Myasischev A.A.

Khmelnitsky National University, Ukraine

COMPUTING CAPABILITIES STM32F429I-DISCO FOR MATRIX MULTIPLICATION

Unmanned cars based on the Autonomous management system. Driving is the automatic and performed using optical sensors, navigation systems, computer algorithms. Solution of these problems is performed in real time and requires a high speed. Currently available powerful 32-bit microcontrollers STM32 with embedded coprocessor. In the work discusses computational capabilities of the microcontroller STM23F429ZIT6. It is installed on the board STM32F429I-DISCO [1]. The microcontroller has a 2 MB Flash memory, 256 Kbytes of RAM and runs at a frequency of 180 MHz. The board STM32F429I-DISCO installed SDRAM memory of 64 Mbps. In the paper compared the time of calculation of matrix multiplication for the microcontroller and the PC with a processor AMD Phenom II X6 1090T(3,2GHz). It uses one core. The result of the work is to be displayed on the LCD screen, as shown in Figure 1. Entering initial data performed with the touch panel display. The keyboard is programmatically created and displayed at the bottom of the display. The input format:

724,724,120,45,700

At the end of the input click on the region CL.

Here:

724 - dimension of the square matrices (C = A * B);

724,120 - row and column indices of the first element of the matrix C (C[724][120]), which is displayed on the screen;

45,700 - row and column indices of the second element of C (C [45] [700]), which is displayed on the screen.

Figure 1 shows the output of the calculation results and the board STM32F429I-DISCO.

Fig.1. STM32F429I-DISCO

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.

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.

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.

Функии

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".

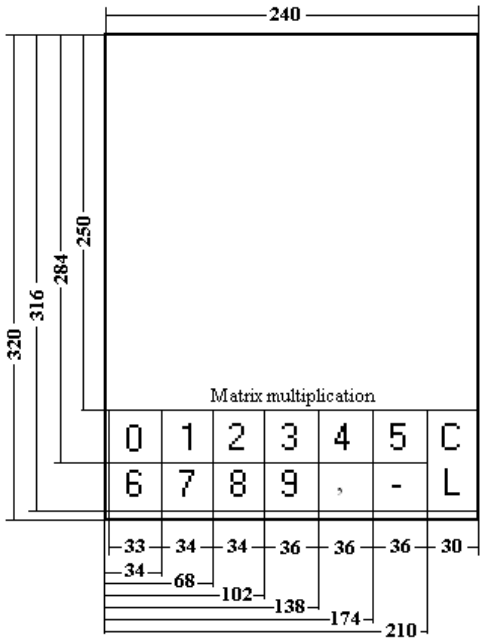

Рис.2. Формирование клавиатуры

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=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 (nxn>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)

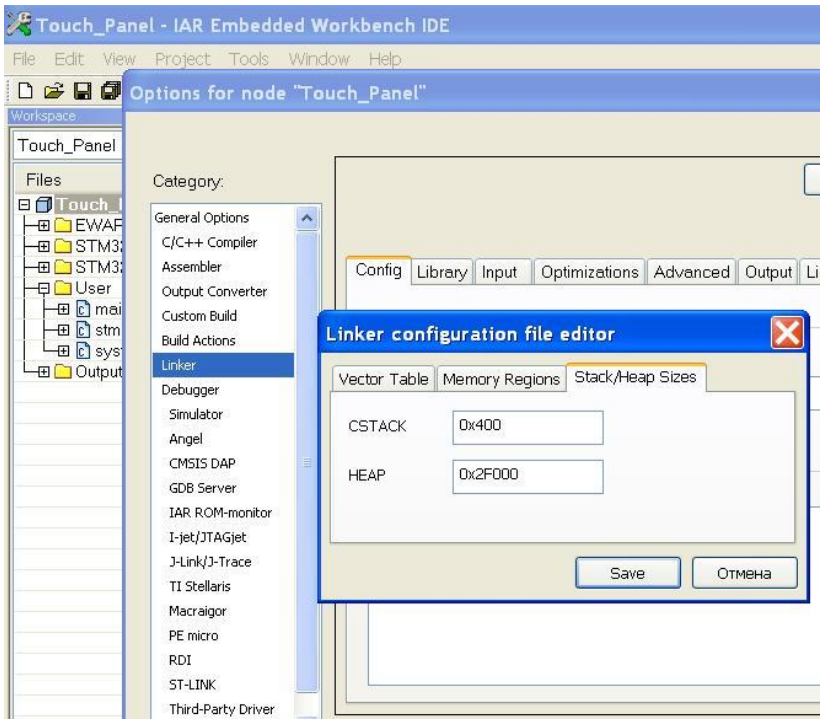

Рис.3. Увеличение HEAP до 0x2F00

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

Выводы

1. 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.
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 приводит к заключению, что динамическая память SDRAM уменьшает быстродействие микроконтроллерной системы примерно в 3.5-4 раза.

Литература

1. 32F429IDISCOVERY. Discovery kit with STM32F429ZI MCU. [Electronic resource]. - Mode of access: http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090, 2013
2. Мясищев 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. 2. Мясищев А.А. Вычислительные возможности STM32. Практика для студентов. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/stm32_1. 2014.