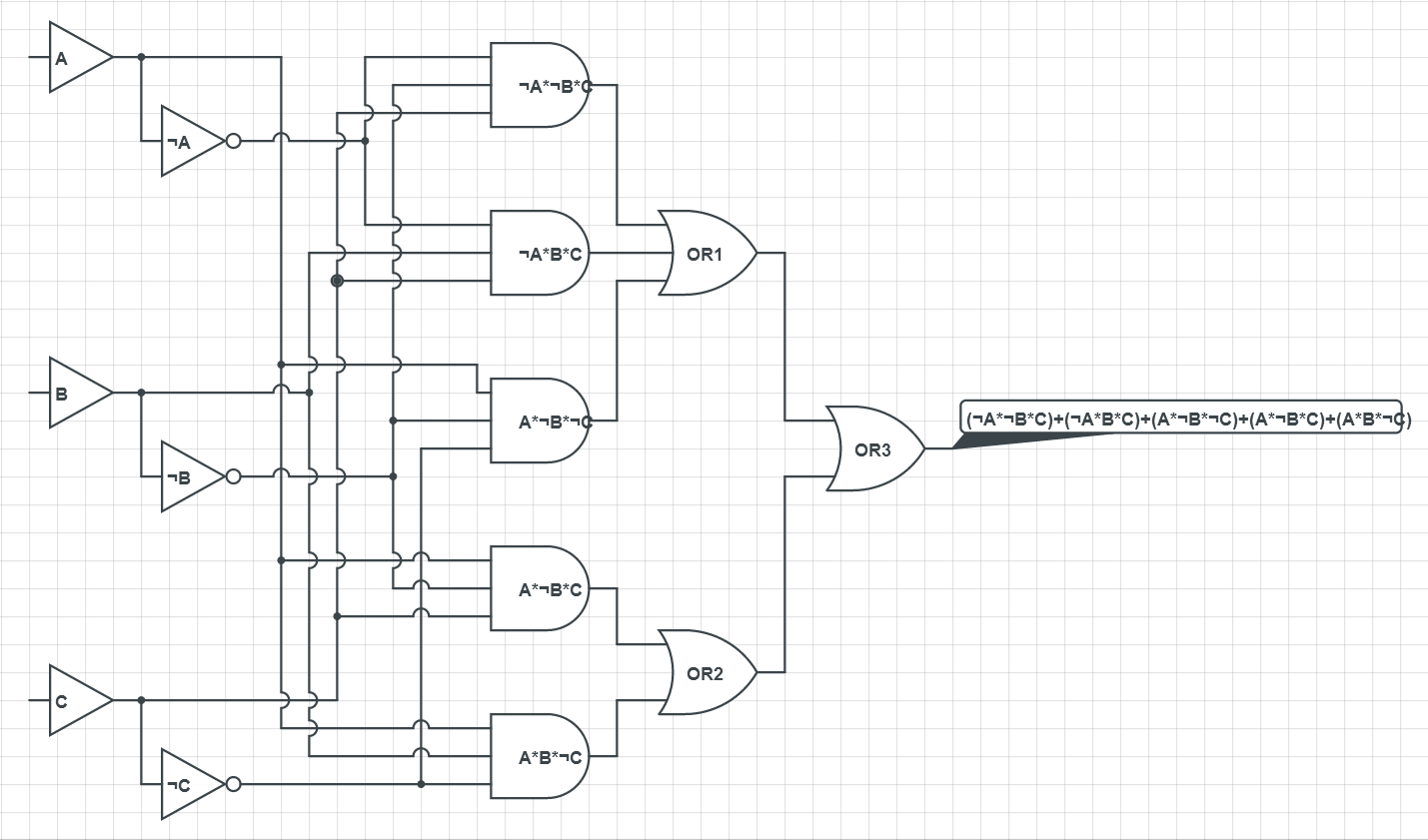
Ford Tang

46564602

ICS 51

Homework 1

1. Translation takes the original code and creates a new program written with the new instructions. After translation, the original code is discarded and the new program is executed. Translation executes faster but is less portable than Interpreting code.  
   For interpreters, each instruction is examined and decoded and executed immediately. Interpreters execute code per instruction making them much more portable, although they execute slower than programs that were translated first.
   1. CPU – Central processing unit. Arithmetic/logic unit, control unit and memory registers combined in a single chip.
   2. PC – Program Counter. Register that points to the next instruction
   3. IR – Instruction Register. Holds the current instruction that is executed.
   4. ALU – Arithmetic/Logic Unit – Performs addition, subtraction and other simple operations.
   5. MAR – Memory Address Register. Works with MDR to map the memory addresses so the registers can use it.
   6. MDR – Memory Data Register. Once MAR places the address of the memory into MDR so the program can write to the memory.
   7. 6 bit instruction + 4 bit register + 4 bit register + 4 bit register = 18 bits
   8. 6 bit instruction + 4 bit register + 25 bit memory address = 35 bits
   9. 97/2 = 48 r 1  
      48/2 = 24 r 0  
      24/2 = 12 r 0  
      12/2 = 6 r 0  
      6/2 = 3 r 0  
      3/2 = 1 r 1  
      ½ = 0 r1  
      92 decimal in binary = 1100001  
        
      97/8 = 12 r 1  
      12/8 = 1 r 4  
      1/8 = 0 r 1  
      92 decimal in octal = 141  
        
      97/16 = 6 r 1  
      6/16 = 0 r 6  
      92 decimal in hexadecimal = 61  
        
      123/2 = 61 r 1  
      61/2 = 30 r 1  
      30/2 = 15 r 0  
      15/2 = 7 r 1  
      7/2 = 3 r 1  
      3/2 = 1 r 1  
      ½ = 0 r 1  
      123 decimal to binary = 1111011  
        
      123/8 = 15 r 3  
      15/8 = 1 r 7  
      1/8 = 0 r 1  
