

基于 FPGA 和 2 位串行分布式算法的实时高速二维 DCT/IDCT 处理器研制¹

向 晖 滕建辅 王承宁

(天津大学电子信息工程学院 天津 300072)

摘 要 本文在 W.Li(1991) 循环斜卷积算法和分布式算法的基础上, 通过软件模拟和具体硬件设计, 利用 FPGA 完成了可用于高清晰度电视核心解码器及其它信号与信息处理系统的 8×8 二维 DCT/IDCT 处理器的全部电路设计工作。它采用一根信号线控制计算 DCT/IDCT, 其输入、输出为 12 位, 内部数据线及内部参数均为 16 位。

关键词 现场可编程门阵列, 二维 DCT/IDCT 处理器, 分布式算法

中图分类号 TN941, TN710

1 引 言

二维 DCT/IDCT 作为目前最广泛采用的变换编码方式, 已被 CCITT 标准、JPEG 标准、MPEG 标准等多种国际图像和视频压缩标准所采用, 并在高清晰度电视、DVD、信息高速公路中得到了广泛应用^[1]。本文研究了二维 DCT/IDCT 处理器硬件电路设计问题。

2 8×8 二维 DCT/IDCT 的快速算法

从 1974 年引入 DCT/IDCT 开始, 为提高 DCT/IDCT 的计算速度, 减少硬件实现的复杂性, 人们提出四类算法: 间接算法、矩阵分解算法、脉动算法、递归算法。本文采用的是 W.Li 的循环斜卷积算法^[2], 该算法是矩阵分解算法的一种。根据 W.Li 循环斜卷积算法, 可以推导出二维 8×8 DCT/IDCT 的数学表达式如下:

2.1 一维 8 阶 DCT 的快速算法

2.1.1 奇项 DCT 的计算

$$\begin{bmatrix} X(1) \\ X(3) \\ -X(7) \\ X(5) \end{bmatrix} = \begin{bmatrix} y(0) & y(1) & y(2) & y(3) \\ -y(3) & y(0) & y(1) & y(2) \\ -y(2) & -y(3) & y(0) & y(1) \\ -y(1) & -y(2) & -y(3) & y(0) \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}, \quad (1)$$

其中

$$\begin{bmatrix} y(3) \\ y(2) \\ y(1) \\ y(0) \end{bmatrix} = \begin{bmatrix} x(2) - x(5) \\ x(4) - x(3) \\ x(1) - x(6) \\ x(0) - x(7) \end{bmatrix}, \quad \text{参数} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} (1/2) \cos(\pi/16) \\ (1/2) \cos(3\pi/16) \\ (1/2) \cos(9\pi/16) \\ (1/2) \cos(27\pi/16) \end{bmatrix}.$$

¹ 1998-01-20 收到, 1998-11-16 定稿

2.1.2 偶项 DCT 的计算

$$\begin{bmatrix} X(2) \\ X(6) \end{bmatrix} = \begin{bmatrix} y_1(0) & y_1(1) \\ -y_1(1) & y_1(0) \end{bmatrix} \begin{bmatrix} C_4 \\ C_5 \end{bmatrix}, \quad \begin{bmatrix} X(4) \\ X(0) \end{bmatrix} = \begin{bmatrix} y_2 \\ y_2' \end{bmatrix} [C_6], \quad (2)$$

其中

$$\begin{bmatrix} y_1(0) \\ y_1(1) \end{bmatrix} = \begin{bmatrix} [x(0) + x(7)] - [x(3) + x(4)] \\ [x(1) + x(6)] - [x(2) + x(5)] \end{bmatrix},$$

$$\begin{bmatrix} y_2 \\ y_2' \end{bmatrix} = \begin{bmatrix} [x(0) + x(7) + x(3) + x(4)] - [x(1) + x(6) + x(2) + x(5)] \\ [x(0) + x(7) + x(3) + x(4)] + [x(1) + x(6) + x(2) + x(5)] \end{bmatrix},$$

所包含参数为

$$\begin{bmatrix} C_4 \\ C_5 \\ C_6 \end{bmatrix} = \begin{bmatrix} (1/2) \cos(\pi/8) \\ (1/2) \cos(3\pi/8) \\ (1/2) \cos(\pi/4) \end{bmatrix}.$$

