一实施例的运行期间所执行的方法。

附图说明

[0048] 图 1 是可根据本发明实施例构建的包括变换引擎的管线式视频解码系统的方块图;

[0049] 图 2 是可根据本发明构建的处理系统(具有视频解码能力)的方块图;

[0050] 图 3 是输入数据块、所述块的已经过行变换的版本、及所述块的已经过行变换及列变换的版本的图式;

[0051] 图 4 是本发明方法的一实施例(其可由本发明系统的一类实施例来执行)的步骤的流程图;

[0052] 图 5 是在实施本发明的图 2 所示电路 7(或电路 11)的典型实施方案中所用的变换电路的方块图;

[0053] 图 6 是本发明管线式视频解码系统的一实施例中各元件的方块图。

具体实施方式

[0054] 下面将参照图 1、2、3、4、5 及 6 来说明本发明的实施例。

[0055] 图 1 是一管线式视频解码系统的系统层次方块图,所述管线式视频解码系统包括如图所示沿系统总线连接的位流引擎 1、变换引擎 3(其可根据本发明构建而成)、及像素重构引擎 5。位流引擎 1 接收包含经处理的频域系数块的经编码视频数据,该等频域系数是通过视频像素块执行 2D 变换(例如二维 DCT)而产生。位流引擎 1 处理经编码的视频数

据,通常是通过对其执行拆开及反量化,并对变换引擎 3 确定所得到的经处理频域系数块序列。变换引擎 3 对所述序列中的每一块执行 2D 反变换,以产生经局部解码视频像素块,并将所述经局部解码视频像素确定至像素重构引擎 5,以供进一步进行视频解码。像素重构引擎 5 响应于所述经局部解码视频像素而产生完全解码的像素,并将所述完全解码的像素确定至系统总线。

[0056] 图 1 所示系统可为具有视频解码能力的便携式媒体播放器或另一种便携式数据处理系统(或并非打算为便携式的数据处理系统)的子系统,且此一子系统可通过构建本发明的变换引擎 3 而根据本发明构建而成。此一处理系统可以且通常将包括图 1 中所未显示的其他元件。

[0057] 预期本发明的某些实施例是由不具有图 1 所示结构的系统构建而成,或者由不具有图 2 所示结构的处理系统构建而成。

[0058] 图 2 是图 1 中变换引擎 3、或另一用于对数据值块执行二维变换(例如 2D 反变换)的装置的典型实施方案的方块图。图 2 包括如图所示相连的行/列变换电路 7、存储缓冲器 9、及列/行变换电路 11。在某些实施方案中,图 2 中的电路 7 经配置以对由其接收的输入数据块序列中每一块的所有行执行行变换并将所得到的数据确定至存储缓冲器 9,且电路 11 经配置以从缓冲器 9 中读取此种数据的列并对从行变换得到的每一数据块的所有列执行列变换。在其他实施方案中,图 2 中的电路 7 经配置以对由其接收的输入数据块序列中每一块的所有列执行列变换并将所得到的数据确定至存储缓冲器 9,且电路 11 经配置以从缓冲器 9 中读取此种数据的行并对从列变换得到的每一数据块的所有行执行行变换。

[0059] 图 3 是可输入至图 2 所示实施方案中的电路 7 的一种类型的输入数据块(标记为“ I ”)、块 I 的已经过行变换(例如在图 2 所示此种实施方案的电路 7 中)的版本(标记为块“ II ”)、及块 I 的已经过行变换及列变换(例如在图 2 所示此种实施方案的电路 7 及电路 11 中)的版本(标记为块“ III ”)的图式。图 3 中的块 I 是一 4×4 频域系数块,其包含四个系数行及四个系数列,其中每一系数均具有标记为“0”的零值,且每一系数均具有标记为“ X ”的非零值。所述行由参考符号“ $R1$ ”至“ $R4$ ”标识,且所述列由参考符号“ a ”至“ d ”标识。

[0060] 典型的传统变换引擎经配置以识别完全由零值组成的(所要变换的每一块的)所有行或列(即“零行”或“零列”)并绕过原本将对每一所识别的零行或零列执行的正常变换运算。此种传统变换引擎将块 I 的行 $R1$ 及 $R3$ 识别为零行并绕过原本将对行 $R1$ 及 $R3$ 执行的变换计算。

[0061] 相比之下,本发明变换引擎的典型实施例(例如实施本发明的图 2 所示引擎的实施方案)则经配置以不仅识别每一所要变换的块的零行及零列(例如块 I 的行 $R1$ 及 $R3$),以绕过原本将对零行执行(假如其未将所述行识别为零行)的行变换计算及绕过原本将对零列执行(假如其未将所述列识别为零列)的列变换计算,而且还能在对非零行及非零列执行变换运算(例如对块 I 的行 $R2$ 及 $R4$ 执行的行变换计算)期间节省额外的功率。所述额外的功率节省是通过如下方式来实现:识别完全由零数据值组成的行的子集(及列的子集)(例如,由列 c 及 d 中的系数组成的块 I 中行 $R2$ 的子集,或者由列 b 及 d 中的系数组成的块 I 的行 $R4$ 的子集),并避免(或以降低功率的方式执行)假如所述子集并不完全由零数据值组成时原本将对每一此种子集执行的变换计算。

[0062] 考虑本发明变换引擎的此一实施例具有图 2 所示结构且经配置以对一 4×4 DCT 变换系数块序列中每一块执行 IDCT (反 DCT) 变换的情形。每一此种块的每一行均由两个低频系数 (例如图 3 中每一块的列“a”及“b”中的系数) 以及两个高频系数 (例如图 3 中每一块的列“c”及“d”中的系数) 组成。此一块的每一列均由两个低频系数 (例如图 3 中每一块的行“R1”及“R2”中的系数) 以及两个高频系数 (例如图 3 中每一块的行“R3”及“R4”中的系数) 组成。在此种情形中, 每一行及每一列的较高频系数具有比较低频系数高的仅由零值组成的机率。

[0063] 本发明变换引擎的一个此种实施例的电路 7 经配置以判定每一 DCT 变换系数块的每一行的这两个最低频系数是否完全由零值组成, 判定每一此种行的这两个最高频系数是否完全由零值组成, 如果每一行的这两个最低频系数完全由零值组成, 则绕过对其进行的行变换计算 (其中假如其并非完全由零值组成, 则原本将以消耗全功率的方式对这些系数执行这些计算) 或者以降低功率的方式执行这些计算, 及如果每一行的这两个最高频系数完全由零值组成, 则绕过对其进行的行变换计算 (其中假如其并非完全由零值组成, 则原本将以消耗全功率的方式对这些系数执行这些计算) 或者以降低功率的方式执行这些计算。例如, 此一实施例的电路 7 可经配置以绕过对块 I 的行 R1、R2、R2、及 R4 中每一者的这两个最高频系数的行变换计算。

[0064] 本发明变换引擎的此种实施例的电路 7 还经配置以判定对其确定的每一 DCT 变换系数块的每一行的这两个最低频分量的偶数系数 (例如, 图 3 所示块 I 的列“b”中的每一系数) 是否是零值, 判定对其确定的每一 DCT 变换系数块的每一行的这两个最低频分量的奇数系数 (例如, 图 3 所示块 I 的列“a”中的每一系数) 是否是零值, 判定对其确定的每一 DCT 变换系数块的每一行的这两个最高频分量的偶数系数 (例如, 图 3 所示块 I 的列“d”中的每一系数) 是否是零值, 以及判定对其确定的每一 DCT 变换系数块的每一行的这两个最高频分量的奇数系数 (例如, 图 3 所示块 I 的列“c”中的每一系数) 是否是零值。这些实施例的电路 7 还配置成以降低功率的方式执行原本对每一被确定为零值的此种系数执行 (以消耗更大功率的方式) 的乘法计算, 以对所述行进行变换。通常, 每一此种降低功率的乘法均通过如下方式执行: 操作一具有第一输入端及第二输入端的乘法电路, 以将零值 (确定至第一输入端) 乘以第二值 (确定至第二输入端), 而不更新确定至第二输入端的值, 从而避免消耗功率 - 原本在确定至第一输入端的值为非零值时, 将消耗功率来使第二输入双态转换 (且第二输入端处更新后的第二值需要乘以此非零值)。