      123 decimal to octal = 173  
        
      123/16 = 7 r 11  
      7/16 = 0 r 7  
      123 in hexadecimal = 7B
   10. 010 = 2  
       110 = 6  
       010 = 2  
       10110010 binary in octal = 262  
         
       0010 =2  
       1011 = B  
       10110010 binary in hexadecimal = B2  
         
       10110010 = 2 + 16 + 32 + 128  
       10110010 binary in decimal = 178  
         
       101 = 5  
       001 = 1  
       001 = 1  
       1001101 binary in octal = 115  
         
       1101 = D  
       0100 = 4  
       1001101 binary in hexadecimal = 4D  
         
       1001101 = 1 + 4 + 8 + 64  
       1001101 binary in decimal = 77
   11. 1101 = D  
       1010 = A  
       1011 = B  
       0010 = 2  
       10101110101101 = 2BAD  
         
       1010 = A  
       1011 = B  
       1100 = C  
       1101 = D  
       1110 = E  
       1111 = F  
       111111101101110010111010 = FEDCBA
       1. X = (¬A Ʌ ¬B Ʌ C) V (¬A Ʌ B Ʌ C) V (A Ʌ ¬B Ʌ ¬C) V (A Ʌ ¬B Ʌ C) V (A Ʌ B Ʌ ¬C)
       2. 
       3. X = (A Ʌ C) V ( A Ʌ B Ʌ ¬C)

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | X |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

* 1. Immediate = Data that is automatically fetched from memory at the same time as instruction is executed.  
     Register = Data that is taken from a register and used in the instruction.  
     Direct = Data that is from a memory address used during instruction.  
     Indirect = Data in the registry is an address to a memory location that contains data.
  2. MOV EAX, [EBX] = moves data that EBX points to into EAX  
     MOV AX, [EBX] = moves the first 16 bits of data that EBX points to into AX  
     MOV AL, [EBX] = moves the first 8 bits of data that EBX points to into AX
     1. 1000 = 0  
        1001 = 255  
        1002 = 255  
        1003 = 255
     2. 1000 = 0  
        1001 = 0  
        1002 = 255  
        1003 = 255
     3. 1000 = 0  
        1001 = 0  
        1002 = 0  
        1003 = 0