Лабораторная работа № 4
SIMD-оптимизация оператора Собела с использованием различных наборов команд

Цель работы: познакомиться с принципами организации и программирования векторных расширений системы команд современных процессоров семейства x86.

Векторные расширения системы команд процессоров семейства x86 позволяют получить выигрыш в скорости обработки однородных данных в соответствии с SIMD-принципом (Single Instruction Multiple Data). Расширения представляют собой группы ассемблерных команд, представленные в таблице.

Таблица. Векторные расширения системы команд

Название	Поддержка процессорами	Год выпуска	Примечание
MMX (Multi Media eXtensions)	с Intel Pentium MMX	1997 г.	–
3DNow!	с AMD K6-2	1998 г.	не поддерживается процессорами Intel
SSE (Streamed SIMD Extensions)	с Intel Pentium III	1999 г.	–
SSE2	с Intel Pentium IV, с AMD Athlon 64	2000 г.	
SSE3	с Intel Pentium IV (ядро Prescott), с AMD Athlon 64 (ядро Venice)	2004 г.	
SSSE3 (Supplemental SSE3)	с Intel Core, с AMD Bobcat	2006 г.	
SSE4 (4.1, 4.2, 4a)	с Intel Core (ядро Penryn)	2008 г.	частично поддерживается процессорами AMD
AVX (Advanced Vector eXtensions)	с Intel Core i7 (Sandy Bridge), с AMD Phenom (Bulldozer)	2011 г.	
AVX2	с Intel Core i7 (Haswell)	2013 г.	

В зависимости от типа расчетного кода (целочисленные вычисления, вычисления с плавающей точкой одинарной или двойной точности) для решения практических задач могут применяться команды различных векторных расширений.

Оператор Собела применяется для выделения границ (контрастных переходов) на изображениях (рис. 1).

Рис. 1. Исходное изображение (слева) и результат применения к нему оператора Собела (справа)

Математически полутоновое изображение I представляет собой матрицу $I = \left\| I_{xy} \right\|_{n \times m}$, в которой каждый элемент I_{xy} является целым числом с диапазоном значений $[0; 255]$. Для каждой точки изображения (за исключением крайних) производится операция свертки 8-связной окрестности точки $\begin{Vmatrix} I_{x-1,\,y-1} & I_{x,\,y-1} & I_{x+1,\,y-1} \\ I_{x-1,\,y} & I_{x,\,y} & I_{x+1,\,y} \\ I_{x-1,\,y+1} & I_{x,\,y+1} & I_{x+1,\,y+1} \end{Vmatrix} = \begin{Vmatrix} A & B & C \\ D & E & F \\ G & H & I \end{Vmatrix}$ с заданной маской $M = \begin{Vmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{Vmatrix}$:

$$d_{xy} = \sum_{i=1}^{3} \sum_{j=1}^{3} I_{x+i-2,\,y+j-2} m_{ij} = \tag{1}$$

$$= m_{11}A + m_{12}B + m_{13}C + m_{21}D + m_{22}E + m_{23}F + m_{31}G + m_{32}H + m_{33}I.$$

При обработке изображения оператором Собела используются две маски:

$$M^{(h)} = \begin{Vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Vmatrix} \text{ и } M^{(v)} = \begin{Vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{Vmatrix},$$

а получаемые с их применением значения сверток $d_{xy}^{(h)}$ и $d_{xy}^{(v)}$ объединяются по следующей формуле:

$$d_{xy} = \left\lfloor \frac{256}{1140} \sqrt{\left(d_{xy}^{(h)}\right)^2 + \left(d_{xy}^{(v)}\right)^2} \right\rfloor,$$

где $\lfloor x \rfloor$ – операция округления вниз (усечения).

При реализации свертки с конкретной маской операции допускают упрощение по сравнению с (1):

$$t_1 = A - I,$$
$$t_2 = C - G,$$
$$H_h = 2(D - F) + t_1 - t_2,$$
$$H_v = 2(B - H) + t_1 + t_2,$$
$$d = \left\lfloor \frac{256}{1140} \sqrt{H_h^2 + H_v^2} \right\rfloor,$$

что уменьшает число выполняемых операций и экономит затраты вычислительного времени.

