Joe Stanley

ECE541 - HWK5

1. Consider the following C code:

```
double A[16], B[16];
double X;
int i;
   .
   .
for(i = 0; i < 16; i++)
   A[i] = X * A[i] + B[i];
```

a) Write a RISC-V assembly program that implements this loop, using the standard (non-vector) instruction set. Count the number of floating point instructions that will be executed in performing this loop.

b) Write a version of this loop using a RISC-V model that has vector instructions. Use the vector op-codes listed in Figure 4.2 . Count the number of floating point instructions that will be executed in performing this code.

c) Write a version of this loop using a RISC-V model that has SIMD instructions. Use the SIMD op-codes described in Chapter 4 (there is no table of these instructions in the chapter, but a syntax is suggested in the chapter). Count the number floating point instructions that will be executed in performing this code.

In [42]:
```
# Calculate Total Number of Instructions
instructions = 2 + (16 * 8)
```

a):

Total number of instructions executed: 130

Number		Instruction	Descriptors	Comment
1		fld	f0,X	Load Scalar X
2		addi	x28,x5,#256	Last Address to Load
3	Loop:	fld	f1,0(x5)	Load A[i]
4		fmul.d	f1,f1,f0	X x A[i]
5		fld	f2,0(x6)	Load B[i]
6		fadd.d	f2,f2,f1	(X x A[i]) + B[i]
7		fsd	f1,0(x6)	Store into A[i]
8		addi	x5,x5,#8	Increment index to A
9		addi	x6,x6,#8	Increment index to B
10		bne	x28,x5,Loop	Check if Done

b):

Total number of instructions executed: 8

Number	Instruction	Descriptors	Comment
1	vsetdcfg	4xFP64	Enable 4 Double Precision FP vregs
2	fld	f0,X	Load Scalar X
3	vld	v0,x5	Load Vector A
4	vmul	v1,v0,f0	Vector-Scalar Multiplication (XxA)
5	vld	v2,x6	Load Vector B
6	vadd	v3,v1,v2	Vector-Vector Add ((XxA) + B)
7	vst	v3,x6	Vector Store Sum
8	vdisable	-	Disable Vector Regs