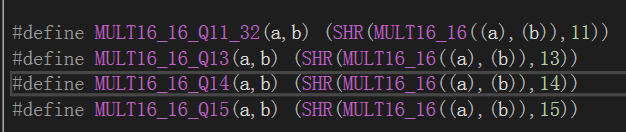
# 定点化数学运算库的结构

## 层次一：skv\_math\_core

两个实数之间的加减乘除，通过宏定义实现定标定点化运算，我们定义了上百个各种情况的宏定义，比如



## 层次二：skv\_math\_middle

1. 两个复数和复数与实数之间的加减乘除，通过内联函数实现。复数的实部虚部加减乘除计算通过调用**层次一skv\_math\_core**中定义的定标定点运算符。比如

static inline Complex Add1(Complex a, Complex b)

static inline Complex Add2(spx\_word16\_t a, Complex b)

static inline Complex Add3(Complex a, spx\_word16\_t b)

1. 向量的点积，也就是乘加和运算，通过正常的函数实现。复数的实部虚部加减乘除以及实数之间的加减乘除计算通过调用**层次一skv\_math\_core**中定义的定标定点运算符，比如

Complex inner\_product\_complex(Complex \*a, Complex \*b, spx\_uint32\_t len);

spx\_word16\_t inner\_product\_real(spx\_word16\_t \*a, spx\_word16\_t \*b, spx\_uint32\_t len);

Complex inner\_product\_complex\_real(Complex \*a, spx\_word16\_t \*b, spx\_uint32\_t len);

Complex inner\_product\_real\_complex(spx\_word16\_t \*a, Complex \*b, spx\_uint32\_t len);

1. 向量之间点乘点除点加点减，也就是对应元素的加减乘除，通过正常的函数实现。复数的实部虚部加减乘除以及实数之间的加减乘除计算通过调用**层次一skv\_math\_core**中定义的定标定点运算符，比如

Complex\* dot\_product\_complex1(Complex \*a, Complex \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_complex2(Complex a, Complex \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_complex3(Complex \*a, Complex b, Complex \*out, spx\_uint32\_t len);

spx\_word16\_t\* dot\_product\_real1(spx\_word16\_t \*a, spx\_word16\_t \*b, spx\_word16\_t \*out, spx\_uint32\_t len);

spx\_word16\_t\* dot\_product\_real2(spx\_word16\_t a, spx\_word16\_t \*b, spx\_word16\_t \*out, spx\_uint32\_t len);

spx\_word16\_t\* dot\_product\_real3(spx\_word16\_t \*a, spx\_word16\_t b, spx\_word16\_t \*out, spx\_uint32\_t len);

Complex\* dot\_product\_complex\_real1(Complex \*a, spx\_word16\_t \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_complex\_real2(Complex a, spx\_word16\_t \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_complex\_real3(Complex \*a, spx\_word16\_t b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_real\_complex1(spx\_word16\_t \*a, Complex \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_real\_complex2(spx\_word16\_t a, Complex \*b, Complex \*out, spx\_uint32\_t len);

Complex\* dot\_product\_real\_complex3(spx\_word16\_t \*a, Complex b, Complex \*out, spx\_uint32\_t len);

## 层次三：skv\_math

矩阵之间的加减乘除运算，矩阵的所有运算通过调用**层次二：skv\_math\_middle** 关于向量直接的运算实现。