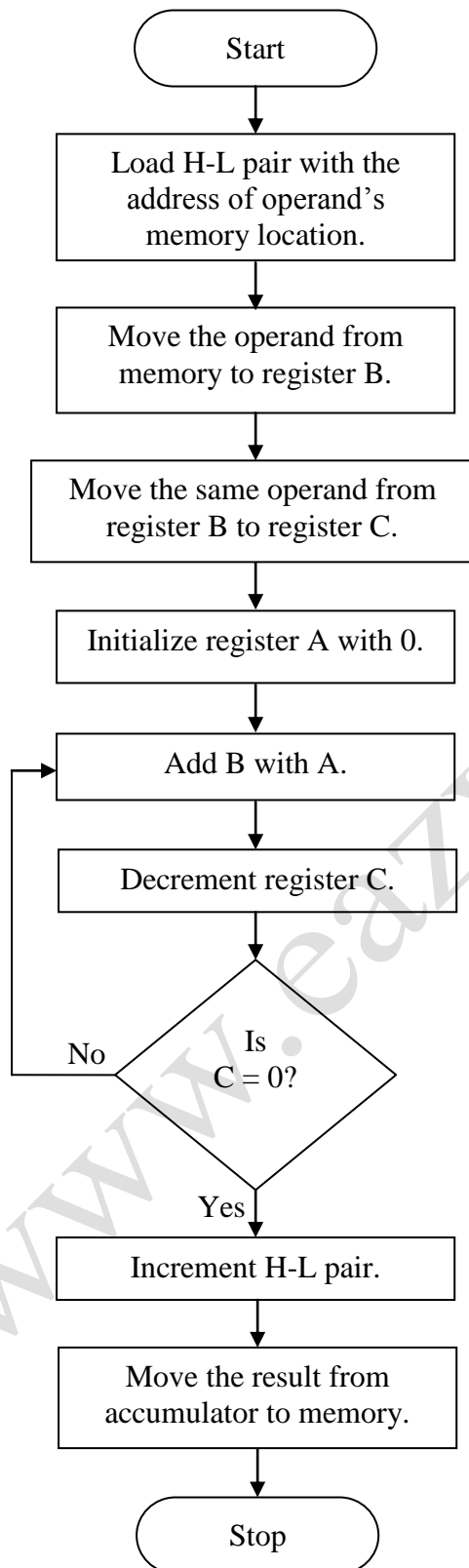


**Program 17:** Find square of an 8-bit number.**Flowchart:**

**Program:**

Address	Mnemonics	Operand	Opcode	Remarks
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	MOV	B, M	46	Move the operand from memory to reg. B.
2004	MOV	C, M	4E	Move the same number from reg. B to reg. C.
2005	MVI	A, 00H	3E	Initialize accumulator with 00H.
2006			00	Immediate value 00H.
2007	ADD	B	80	Add B with A.
2008	DCR	C	0D	Decrement reg. C (counter).
2009	JNZ	2007H	C2	Jump back to address 2007H if C $\neq$ 0.
200A			07	Lower-order of 2007H.
200B			20	Higher-order of 2007H.
200C	INX	H	23	Increment H-L pair.
200D	MOV	M, A	77	Move the result from accumulator to memory.
200E	HLT		76	Halt.

**Explanation:**

- This program finds the square of an 8-bit number stored in memory location 3000H.
- The square of a number is found by multiplying it by itself.
- Therefore, the number is added with itself and is also used as counter.
- Let us assume that the operands stored at memory location 3000H is 03H.
- Then, by using successive addition method, we get  $03H + 03H + 03H = 09H$ .
- Initially, H-L pair is loaded with the address of the operand.
- The operand is moved to register B from memory location 3000H and then it is copied to register C.
- Accumulator is initialized to 00H.
- Register B is added with accumulator and the result is stored in accumulator.
- Register C (counter) is decremented by 1.
- Then, counter is checked for zero. If it hasn't become zero yet, then register B is again added with accumulator, and counter is again checked for zero.
- If counter becomes zero, then H-L pair is incremented and the result is moved from accumulator to memory location 3001H.

**Output:**

**Before Execution:**

3000H: 03H

**After Execution:**

3001H: 09H

*www.eazynotes.com*