

ВЫЧИСЛИТЕЛЬНЫЕ ВОЗМОЖНОСТИ МИКРОКОНТРОЛЛЕРОВ STM32F4

В работе изучаются вычислительные возможности 32-х разрядных микроконтроллеров STM32 для расчета произведения заполненных прямоугольных матриц и вычисления определенного интеграла методом прямоугольников. Показана большая потеря производительности при работе микроконтроллера с памятью SDRAM при матричных вычислениях. Отмечается эффективная работа сопроцессора с плавающей точкой микроконтроллера.

Ключевые слова: микроконтроллер STM32, FPU, STM32F429I-DISCO, STM32F4Discovery, SDRAM, IAR Embedded Workbench for ARM.

Введение. Создание автономных транспортных средств (роботов) связано с решением задач компьютерного зрения. Классической задачей в компьютерном зрении является определение, содержат ли видеоданные некоторый характерный объект, особенность или некоторую активность. Решение таких задач связано с необходимостью проведения вычислительных операций в режиме реального времени. Обычно для этих целей используются вычислительные системы, которые находятся за пределами робота и он из-за

этого не является полностью автономным. Однако в настоящее время появляются довольно мощные 32 - разрядные микроконтроллеры, которые имеют даже встроенные сопроцессоры для выполнения операций с вещественными числами.

Постановка задачи. В работе подробно рассматриваются вычислительные возможности микроконтроллера STM32F429ZIT6. Он установлен на плате STM32F429I-DISCO [1]. Микроконтроллер имеет 2МБайт флеш памяти и 256КБайт оперативной памяти и работает с частотой 180МГц. На плате STM32F429I-DISCO установлена память SDRAM размером 64МБит. В работе сравнивается время расчета произведения квадратных матриц для микроконтроллера и компьютера на процессоре AMD Phenom II X6 1090T с частотой ядра 3.2Гц. Используется одно ядро. Результат работы должен быть выведен на LCD дисплей, как представлено на рисунке 1. Ввод исходных данных выполняется с touch панели дисплея. Клавиатура создана программно и высвечивается внизу дисплея. Формат ввода должен быть следующий:

724,724,120,45,700

Окончанием ввода (моделирование клавиши Enter) является нажатие на область экрана CL.

Здесь:

724 – размерность квадратных матриц ($C = A * B$);

724,120 – индексы строки и столбца первого элемента матрицы C ($C[724][120]$), который выводится на экран;

45,700 - индексы строки и столбца второго элемента матрицы C ($C[45][700]$), который выводится на экран.

Также приводятся результаты изучения производительности расчета определенного интеграла методом прямоугольников для микроконтроллера STM32F407VGT6, установленного на плате STM32F4Discovery.

Результаты. Для решения задачи необходимо установить пакеты программ IAR Embedded Workbench for ARM, ST-LINK/V2 USB driver for Windows, STSW-STM32138[2]. Тестовая программа написана на основе проекта Touch_Panel.eww, который включен в файл STSW-STM32138.

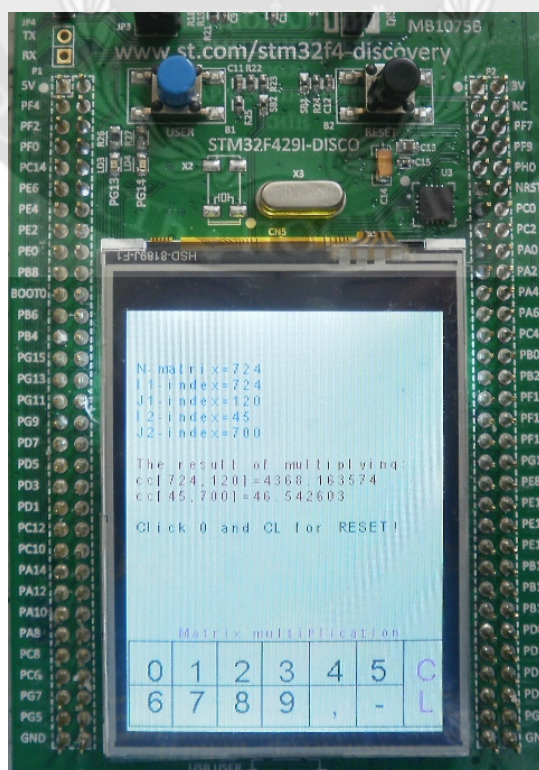


Рис. 1. Плата STM32F429I-DISCO

Формирование клавиатуры на touch - панели выполняется в 2 этапа (рис.2):

