ECE154A — Discussion 04

George Higgins Hutchinson

October 22, 2021

Keep your eyes open for. . .

- PSet 2: due Friday, October 22
- Lab 2: due Friday, October 22 (autograder bug is causing some timeouts. this is my fault, not yours, and you won't be marked down if it's the only problem.)
- Midterm: Thursday, October 28

MIPS register/calling convention

Calling Convention: etiquette between functions (which may have different authors!) at the machine level.

Callee-saved in bold in this table:

$0	$zero	Always 0 (wire to ground)
$1	$at	The Assembler Temporary, used when expanding pesudo-ops.
$2-3	$v0-1	The return value of a function call (use the stack if you need more than two words)
$4-7	$a0-4	The arguments to a function call (use the stack if you need more than 4 words)
$8-15,24,25	$ t0-9	The temporary registers
$16-23	**$s0-7**	**The Saved Registers**
$26,27	$k0,1	Kernel Reserved registers
$28	**$gp**	Globals pointer, for addressing static memory
$29	**$sp**	Stack Pointer
$30	**$fp (or $s8)**	Frame Pointer for language VMs (python, java, etc).
$31	**$ra**	Return address from a function call.

More MIPS decoding: from MT1-FA2016

```
       addi $v0 $zero 0
loop:  addi $t0 $a0 1
       add $v0 $v0 $t0
       srl $a0 $a0 1
       bne $a0 $zero loop
```

Translate this code to C, and explain what it's doing. What's the best-case and worst-case runtime?

MIPS decoding solution

```
int count, data;
count = 0;
{
  count += data & 1;
  data >> 1;
} while (data != 0)
```

This code counts the ones in the provided bitstring!

Best case: $1 + 1 * 4 = 5$ instructions, when data $= 0$ initially.

Worst case: $1 + 32 * 4 = 129$ instructions, when data $=$ -1 initially.

MIPS math: from MT1 FA'17

```
        addi  $v0, $0, -1
LOOP:   bne   $a0, $0, ELSE
        j     DONE
ELSE:   srl   $a0, $a0, 1
        addi  $v0, $v0, 1
        j     LOOP
DONE:
```

- Explain what this code does, in english or mathmatical expression. Worst case dynamic count?
- re-write this algorithm for minimum dynamic count – what can you achieve?

MIPS math: from MT1 FA'17

```
        addi   $v0, $0, -1
LOOP:   bne    $a0, $0, ELSE
        j      DONE
ELSE:   srl    $a0, $a0, 1
        addi   $v0, $v0, 1
        j      LOOP
DONE:
```

- Explain what this code does, in english or mathmatical expression. Worst case dynamic count?
  $v0 = \mathrm{floor}(\log_2(a0))$
  In the worst case, runs $1 + 32 * 4 + 2 = 131$ instructions.
- re-write this algorithm for minimum dynamic count – what can you achieve?

MIPS math: solution part 2

Streamlined code:

```
         addi   $v0, $0, -1
         beq    $a0, $0, DONE
 LOOP:   srl    $a0, $a0, 1
         addi   $v0, $v0, 1
         bne    $a0, $0, LOOP
 DONE:
```

Worst-case: $2 + 3 * 32 = 98$ instructions.