[0065] 本发明变换引擎的此一实施例的电路 11 还较佳经配置以判定从缓冲器 9 读取的每一 4×4 经局部变换系数块的每一列的第一半 (即前两个数据值) 是否完全由零值组成, 判定每一此种列的第二半 (后两个数据值) 是否完全由零值组成, 如果每一列的第一半完全由零值组成, 则绕过对其进行的列变换计算 (其中假如其并非完全由零值组成, 则原本将以消耗全功率的方式对这些值执行这些计算) 或以降低功率的方式执行这些计算, 以及如果每一列的第二半完全由零值组成, 则绕过对其进行的列变换计算 (其中假如其并非完全由零值组成, 则原本将以消耗全功率的方式对这些值执行这些计算) 或以降低功率的方式执行这些计算。

[0066] 本发明变换引擎的此一实施例的电路 11 还经配置以判定从缓冲器 9 读取的每一 4×4 经局部变换系数块的每一列的第一半中的偶数数据值 (例如图 3 所示块 II 的行 R2 中

的每一值) 是否是零值, 判定每一此种列的第一半中的奇数数据值 (例如图 3 所示块 II 的行 R1 中的每一值) 是否是零值, 判定每一此种列的第二半中的偶数数据值 (例如图 3 所示块 II 的行 R4 中的每一系数) 是否为零值, 以及判定每一此种列的第二半中的奇数数据值 (例如图 3 所示块 II 的行 R3 中的每一系数) 是否为零值。这些实施例的电路 11 还配置成以降低功率的方式对被确定为零值的每一此种数据值执行原本将执行 (以消耗更大功率的方式) 的乘法计算, 以对所述列进行变换。通常, 每一此种降低功率的乘法均通过如下方式执行: 操作一具有第一输入端及第二输入端的乘法电路, 以将零值 (确定至第一输入端) 乘以第二值 (确定至第二输入端), 而不更新确定至第二输入端的值, 从而避免消耗功率 - 原本在确定至第一输入端的值为非零值时, 将消耗功率来使第二输入双态转换 (且第二输入端处更新后的第二值需要乘以此非零值)。

[0067] 在上述实例的其中每一行及列均包含 8 个数据值的变化形式中, 每一判定一行 (或列) 的第一 (或第二) 半中的奇数 (或偶数) 数据值是否是零的步骤均被一判定所述行 (或列) 的所述一半的第一 (或第二) 半中的数据值是否完全由零组成的步骤所取代, 且所述引擎经配置以绕过对完全由零值组成的每一半行 (或半列) 的每一半的行 (或列) 变换计算 (其中假如这些值并非安全由零值组成, 则原本将以全功率对其执行这些计算) 或者以降低功率的方式执行这些计算。较佳地, 所述引擎还经配置以判定每一此种半行 (或半列) 中的奇数值是否为零值, 判定每一此种半行 (或半列) 的每一数据值是否为零值, 及以降低功率的方式对每一此种数据值执行原本将执行 (以消耗更大功率的方式) 的乘法计算, 以对相关行或列进行变换。

[0068] 在这两个上述实例中任一实例的变化形式中, 包含要由电路 7 或 11 进行变换的块的每一行或列 (或行或列的子集) 的数据值均在判定每一包含所述行或列 (或其子集) 的分区的不同子集是否完全由零值组成 (或为零值) 之前或期间进行重新排序。例如, 如果一行 (或列) 是由值 x_i (其中 i 是处于 $0 \leq i \leq N-1$ 范围内的整数, 且 N 为偶数整数) 组成, 所述分区是由所述行 (列) 的不同的第一及第二子集组成, 所述第一子集是由所述行 (列) 的不同的第三及第四子集组成, 且所述第二子集是由所述行 (列) 的不同的第五及第六子集组成, 则所述第一子集由值 x_i (其中 i 是处于 $0 \leq i \leq (2/N)-1$ 范围内的整数) 组成, 所述第二子集由值 x_i (其中 i 是处于 $2/N \leq i \leq N-1$ 范围内的整数) 组成,

[0069] 所述第三子集可由值 x_i (其中 i 处于 $0 \leq i \leq (N/4)-1$ 范围内) 组成, 或者其可由第一子集的偶数值组成, 抑或其可由第一子集的经重新排序版本的偶数值组成,

[0070] 所述第四子集可由值 x_i (其中 i 处于 $N/4 \leq i \leq (N/2)-1$ 范围内) 组成, 或者其可由第一子集的奇数值组成, 抑或其可由第一子集的经重新排序版本的奇数值组成,

[0071] 所述第五子集可由值 x_i (其中 i 处于 $N/2 \leq i \leq (3N/4)-1$ 范围内) 组成, 或者其可由第二子集的偶数值组成, 抑或其可由第二子集的经重新排序版本的偶数值组成, 以及

[0072] 所述第六子集可由值 x_i (其中 i 处于 $3N/4 \leq i \leq N-1$ 范围内) 组成, 或者其可由第二子集的奇数值组成, 抑或其可由第二子集的经重新排序版本的奇数值组成。

[0073] 再次参见图 3 所示块 I 的行 R4, 该行 R4 中的偶数系数是零值, 且该行中的奇数系数并非完全由零值组成。为对该行 R4 进行变换, 本发明变换引擎的某些实施例将通过阻止在对所述行的偶数系数执行行变换运算期间阻止用于执行此种行变换运算的乘法器的输入双态转换来降低功率消耗。此有助于降低乘法器在变换期间所消耗的切换功率 (即降低

动态功率消耗)。

[0074] 再次参见图 3,在块 I 的行 R4 中,整个行中仅第一系数是非零(经编码的)系数。本发明变换引擎的较佳实施例经配置以辨别此种类型(即其中仅第一数据值为非零)的行或列,并以一种所消耗功率小于为对完全由非零值组成的行或列进行变换所需功率的方式对此种行或列进行变换。例如,(本发明变换引擎的具有图 2 所示结构的实施例的)电路 7 的较佳实施方案配置成以降低功率的方式对其中仅第一值为非零值的输入数据行执行行变换—仅响应于行中的非零系数且不对行中所有数据值(此将消耗全功率)执行正常行变换,并将所得到的经局部变换的数据值存储于缓冲器 9 中。

[0075] 本发明变换引擎的较佳实施例通过如下方式来降低功率消耗:检测所要变换的块的各单独数据值(例如,在逐个系数的基础上对要经历直接余弦反变换的 DCT 系数块的各单独系数执行零检测),且对于被判定为零值的所有数据值,均确保使被用于变换至少一个所述零值的至少一个乘法器的至少一个(且较佳每一个)输入均不会自其先前值发生切换。此会降低引擎中的动态功率消耗。

[0076] 在本发明变换引擎的具有图 2 所示结构的实施例的较佳实施方案中,电路 7 将其所产生的经局部变换的数据值存储于内部缓冲器 9 中。电路 11 从缓冲器 9 中读取该数据,并对该数据的每一列(或行)执行列(或行)变换。通常,电路 11 可在缓冲器中存在至少一列(行)数据且所述数据可供用于变换时,即刻执行列(行)变换计算。因此,在较佳实施例中,电路 7 经配置以对其所执行的行(列)变换运算重新排序,以尽可能快地产生(并存储于缓冲器 9 中)恰好足够多的经局部变换的数据,以使电路 11 对所述数据的完整列(行)执行列(行)变换。通过将电路 7 配置成以此种重新排序(例如交错的)方式执行行(列)变换,可降低系统的功率消耗(例如,通过使进行内部缓冲器存储所消耗的功率最小化)。

[0077] 另一选择为,当根据本发明变换引擎的具有图 2 所示结构的实施例来执行对称变换时,以一种能利用变换的对称性的方式配置电路 7,以执行足以在执行额外的行变换计算之前产生(并存储于缓冲器 9 中)两列数据的行变换计算、或者执行足以在执行额外的列变换计算之前产生(并存储于缓冲器 9 中)两行数据的列变换计算。电路 7 的此种实施例所执行的行(或列)变换继续尽可能快地产生(并存储于缓冲器 9 中)恰好足够多的经局部变换数据,以使电路 11 对其他完整的经缓冲数据列(行)执行列(行)变换。关于是将电路 7 构建成执行行变换计算以在执行额外行变换计算之前输出一列数据还是两列数据的决策取决于性能与缓冲存储器要求之间的折衷。应了解,如果将电路 7 构建成以使缓冲器 9 仅需要具有存储一列经局部变换数据的容量的方式执行行变换,则使缓冲器 9 的所需容量最小化的优点将通常以降低性能为代价(例如用于使本发明变换引擎对完整数据值块执行反变换的循环数量增大)。