2.2 一维 8 阶 IDCT 的计算

2.2.1 利用奇项 DCT

$$\begin{bmatrix} S_2(0) \\ S_2(1) \\ -S_2(3) \\ S_2(2) \end{bmatrix} = \begin{bmatrix} y(0) & y(1) & y(2) & y(3) \\ -y(3) & y(0) & y(1) & y(2) \\ -y(2) & -y(3) & y(0) & y(1) \\ -y(1) & -y(2) & -y(3) & y(0) \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}, \quad (3)$$

其中 $\begin{bmatrix} y(3) \\ y(2) \\ y(1) \\ y(0) \end{bmatrix} = \begin{bmatrix} x(5) \\ -x(7) \\ x(3) \\ x(1) \end{bmatrix}$, 参数 $C_0 - C_3$ 同 DCT 的计算.

2.2.2 利用偶项 DCT

$$\begin{bmatrix} S_{2b}(0) \\ S_{2b}(1) \end{bmatrix} = \begin{bmatrix} X(2) & X(6) \\ -X(6) & X(2) \end{bmatrix} \begin{bmatrix} C_4 \\ C_5 \end{bmatrix}, \quad \begin{bmatrix} S_{2bb} \\ S_{1bb} \end{bmatrix} = \begin{bmatrix} X(4) \\ X(0) \end{bmatrix} [C_6], \quad (4)$$

其中参数 C_4, C_5, C_6 同 DCT.

2.2.3 后处理得到各项 IDCT

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} S_1(0) + S_2(0) \\ S_1(1) + S_2(1) \\ S_1(2) + S_2(2) \\ S_1(3) + S_2(3) \\ S_1(3) - S_2(3) \\ S_1(2) - S_2(2) \\ S_1(1) - S_2(1) \\ S_1(0) - S_2(0) \end{bmatrix}, \quad (5)$$

其中

$$\begin{bmatrix} S_1(0) \\ S_1(1) \\ S_1(2) \\ S_1(3) \end{bmatrix} = \begin{bmatrix} S_{2b}(0) + (S_{2bb} + S_{1bb}) \\ S_{2b}(1) + (S_{1bb} - S_{2bb}) \\ (S_{1bb} - S_{2bb}) - S_{2b}(1) \\ (S_{2bb} + S_{1bb}) - S_{2b}(0) \end{bmatrix}.$$

之后, 我们用软件对 W.Li 循环斜卷积算法进行了实现, 软件模拟显示, 处理前后数据差值在 ± 1 之间, 处理前后图像信噪比为 40.25dB, 从而说明用这种算法设计 DCT/IDCT 实时硬件处理图像时, 其误差是在可接受范围以内的。

3 8×8 二维 DCT/IDCT 处理器硬件电路设计

采用矩阵分解算法, 即以一维 8 项行变换 DCT/IDCT 加一维 8 项列变换 DCT/IDCT(行列次序可以任意) 实现二维 DCT/IDCT 的硬件框架如图 1 所示。

利用 W.Li 循环斜卷积算法设计的二维 DCT/IDCT 处理器的硬件电路结构如图 2 所示。

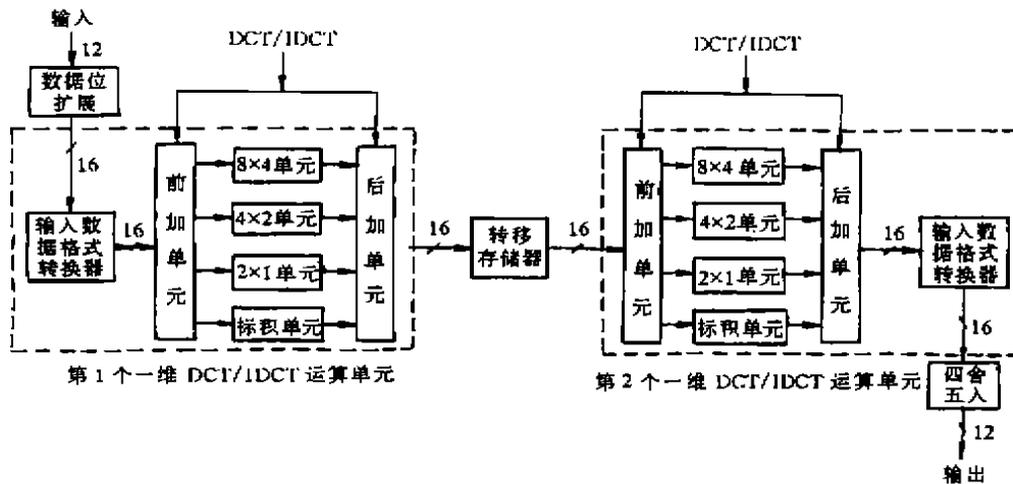
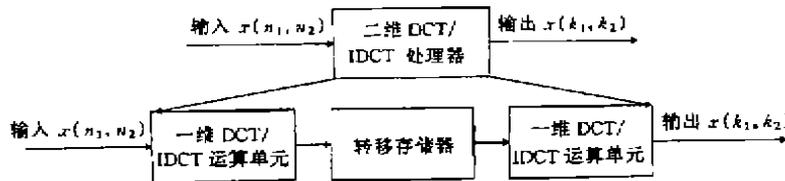


