CDA 3101 - Computer Organization 2

First Day Quiz - Solution

January 15, 2019

1. Write the printf statement to print the double variable "val", right justified by 15 spaces, with 4 digits after the decimal point.

printf("%15.41f n", val);

2. What is the logic equation represented by the following circuit?



 $\overline{Y} + X.\overline{Z}$

The expression is:

$$(\overline{X}+Y).\overline{(Y+(X.\overline{Z}))}+X.\overline{(Y.Z)}$$

You can reduce it directly or use a truth table and a K-Map. Here, we will reduce using the Boolean Algebra laws. The truth table and the K-Map have been left as an exercise for you.

$$(\overline{X}+Y).\overline{(Y+(X.\overline{Z}))}+X.\overline{(Y.Z)}$$

Using De Morgan's Law:

$$(\overline{X} + Y).(\overline{Y}.\overline{(X.\overline{Z})}) + X.(\overline{Y}.\overline{Z})$$

Using De Morgan's Law again:

$$(\overline{X} + Y).(\overline{Y}.(\overline{X} + Z)) + X.(\overline{Y}.\overline{Z})$$

Using Distributive Law:

$$(\overline{X} + Y).(\overline{X}.\overline{Y} + \overline{Y}.Z) + (X.\overline{Y} + X.\overline{Z})$$

Using Distributive Law again:

$$\overline{X}.\overline{X}.\overline{Y} + \overline{X}.\overline{Y}.Z + \overline{X}.Y.\overline{Y} + Y.\overline{Y}.Z + X.\overline{Y} + X.\overline{Z}$$

Using Identity, Idempotent and Inverse Laws:

 $\overline{X}.\overline{Y} + \overline{X}.\overline{Y}.Z + 0 + 0 + X.\overline{Y} + X.\overline{Z}$

Using Distributive Law:

$$\overline{X}.\overline{Y}.(1+Z) + X.\overline{Y} + X.\overline{Z}$$

Using Identity and Idempotent Laws:

$$\overline{X}.\overline{Y} + X.\overline{Y} + X.\overline{Z}$$

Using Distributive Law:

 $(\overline{X} + X).\overline{Y} + X.\overline{Z}$

Using Identity, Idempotent and Inverse Laws:

 $\overline{Y} + X.\overline{Z}$

3. Convert the following C code snippet to MIPS

```
int x = 0;
char arr[20] = "Hello";
int l= 5;
while(x<1)
{
    printf("%c",arr[x]);
    x++;
}
```

This is one approach. Your solution might be slightly different.

```
.data
        .asciiz "Hello" #Store the string
arr:
        .text
main:
       li $t0, 0
                       # t0 is x. Initialize with 0
       li $t1, 5
                        # t1 is l. Initialize with 5
                        # Load starting address of string in s0
       la $s0, arr
       li $v0, 11
                        # 11 is code to print a character
Loop:
       lb $a0, 0($s0) # Load byte in address in s0 into a0
       syscall
                        # This will print the character
       addi $t0,$t0,1 # Increment t0
        addi $s0,$s0,1 # Increment s0
       blt $t0,$t1,Loop # We have more stuff to print. Go back
       li $v0,10
                        # 10 is code for halt
        syscall
```

4. Fill in the single-cycle datapath below so that it implements the load word instruction. An example store word instruction is lw s0, 4(s1). Furthermore, indicate the bit size of each of the input and output lines being used in the datapath. You can ignore control lines.