[0078] 为根据本发明执行传统的基于 H264 的变换(无需进行实际乘法便可执行的整数变换),本发明的变换引擎可不构建有乘法器。可使用移位器而非乘法器来执行基于 H264 的变换。甚至在这些实施方案中,根据本发明绕过变换电路也可使变换计算加速(并降低其所消耗的能量)。

[0079] 接下来参照图 4 所示流程图更详细地说明本发明的一类实施例。图 4 所示流程图涉及一种对一输入数据块序列中的每一块执行的 2D 变换(例如,离散余弦反变换),其中

每一块均包括输入数据值行及列,且所述 2D 变换包括行变换及列变换。根据本发明,所述 2D 变换通过如下方式执行:对每一块的所有行执行行变换以产生经局部变换的数据块,并然后对所述经局部变换的数据块的每一列执行列变换;或者通过如下方式执行:对每一块的所有列执行列变换以产生经局部变换的数据块,并然后对所述经局部变换的数据块的每一行执行行变换。图 4 所示流程图描述对数据值行执行的此种行变换或对数据值列执行的此种列变换(例如,其描述通过实施本发明的图 2 所示电路 7 的实施方案对频率系数行或列执行变换,或者通过实施本发明的图 2 所示电路 11 的实施方案对从缓冲器 9 读取的经局部变换数据值行或列执行变换)。为简明起见,下文对图 4 的说明将假定变换数据值行。应了解,在本说明中所提及的每一处“行”可由提及“列”来取代,以描述本发明的其他实施例。要根据本发明进行反变换的典型块每行具有四个系数(例如,如果所述反变换是基于 H264 的变换)或每行八个系数(例如,如果所述反变换是对 8×8 频率系数块进行的 IDCT)。图 4 适用于这两种情形。

[0080] 在执行图 4 中所示的步骤之前,本发明典型实施例的变换引擎判定所要变换的数据块是否完全由零值组成。在判定所述块完全由零值组成时,所述变换引擎使得绕过对所述块的变换运算(行变换与列变换运算二者)(例如其响应于所述块而输出零,而不对所述块的值执行实际变换运算)。如果所述变换引擎判定所要变换的数据块包括至少一个非零值,则其启动行变换(即其执行图 4 中的步骤 20)。

[0081] 在根据图 4 的流程图启动(步骤 20)所要执行的行变换后,从缓冲存储器读取所要变换的第一行数据值。缓冲存储器可为耦接至图 2 中电路 7 的输入端的缓冲存储器(如果电路 7 执行图 4 的流程图),或者其可为图 2 的缓冲存储器 9(如果电路 11 执行图 4 的流程图)。

[0082] 在步骤 24 中,所述变换引擎判定当前行(在最近一次执行步骤 22 期间所读取的行)是否包含至少一个非零值。若为否,则所述变换引擎输出指示所述行的经行变换版本的零值(步骤 25)(而不对所述行的数据值执行实际的行变换计算),并从缓冲存储器读取下一行(步骤 22 的下次执行)。如果当前行包括至少一个非零值,则所述引擎判定(在步骤 26 中)当前行的第一半(例如由四个数据值组成的行的前两个数据值)是否包括至少一个非零值。

[0083] 如果当前行的第一半不包括至少一个非零值,则所述引擎输出指示所述行的第一半的经行变换版本的零值(步骤 28)(而不对所述行的第一半的数据值执行实际的行变换计算),并判定(在步骤 40 中)当前行的第二半(例如由四个数据值组成的行的后两个数据值)是否包括至少一个非零值。

[0084] 如果当前行的第一半包括至少一个非零值,则所述引擎判定(在步骤 30 中)所述当前行的第一半的第一半是否包括至少一个非零值,并判定(在步骤 32 中)当前行的第一半的第二半是否包括至少一个非零值。

[0085] 如果当前行的第一半的第一半包括至少一个非零值(例如,如果其由单个非零值组成,或者由包括至少一个非零值在内的两个值组成),则所述引擎对所述行的第一半的第一半的数据值执行(在步骤 36 中)行变换计算,并将所得到的经变换数据存储(在步骤 39 中)于缓冲存储器中。用于执行步骤 39 的缓冲存储器可为耦接至图 2 所示电路 11 的输出端的缓冲存储器(如果电路 11 执行图 4 的流程图),或者图 2 的缓冲存储器 9(如果电路 7

执行图 4 的流程图)。

[0086] 如果当前行第一半的第一半不包括非零值,则所述引擎执行步骤 37,其中其:

[0087] (a) 输出一个或多个零来指示当前行第一半的第一半的经行变换版本(而不对所述数据值或当前行第一半的第一半的值执行实际的行变换计算)。然后,将每一此种零值存储(在步骤 39 中)于缓冲存储器中;或

[0088] (b) 输出通过以降低功率的方式对当前行第一半的第一半的数据值执行行变换计算所产生的至少一个数据值。然后,将每一此种数据值存储(在步骤 39 中)于缓冲存储器中。例如,所述引擎可通过如下方式进行此种作业:操作具有第一输入端集合(包括至少一个输入端)及第二输入端集合(包括至少一个输入端)的乘法电路,对所述第一输入端集合确定至少一个常数且对所述第二输入端集合确定当前行第一半的第一半的每一数据值,以对当前行第一半的第一半的每一数据值执行至少乘法运算,而不更新对第一输入端集合与第二输入端集合中至少一者所确定的值,以避免消耗功率 - 而原本对所述乘法电路的相关输入端进行双态转换则会消耗功率。在所述引擎正在执行 IDCT 的情况下,对第一输入端集合确定的每一常数均为余弦常数,且在一种实施方案中,每当对第二输入端集合确定的数据值是零值时,所述引擎均阻止更新(相对于其先前值发生变化)要乘以所述零值的余弦常数,因为无论所述余弦常数的值如何,所述乘法的结果均将为 0。在另一实施方案中,所述引擎阻止更新对第一输入端集合所确定的余弦常数(要乘以零数据值),并阻止更新对第二输入端集合中对应输入端所确定的数据值(以阻止两个乘法器输入端中的任一个双态转换),并确定一零值(例如将零值多路复用至处理管线的输出端)以指示将所述零数据值乘以所述余弦常数的结果(而不实际将这两个操作数相乘在一起);或

[0089] (c) 输出一个或多个零来指示当前行第一半的第一半中至少一个值的经行变换版本(而不对这一或这些数据值执行实际的行变换计算),并输出通过以降低功率方式对当前行第一半的第一半中至少一个其他数据值执行行变换计算而产生的至少一个数据值。然后,将每一此种零值及输出值存储(在步骤 39 中)于缓冲存储器中。

[0090] 如果当前行第一半的第二半包括至少一个非零值(例如,如果其由单个非零值组成或者由包括至少一个非零值在内的两个值组成),则所述引擎对所述行的第一半的第二半的数据值执行(在步骤 34 中)行变换计算,并将所得到的经变换数据存储(在步骤 39 中)于缓冲存储器中。

[0091] 如果当前行的第一半的第二半不包括非零值,则所述引擎执行步骤 35,其中其:

[0092] (a) 输出一个或多个零来指示当前行第一半的第二半的经行变换版本(而不对所述数据值或当前行第一半的第二半的值执行实际的行变换计算)。然后,将每一此种零值存储(在步骤 39 中)于缓冲存储器中;或

[0093] (b) 输出通过以降低功率的方式对当前行第一半的第二半的数据值执行行变换计算所产生的至少一个数据值。然后,将每一此种数据值存储(在步骤 39 中)于缓冲存储器中。例如,所述引擎可通过如下方式进行此种作业:操作具有第一输入端集合(包括至少一个输入端)及第二输入端集合(包括至少一个输入端)的乘法电路,对所述第一输入端集合确定至少一个常数且对所述第二输入端集合确定当前行第一半的第二半的每一数据值,以对当前行第一半的第二半的每一数据值执行至少乘法运算,而不更新对第一输入端集合与第二输入端集合中至少一者所确定的值,以避免消耗功率 - 而原本对所述乘法电路的相