图 2 二维 DCT/IDCT 处理器电路结构框图

在二维 DCT/IDCT 硬件电路设计中, 采用了基于分布式算法的字并 / 位串设计方法, 即每行或每列的 8 个字并行输入, 每个字每一时钟周期只输入 2 位, 8 个时钟周期串行输入全部 16 位。同时, 为提高处理器的系统时钟频率, 在整个电路设计中广泛采用了流水线设计方式。按上述算法, DCT 和 IDCT 变换可用同样的硬件结构, 只需一根信号线就可决定是计算 DCT 还是计算 IDCT, 其不同之处仅在于一维 DCT 运算须经前加单元, 而一维 IDCT 运算则须经后加单元; 以及 8×4 、 4×2 、 2×1 单元的输入不同。

4 各硬件单元电路设计

4.1 输入、输出数据格式转换器

输入、输出数据格式转换器主要完成数据格式的转换功能。其中输入数据格式转换器是将字串 / 位并的输入格式转化为字并 / 位串的数据传输格式；而输出数据格式转换器则是将字并 / 位串的数据传输格式重新转换成字串 / 位并的数据形式输出。输入、输出数据格式转换器的电路结构框图如图 3 所示。

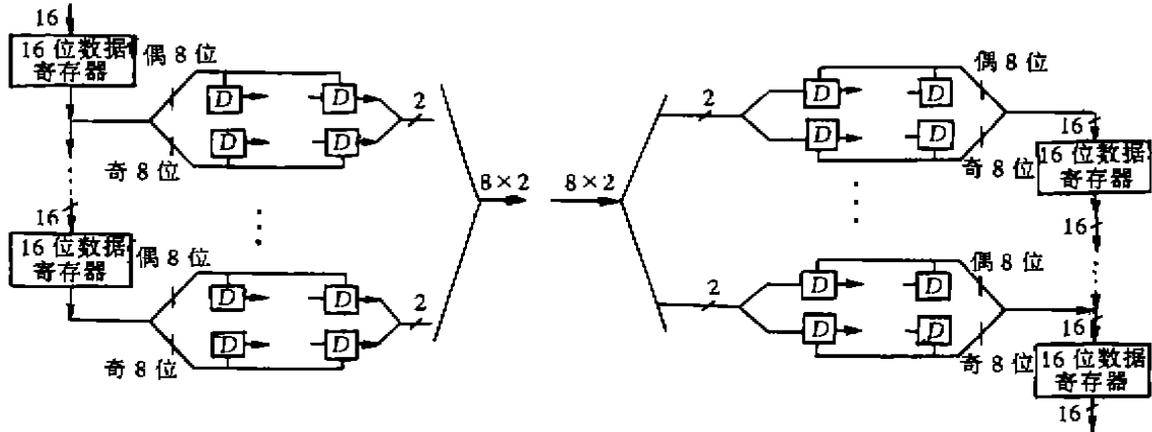


图 3 输入、输出数据格式转换器

4.2 中间处理单元

中间处理单元包括 8×4 、 4×2 、 2×1 标积单元，其结构如图 4 所示。主要功能是完成 W.Li 循环斜卷积算法中所包含的大量 16 位 16 位的乘法运算及相应的累加运算。在该部分电路中，循环斜卷积的实现采用了目前国际上比较流行的基于 ROM 查表法的分布式算法来完成，从而大大降低了电路的复杂程度，节省了硬件资源，提高了电路的可靠性。中间处理单元是整个一维 IDCT 运算单元中最复杂也是设计最困难的一个部分。

