Rodrigo Becerril Ferreyra

Student ID 017584071

CECS 440 Section 02

05 October 2021

# Midterm 1

## Question 1

1. The formula for total time taken is (instructions/program) * CPI * (seconds/clock cycle). Applying the hardware optimization would change the time by (1-0.10) * (1-0.10) * 1.14 = 0.9234.
2. The new program is 1/0.9234 = 1.08 times faster than the old program. This hardware optimization is definitely worth applying.

## Question 2

```
bne $s0, $s1, ELSE
add $s2, $s0, $zero
j EXIT
ELSE:
    sub $s2, $s0, $s1
EXIT:
```

# Question 3

Free question.

# Question 4

a) Register addressing lets you choose from the registers. Since the register fields are five bits wide, they can go from 0 to 31, which covers all 32 registers.

b) This is an example of immediate addressing. The immediate field is 16 bits wide, but it is signed, so one can go from -32768 to 32767.

c) This uses PC-relative addressing. The address field is 16 bits wide; this means that MIPS can add anything from -32768 to 32767 (or about 32768/4 = 8 KiB of instructions in either direction) to the PC.

d) This is pseudodirect addressing. The address field is 26 bits wide, so the range is 0 to 67 108 863. The address field is multiplied by 4 then concatenated to the PC.

e) Load and store instructions use base addressing. The address field is 16 bits wide, so it can go from -32768 to 32767.

# Question 5

a) The number of wafers in this case is 280 (assuming 100% yield).

b) Assuming the area of the wafer is a = 70 686 $mm^2$ (area of a circle) and the total area of the dies is y = 280 * 20.7 mm * 10.5 mm = 60 858 $mm^2$, the area of the die that is wasted is (a – y)/a = 13.9%

# Question 6

The following is the complete first line. The `bne` instruction jumps to `0x00450014` which is the `nop` instruction.

```
0x0044FFFC    bne    0x00450000
```

# Question 7

(a)

```
lw $t0, 12($s7)   # get B[3]
sll $t0, $t0, 4   # allign address
add $t0, $t0, $s6 # get address of A[t0]
lw $t0, 0($t0)    # get A[t0]
sub $s0, $s1, $t0 # f = g - A[t0]
```

(b)

```
f = A[2];
```

(c)

```
lui $t7, 0xFF00
ori $t7, $t7, 0x1234
```

# Problem 8

(a) The value is 36 (decimal).

(b) This question has a typo: I assumed that `$1` means `$s1` (because `$1` is actually `$at`, which is reserved for the assembler). With that in mind, the value is 28 (decimal).

(c) The value is 40 (decimal).

# Problem 9

```
add $sp, $sp, -8 # make space
sw $ra, 4($sp)   # save $ra
sw $s0, 0($sp)   # save $s0
```