# 组会报告

徐益

2018 年 12 月 3 日

## 1 工作内容

1. 收数据程序结构化；
2. 编写并测试 RZF 预编码矩阵生成程序；
3. 根据当前架构修改 PHY-MAC 对接程序及多线程系统。

## 2 data_collection 程序结构化

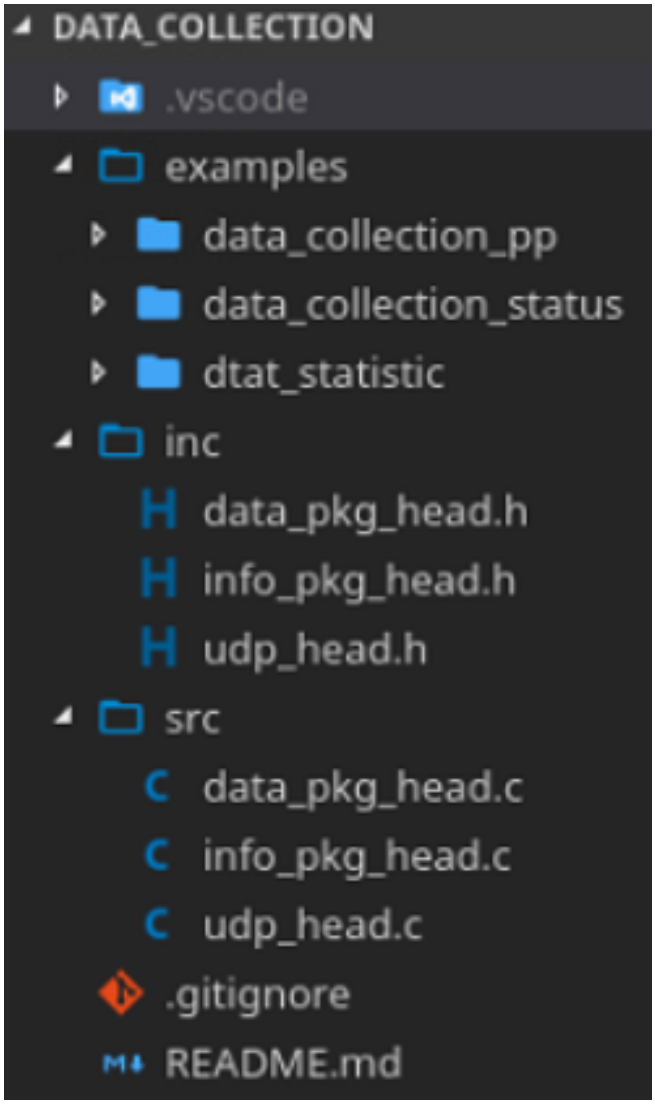


图 1: data_collection 程序结构

# 3 编写并测试 RZF 预编码矩阵生成程序

## 3.1 RZF 预编码

$$W_{RZF} = \beta H(H^H H + \xi I_K)^{-1} \tag{1}$$

## 3.2 核心函数

```
void cblas_cgemm (const CBLAS_LAYOUT Layout, const CBLAS_TRANSPOSE transa, const
CBLAS_TRANSPOSE transb, const MKL_INT m, const MKL_INT n, const MKL_INT k, const void
*alpha, const void *a, const MKL_INT lda, const void *b, const MKL_INT ldb, const void
*beta, void *c, const MKL_INT ldc);
```

### Include Files

- mkl.h

### Description

The ?gemm routines compute a scalar-matrix-matrix product and add the result to a scalar-matrix product, with general matrices. The operation is defined as

```
C := alpha*op(A)*op(B) + beta*C,
```

where:

op(*X*) is one of op(*X*) = *X*, or op(*X*) = $X^T$, or op(*X*) = $X^H$,

*alpha* and *beta* are scalars,

*A*, *B* and *C* are matrices:

op(*A*) is an *m*-by-*k* matrix,

op(*B*) is a *k*-by-*n* matrix,

图 2: 矩阵运算函数

```
lapack_int LAPACKE_cgesv (int matrix_layout , lapack_int n , lapack_int nrhs ,
lapack_complex_float * a , lapack_int lda , lapack_int * ipiv , lapack_complex_float *
b , lapack_int ldb );
```

### Include Files

- mkl.h

### Description

The routine solves for *X* the system of linear equations *A***X* = *B*, where *A* is an *n*-by-*n* matrix, the columns of matrix *B* are individual right-hand sides, and the columns of *X* are the corresponding solutions.

The *LU* decomposition with partial pivoting and row interchanges is used to factor *A* as *A* = *P***L***U*, where *P* is a permutation matrix, *L* is unit lower triangular, and *U* is upper triangular. The factored form of *A* is then used to solve the system of equations *A***X* = *B*.

图 3: 线性方程求解函数

### 3.3 表达式变换

$$W_{RZF} = H(\frac{1}{\beta}H^H H + \frac{\xi}{\beta}I_K)^{-1} \tag{2}$$

$$W_{RZF} = HM^{-1} \tag{3}$$

$$W_{RZF}M = H \tag{4}$$

$$M^T = W_{RZF}^T H^T \tag{5}$$

### 3.4 测试结果

```
========================================
2x2:
Time:              0.0113s
Throughput:     88817.84times/s
========================================
4x4:
Time:              0.0024s
Throughput:    415627.62times/s
========================================
8x8:
Time:              0.0066s
Throughput:    152207.00times/s
========================================
16x16:
Time:              0.0116s
Throughput:     86535.13times/s
========================================
32x32:
Time:              0.1258s
Throughput:      7946.16times/s
========================================
64x64:
Time:              0.1771s
Throughput:      5647.13times/s
========================================
128x128:
Time:              0.2832s
Throughput:      3531.32times/s
========================================
```

图 4: 测试结果

## 4 根据当前架构修改 PHY-MAC 对接程序及多线程系统