Вычисление значений t_1, t_2, H_h и H_v рациональнее производить как целочисленные операции, а вычисление значения d – как операции над вещественными аргументами одинарной точности. Подобное разделение операций дает возможным использование следующих связок расширений в зависимости от используемого процессора:

1. MMX + FPU;
2. MMX + 3DNow;
3. MMX + SSE;
4. SSE + SSE2;
5. SSE2 + AVX;
6. AVX + AVX2.

С использованием команд первого из указанных расширений производятся целочисленные операции, второго – вещественные.

При векторной обработке выигрыш в скорости достигается за счет параллельной обработки нескольких точек изображения в зависимости от того, какое число операндов помещается в регистре (2, 4 или 8). На рис. 2 показан пример для параллельной векторной обработки 4 операндов:

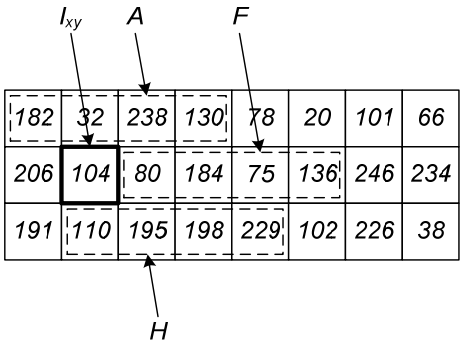

Рис. 2. Выборка операндов группами по 4 из памяти

Схема загрузки исходных данных из памяти представлена на рис. 3 (на примере MMX расширения).

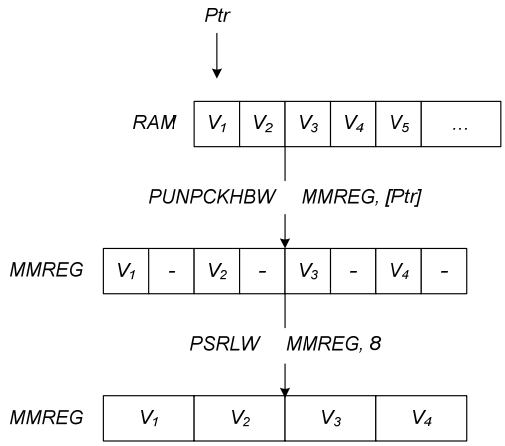

Рис. 3. Загрузка исходных данных из памяти

Ей соответствует следующий ассемблерный код:

```
punpckhbw mm0, [esi]
psrlw     mm0, 8
```

Схема вычисления значений сверток приведена на рис. 4 (на примере MMX расширения).

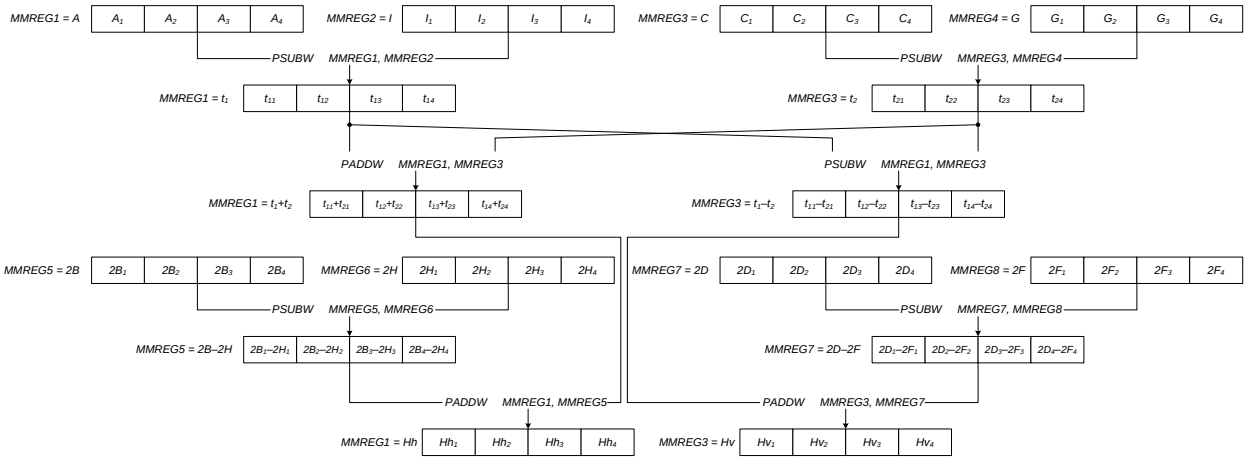

Рис. 4. Схема вычисления значений сверток

Схема преобразования целочисленных значений в вещественные одинарной точности приведена на рис. 5.