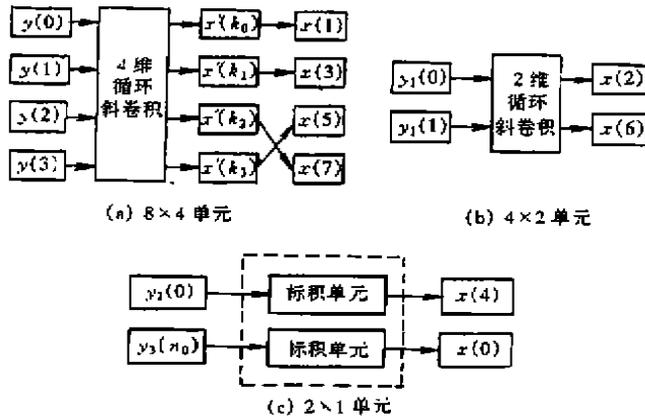


图 4

我们以 8×4 单元为例说明中间处理单元的电路设计。 8×4 单元是一个 4 维循环斜卷积器。 8×4 单元主要完成 (1) 式和 (3) 式所示矩阵的求解运算。每个时钟周期， 8×4 单元从后

级接收 4 个 2 位数据, 并输出 4 个 2 位数据。每个 8×4 单元包含 4 个电路结构完全相同的电路模块。其具体电路结构如图 5 所示。

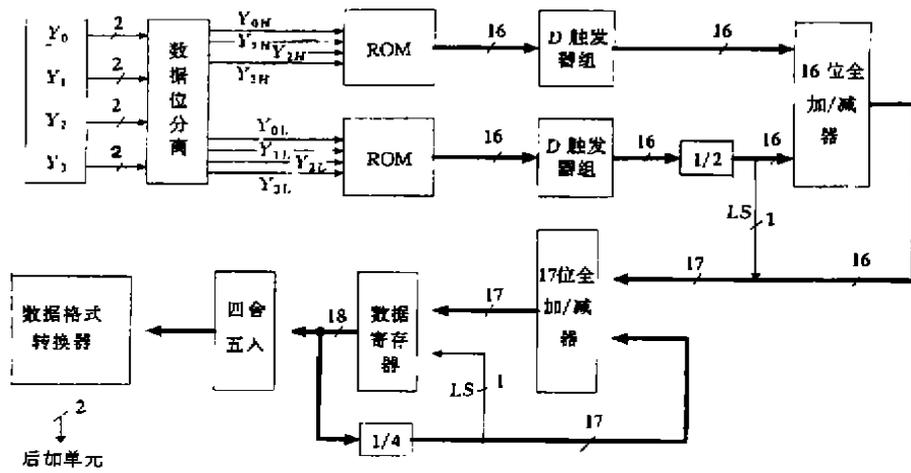


图 5 8×4 单元中一个电路模块的电路结构框图

字并 / 位串数据 $Y_0 \sim Y_3$ 经数据位分离后成为高四位, 低四位分别作为两个 16×16 ROM 的地址输入。两个 16×16 ROM 的输出则经过两次移位累加及四舍五入处理之后, 被送往数据格式转换器重新还原为位串行格式输出。 8×4 单元中的 4 个电路模块中的任何一个均包含两个具有相同存储内容的 ROM, 但不同电路模块的 ROM 存储内容是不同的。 8 位地址线 $Y_{0H} \sim Y_{3H}$, $Y_{0L} \sim Y_{3L}$ 是 4 个电路模块的公共地址线。 ROM 中存储的是按分布式算法预先计算好的可能出现的各个部分积。 我们可以利用不同的地址, 选择出相应的数据。 由图 5, 我们可以看出 8×4 单元每一电路模块的一个基本操作就是不断从 ROM 中读取数据并与上一周期计算所得结果进行累加。 在实际硬件设计中, 这种累加运算是通过采用 FPGA 器件库中的带快速进位逻辑的全加器完成。 带快速进位逻辑的全加器工作速度可以达到很高, 且延时固定, 便于电路中流水线的设计。