关输入端进行双态转换则会消耗功率。在所述引擎正在执行 IDCT 的情况下,对第一输入端集合确定的每一常数均为余弦常数,且在一种实施方案中,每当对第二输入端集合确定的数据值是零值时,所述引擎均阻止更新(相对于其先前值发生变化)要乘以所述零值的余弦常数,因为无论所述余弦常数的值如何,所述乘法的结果均将为零。在另一实施方案中,所述引擎阻止更新对第一输入端集合所确定的余弦常数(要乘以零数据值),并阻止更新对第二输入端集合中对应输入端所确定的数据值(以阻止两个乘法器输入端中的任一个双态转换),并确定一零值(例如将零值多路复用至处理管线的输出端)以指示将所述零数据值乘以所述余弦常数的结果(而不实际将这两个操作数相乘在一起);或

[0094] (c) 输出一个或多个零来指示当前行第一半的第二半中至少一个值的经行变换版本(而不对这一或这些数据值执行实际的行变换计算),并输出通过以降低功率方式对当前行第一半的第二半中至少一个其他数据值执行行变换计算而产生的至少一个数据值。然后,将每一此种零值及输出值存储(在步骤 39 中)于缓冲存储器中。

[0095] 当在步骤 34 及 36、步骤 34 及 37、步骤 35 及 36、或者步骤 35 及 37 中所产生或确定的所有数据值均已存储(步骤 39)于缓冲存储器中之后,所述引擎判定(在步骤 40 中)当前行的第二半是否包括至少一个非零值。如果当前行的第二半不包括至少一个非零值,则所述引擎输出零(步骤 41)以指示所述行的第二半的经行变换版本(而不对所述行的所述第二半的数据值执行实际的行变换计算)并从缓冲存储器读取所要执行的下一行(另一次执行步骤 22)。

[0096] 如果当前行的第二半包括至少一个非零值,则所述引擎判定(在步骤 42 中)当前行的第二半的第一半是否包括至少一个非零值,并判定(在步骤 44 中)当前行的第二半的第二半是否包括至少一个非零值。

[0097] 如果当前行的第二半的第一半包括至少一个非零值(例如,如果其由单个非零值组成,或者由包括至少一个非零值在内的两个值组成),则所述引擎对所述行的第二半的第一半的数据值执行(在步骤 48 中)行变换计算,并将所得到的经变换数据存储(在步骤 50 中)于缓冲存储器中。用于执行步骤 50 的缓冲存储器可为耦接至图 2 所示电路 11 的输出端的缓冲存储器(如果电路 11 执行图 4 的流程图),或者图 2 的缓冲存储器 9(如果电路 7 执行图 4 的流程图)。

[0098] 如果当前行第二半的第一半不包括非零值,则所述引擎执行步骤 49,其中其:

[0099] (a) 输出一个或多个零来指示当前行第二半的第一半的经行变换版本(而不对所述数据值或当前行第二半的第一半的值执行实际的行变换计算)。然后,将每一此种零值存储(在步骤 50 中)于缓冲存储器中;或

[0100] (b) 输出通过以降低功率的方式对当前行第二半的第一半的数据值执行行变换计算所产生的至少一个数据值。然后,将每一此种数据值存储(在步骤 50 中)于缓冲存储器中。例如,所述引擎可通过如下方式进行此种作业:操作具有第一输入端集合(包括至少一个输入端)及第二输入端集合(包括至少一个输入端)的乘法电路,对所述第一输入端集合确定至少一个常数且对所述第二输入端集合确定当前行第二半的第一半的每一数据值,以对当前行第二半的第一半的每一数据值执行至少乘法运算,而不更新对第一输入端集合与第二输入端集合中至少一者所确定的值,以避免消耗功率-而原本对所述乘法电路的相关输入端进行双态转换则会消耗功率。在所述引擎正在执行 IDCT 的情况下,对第一输入端

集合确定的每一常数均为余弦常数,且在一种实施方案中,每当对第二输入端集合确定的数据值是零值时,所述引擎均阻止更新(相对于其先前值发生变化)要乘以所述零值的余弦常数,因为无论所述余弦常数的值如何,所述乘法的结果均将为0。在另一实施方案中,所述引擎阻止更新对第一输入端集合所确定的余弦常数(要乘以零数据值),并阻止更新对第二输入端集合中对应输入端所确定的数据值(以阻止任一乘法器输入端的双态转换),并确定一零值(例如将零值多路复用至处理管线的输出端)以指示将所述零数据值乘以所述余弦常数的结果(而不实际将这两个操作数相乘在一起);或

[0101] (c) 输出一个或多个零来指示当前行第二半的第一半中至少一个值的经行变换版本(而不对这一或这些数据值执行实际的行变换计算),并输出通过以降低功率方式对当前行第二半的第一半中至少一个其他数据值执行行变换计算而产生的至少一个数据值。然后,将每一此种零值及输出值存储(在步骤50中)于缓冲存储器中。

[0102] 如果当前行第一半的第二半包括至少一个非零值(例如,如果其由单个非零值组成或者由包括至少一个非零值在内的两个值组成),则所述引擎对所述行的第二半的第二半的数据值执行(在步骤46中)行变换计算,并将所得到的经变换数据存储在(在步骤50中)于缓冲存储器中。

[0103] 如果当前行的第二半的第二半不包括非零值,则所述引擎执行步骤47,其中其:

[0104] (a) 输出一个或多个零来指示当前行第二半的第二半的经行变换版本(而不对所述数据值或当前行第二半的第二半的值执行实际的行变换计算)。然后,将每一此种零值存储在(在步骤50中)于缓冲存储器中;或

[0105] (b) 输出通过以降低功率的方式对当前行第二半的第二半的数据值执行行变换计算所产生的至少一个数据值。然后,将每一此种数据值存储在(在步骤50中)于缓冲存储器中。例如,所述引擎可通过如下方式进行此种作业:操作具有第一输入端集合(包括至少一个输入端)及第二输入端集合(包括至少一个输入端)的乘法电路,对所述第一输入端集合确定至少一个常数且对所述第二输入端集合确定当前行第二半的第二半的每一数据值,以对当前行第二半的第二半的每一数据值执行至少乘法运算,而不更新对第一输入端集合与第二输入端集合中至少一者所确定的值,以避免消耗功率-而原本对所述乘法电路的相关输入端进行双态转换则会消耗功率。在所述引擎正在执行IDCT的情况下,对第一输入端集合确定的每一常数均为余弦常数,且在一种实施方案中,每当对第二输入端集合确定的数据值是零值时,所述引擎均阻止更新(相对于其先前值发生变化)要乘以所述零值的余弦常数,因为无论所述余弦常数的值如何,所述乘法的结果均将为零。在另一实施方案中,所述引擎阻止更新对第一输入端集合所确定的余弦常数(要乘以零数据值),并阻止更新对第二输入端集合中对应输入端所确定的数据值(以阻止两个乘法器输入端中的任一个双态转换),并确定一零值(例如将零值多路复用至处理管线的输出端)以指示将所述零数据值乘以所述余弦常数的结果(而不实际将这两个操作数相乘在一起);或

[0106] (c) 输出一个或多个零来指示当前行第二半的第二半中至少一个值的经行变换版本(而不对这一或这些数据值执行实际的行变换计算),并输出通过以降低功率方式对当前行第二半的第二半中至少一个其他数据值执行行变换计算而产生的至少一个数据值。然后,将每一此种零值及输出值存储在(在步骤50中)于缓冲存储器中。

[0107] 当在步骤46及48、步骤46及49、步骤47及48、或者步骤47及49中所产生或确

定的所有数据值均已存储（步骤 50）于缓冲存储器中之后，所述引擎判定（在步骤 52 中）是否已对当前块中的所有行执行行变换。如果已对当前块中的所有行执行行变换，则所述引擎进入一种使其停止行变换运算的状态（步骤 54）。然后，其可再次执行步骤 20，以开始处理所要执行的下一数据值块。如果尚未对当前块中的所有行执行行变换，则所述引擎从缓冲存储器中读取（所要变换的当前块）的下一行（再一次执行步骤 22）。

