Name: Han Choi

GT Number: 903598626

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

1. Are you planning on taking courses for which CS 2200 is a prerequisite:
X Networking __ Operating Systems __ Hardware Arch.

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:

```
Void Swap (int *a, int *b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

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

a. Combines files to be compiled together.

b. Creates an executable name

c. Uses information from a description file

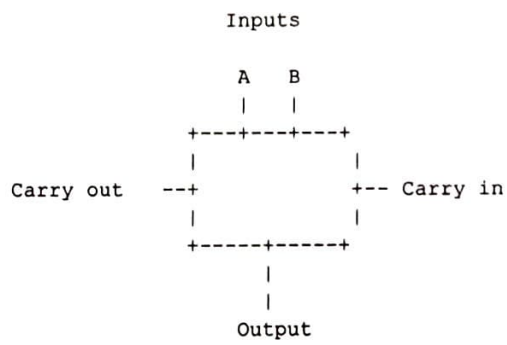
4. Write 42 in binary and in hexadecimal

binary: 101010

hexidecimal: X002A

Write 42.25 in binary (Not IEEE Floating Point) binary: 101010.01

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:

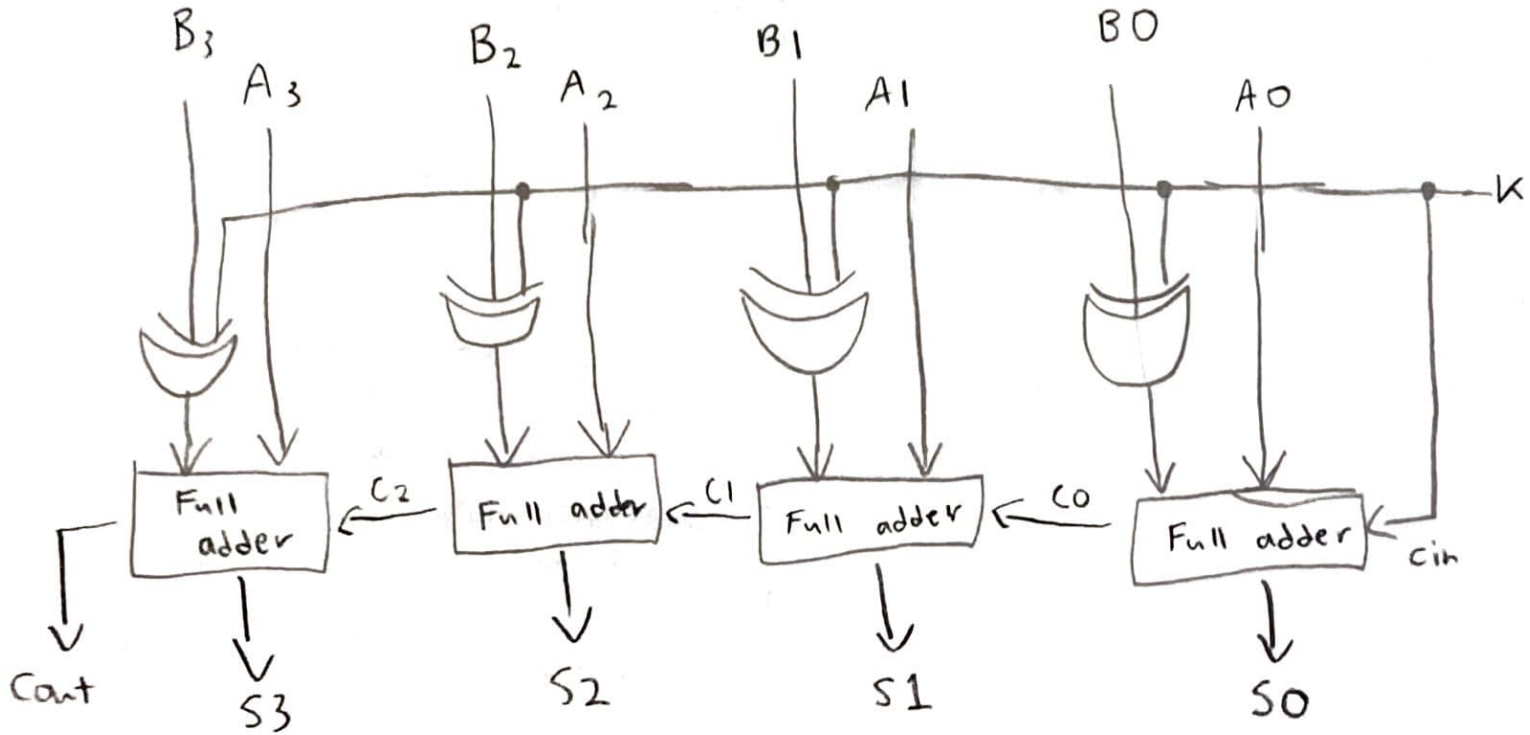
```
load   r1, #2
load   r2, #3
add    r3, r1, r2
store  r3, result

result: word  0
```

It loads 2 and 3 into r1 and r2 respectively then adds the result into r3 which gets stored in results.

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;
}
```

You need to check whether newNode is null after mallocing or you run into a Segmentation fault.


B3
A3
B2
A2
B1
A1
B0
A0
K
Full adder
Full adder
Full adder
Full adder
C2
C1
C0
Cin
Cout
S3
S2
S1
S0