1. Внизу экрана прорисовывается клавиатура. Выполняется функцией void TP_Config()[2];

2. В начале цикла while() выделяются зоны touch панели, при нажатии на которые вводится соответствующий символ. На рисунке 2 показаны зоны экрана по координатам.

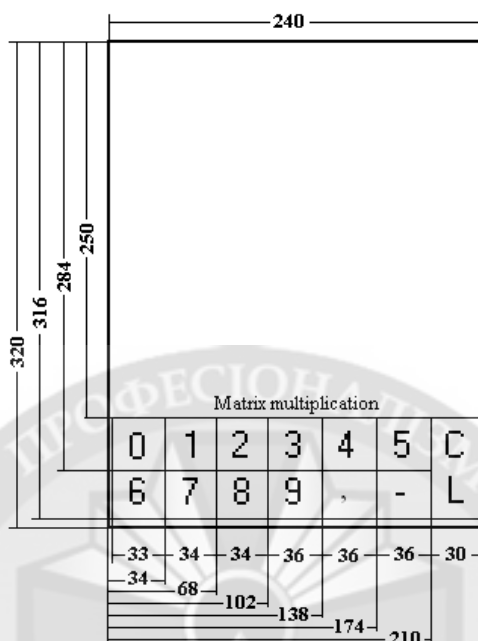


Рис. 2. Формирование клавиатуры

Входящие в программу функции работы со шрифтами, линиями, установки цвета представлены в файле stm32f429i_discovery_lcd.c.

Например, функция

LCD_DrawLine(1, 250, 239, LCD_DIR_HORIZONTAL) - рисует линию в горизонтальном направлении длиной 239 пикселей с начальными координатами x=1, y=250 пикселей.

Функции

LCD_SetFont(&Font8x12),
LCD_SetTextColor(LCD_COLOR_RED)

задают размер шрифта 8x12, который будет выведен на дисплей красным цветом.

Функция

LCD_DisplayChar(LCD_LINE_11, 14, 0x30)

выведет на 11 строке с координатой x=14 символ 0.

Функция

LCD_DisplayStringLine(LINE(30), (uint8_t*)" Matrix multiplication");

на 30-й линии выведет строку "Matrix multiplication".

Перемножение двух квадратных матриц выполнялось двумя способами:

1. По стандартному алгоритму в соответствии с известной программой на ФОРТРАНЕ

do 2 i=1,n

do 2 j=1,n

cc(i,j)=0.

do 3 k=1,n

3 cc(i,j)=cc(i,j)+aa(i,k)*bb(k,j)

2 continue

2. По алгоритму с подстановкой линейного массива для ускорения работы с памятью SDRAM

```

do 2 i=1,n
do m=1,n
a(m)=aa(i,m)
end do
do 2 j=1,n
cc(i,j)=0.
do 3 k=1,n
3 cc(i,j)=cc(i,j)+a(k)*bb(k,j)
2 continue

```

В программе вместо двумерных массивов используются одномерные массивы с подстановкой индексов:

```
cc(i,j) = cc(i + (j-1)*n)
```

Здесь i, j - индексы массива, n - размер квадратной матрицы.

Запись произвольного вещественного числа в память SDRAM в программе выполняется командой:

```
*(float*) (a +4*(i+(j-1)*n)) = 23.890
```

Здесь

a – адрес первой ячейки памяти SDRAM, с которой последовательно записывается весь одномерный массив. В программе этот адрес задан так:

```
#define a 0xD0100000
```

4 – выполняется умножение индекса массива на 4, так как вещественное число занимает в памяти 4 байта.

23.890 – произвольное вещественное число, записанное по адресу, который вычисляется для конкретных значений индексов i и j .