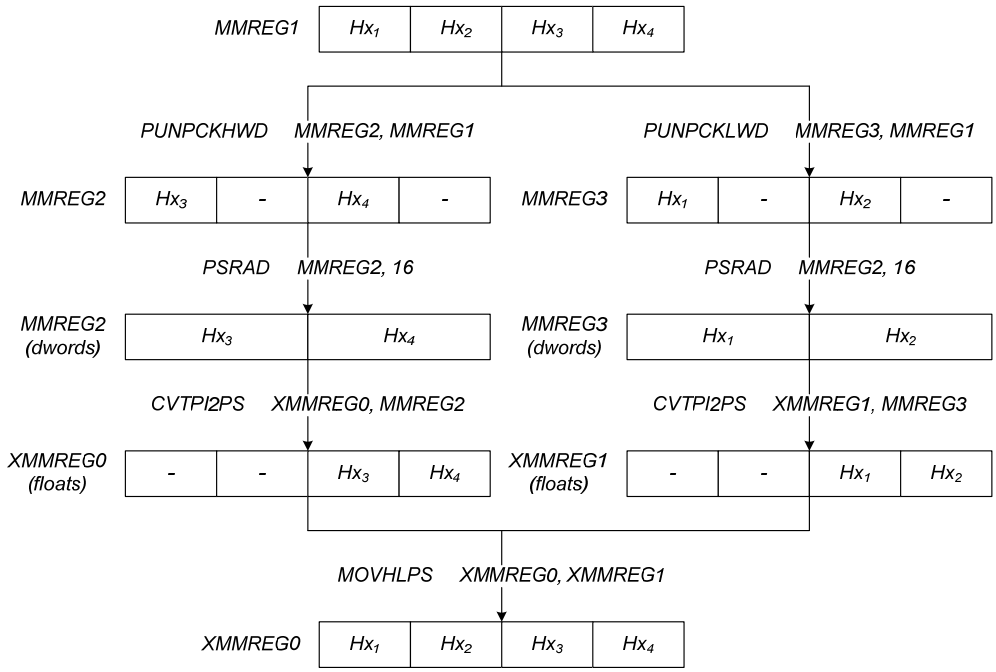

Рис. 5. Передача данных MMX→SSE

На рис. 6 показана схема вычисления искомого значения оператора Собела (на примере SSE расширения).

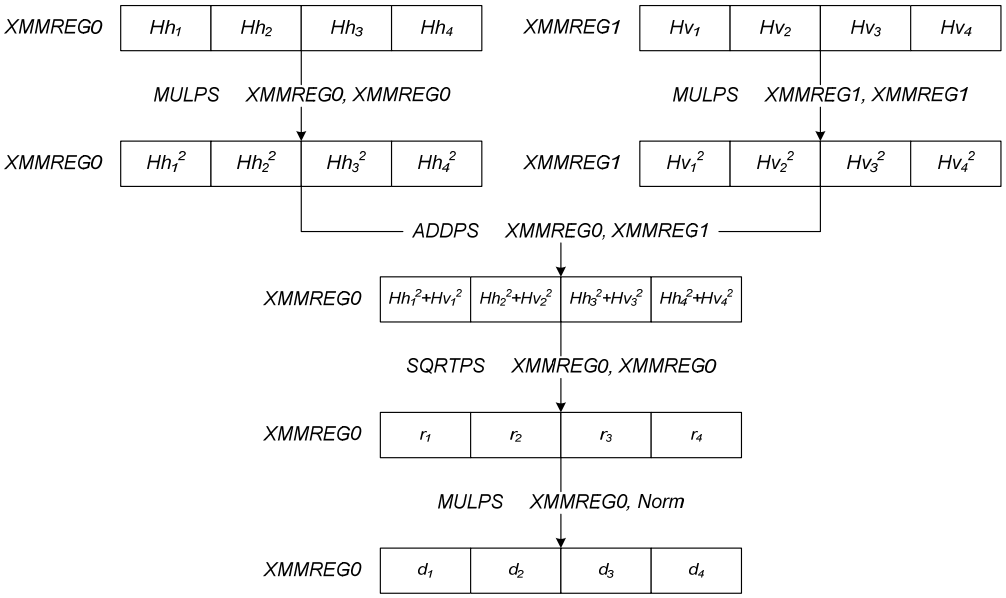

Рис. 6. Вычисление значения оператора Собела

Полученное значение является вещественным, перед записью в память оно должно быть преобразовано в целое. Схема преобразования приведена на рис. 7 (на примере связки SSE+MMX).

