

Make a Processor  
11/24/15  
COMP 399

Registers:

A - 000 : General Purpose Register  
B - 001 : General Purpose Register  
C - 010 : General Purpose Register  
D - 011 : General Purpose Register  
Sp - 100 : Stack Pointer  
LC - 101 : Loop Counter  
cmpf - 110 : Comparison flag – greater = 1, equal = 0, less = -1  
mema - 111 : Holds a memory address

Instructions:	encoding,	register,	address,	insignificant bit	
PUTC	constant, register	0000	0rrr	cccc	cccc
Puts the given constant (0-255) into specified register.					
COPY	source register, destination register	0001	0000	0rrr	0rrr
Copies contents of first register to the second register.					
SWAP	register, register	0010	0000	0rrr	0rrr
Swaps the values of the given registers.					
ADD	register1, register2 → register1	0011	0000	0rrr	0rrr
Adds the value in both registers and stores result in the first.					
SUB	register1, register2 → register1	0100	0000	0rrr	0rrr
Subtracts value in register2 from register1 and stores result in register1					
INC	register	0101	0rrr	0000	0000
Increments the value in the specified register.					
DEC	register	0110	0rrr	0000	0000
Decrements the value in the specified register.					
CMP	register1, register2 → cmpf	0111	0000	0rrr	0rrr
Compares register1 to register2: 1-greater, 0-equal, -1-less.					
JMP	address	1000	0000	aaaa	aaaa
Unconditionally jumps to given address					
JMPG	address	1001	0000	aaaa	aaaa
Jumps to given address if value in cmpf is 1.					
JMPL	address	1010	0000	aaaa	aaaa
Jumps to address if value in cmpf is -1.					
JMPE	address	1011	0000	aaaa	aaaa
Jumps to address if value in cmpf is 0.					
JNE	address	1100	0000	aaaa	aaaa
Jumps to address if value in cmpf is not 0.					
SAVE	register, address	1101	0rrr	aaaa	aaaa
Saves value in given register at address.					
OUT	address	1110	0000	aaaa	aaaa
Outputs the value at given memory address.					
NOP		1111	0000	aaaa	aaaa

C Code:

```

char x, n, i;
n=5;

x=0;
for (i=0; i!=n; i++)
    x=x+i;    //at the end, x = 1+2+3+4+5 = 15 (or hex 0f)
while(true); //program halts here

```

#### Instruction Set:

Address		Instruction	Arguments	Byte Code				Hex Bytecode	
binary	hex								
0000	0000	00	PUTC	5, A	0000	0000	0000	0101	00 05
0000	0010	02	PUTC	0, B	0000	0001	0000	0000	01 00
0000	0100	04	PUTC	0, LC	0000	0101	0000	0000	05 00
0000	0110	06	CMP	LC, A	0111	0000	0101	0000	70 50
0000	1000	08	JMPE	10	1011	0000	0001	0000	b0 10
0000	1010	0A	ADD	B, LC	0011	0000	0001	0101	30 15
0000	1100	0C	INC	LC	0101	0101	0000	0000	55 00
0000	1110	0E	JMP	06	1000	0000	0000	0110	80 06
0001	0000	10	SAVE	B, 0xA0	1101	0001	1010	0000	d1 a0
0001	0010	12	OUT	0xA0	1110	0000	1010	0000	e0 a0
0001	0100	14	JMP	14	1000	0000	0001	0100	80 14