	Cod. Operação	Reg. a	Reg. b	Reg. c	Const
	4bits	4bits	4bits	4bits	16 bits
NOP	0000	0000	0000	0000	0000
Add	0001	0010	0011	0001	0000
Sub	0010	0011	0100	0010	
And	0011				
Or	0100				
Xor	0101				
Not	0110				
Sll	0111				
Srl	1000				
Ori	1001				
Xori	1010				
Addi	1011				
Show	1100				
Jump	1101				
Branch	1110				
Halt	1111				

OPCODE	
NOP	0000
Add	0001
Sub	0010
And	0011
Or	0100
Xor	0101
Not	0110
Sll	0111
Srl	1000
Ori	1001
Xori	1010
Addi	1011
Show	1100
Jump	1101
Branch	1110
Halt	1111

Tabela de Memória CONTROLE

	Write	ula	branch	IP_mux	halt	SelB	Show	**Bin**	Hex
NOP	0	000	0	0	0	0	0	000000000	0
Add	1	000	0	0	0	0	0	100000000	100
Sub	1	001	0	0	0	0	0	100100000	120
And	1	010	0	0	0	0	0	101000000	140
Or	1	011	0	0	0	0	0	101100000	352
Xor	1	100	0	0	0	0	0	110000000	180
Not	1	101	0	0	0	0	0	110100000	1A0
Sll	1	110	0	0	0	0	0	111000000	1C0
Srl	1	111	0	0	0	0	0	111100000	1E0
Ori	1	011	0	0	0	1	0	101100010	162
Xori	1	100	0	0	0	1	0	110000010	182
Addi	1	000	0	0	0	1	0	100000010	102
Show	0	000	0	0	0	0	1	000000001	1
Jump	0	000	0	1	0	0	0	000001000	8
Branch	1	001	1	1	0	0	0	100111000	138
Halt	0	000	0	1	1	0	0	000001100	C

#assembly code	# r0 = 0000	# r6 = 0110
# N = r1	# r1 = 0001	# r7 = 0111
# first = r2	# r2 = 0010	# r8 = 1000
# next = r3	# r3 = 0011	# r9 = 1001
# second = r4	# r4 = 0100	
# i = r5	# r5 = 0101	

Tabela de Memória CONTROLE

	OP CODE dez	Binario					Hex	Assembly	
		Código da oper	Registrador a	Registrador b	Registrador c	Constante			
	SOMA SIMPLES 3+2								
	NOP	0000	0000	0000	0000	0000000000000000	0		
	ADDi	1011	0000	0000	0001	0000000000000011	B0010003		
	ADDi	1011	0000	0000	0010	0000000000000010	B0020002		
	ADD	0001	0001	0010	0011	0000000000000000	11230000		
	SHOW	1100	0011	0000	0000	0000000000000000	C3000000		
	Fibi.asm								
	OP CODE dez	Binario					Hex	Assembly	
1	SHOW	1100	0000	0000	0000	0000000000000000	C0000000	inicializa o show	
2	ADDi	1011	0000	0000	0001	0000000000000000	B0010000	addi r0 r0 r1 10	#Defina N como o número desejado de termos da sequência Fibonacci
3	ADDi	1011	0000	0000	0010	0000000000000000	B0020000	addi r0 r0 r2 0	#Inicializa first
4	ADDi	1011	0000	0000	0011	0000000000000000	B0030000	addi r0 r0 r2 0	#Inicializa next
5	ADDi	1011	0000	0000	0100	0000000000000001	B0040001	addi r0 r0 r4 1	#Inicializa Second
6	ADD	0001	0001	0000	0101	0000000000000000	11050000	add r1 r0 r5	#Inicializa i= N
7	BRANCH	1110	0000	0101	0000	pula linha 14			
8	SHOW	1100	0000	0000	0000	0000000000000000	C0000000	show r1	#Show N
9	ADD	0001	0010	0100	0011	0000000000000000	12430000	add r2 r4 r3	#next = first + second;
10	ADD	0001	0000	0100	0010	0000000000000000	10420000	add r0 r4 r2	# first = second;
11	ADD	0001	0000	0011	0100	0000000000000000	10340000	add r0 r3 r4	#second = next;
12	SUB	0010	0000	0000	0101	0000000000000001	20050001	subi r0 r0 r5 -1	# i = i-1;
13	JUMP	1101	0000	0000	0000	pula linha 7			
14	HALT	1111	0000	0000	0000	0000000000000000	F0000000		