4.3 前、后加单元

前加单元和后加单元分别是一维 DCT/IDCT 运算单元的数据前和数据后处理单元。 由于采用 2 位串行运算方式, 所以前、后加单元全部由 2 位带进位或借位存储功能的串行全加 / 减器组成。 前、后加单元的总结构框图如图 6 所示。

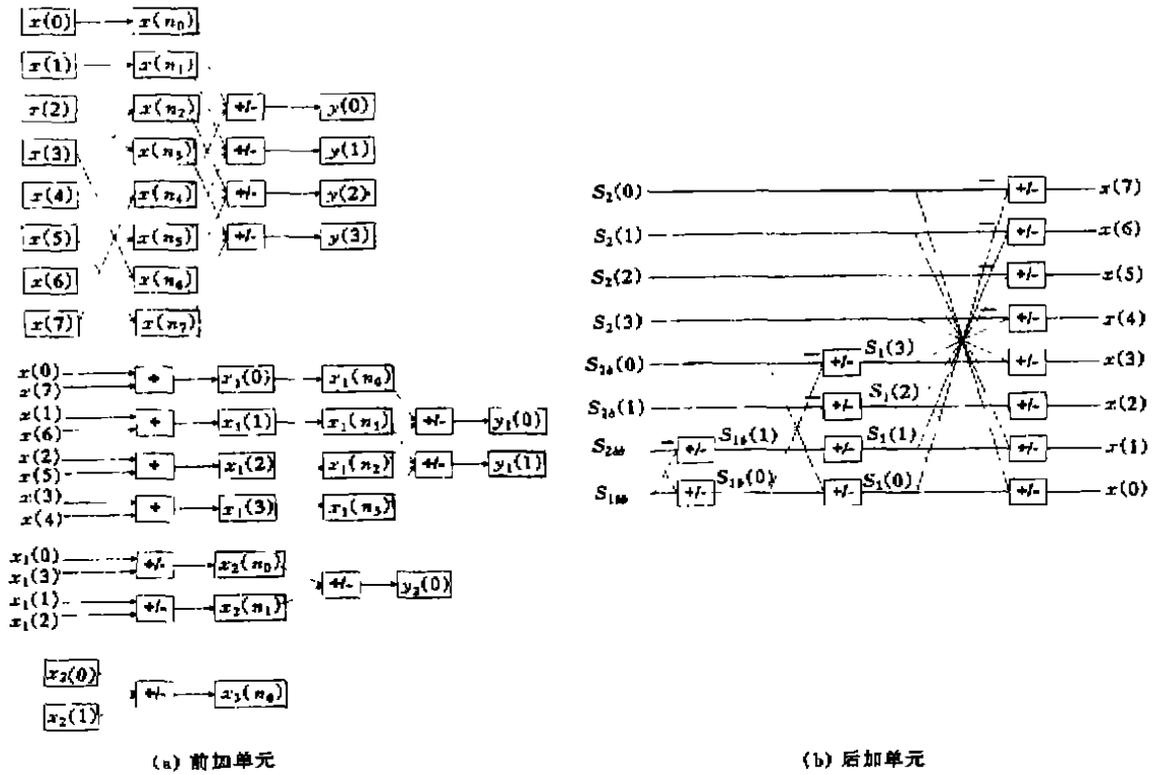


图 6

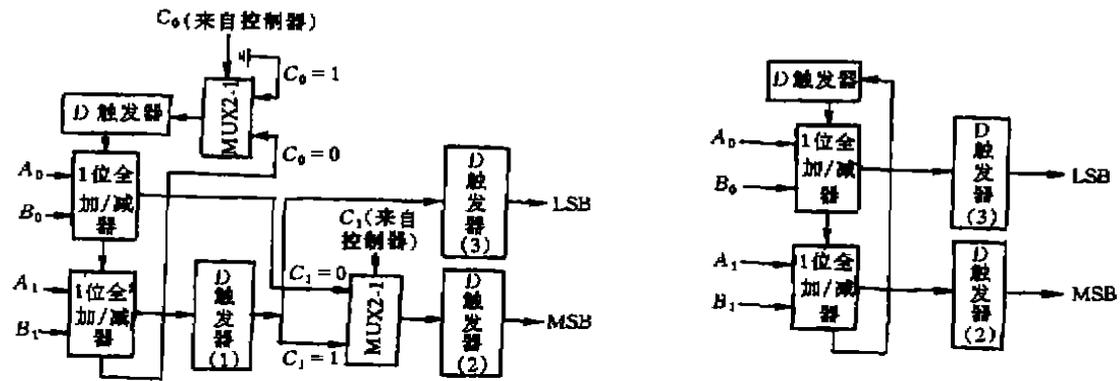


图 7 (a) 改进 2 位串行加 / 减法器 (b) 传统 2 位串行加 / 减法器

为适应前、后加单元多级连续相加(减)的结构,并在确保不溢出的情况下,有效地提高前、后加单元的运算精度,我们采用了如图 7 所示的 2 位串行加 / 减法器。该 2 位串行全加 / 减法器具有自动将相加结果除 2 并进行符号位扩展以防止在下级相加时产生溢出的功能。该电路对溢出的处理(包括移位和符号位扩展)均是在每次加(减)运算完成之后进行的,从而有效地提高了加(减)运算的精确度。它的具体工作过程如下:控制信号 C_1 在每一 8 时钟周期的第一个周期被置为高电平,而在其余 7 个时钟周期则保持低电平。这样,当 C_1 为高电平时,上一次

相加所得数据的符号位就从 D 触发器 (1) 被置入 D 触发器 (2) 和 D 触发器 (3) 。这等于于进行了一次符号位扩展。同时, 当前相加所得数据的最低位则因 C_1 控制信号而被 2 选 1 选择器所阻塞, 且其第二低位被储存在 D 触发器 (1) 中。当下一个时钟周期来临时, 控制信号 C_1 变为低电平, 当前相加所得数据的第三低位被送入 D 触发器 (2) 中, 而第二低位则由 D 触发器 (1) 被置入 D 触发器 (3) 中。这样, 下一级所接收到的数据最低位即为相加结果的第二低位, 等价于完成了一次除 2 操作。这种操作反复进行直到 C_1 重新被置为高电平为止。