В программе для каждой из трех матриц (aa , bb и cc) выделяется по $0x2000000$ (2097152)байта памяти. Учитывая, что вещественное число занимает 4байта, максимальное количество элементов одномерного массива будет равно $2097152/4=524288$, а размерность квадратной матрицы $n=\text{SQRT}(524288) = 724$.

Пример фрагмента программы расчета произведения:

```

// Адреса начала массивов aa, bb, cc в SDRAM
#define aa 0xD0100000
#define bb 0xD0300000
#define cc 0xD0500000
// Очистка SDRAM памяти с 0xD0100000 по 0xD0700000 адреса
for(int ie=0xD0100000;ie<=0xD0700000;ie++) *(uint32_t*) (ie)=0x00;
// Выделение массива a в памяти SRAM
a = (float *)calloc(n1, sizeof(float));
// Умножение матриц по первому варианту
for(im=1;im<=nn;im++) { for(jm=1;jm<=nn;jm++)
{ *(float*) (aa +4*(im+(jm-1)*nn)) =1.0f*((float)(im*jm));
*(float*) (bb +4*(im+(jm-1)*nn))=1.0f/(*(float*) (aa +4*(im+(jm-1)*nn)));
*(float*) (cc +4*(im+(jm-1)*nn))=0.0f; } }
// Умножение матриц. по второму варианту.
//Здесь делается подстановка одномерного массива a[]
//Это приводит к повышению производительности
// для больших матриц (nхn>500элементов)
for(im=1;im<=nn;im++) {
for(int m2=1;m2<=nn;m2++) a[m2]=*(float*) (aa +4*(im+(m2-1)*nn));
for(jm=1;jm<=nn;jm++) { *(float*) (cc +4*(im+(jm-1)*nn))=0.0f;
for(k=1;k<=nn;k++)
*(float*) (cc +4*(im+(jm-1)*nn))=*(float*) (cc +4*(im+(jm-1)*nn))
+a[k]*(*(float*) (bb +4*(k+(jm-1)*nn))); } }

```

Перед компиляцией необходимо увеличить значение "кучи" (HEAP) с 0x200 (512Байт) до 0x2F000 (192521Байт) для того, чтобы с помощью функции calloc можно было выделить максимум памяти SRAM, которая находится внутри микроконтроллера. В этой памяти в программе создается промежуточный массив для ускорения матричного умножения (рис. 3).

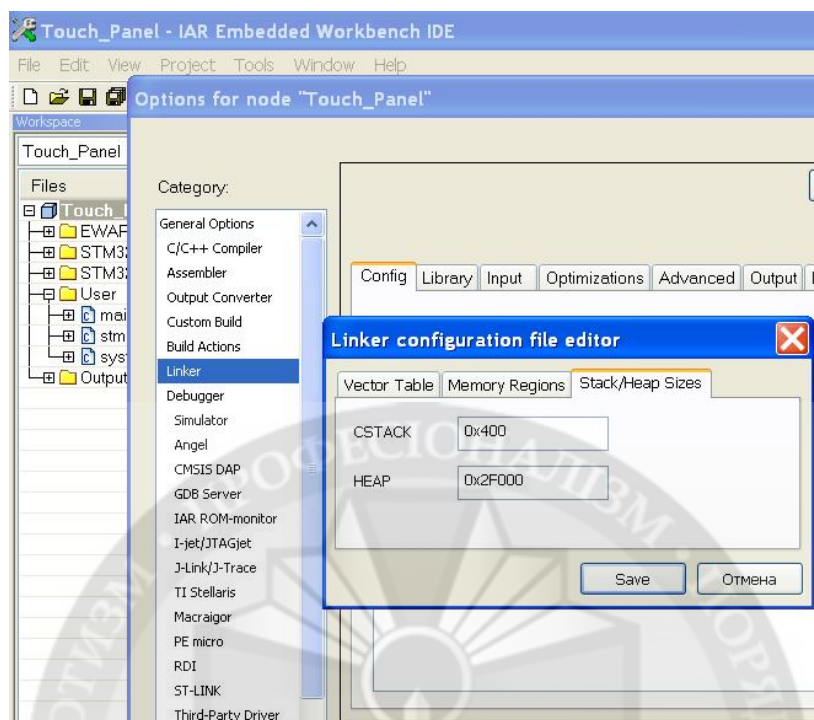


Рис. 3. Увеличение HEAP до 0x2F00

В таблице 1 представлены результаты расчета умножения квадратных матриц для микроконтроллера по алгоритмам с подстановкой линейного массива и без подстановки. Указано время расчета в секундах. Представлены результаты решения этой задачи на компьютере с процессором AMD Phenom II X6 1090T (3.2ГГц). Эффективность подстановки - это отношение времени счета без подстановки линейного массива к времени счета с его подстановкой.