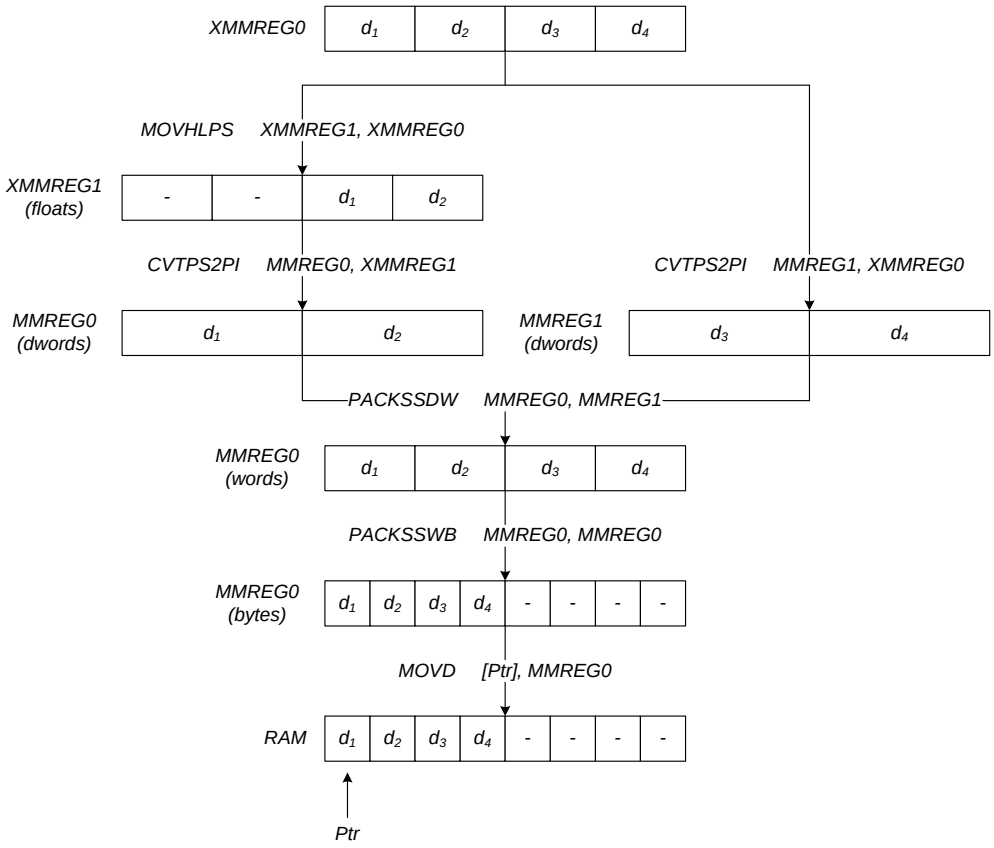

Рис. 7. Округление результата и запись в память

Задание. В соответствии с индивидуальным вариантом задания реализовать обработку изображения оператором Собела с использованием указанных преподавателем связок расширений. Убедиться в правильности выполняемых действий путем сопоставления результатов расчета высокоуровневой реализации с SIMD-оптимизированной версией кода. Протестировать разработанную программную реализацию на 3 различных процессорах для всех указанных связок расширений, определить необходимые затраты времени. Определить выигрыш во времени обработки SIMD-оптимизированных реализацией по сравнению с исходной высокоуровневой реализацией.

Содержание отчета.

1. Титульный лист.
2. Цель работы.
3. Описание условия решаемой задачи в соответствии с индивидуальным вариантом
4. Листинг программы.
5. Результаты сопоставления оптимизированной и высокоуровневой версий кода.
6. Результаты измерения времени выполнения высокоуровневого и оптимизированного фрагмента программы, оценка достигнутого выигрыша во времени.
7. Выводы.