[0108] 图 5 是在实施本发明的图 2 所示电路 7（或电路 11）的典型实施方案中所用变换电路的方块图。设计用于变换 4×4 数据块的电路 7（或 11）的实施方案将通常包括一个单元的图 5 所示电路，以用于变换所要变换的每一行或列的四个数据值 I_0 、 I_1 、 I_2 及 I_3 。设计用于变换 8×8 数据块的电路 7（或 11）的实施方案将通常包括两个单元的图 5 所示电路：一个用于变换所要变换的每一行或列的第一半的四个数据值 I_0 、 I_1 、 I_2 及 I_3 ；及完全相同的一个，以用于变换所要变换的每一行或列的第二半的四个数据值（ I_4 、 I_5 、 I_6 及 I_7 ）（使用第二组余弦常数 C_4 、 C_5 、 C_6 及 C_7 取代图 5 中所示的常数 C_0 、 C_1 、 C_2 及 C_3 ）。

[0109] 图 5 所示的每一乘法单元 M_1 、 M_2 、 M_3 及 M_4 均包括一第一输入端（对其确定余弦常数 C_0 、 C_1 、 C_2 及 C_3 中的一个）、一第二输入端（对其确定数据值 I_0 、 I_1 、 I_2 及 I_3 中的一个）、以及一输出端，其在所述输出端上确定一标识这两个操作数的乘积的数据值。图 5 所示电路还包括如图所示连接的预处理及控制逻辑 100（将称作预处理逻辑 100）、加法电路 A_1 、 A_2 、 A_3 、 A_4 及 A_5 、减法电路 S_1 、及延迟电路 A 及 B （构建为正反器），且经配置以对其确定的每一组四个输入值执行管线式行变换或列变换，以响应于每一输入值而产生一输出值 Y_0 （或 Y_1 ）。在运行期间，延迟电路 A 通常使将加法电路 A_3 的输出确定至加法电路 A_5 、减法电路 S_1 、及加法电路 A_3 中每一被选定者（由逻辑电路 100 所确定的控制信号来确定）延迟一个管线式变换循环。在运行期间，延迟电路 B 通常使将加法电路 A_4 的输出确定至加法电路 A_5 、减法电路 S_1 、及加法电路 A_4 中每一被选定者（由逻辑电路 100 所确定的控制信号来确定）延迟一个管线式变换循环。

[0110] 当在经设计以变换 4×4 数据块的电路 7（或 11）的实施方案中包含图 5 所示电路时，处理逻辑 100 经配置以执行图 4 所示的零（或非零）值判定运算。当在经设计以变换 8×8 数据块的电路 7（或 11）的实施方案中包含两个图 5 所示单元时，每一单元 100 均经配置以对所要变换的每一数据行或列的一不同子集（例如一半）执行图 4 所示的零（或非零）值判定运算。在电路 7（或 11）的任一实施方案中，预处理逻辑 100 均经配置以从相关缓冲存储器（例如图 2 所示缓冲器 9 或者耦接至图 2 所示电路 7 的输入端的缓冲存储器）中读取所要变换的输入数据值，并产生（并确定至图 5 中其他元件中的适当元件）为执行图 4 所要求的其他作业所需的控制信号。例如，如果逻辑 100 判定系数 1_0 是零值（例如在图 4 的步骤 30 中），则其可确定一控制信号，以使在前一操作循环期间确定至乘法器 M_1 的系数 C_0 在其中乘法器 M_1 要将零值系数 1_0 乘以常数的每一后续循环期间继续确定至乘法器 M_1 。

[0111] 在典型实施方案中，预处理逻辑 100 在接收（例如从缓冲器 9）到所要变换的每一组四个新输入数据值时均将输出值 Y_0 及 Y_1 复位至零。在此种情形中，当预处理逻辑 100 响应于其中一个输入值而判定应输出零值（即 Y_0 或 Y_1 的零值）时（例如在图 4 的步骤 25、28、41、35、37、47、或 49 中），其可在其中电路元件 M_1 、 M_2 、 M_3 、 M_4 、 A_1 、 A_2 、 A_3 、 A_4 、 A 、 B 、 A_5 、及 S_1 原本将运行来变换该输入值的循环期间阻止这些元件的输入端的所有双态转换，并转而

使 Y0 (或 Y1) 的复位后的零值确定至相关缓冲器 (例如确定至缓冲存储器 9 或耦接至电路 11 的输出端)。

[0112] 在需要对所要变换的数据值重新排序的实施例, 预处理单元 100 较佳经配置以执行此种重新排序。例如, 为执行类似于参照图 4 所述的变换、但其中每一判定半行或半列 (或半行或半列的一半) 的第一 (或第二) 半中的数据值是否包括至少一个非零值的步骤被取代为判定半行或半列 (或半行或半列的一半) 中的偶数 (或奇数) 数据值是否包括至少一个非零值的步骤, 预处理单元 100 可将所要评估的数据值重新排序, 以便执行所要求的判定。例如, 单元 100 可判定经重新排序的半行或经重新排序的半列 (或经重新排序的半行或经重新排序的半列的一半) 的第一 (或第二) 半中的数据值是否包括至少一个非零值, 从而判定所述半行或半列的未重新排序的版本 (或半行或半列的未重新排序的版本的一半) 中的偶数 (或奇数) 数据值是否包括至少一个非零值。预期预处理单元 100 的某些实施方案可对行或列 (或不完整的行或不完整的列) 的并非为半行 (或不完整半行) 或半列 (或不完整半列) 的子集执行重新排序。

[0113] 图 6 是本发明管线式视频解码系统的一实施例各元件的方块图。涵盖图 6 所示设计的用于本发明其他实施例中的实施方案的各种变化形式 (例如可省略零检测电路 132、134 及 135-138, 且其功能可由零检测电路 130 或其他零检测电路执行)。图 6 中的系统包括如图所示相连接的预处理单元 120、变换引擎 122 及后处理单元 124。

[0114] 单元 120 耦接成接收一系列输入数据值块 (例如已通过对视频像素块执行二维 DCT 或反 DCT 或其他 2D 变换而产生的输入频域系数)。单元 120 还耦接成在图 6 所示系统中经过第一遍之后接收从后处理单元 124 输出的数据值 (在图 6 中标识为“来自第一级的输出”及“第一级输出”), 以使这些数据值 (其视需要在从单元 124 输出之后及确定至单元 120 之前在缓冲存储器中进行缓冲) 可在图 6 所示电路中经过第二遍期间经历第二级处理 (例如, 以在各数据值列在图 6 所示系统中经过第一遍期间已经历列变换时对数据值行执行行变换)。

[0115] 单元 120 的多路复用器 129 选择输入数据值 (尚未经过图 6 中的处理) 或者从后处理单元 124 输出的数据值 (已在前一遍经过图 6 所示系统时经受处理), 并将所选值确定至零检测逻辑 130。使所选值经过零检测逻辑 130 (视需要, 当逻辑 130 判定所选值完全由零值组成时除外)。然后, 视需要, 单元 120 内的其他电路对直通过逻辑 130 的值执行初步处理作业, 以产生经预处理的数据值 (这些作业是在需要时执行以实施将要使用图 6 所示系统实施的特定变换)。单元 120 将经预处理的数据值 (如果不对逻辑 130 的输出执行初步处理, 则其与逻辑 130 的输出完全相同) 确定至变换引擎 122。

[0116] 预处理单元 120 通常经配置以将其确定的每一数据值集合分成适于由引擎 122 执行所要实施的变换的子集。单元 120 分离成子集的进行方式将依赖于所要实施的变换 (例如 8×8 DCT, 8×8 IDCT, 或者 8×8 哈达玛), 并可取决于从外部单元确定至单元 120 的控制信号, 以便可根据所要实施的变换来定制子集选择。

[0117] 变换引擎 122 对其从单元 120 接收的每一数据值块执行 2D 变换, 以产生经变换的数据值块 (例如局部经解码的视频像素块), 并将已变换的数据值确定至后处理单元 124 以供视需要进一步处理。更具体而言, 变换引擎 122 经配置以对从单元 120 对其确定的每一组四个数据值执行管线式行变换或列变换, 以因应这四个数据值中的每一者而产生输出

