M4 DSP Matrix Functions

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	本范例代码使用M4内核DSP计算矩阵运算
BSP 版本	M480 Series BSP CMSIS V3.04.000
开发平台	NuMaker-PFM-M487 Ver 3.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 功能介绍

1.1 简介

CMSIS DSP函式库有许多关于矩阵的运算包含：

1. 矩阵加法运算
2. 矩阵减法运算
3. 矩阵乘法运算
4. 逆矩阵
5. 矩阵缩放
6. 转置矩阵

在此范例程序使用CMSIS DSP函式库进行转置矩阵、矩阵乘法和反矩阵运算，用户可以直接使用这些函式，来实现自己的数学方程式运算。程序内比较了有无使用DSP计算时间的差异，并计算效率提升比率。

1.2 原理

用户须先用arm_matrix_instance_f32宣告矩阵变量，再来呼叫函式arm_mat_init_f32()对使用到的矩阵初始化设定，定义矩阵直行与横列个数以及矩阵数值。完成设定后即可对各矩阵进行运算，其程序设定如下表。

arm_mat_init_f32(arm_matrix_instance_f32 *S, uint16_t nRows, uint16_t nColumns, float32_t *pData)		
参数：	*S	[in, out] 指向所宣告的浮点数矩阵结构
	nRows	[in] 矩阵直行个数
	nColumns	[in] 矩阵横列个数
	*pData	[in] 矩阵数据
回传值：		无

arm_mat_add_f32(const arm_matrix_instance_f32 *pSrcA, const arm_matrix_instance_f32 *pSrcB, arm_matrix_instance_f32 *pDst)		
参数：	*pSrcA	[in] 输入被加数矩阵
	*pSrcB	[in] 输入加数矩阵
	*pDst	[out] 输出结果矩阵
回传值：		成功显示ARM_MATH_SUCCESS, 矩阵不相合则显示 ARM_MATH_SIZE_MISMATCH

arm_mat_sub_f32(const arm_matrix_instance_f32 *pSrcA, const arm_matrix_instance_f32 *pSrcB, arm_matrix_instance_f32 *pDst)		
参数：	*pSrcA	[in] 输入被减数矩阵
	*pSrcB	[in] 输入减数矩阵
	*pDst	[out] 输出结果矩阵
回传值：		成功显示ARM_MATH_SUCCESS, 矩阵不相合则显示 ARM_MATH_SIZE_MISMATCH

arm_mat_mult_f32(const arm_matrix_instance_f32 *pSrcA, const arm_matrix_instance_f32 *pSrcB, arm_matrix_instance_f32 *pDst)		
参数：	*pSrcA	[in] 输入被乘数矩阵
	*pSrcB	[in] 输入乘数矩阵
	*pDst	[out] 输出结果矩阵
回传值：		成功显示ARM_MATH_SUCCESS, 矩阵不相合则显示 ARM_MATH_SIZE_MISMATCH

arm_mat_inverse_f32(const arm_matrix_instance_f32 *pSrc, arm_matrix_instance_f32 *pDst)		
参数：	*pSrc	[in] 输入欲计算矩阵
	*pDst	[out] 输出逆矩阵
回传值：		成功显示ARM_MATH_SUCCESS, 矩阵不为方阵则显示ARM_MATH_SIZE_MISMATCH, 若矩阵不可逆显示ARM_MATH_SINGULAR

arm_mat_scale_f32(const arm_matrix_instance_f32 *pSrc, float32_t scale, arm_matrix_instance_f32 *pDst)		
参数：	*pSrc	[in] 输入欲计算矩阵
	scale	[in] 缩放参数
	*pDst	[out] 输出结果矩阵
回传值：		成功显示ARM_MATH_SUCCESS

arm_mat_trans_f32(const arm_matrix_instance_f32 *pSrc, arm_matrix_instance_f32 *pDst)

参数：	*pSrc	[in] 输入欲计算矩阵
	*pDst	[out] 输出转置后矩阵
回传值：		成功显示ARM_MATH_SUCCESS

1.3 执行结果

执行后会打印出以下信息

```
+----------------------------------------+
|     DSP Matrix functions Sample Code   |
+----------------------------------------+

Calculating time with DSP instruction is 0.040667 ms

Calculating time without DSP instruction is 0.243083 ms

Efficiency increase rate is 5.98
```

