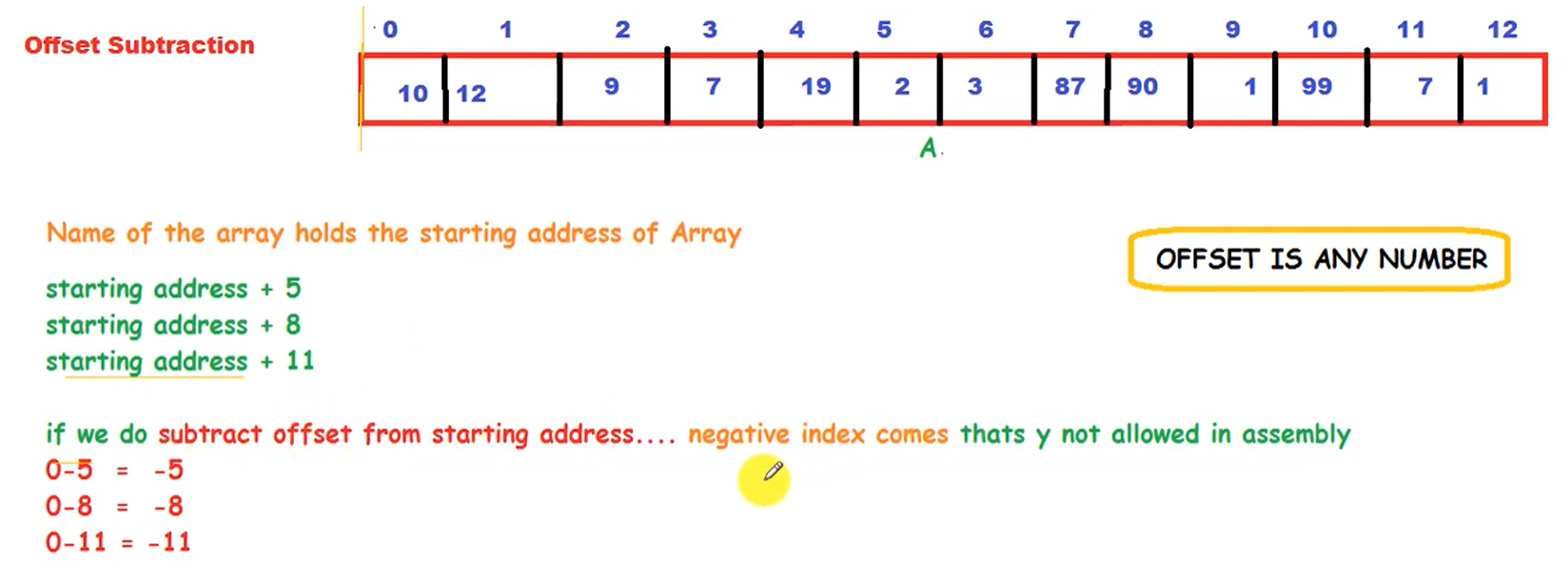


A screenshot of a computer

Description automatically generated



**Addressing Modes**

Addressing modes in assembly language specify how the operand of an instruction is determined. Different modes allow flexibility in how data is accessed or manipulated. There are **four registers** in iAPX88 architecture that can hold address of data, and they are **BX, BP, SI, and DI**.

**1. Immediate Addressing Mode**

* **Syntax**: MOV AX, 5
* The operand is a constant or immediate value.
* **Example**: MOV AX, 0x1234 moves the immediate value 0x1234 directly into the AX register.
* **Usage**: For loading constants into registers.

**2. Register Addressing Mode**

* **Syntax**: MOV AX, BX
* The operand is a register, and data is transferred between registers.
* **Example**: MOV AX, BX moves the content of register BX into AX.
* **Usage**: For operations involving registers directly.

**3. Direct Addressing Mode**

* **Syntax**: MOV AX, [1234H] **or** MOV AX, [num1]
* The source operand is a memory address, and the instruction accesses data stored at that address.
* **Example**: MOV AX, [1234H] moves the value stored at memory address 1234H into AX.
* **Usage**: For reading or writing specific memory locations.

**4. Direct Addressing with Displacement (Offset) Addressing Mode**

* **Syntax:** MOV AX, [num+2]
* num is a label or constant that represents a memory address.
* num + 2 means that the processor will access the memory location starting at num, but with a displacement of 2 bytes added to the base address.
* The data at this new address (num + 2) will be moved into AX.

**5. Based Register Indirect Addressing Mode**

* **Syntax**: MOV AX, [BX] **or** MOV [BP], AL
* The memory address is specified indirectly through a register. BX is allowed to hold an address but BL or BH are not. Address is 16bit and must be contained in a 16bit register.
* **Example**: MOV AX, [BX] moves the value stored at the memory address in BX into AX.
* **Usage**: Allows for dynamic memory access where the address is stored in a register.
* We used the BX register for it and the B in BX and BP stands for base therefore we call register indirect memory access using BX or BP, “based addressing.”
* When SI or DI is used we name the method “indexed addressing.”

**6. Based Register Indirect + Offset Mode**

* **Syntax:** ADD AX, [BX+2]
* **Base Register:** BX holds a memory address.
* **Offset/Displacement:** +2 is the offset added to the contents of BX.
* **Effective Address:** The sum of BX and the offset 2 forms the final address where the operand is located.
* The value stored at this effective address is then added to the AX register.