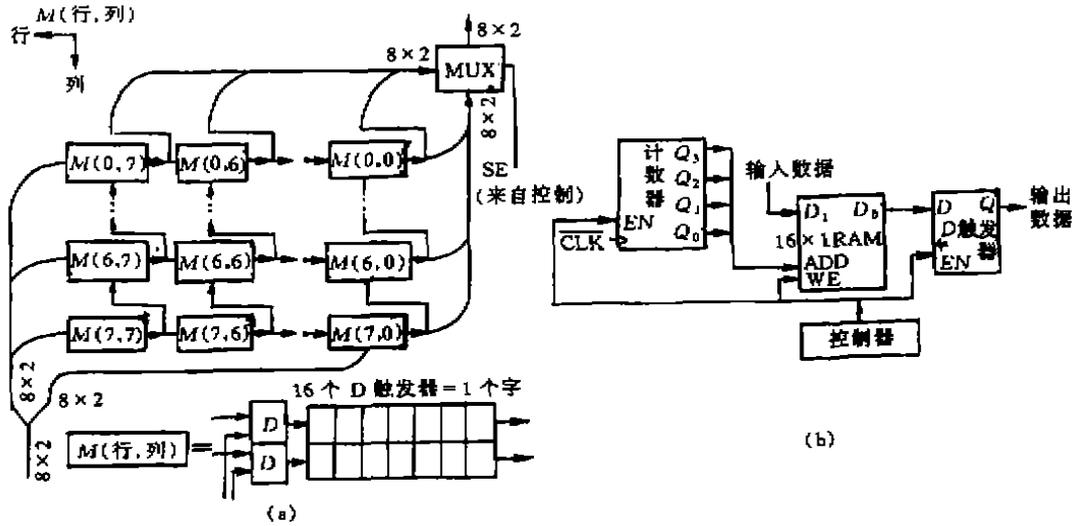


图 8 (a) 转移存储器 (b) 基于 16×1RAM 的 16 位移位寄存器

4.4 转移存储器

转移存储器是采用矩阵分解算法的二维 DCT/IDCT 处理器中的一个必不可少的电路单元。转移存储器的主要功能是实现一个 8×8 矩阵的转置。矩阵中的每一个数据均为 16 位。转移存储器的电路结构框图如图 8(a) 所示。由图 8(a) 可以看出, 转移存储器由 64 个“M”单元组成, 其中每个“M”单元又是一个由 16 个 D 触发器组成的特殊移位寄存器。因此, 整个转移存储器一共使用了 1024 个 D 触发器。每个“M”单元的最左边两个 D 触发器可以接收来自左边或底部的输入数据。该转移存储器的工作过程是这样的: (1) 第一个 64 时钟周期: 来自第一个一维 IDCT 单元的输入数据按照字并 / 位串和由下至上的方式由“M”单元的最左侧, 按行输入方式顺序存储在 64 个“M”单元中; (2) 第二个 64 时钟周期: 来自第一个一维 IDCT 单元的输入数据则按字并 / 位串方式由“M”单元的最左侧, 按列输入方式顺序存储在 64 个“M”单元中。同时, 多路器在 SE 控制下, 将第一个 64 时钟周期存储在转移存储器中的 8×8 矩阵在最右端按列输出。这些输出数据即是转置处理后的一维 IDCT 值; (3) 第三个 64 时钟周期: 多路器在 SE 控制下, 将第二个 64 时钟周期储存在转移存储器中的 8×8 矩阵按转置形式以行输出, 同时又按行接收一个新的 8×8 的一维 IDCT 矩阵。此后, 上述三个过程反复进行, 共同配合完成将 8×8 矩阵进行转置的功能。图 8 所示的转移存储器具有高度的紧凑性和很简单的控制信号, 与前级电路配合十分巧妙, 且完全实现了流水线操作。具有结构简单, 工作频率高的突出优点。由前所述, 采用图 8 所示结构的转移存储器至少应包含 1024 个 D 触发器, 这个数目相对于 FPGA 来说是十分庞大的, 因为每个 CLB(Configurable Logic Block) 仅包含两个 D 触发器。因此, 要用 FPGA 完成二维 IDCT 的设计并实现流水线分级操作就必须解决转移存储器的 D 触发器资源占用问题。为解决上述问题, 本文采用了基于 FPGA

的片内 RAM 的移位寄存器来设计转移存储器。经仿真实验证实,其效果很好。本文设计的基于 16×1 RAM 的 16 位移位寄存器电路框图如图 8(b) 所示。该 16 位移位寄存器的工作过程大致是这样的:新的输入数据每次总被写入 16×1 RAM 中前一次被读取数据所在单元,此过程反复进行,即可实现与移位操作类似的过程。