值。此种每一组四个数据值均通常为—数据值块中的一行（或列），或者—数据值块中一行（或列）的子集。

[0118] 后处理单元 124 视需要（即当适合于图 6 所示系统正执行的特定变换时）对从引擎 122 接收的已变换值执行后处理作业，以产生输出数据值。所述输出数据值可从单元 124 确定至系统总线及 / 或反馈至单元 120 的至少一个输入端（或反馈至其被确定至单元 120 的至少一个输入端时所带来的缓冲存储器）。通常，单元 124 执行舍入、饱和、及最终填充（如果需要）。单元 124 所执行的具体后处理作业依赖于图 6 所示电路是正在对单元 122 的输入执行第一级还是第二级处理。

[0119] 图 6 所示变换引擎 122 经设计以变换 4×4 数据块，包括通过使用一组常数 C_0 、 C_1 、 C_2 及 C_3 来变换所要变换的每一行或列的四个数据值（ I_0 、 I_1 、 I_2 、及 I_3 ）。在本发明的某些其他实施例中，用于变换 8×8 数据块的电路将包括两个变换引擎 122：一个引擎用于变换所要变换的每一行或列的第一子集（例如第一半）的四个数据值 I_0 、 I_1 、 I_2 及 I_3 ；及完全相同的一引擎，用于变换所要变换的每一行或列的第二子集（例如第二半）的四个数据值（ I_4 、 I_5 、 I_6 及 I_7 ）（使用第二组常数 C_4 、 C_5 、 C_6 及 C_7 取代图 6 中块 122 所示的常数 C_0 、 C_1 、 C_2 及 C_3 ）。所要使用的常数（例如 C_0 、 C_1 、 C_2 及 C_3 ）可由从外部单元确定至引擎 122 的控制信号决定，以便可根据所要执行的变换来定制常数选项。通常，预处理单元 120 的零检测逻辑 130 经配置以判定所要变换的整个数据块是否完全由零值组成（所要变换的 4×4 数据块的所有四个行或列完全由零值组成）。例如，其可包括移位寄存器，以提供足够的延迟在将块中的第一行（列）传送至下游电路之前对所述块的所有行（列）执行此种判定。在判断出所述块完全由零值组成后，逻辑 130 将适当的控制位确定至图 6 所示系统的其他元件，所述系统元件使得绕过（例如如上文所述）原本将由引擎 122 随后对所述块执行的变换运算，并通常还使得响应于所述块而从单元 124 的多路复用器 139 输出预定值（例如零），而不由引擎 122 与单元 124 中的一者或两者对所述块的值执行实际的变换运算。

[0120] 无论预处理单元 120 的零检测逻辑 130 是否如在前一段中所述配置成判定整个数据块是否完全由零值组成，逻辑 130 均配置成对从多路复用器 129 确定至逻辑 130 的每一单独行（或列）的数据值执行零检测，以判定是否此一行（或列）的所有数据值均为零值（即其执行图 4 中的步骤 24）。当每一此种行（或列）均由四个数据值组成时，逻辑 130 判定所述行（列）是否由四个零位组成。如果逻辑 130 判定一行（列）完全由零位组成时，其确定适当的控制位，所述适当的控制位使得绕过原本随后将由引擎 122 对所述行（或列）执行的变换运算，并通常还响应于所述行（列）而使得从单元 124 的多路复用器 139 输出预定值（例如零），而不由引擎 122 与单元 124 中的一者或两者对所述行（列）的值执行实际变换运算。

[0121] 在某些实施方案中，逻辑 130 判定是否从多路复用器 129 对其确定的每一单独数据值行（或列）的第一数据值是非零值、而所述行（或列）的所有其他值均为零值。响应于识别出一行（列）除初始非零值之外由零值组成，逻辑 130 的此一实施方案确定适当的控制位，所述适当的控制位使得绕过原本随后将由引擎 122 对所述行（或列）执行的变换运算，并通常还响应于所述行（列）而使得从单元 124 的多路复用器 139 输出预定值（例如零），而不由引擎 122 与单元 124 中的一者或两者对所述行（列）的值执行实际变换运算。

[0122] 引擎 122 的零检测逻辑 132 经配置以对从单元 120 对引擎 122 确定的每一行（或列）的第一子集（例如第一半，其为前两个数据值）执行零检测，以判定是否此第一子集的所有数据值均为零值（例如，其执行图 4 中的步骤 26）。如果逻辑 132 判定第一子集（例如半行或半列）完全由零位组成，则其确定适当的控制位，所述适当的控制位使得绕过原本随后将由引擎 122 对第一子集执行的变换运算，并通常还响应于所述第一子集而使得从引擎 122 的多路复用器 140 输出预定值（例如零），而不由引擎 122 对所述第一子集的值执行实际变换运算。如果逻辑 132 判定出当前行（列）的第一子集（例如第一半）包括至少一个非零值，则零检测逻辑单元 135 及 136 对第一子集的第一及第二值执行零检测。

[0123] 如果逻辑 135 判定当前行（列）的第一子集的第一数据值是零值，则其对乘法电路 141（变换常数 C0 确定至其第一输入端，且当前行或列的零数据值确定至其第二输入端）确定一控制位，以使电路 141 对确定至其第二输入端的值执行乘法运算，而不更新先前确定至其第一输入端的值，以避免消耗功率 - 而原本为将第一输入端双态转换至常数 C0 的已更新的值则会消耗功率。

[0124] 类似地，如果逻辑 136 判定当前行（列）的第一子集的第二数据值是零值，则其对乘法电路 142（变换常数 C1 确定至其第一输入端，且当前行或列的该零数据值确定至其第二输入端）确定一控制位，以使电路 142 对确定至其第二输入端的值执行乘法运算，而不更新先前确定至其第一输入端的值，以避免消耗功率 - 而原本为将第一输入端双态转换至常数 C1 的已更新的值则会消耗功率。

[0125] 引擎 122 的零检测逻辑 134 经配置以对从单元 120 对引擎 122 确定的每一行（或列）的第二子集（例如第二半，其为第二两个数据值）执行零检测，以判定是否此第二子集的所有数据值均为零值（例如，其执行图 4 中的步骤 40）。如果逻辑 134 判定第二子集完全由零位组成，则其确定适当的控制位，所述适当的控制位使得绕过原本随后将由引擎 122 对第二子集（例如半行或半列）执行的变换运算，并通常还响应于所述第二子集而使得从引擎 122 的多路复用器 141 输出预定值（例如零），而不由引擎 122 对所述第二子集的值执行实际变换运算。如果逻辑 134 判定出当前行（列）的第二子集包括至少一个非零值，则零检测逻辑单元 137 及 138 对第二子集的第一及第二值执行零检测。

[0126] 如果逻辑 137 判定当前行（列）的第二子集的第一数据值是零值，则其对乘法电路 143（变换常数 C2 确定至其第一输入端，且当前行或列的零数据值确定至其第二输入端）确定一控制位，以使电路 143 对确定至其第二输入端的值执行乘法运算，而不更新先前确定至其第一输入端的值，以避免消耗功率 - 而原本为将第一输入端双态转换至常数 C2 的已更新的值则会消耗功率。

[0127] 类似地，如果逻辑 138 判定当前行（列）的第二子集的第二数据值是零值，则其对乘法电路 144（变换常数 C3 确定至其第一输入端，且当前行或列的该零数据值确定至其第二输入端）确定一控制位，以使电路 144 对确定至其第二输入端的值执行乘法运算，而不更新先前确定至其第一输入端的值，以避免消耗功率 - 而原本为将第一输入端双态转换至常数 C3 的已更新的值则会消耗功率。

[0128] 在运行期间控制图 6 中的每一多路复用器 129、140、141、150、151、152 及 153，以在图 6 所示系统的当前管线式作业循环期间使被确定至其输入端上的其中一个值直通至其输出端。图 6 所示的每一电路 156 及 157 均为加法电路，其在运行期受到控制，以在图 6 所

示系统的当前作业循环期间在其输出端上确定被确定至其输入端的值的和。图 6 所示电路 154 及 155 是延迟电路（构建为正反器），其具有适合由图 6 所示系统执行的具体变换的延迟。