Таблица 1

	Микроконтроллер STM23F429ZIT6				AMD Phenom II X6 1090T (3.2ГГц)			
	200x200	300x300	500x500	724x724	200x200	300x300	500x500	724x724
Без подстановки	4	13	61	184	0.06	0.17	0.78	2.96
С подстановкой	3	9	40	120	0.05	0.13	0.59	1.76
Эффект. подстан.	1.33	1.44	1.53	1.53	1.20	1.30	1.32	1.68

В работе также было рассмотрено быстродействие процессора STM32F407VGT6, установленного на плате STM32F4Discovery фирмы STMicroelectronics. В качестве вычислительной задачи рассматривался расчет определенного интеграла который равен числу π :

$$\int_0^1 \frac{4}{1+x^2} dx = 4 \cdot \arctg(x) \Big|_0^1 = \pi.$$

Согласно правилу прямоугольников интеграл был представлен суммой площадей элементарных прямоугольников:

$$\pi \approx h \cdot \sum_{i=1}^n \left(\frac{4}{1+x_i^2} \right); \quad h = \frac{1}{n}; \quad x_i = \left(i - \frac{1}{2} \right) \cdot h.$$

На рис. 4 представлен пример работы программы для одинарной и двойной точности.

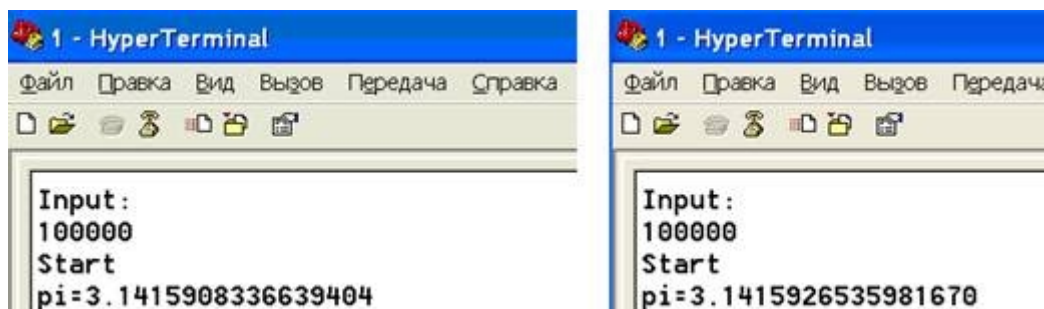


Рис. 4. Пример расчета для одинарной и двойной точности (pi=3.141592653589793238...)

Расчеты выполнялись с максимальной оптимизацией и без оптимизации (устанавливается опциями для компилятора), с сопроцессором(FPU) и без сопроцессора, для одинарной и двойной точности в средах разработки IAR Embedded Workbench for ARM ver. 6.70 и CoCoX ver. 1.7.5 с рекомендованным для него GNU Tools for ARM Embedded Processors Version: 4.8. Выполнено сопоставление с процессором AMD Phenom II X6 1090T (3.2ГГц), для которого задача запускалась на одном ядре. Результаты представлены в таблице 2. Дано время выполнения расчетов в секундах.

Таблица 2

	IAR Embedded Workbench for ARM STM32F407VGT6(168МГц)				CooCox ver. 1.7.5. STM32F407VGT6(168МГц)				AMD Phenom II X6 1090T, Linux Ubuntu, gcc.	
	Без оптим., без FPU	С оптим., без FPU	Без оптим., с FPU	С оптим., с FPU	Без оптим., без FPU	С оптим., без FPU	Без оптим., с FPU	С оптим., с FPU	Без оптим.	С оптим.
Одннар. точность	35с.	35с.	4с.	4с.	58с.	55с.	7с.	4с.	0.19с	0.11с
Двойная точность	-	65с.	-	82с.	-	155с.	-	153с.	-	-

Выводы.

