Chimera-2016-I Emulator Assignment

Practical 4 - Inc and logic

CANS Tech INC

Implementing the INCA Instruction

Once again inside the Group_1 function switch add

```
case 0x95: // INCA
    CODE HERE
    break;
```

INCA		Addressing	Opcode
Increment Memory or Accumulator		A	0x95
Flags:	T - - - T - - -		
notes			

We simply increment Register A...

```
++Registers[REGISTER_A];
```

And check the N,Z flags...

```
set_flag_n(Registers[REGISTER_A]);
set_flag_z(Registers[REGISTER_A]);
```

Implementing the INX Instruction

Once again inside the Group_1 function switch add

```
case 0x4B: // INX
    CODE HERE
    break;
```

INX		Addressing	Opcode
Increments register X		impl	0x4B
Flags:	T - - - - - - -		
notes			

Just add...

++Index_Registers[REGISTER_X];

Don't forget about the flags as always, which is just the Z flag...

set_flag_z(Index_Registers[REGISTER_X]);

Implementing the AND Instruction

Once again inside the Group_1 function switch add

```
case 0xBC: // AND
    CODE HERE
    break;
```

AND		Addressing	Opcode
Register bitwise and with Accumulator		A-B	0xBC
		A-C	0xCC
Flags:	T - - - T - - -	A-L	0xDC
notes		A-H	0xEC
		A-M	0xFC

Steps...

1. Copy your addiction op code
2. Replace the + with &
3. Remove adding the carry
4. Remove the code that sets the Carry flag

Implementing the CLRA Instruction

Once again inside the Group_1 function switch add

```
case 0x9F: // CLRA
    CODE HERE
    break;
```

CLRA		Addressing	Opcode
Clear Memory or Accumulator		A	0x9F
Flags:	1 - - - 0 - - 0		
notes			

Add...

Registers[REGISTER_A] = 0;

Then add the code to set and clear the relivate flags remembering:
Flags = Flags | flag_to_be_set
Flags = Flags & (0xFF - flag_to_be_cleared)

Compile and run your code to see how many marks you have!

Now you can implement
AND, INC, CLR, DEX, INX, DEY, INY,

Now is a good time to catch up if you find yourself falling behind!

Questions?