5 仿真结果及分析

5.1 二维 DCT/IDCT 处理器功能仿真结果

由于篇幅所限,此处仅将电路的实际输出转换为十进制形式以表格方式体现出来,而未能给出电路的详细仿真波形图。

在实际二维 DCT 处理器功能仿真实验中,我们连续输入 2 个 8×8 矩阵块,以验证二维 DCT 处理器计算结果是否正确及是否能够做到连续正确的工作。由于篇幅所限,此处仅给出一个 8×8 矩阵块的处理结果,如表 1 所示(按数据输入/输出的先后顺序)。

表 1

(a) 二维 DCT 处理器输入数据表

序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据
1	59	9	-214	17	-142	25	255	33	231	41	207	49	187	57	-232
2	59	10	-214	18	-142	26	255	34	231	42	207	50	187	58	-232
3	59	11	-214	19	-142	27	255	35	231	43	207	51	187	59	-232
4	59	12	-214	20	-142	28	255	36	231	44	207	52	187	60	-232
5	59	13	-214	21	-142	29	255	37	231	45	207	53	187	61	-232
6	59	14	-214	22	-142	30	255	38	231	46	207	54	187	62	-232
7	59	15	-214	23	-142	31	255	39	231	47	207	55	187	63	-232
8	59	16	-214	24	-142	32	255	40	231	48	207	56	187	64	-232

(b) 输入数据块经硬件处理后的结果(保留一位小数)

序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据
1	350.7	9	0	17	0	25	0	33	0	41	0	49	0	57	0
2	-335.6	10	0	18	0	26	0	34	0	42	0	50	0	58	0
3	-910.8	11	0	19	0	27	0	35	0	43	0	51	0	59	0
4	917.8	12	0	20	0	28	0	36	0	44	0	52	0	60	0
5	274.9	13	0	21	0	29	0	37	0	45	0	53	0	61	0
6	716.5	14	0	22	0	30	0	38	0	46	0	54	0	62	0
7	-236.5	15	0	23	0	31	0	39	0	47	0	55	0	63	0
8	-48.2	16	0	24	0	32	0	40	0	48	0	56	0	64	0

