Name: Vidit Pokharna
GT Number: 903772087

0. Which prereq did you take and when? CS2110 CS2130
Fall Spring Summer 98 98 99 00 01 02 03 04 05 06 07 08 ... (23)

1. Are you planning on taking courses for which CS 2200 is a prerequisite:
__ Networking __ Operating Systems __ Hardware Arch. ✓ No

2. Write a function in C called swap that will swap two ints:

It will be called like this:

```
int a = 42;
int b = 78;
/* Call to swap goes here */
printf("%d %d\n", a, b);
```

The output would be: 78 42

Note: a and b are not global variables. That is your function must be able to be called with different pairs of variables.
Write swap here:

```
int c = b;
b = a;
a = c;
```

3. What does "make" do? Your answer should include three major items.

a. tracks dependencies between files

b. invokes compiler to create executables

c. helps run unit tests

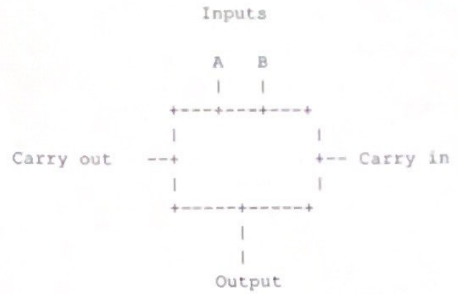
4. Write 42 in binary and in hexadecimal

binary: 0b101010

hexidecimal: 0x2A

Write 42.25 in binary (Not IEEE Floating Point) binary: 0b101010.01

5. Given a full adder as a building block:



Design a 4 bit arithmetic functional unit which will implement addition (A+B) and 2's complement subtraction (A-B)

Extra credit: Provide for overflow detection. (Use the back of the paper for your answer)

6. What do you suppose this does:

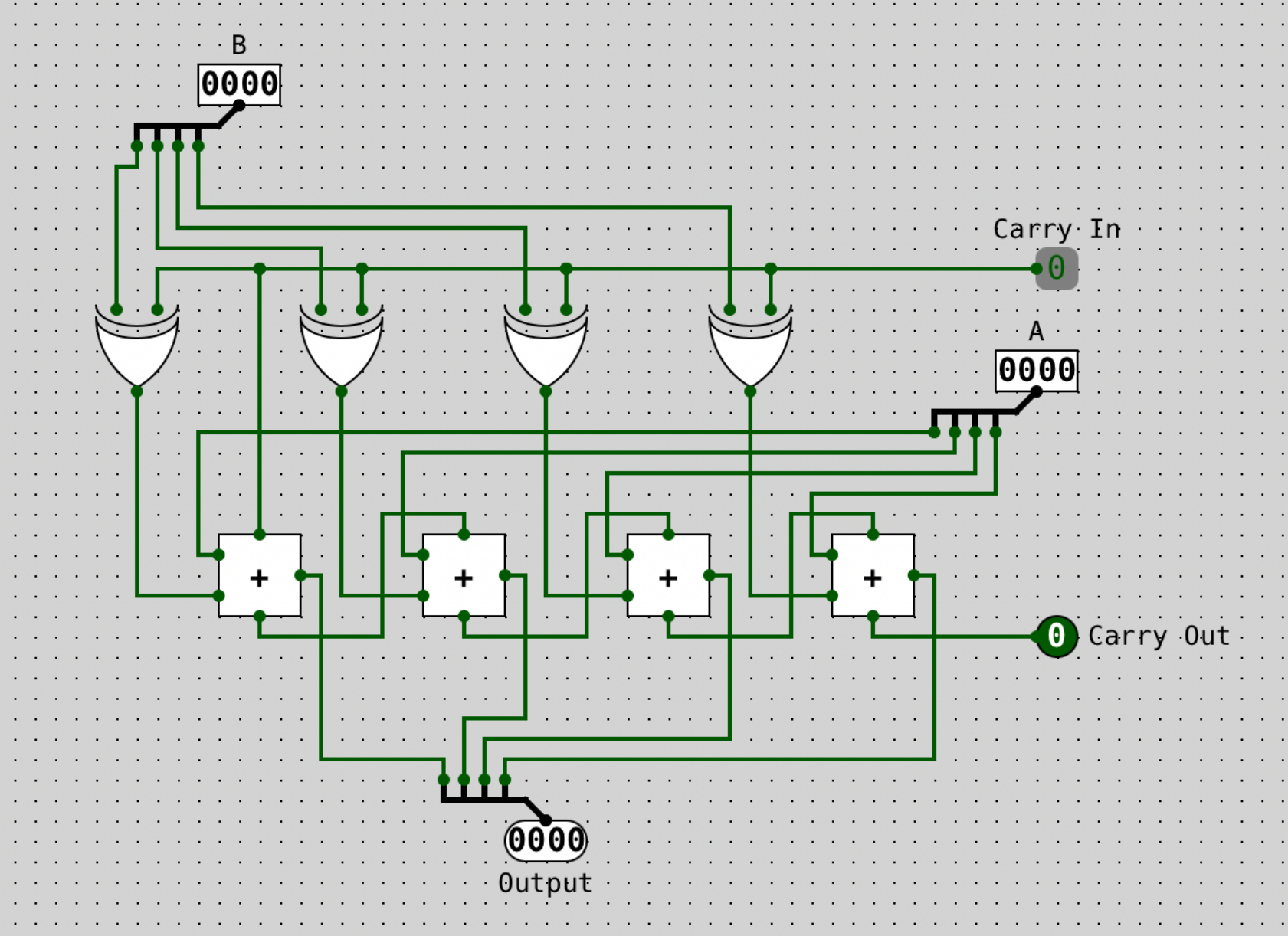
```
load   r1, #2
load   r2, #3
add    r3, r1, r2
store  r3, result

result: word  0
```

This code stores 2 into R1 and 3 into R2, add them together and stores in R3, and stores the value in R3 into the label result.

7. What do you think about this:

```
Plan* createNewSelectionNode(
  Cond* condition,
  char* relation)
{
  Plan* newNode;
  newNode=(Plan*)malloc(sizeof(Plan));
  newNode->op=SELECTION;
  newNode->condParams[0]=condition;
  newNode->tableP1=NULL;
  newNode->tableP2=NULL;
  newNode->table1=relation;
  free(newNode);
  return newNode;
}
```

It describes a method with 2 parameters, creating and returning a Plan* object while updating the object's (which is a node) traits


B
0000
Carry In
0
A
0000
+
+
+
+
0
Carry Out
0000
Output