2 代码介绍

使用CMSIS DSP函数库进行转置矩阵、矩阵乘法和反矩阵运算：

```
/* Initialize A Matrix Instance with numRows, numCols and data array(A_f32) */
arm_mat_init_f32(&A, numRows, numCols, A_f32);

/* Initialize Matrix Instance AT with numRows, numCols and data array(AT_f32) */
arm_mat_init_f32(&AT, numRows, numCols, AT_f32);

/* calculation of A transpose with DSP instruction */
arm_mat_trans_f32(&A, &AT);

/* Initialize ATMA Matrix Instance with numRows, numCols and data array(ATMA_f32) */
arm_mat_init_f32(&ATMA, numRows, numCols, ATMA_f32);

/* calculation of AT Multiply with A with DSP instruction */
arm_mat_mult_f32(&AT, &A, &ATMA);

/* Initialize ATMAI Matrix Instance with numRows, numCols and data array(ATMAI_f32) */
arm_mat_init_f32(&ATMAI, numRows, numCols, ATMAI_f32);

/* calculation of Inverse ((Transpose(A) * A) with DSP instruction */
arm_mat_inverse_f32(&A, &ATMAI);
```

接着使用CPU进行相同的计算：

```
/* Matrix Inverse (number of rows) */
void MatrixInverse(int n)
{
    uint32_t i,j,k;
    double t;
    for(i=0; i<n; ++i) {
        for(j=0; j<n; ++j) {
            A1_f32[i][n+j] = 0;
        }
    }
    for(i=0; i<n; ++i) {
        A1_f32[i][n+i] = 1;
    }
    for(i=0; i<n; ++i) {
        if(A1_f32[i][i] == 0)
```

```
            for(j=i+1; j<n; ++j)
                if(A1_f32[j][i] != 0) {
                    for(k=i; k<n*2; ++k)
                        swap(&A1_f32[i][k], &A1_f32[j][k]);
                    break;
                }
        /* if Matrix inverse not exist*/
        if(A1_f32[i][i] == 0) break;

        t = A1_f32[i][i];
        for(k=i; k<n*2; ++k) {
            A1_f32[i][k] /= t;
        }
        for(j=0; j<n; ++j) {
            if(i != j && A1_f32[j][i] != 0) {
                double t = A1_f32[j][i];
                for(k=i; k<n*2; ++k) {
                    A1_f32[j][k] -= A1_f32[i][k] * t;
                }
            }
        }
    }
}
/* Matrix Transpose (number of rows, number of columns) */
void MatrixTranspose(int m, int n)
{
    uint32_t i, j;
    for(i=0; i<n; i++) {
        for(j=0; j<m; j++) {
            resultT[i][j] = A1_f32[j][i];
        }
    }
}
/* Matrix Multiplication (number of rows, number of columns) */
void MatrixMult(int m, int n, int p)
{
    uint32_t i, j, k;
    for(i=0; i<m; i++) {
        for(j=0; j<p; j++) {
            resultM[i][j] = 0;
            for(k=0; k<n; k++) {
```

```
                resultM[i][j] = resultM[i][j] + resultT[i][k] * A1_f32[k][j];
            }
        }
    }
}
```

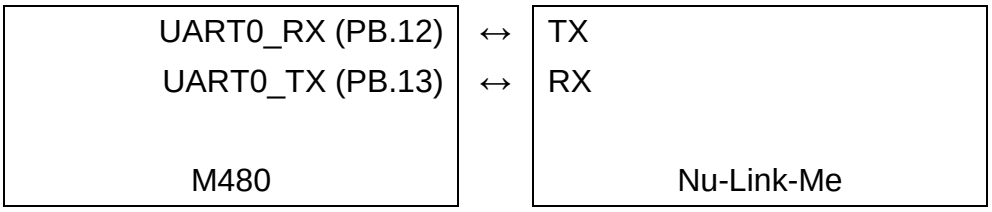
把计数器换成时间，其中计数器时钟源为HXT 12MHz：

```
/* Calculate the time, timer clock source is 12M, unit is ms */
DSPCalTime = (DSPCalTime/12000000) * 1000;
CalTime = (CalTime/12000000)* 1000;
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - M480 Series BSP CMSIS V3.04.000
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuMaker-PFM-M487 or other M480 Development Board
  - 示意图

M480 的 UART0_RX(PB.12)、UART0_TX(PB.13)连接至 Nu-Link Me，打印讯息。
设置终端机的 COM Port 与 Baud，COM Port 的编号可在设备管理器中找到「NuBridge Virtual Com Port (COMX)」，Baud 设置为 115200。



4 目录信息

- EC_M480_DSP_Matrix_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 DSP_Matrix.uvproj
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Jun. 25, 2019	1.00	1. 初始发布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.