



# Download worksheet7

Please mute yourself when not asking questions



# CS 152: Discussion Section 7

## Multithreading and Vectors

Yue Dai, Albert Ou  
03/20/2020



# Administrivia

- Problem Set 4 due 10:30am on Fri, April 3rd
- Lab 3 due April 6th
- Spring break is next week
  
- Spectre/Meltdown discussion during Albert's OH (next Mon?)



## New Office Hours

Monday (Yue) and Thursday (Albert) afternoons

- All weekday afternoons seem equally popular, but based on survey responses this works for the vast majority



# Agenda

- PS3 Review
- Multithreading
- Vectors



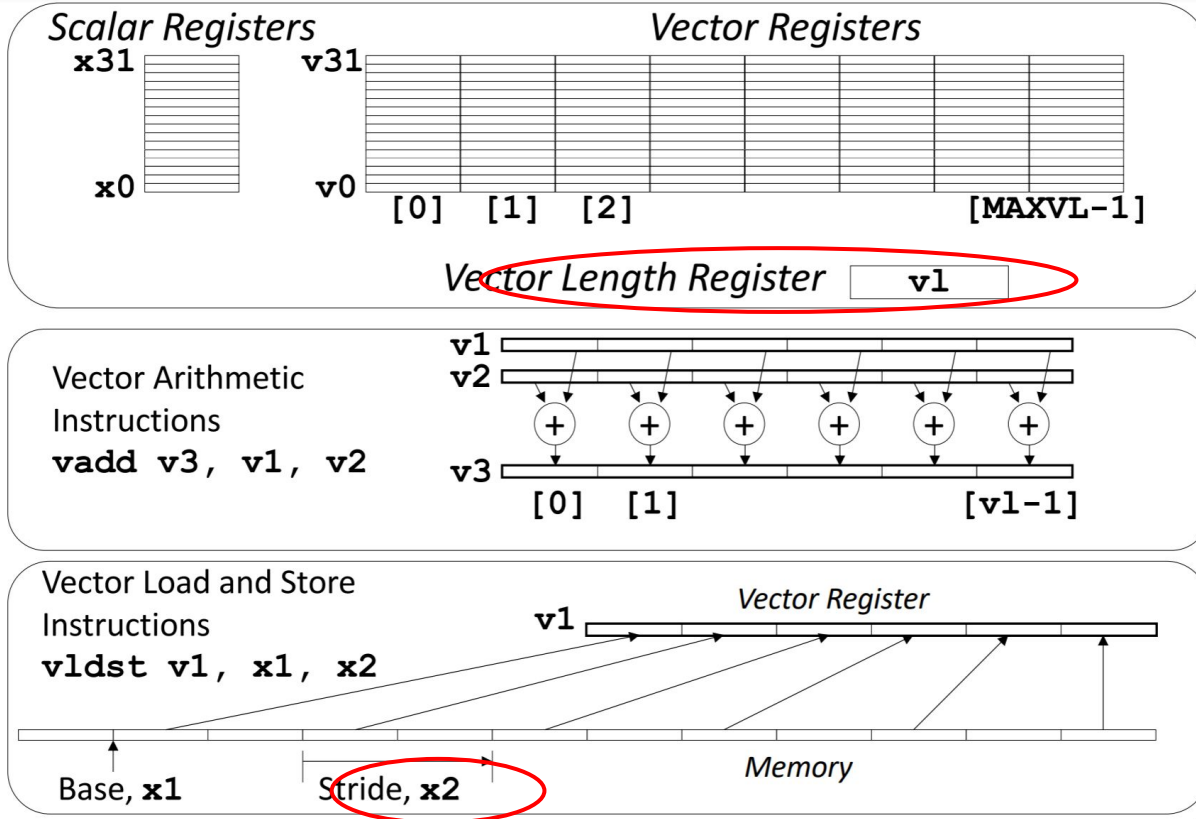
# Multithreading

- Fine-grained multithreading
  - Switch between threads on each clock cycle
- Coarse-grained multithreading
  - Switch threads only on costly stalls
- Simultaneous multithreading
  - Interleave multiple threads in multiple issue slots with no restrictions

# Multithreading



# Vectors





# Vector Programming

```
# C code
for (i=0; i<64; i++)
    C[i] = A[i] + B[i];
```

```
# Scalar Code
    li x4, 64
loop:
    fld f1, 0(x1)
    fld f2, 0(x2)
    fadd.d f3,f1,f2
    fsd f3, 0(x3)
    addi x1, 8
    addi x2, 8
    addi x3, 8
    subi x4, 1
    bnez x4, loop
```

```
# Vector Code
    li x4, 64
    setv1 x4
    vld v1, x1
    vld v2, x2
    vadd v3,v1,v2
    vst v3, x3
```



## Vector Stripmining

```
for (i = 0; i < N; i++) {  
    C[i] = A[i] + B[i];  
}
```

```
    andi x4, xN, 63    # N mod 64  
    setvl x4  
loop:  
    vld v1, x1         # Load A  
    vld v2, x2         # Load B  
    vadd v3, v1, v2    # C = A + B  
    vst v3, x3         # Store C  
    sub xN, xN, x4     # Subtract elements  
    slli x4, x4, 3     # Multiply by elt size  
    add x1, x1, x4     # Bump pointers  
    add x2, x2, x4  
    add x3, x3, x4  
    li x4, 64  
    vsetvl x4          # Reset full vlen  
    bgtz xN, loop
```



## Vector Stripmining #2

Alternative vsetvl instruction that returns the actual vector length:

```
vsetvl <actual>, <requested>
```

- $\text{actual} = \min(\text{requested}, \text{MAXVL})$
- Suppose  $\text{MAXVL} = 64$ :
  - $\text{VLEN} = 64$  for  $\text{requested} \geq 64$
  - $\text{VLEN} = \text{requested}$  for  $\text{requested} < 64$

loop:

```
setvl x4, xN
vld v1, x1      # Load A
vld v2, x2      # Load B
vadd v3, v1, v2 # C = A + B
vst v3, x3      # Store C
sub xN, xN, x4   # Subtract elements
slli x4, x4, 3   # Multiply by elt size
add x1, x1, x4   # Bump pointers
add x2, x2, x4
add x3, x3, x4
bgtz xN, loop
```