

## CS 273 HW 2 - Ziad Arafat

1. Assuming we use two's complement

- 000000000000 is the 0 value so we will have a higher range for negatives.
  - Because we don't have to represent negative 0
- So we can calculate the rest as  $2^{11} = 2048$  for each sign
- So it will be  $-2048 \dots +2047$
- If we count zero as positive then it's 2048 negatives and 2048 positives.

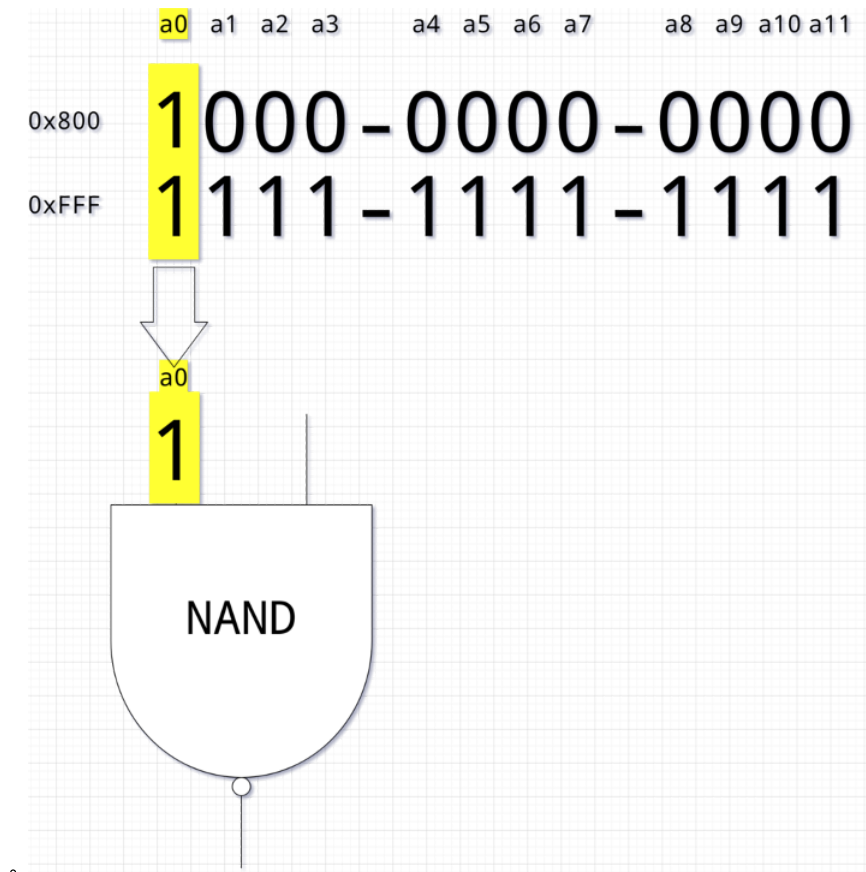
2. Attached the code to the bottom of this file.

3. The modern computer follows a more useful architecture based on the Von Neumann machine and improved by the Harvard and the Princeton architectures. It's not just a simple single-purpose computer that does math computations.

- Lots of general purpose components
- Sometimes several processing units
- Many ways to use I/O
- In general modern PCs are general purpose whereas the simple calculator is made to do one thing.

4.  $2^{16}$  possible addresses

5. Created in draw.io



6. Questions

- Programmable Flash Memory
- Mine is a 328P so it has 32k Bytes of flash
- SRAM and EEPROM
  - The CPU has built in access to SRAM which can be used to temporarily store data.
  - We also have an EEPROM memory on the BUS which could be used to store data in the long term.
- On the 328P
  - SRAM: 2K Bytes
  - EEPROM: 1K Bytes

```
// I added the hex version of the function below the
// original and called it instead of the original on x1-x3
```

```

//
// Declare the things that exist in our assembly code
//
/*
extern "C" {
    byte x1;
    byte x2;
    byte x3;
    void addition();
}
*/

extern byte x1, x2, x3; // import x1 x2 x3
extern "C" void addition(void); // import the addition program as a function

//
// function to read a 2-digit decimal value from user
//
byte read2DigitValue()
{
    byte inch; int val;
    Serial.println("Enter a 2-digit decimal value:");

    while (!Serial.available()) delay(100); // Wait for the user to enter something

    inch = Serial.read(); // As soon as they do read what they entered
    val = (inch - '0') * 10;

    while (!Serial.available()) delay(100); // Do it again for the next digit

    inch = Serial.read(); // read what the digit is
    val += (inch - '0');
    Serial.print("It's decimal value entered is ");
    Serial.println(val, DEC);
    return (byte) val;
}

byte readHexValue()
{
    byte inch; int val;
    Serial.println("Enter a 2-digit hexadecimal value:");

    while (!Serial.available()) delay(100); // Wait for the user to enter something

    inch = Serial.read(); // As soon as they do read what they entered
    if (inch >= '0' && inch <= '9') val = (inch - '0') * 16;
    else if (inch >= 'a' && inch <= 'f') val = (inch - 'a' + 10) * 16;
    else if (inch >= 'A' && inch <= 'F') val = (inch - 'A' + 10) * 16;

    while (!Serial.available()) delay(100); // Do it again for the next digit

    inch = Serial.read(); // read what the digit is
    if (inch >= '0' && inch <= '9') val += (inch - '0');
    else if (inch >= 'a' && inch <= 'f') val += (inch - 'a' + 10);
    else if (inch >= 'A' && inch <= 'F') val += (inch - 'A' + 10);
    Serial.print("The hex value entered is ");
    Serial.println(val, HEX); // print out the value they entered
    Serial.print("It's decimal value is ");
    Serial.println(val, DEC);
    return (byte) val;
}

//
// Arduino-required setup function (called once)
//
void setup()
{
    pinMode(13, OUTPUT); // set the LED to be automatically off
    digitalWrite(13, LOW);

    //
    // Initialize serial communications
    //
    Serial.begin(9600); // Start listening on the Serial console
    //
    // Read three values from user, store in global vars
    //
    x1 = readHexValue(); // read 2 digit values into each memory address
    x2 = readHexValue();
    x3 = readHexValue();
    //
    // Call our assembly code
    //
    addition();
    //
    // print out value of x2 variable
    Serial.println("After addition()");
    Serial.print("The value is ");

```

```
    Serial.println((int) x3,DEC);
}

//
// Arduino-required loop function (called infinitely)
//
void loop()
{
    // prints a * every 20 seconds for some reason
    delay(20000); // 20,000 millisecs == 20 seconds
    Serial.println("*");
}
```