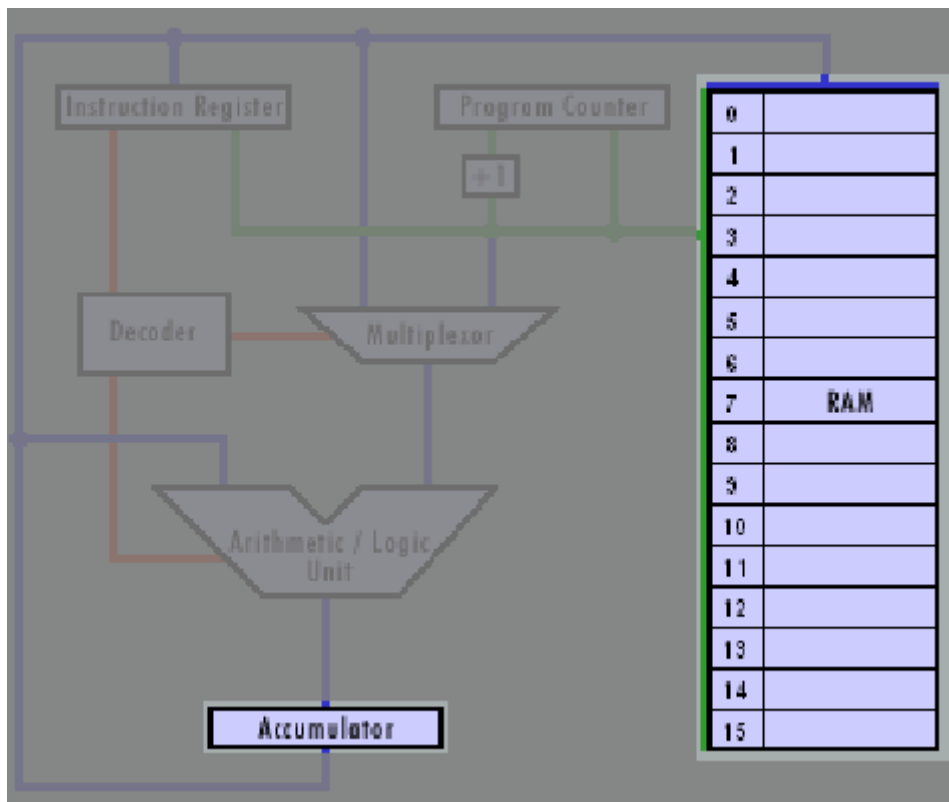
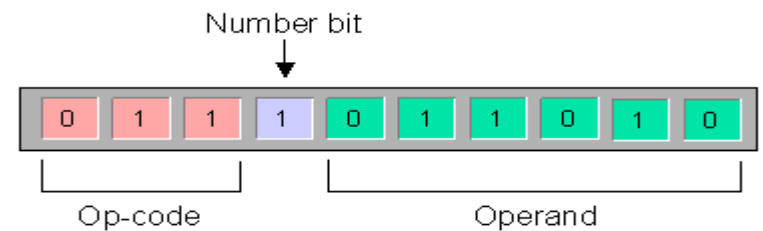


<i>Op-code</i>	<i>Mnemonic</i>	<i>Function</i>	<i>Example</i>
001	LOAD	Load the value of the operand into the Accumulator	LOAD 10
010	STORE	Store the value of the Accumulator at the address specified by the operand	STORE 8
011	ADD	Add the value of the operand to the Accumulator	ADD #5
100	SUB	Subtract the value of the operand from the Accumulator	SUB #1
101	EQUAL	If the value of the operand equals the value of the Accumulator, skip the next instruction	EQUAL #20
110	JUMP	Jump to a specified instruction by setting the Program Counter to the value of the operand	JUMP 6
111	HALT	Stop execution	HALT



<i>Op-code</i>	<i>Mnemonic</i>	<i>Function</i>	<i>Example</i>
001	LOAD	Load the value of the operand into the Accumulator	LOAD 10
010	STORE	Store the value of the Accumulator at the address specified by the operand	STORE 8
011	ADD	Add the value of the operand to the Accumulator	ADD #5
100	SUB	Subtract the value of the operand from the Accumulator	SUB #1
101	EQUAL	If the value of the operand equals the value of the Accumulator, skip the next instruction	EQUAL #20
110	JUMP	Jump to a specified instruction by setting the Program Counter to the value of the operand	JUMP 6
111	HALT	Stop execution	HALT

Op-code	Mnemonic	Function	Example
001	LOAD	Load the value of the operand into the Accumulator.	LOAD 10
010	STORE	Store the value of the Accumulator at the address specified by the operand	STORE 8
011	ADD	Add the value of the operand to the Accumulator	ADD #5
100	SUB	Subtract the value of the operand from the Accumulator	SUB #1
101	EQUAL	If the value of the operand equals the value of the Accumulator, skip the next instruction	EQUAL #20
110	JUMP	Jump to a specified instruction by setting the Program Counter to the value of the operand	JUMP 6
111	HALT	Stop execution	HALT



#	Machine code	Assembly code	Description
0	001 1 000010	LOAD #2	Load the value 2 into the Accumulator
1	010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
2	001 1 000101	LOAD #5	Load the value 5 into the Accumulator
3	010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
4	001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
5	011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
6	010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
7	111 0 000000	HALT	Stop execution

# Sua Missão

- Desenvolver um emulador para esta arquitetura! (2,0)
- Pontos extras:
  - Leitura do programa no formato intel HEX (1,0)
  - Desenvolvimento de um montador
    - Transforma código no formato intel HEX (1,0)
- Detalhes do formato intel HEX
  - [http://en.wikipedia.org/wiki/Intel\\_HEX](http://en.wikipedia.org/wiki/Intel_HEX)

#	Machine code	Assembly code	Description
0	001 1 000101	LOAD #5	These two operations set the count value to five
1	010 0 001111	STORE 15	
2	001 1 000000	LOAD #0	Initialize the count to zero
3	101 0 001111	EQUAL 15	Test to see if count is complete; if yes, skip next instruction and go to instruction 5; if no, go to next instruction
4	110 1 000110	JUMP #6	Set Program Counter to 6
5	111 0 000000	HALT	Stop execution
6	011 1 000001	ADD #1	Increment the count in the Accumulator
7	110 1 000011	JUMP #3	Set Program Count to 3

Contador  
1-> 5

#	Machine code	Assembly code	Description
0	001 1 000010	LOAD #2	Load the value 2 into the Accumulator
1	010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
2	001 1 000101	LOAD #5	Load the value 5 into the Accumulator
3	010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
4	001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
5	011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
6	010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
7	111 0 000000	HALT	Stop execution

X = 2  
Y = 5  
X + Y = Z