[0129] 本发明的某些替代实施例由与图 6 相同的电路组成或包括所述电路，只是省略了图 6 中的逻辑电路 132、134、135、136、137 及 138，且所省略电路的功能由图 6 中逻辑电路 130 的修改型式来执行。具体而言，逻辑 130 的此种修改型式通常将经配置以判定所要变换的整个数据块是否完全由零值组成（即所要变换的 4×4 数据块的所有四个行或列是否完全由零值组成）、所要变换的块的每一行（或列）数据的每一半是否完全由零值组成、及所要变换的块的每一半行（或列）数据的每一数据值是否为零值，并响应于这些判定而（对系统的其他元件）确定由图 6 所示逻辑电路 130、132、134、135、136、137、及 138 所确定的上述类型的控制信号。

[0130] 图 6 所示系统可为便携式媒体播放器或另一种便携式数据处理系统（或并非打算为便携式的数据处理系统，例如具有视频解码能力的系统）的子系统。此种媒体播放器或其他处理系统可以且通常将包括图 6 中所未显示的其他元件。

[0131] 应了解，为将本发明的各种实施例构建成对具有诸多不同格式中任何格式的数据块执行诸多不同 2D 变换中的任何变换，将构建参照图 4 所示及所述的具体步骤以及参照图 5 及 6 所示及所述的具体结构的变化形式。