1. Для расчетов предпочтение необходимо отдать среде разработки IAR Embedded Workbench for ARM.

2. Сопроцессор микроконтроллера STM32 работает только для вещественных чисел с одинарной точностью(4 байта).

3. Анализ показал, что компьютер на базе процессора AMD Phenom II X6 1090T (3.2ГГц) работает на матричных операциях примерно в 70 раз быстрее микроконтроллера с памятью SDRAM. Однако если компиляцию на компьютере выполнить с ключом оптимизации -O2, то компьютер будет работать примерно в 255 раз быстрее микроконтроллера. При расчете интеграла компьютер без оптимизации опережает микроконтроллер в 21 раз.

2. В случае подстановки одномерного массива в программу расчета матричного произведения скорость счета увеличивается примерно в 1.5-1.6 раза для больших массивов. Аналогичная ситуация имеет место при расчете на компьютере. Таким образом, в микроконтроллере, как и в компьютере эффективно используется кэш память.

3. Анализ результатов расчета интеграла [3] и матричного умножения на микроконтроллере [2] приводит к заключению, что динамическая память SDRAM уменьшает быстродействие микроконтроллерной системы примерно в 3.5-4 раза.

ЛИТЕРАТУРА:

1. STM32F429IDISCOVERY. Discovery kit with STM32F429ZI MCU. [Electronic resource]. - Mode of access: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090>, 2013

2. Мясичев А.А. Вычислительные возможности платы STM32F429I-DISCO для матричного умножения. [Electronic resource]. - Mode of access: http://webstm32.sytes.net/stm32_web/stm32_4.html. 2014.

3. Мясичев А.А. Вычислительные возможности STM32F4Discovery, ввод – вывод в UART с помощью scanf и printf для IAR. Практика для студентов. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/stm32_1. 2014.

REFERENCES:

1. STM32F429IDISCOVERY. Discovery kit with STM32F429ZI MCU. [Electronic resource]. - Mode of access: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090>, 2013

2. Myasishchev A.A. Vychislitel'nye vozmozhnosti platy STM32F429I-DISCO dlya matrichnogo umnozheniya. [Electronic resource]. - Mode of access: http://webstm32.sytes.net/stm32_web/stm32_4.html. 2014.

3. Myasishchev A.A. Vychislitel'nye vozmozhnosti STM32F4Discovery, vvod – vyvod v UART s pomoshch'yu scanf i printf dlya IAR. Praktika dlya studentov. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/stm32_1. 2014.

Без рецензії.

**д.т.н., проф. Мясичев О.А., д.т.н., проф. Ленков С.В.
ОБЧИСЛЮВАЛЬНІ МОЖЛИВОСТІ МІКРОКОНТРОЛЕРІВ STM32F4**

У роботі вивчаються обчислювальні можливості 32-х розрядних мікроконтролерів STM32 для розрахунку твори заповнених прямокутних матриць і обчислення певного інтеграла методом прямокутників. Показана велика втрата продуктивності при роботі мікроконтролера з пам'яттю SDRAM при матричних обчисленнях. Відзначається ефективна робота співпроцесора з плаваючою точкою мікроконтролера.

Ключові слова: Мікроконтролер STM32, FPU, STM32F429I-DISCO, STM32F4Discovery, SDRAM, IAR Embedded Workbench for ARM.

prof. Myasischev A.A., prof. Lenkov S.V.

COMPUTATIONAL CAPABILITIES OF MICROCONTROLLERS STM32F4

We study the computational capabilities of 32-bit STM32 microcontroller to calculate the product-filled rectangular matrices and calculate a definite integral by rectangles. It shows more loss of productivity at work microcontroller with SDRAM during matrix calculations. It noted the effective work of the coprocessor floating-point microcontroller.

Keywords: microcontroller STM32, FPU, STM32F429I-DISCO, STM32F4Discovery, SDRAM, IAR Embedded Workbench for ARM.