COMP 122/L Final Practice Exam

Final Exam Topics:

- The use of memory on MIPS (with the `lw` and `sw` instructions) and branches (conditionals and loops)
- Truth tables
- Circuits
- Boolean formulas
- De Morgan's laws
- Karnaugh maps (K-maps), including those with *don't cares*

In addition to this review, you should also review:

- Related Labs (Labs 4 - 6)
- All the handouts

This review itself is not intended to be comprehensive

1.) Consider the following `.data` section of a MIPS program:

```
.data
array:
  .word 7, 2, 4
```

Finish this MIPS program so that it prints out every element of this array, WITHOUT using a loop. As a hint, you'll need to use different offsets in your `lw` instructions. Don't forget to terminate the program.

2.) Consider the following `.data` section of a MIPS program:

```
.data
array:
  .word 7, 2, 4, 8
copy:
  .word 0, 0, 0, 0
```

Finish this MIPS program so that `copy` will contain a copy of the contents of `array`. You MUST use a loop. Don't forget to terminate the program.

3.) Consider the following C-like code:

```
int array[] = {4, 8, 9, 1, 0, 5};
for (int index = 0; index < 6; index += 2) {
  int temp = array[index];
  array[index] = array[index + 1];
  array[index + 1] = temp;
}
```

This code started to be translated to MIPS assembly as follows:

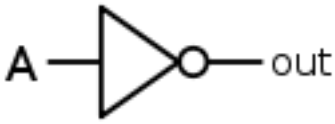
```
.data
array:
  .word 4, 8, 9, 1, 0, 5
```

Complete the translation of this code. Don't forget to terminate the program.

4.) What component is shown below?

5.) What component is shown below?

6.) What component is shown below?



7.) Draw the circuit corresponding to the following sum-of-products equation:

```
R = !A!B + AB
```

8.) Consider the following sum-of-products equation:

```
R = !ABC + ABC + A!B!C
```

8.a.) Write the equation as a truth table.

8.b.) Simplify it using a Karnaugh map.

9.) Consider the truth table below, which includes *don't cares*:

A	B	C	D	U
0	0	0	0	1
0	0	0	1	X
0	0	1	0	0
0	0	1	1	1
0	1	0	0	X
0	1	0	1	1
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	X
1	1	1	1	0

9.a.) Write out the unoptimized sum-of-products equation corresponding to this truth table. As a hint, *don't cares* can be skipped over.

9.b.) Using a K-map, derive an optimized equivalent sum-of-products equation that exploits *don't cares* where appropriate.

10.) Consider the truth table below, which includes *don't cares*:

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	X
0	0	1	1	X
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	X
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	1
1	1	0	1	0
1	1	1	0	X
1	1	1	1	1

Using a K-map, derive an optimized equivalent sum-of-products equation that exploits *don't cares* where appropriate.