表 2

(a) 二维 IDCT 处理器输入数据表(四舍五人)

序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据
1	351	9	0	17	0	25	0	33	0	41	0	49	0	57	0
2	-336	10	0	18	0	26	0	34	0	42	0	50	0	58	0
3	-911	11	0	19	0	27	0	35	0	43	0	51	0	59	0
4	918	12	0	20	0	28	0	36	0	44	0	52	0	60	0
5	275	13	0	21	0	29	0	37	0	45	0	53	0	61	0
6	717	14	0	22	0	30	0	38	0	46	0	54	0	62	0
7	-237	15	0	23	0	31	0	39	0	47	0	55	0	63	0
8	-48	16	0	24	0	32	0	40	0	48	0	56	0	64	0

在二维 IDCT 处理器功能仿真实验中,我们直接将上述二维 DCT 处理器输出的 2 个 8×8 矩阵块作为二维 IDCT 处理器的输入,以验证二维 IDCT 处理器计算结果是否正确及是否能够

做到连续正确的工作。由于篇幅所限, 此处仅给出一个 8×8 矩阵块的处理结果。如表2所示(按数据输入/输出的先后顺序)。

(b) 输入数据块经硬件处理后的结果

序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据	序号	数据
1	58	9	-214	17	-142	25	255	33	231	41	207	49	187	57	-232
2	58	10	-214	18	-142	26	255	34	231	42	207	50	187	58	-232
3	58	11	-214	19	-142	27	255	35	231	43	207	51	187	59	-232
4	58	12	-214	20	-142	28	255	36	231	44	207	52	187	60	-232
5	58	13	-214	21	-142	29	255	37	231	45	207	53	187	61	-232
6	58	14	-214	22	-142	30	255	38	231	46	207	54	187	62	-232
7	58	15	-214	23	-142	31	255	39	231	47	207	55	187	63	-232
8	58	16	-214	24	-142	32	255	40	231	48	207	56	187	64	-232

由表1和表2, 可以看出, 所有模拟结果之间的误差均 $\leq \pm 1$, 因此可以得出结论: 该二维DCT/IDCT处理器的计算精度是符合要求的; 而且通过连续输入两个数据块可知, 该二维DCT/IDCT处理器是可以做到连续正确工作的。

6 结束语

本文设计的二维DCT/IDCT处理器利用Xilinx公司的XC4044EX系列芯片实现时占用约1520个CLB, 系统时钟频率为14MHz。如果转移存储器采用D触发器组进行设计的话, 预计系统时钟频率应能达到50MHz以上。

参 考 文 献

- [1] 卢焯, 等. 基于FPGA的实时高速二维DCT/IDCT处理器. 微电子学, 1996, (1): 15-19.
- [2] Li W. A New Algorithm to Compute the DCT and its Inverse. IEEE Trans.on SP, 1991, SP-39(6): 1305-1313.
- [3] Xilinx Inc. The Programmable Logic Data Book. 1993.
- [4] 孟宪元. 可编程专用集成电路原理、设计和应用. 电子工业出版社, 1995.

A REAL-TIME 2-D DCT/IDCT PROCESSOR USING FPGAs

Xiang Hui Teng Jianfu Wang Chengning

(Electronic Information Engineering College, Tianjin University, Tianjin 300072)

Abstract Based on the skew-circular convolution distributed algorithm presented by W.Li(1991). A 8×8 2-D DCT/IDCT processor has been designed using FPGAs, which can be used for HDTV's decoder or other signal and information processing systems. It can be used to calculate either DCT or IDCT depending on a single control line. All of the input/output are 12-bit and the internal data bus and internal parameters are 16-bit.

Key words FPGA(Field Programmable Gate Array), 2-D DCT/IDCT processor, Distributed algorithm

向晖: 男, 1973年生, 硕士, 主要从事高清晰度电视视频解码器及CDMA移动通信系统的研究。

王承宁: 男, 1972年生, 博士生, 主要从事高清晰度电视视频解码器及视频压缩技术的研究。

滕建辅: 男, 1954年生, 教授, 博士生导师, 主要从事滤波器理论与设计及信号与信息处理的研究。