[0132] 还应了解，尽管在本文中是例示及说明本发明的某些实施例，然而本发明是由申请专利范围界定且不受限于本文所述及所示的具体实施例。

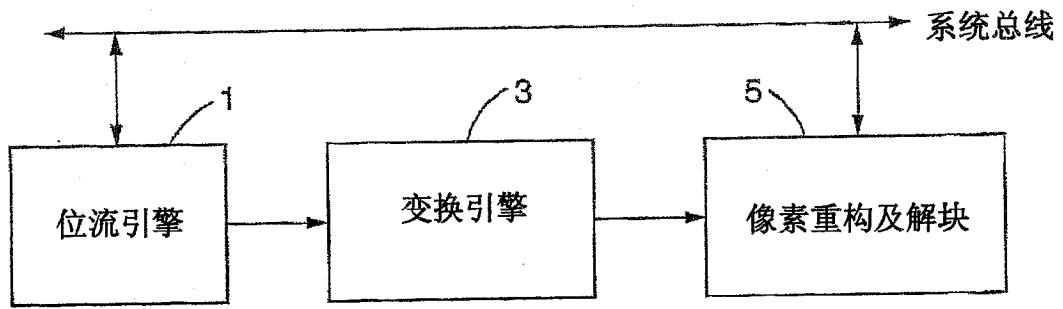


图 1

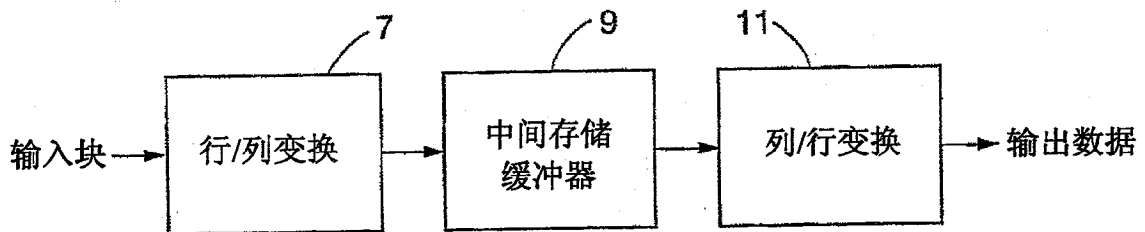


图 2

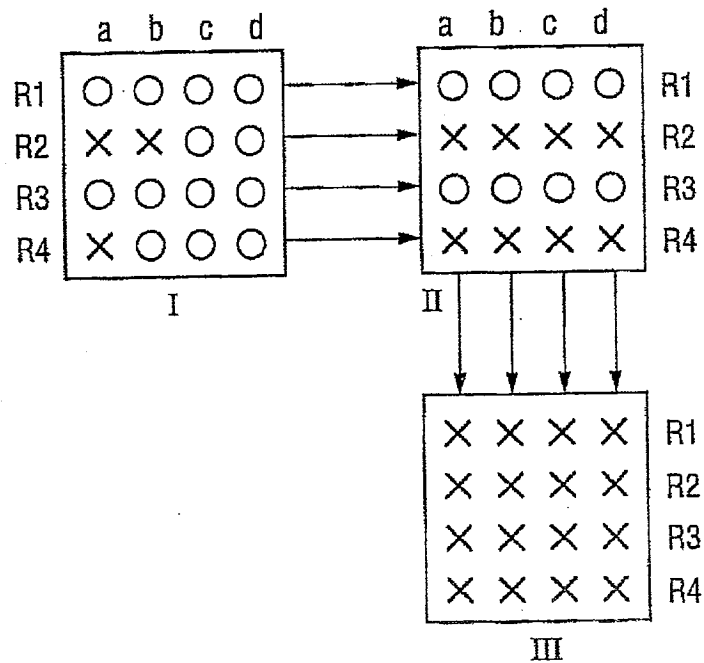


图 3

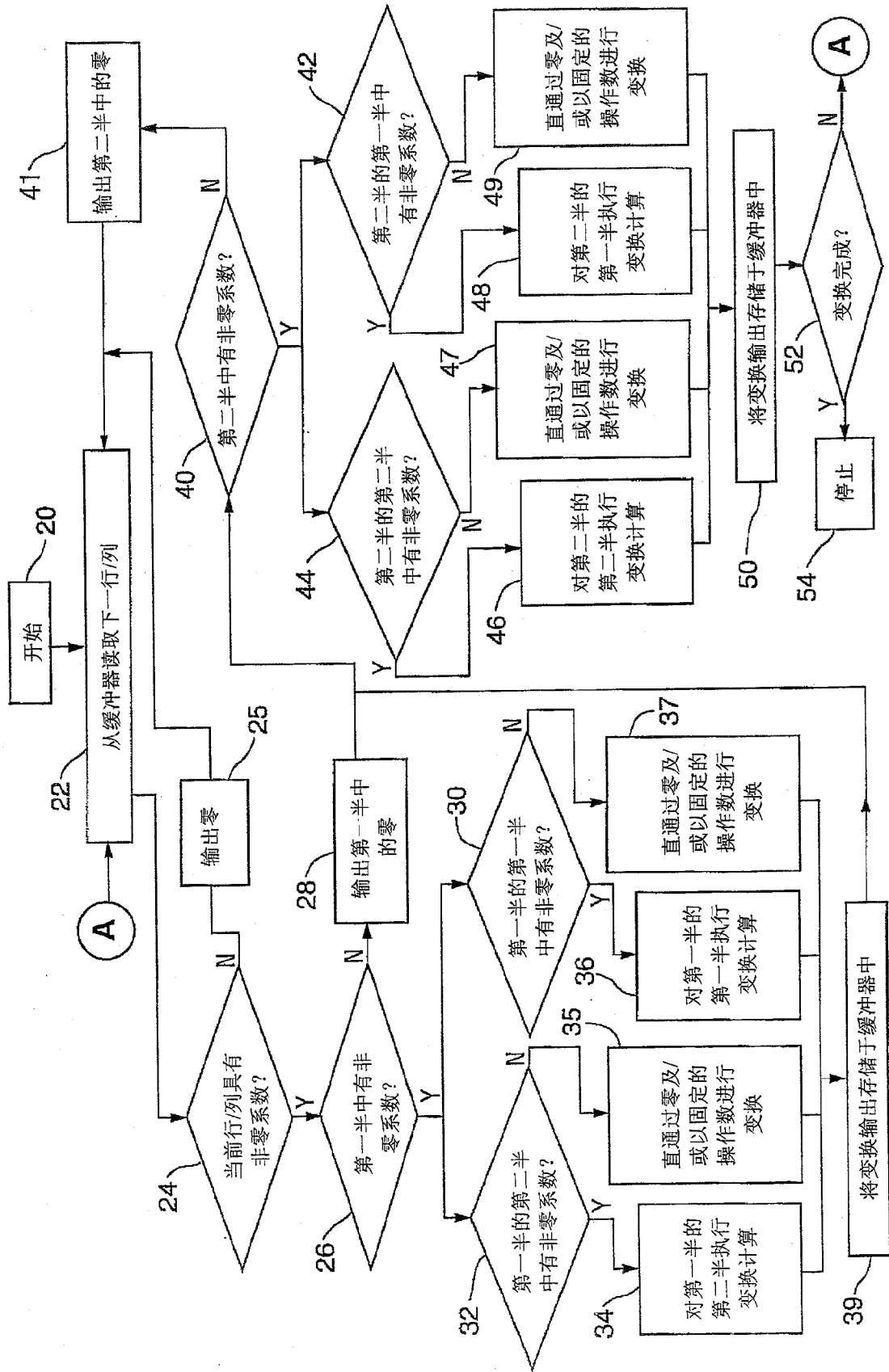


图 4

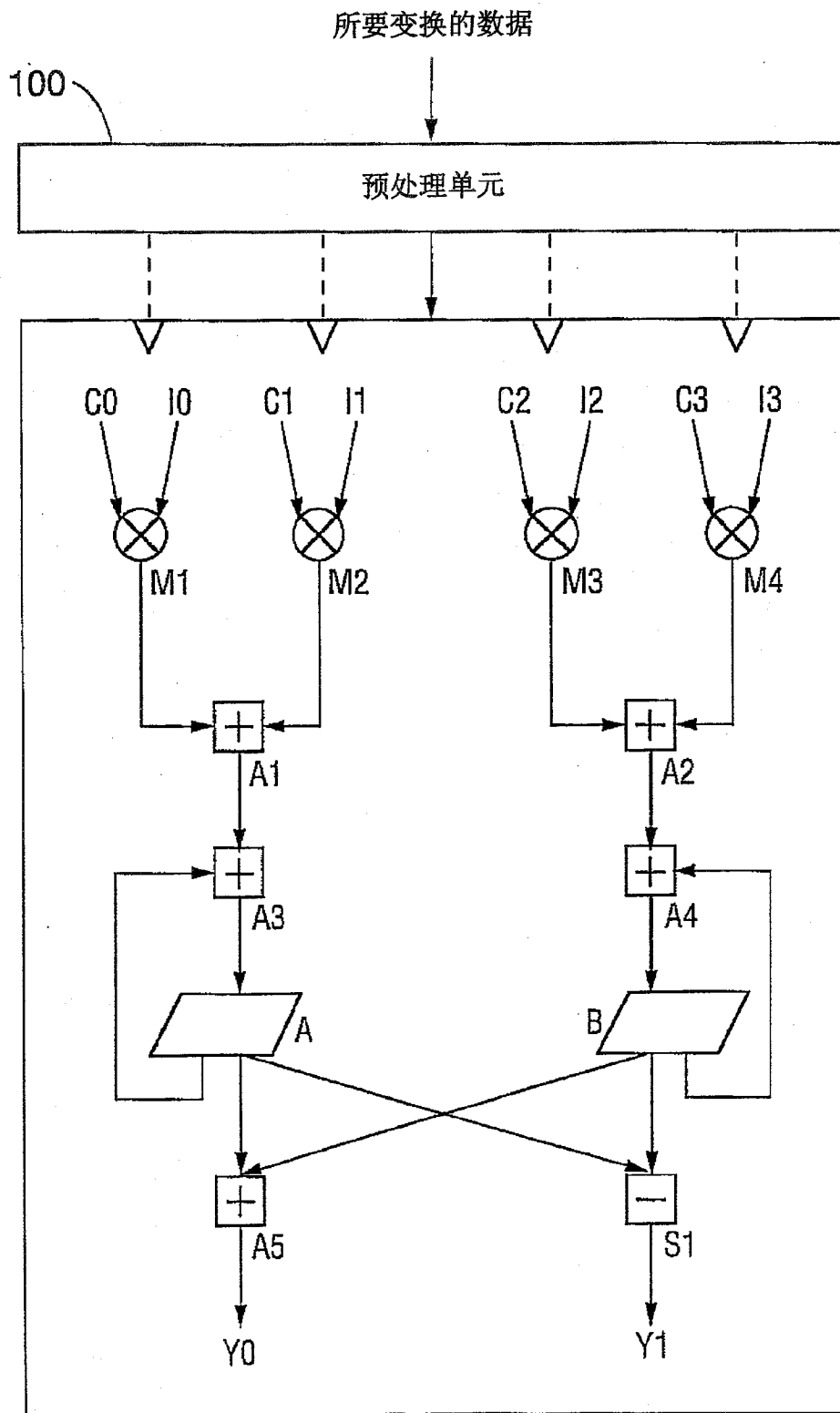


图 5

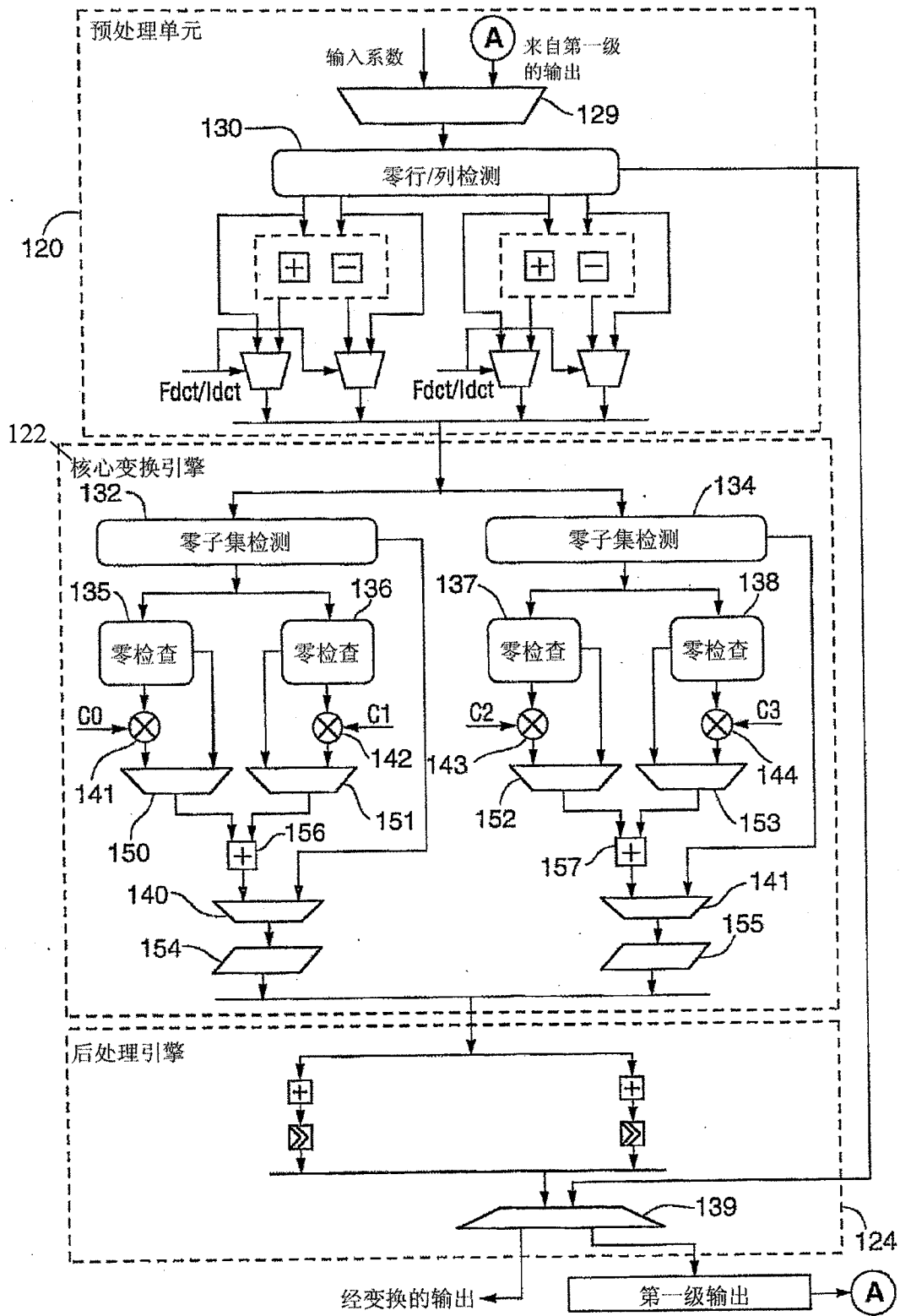


图 6