BRANCH OPERATIONS OF 8051 (SJMP, AJMP AND L JMP)

SHORT JUMP

Syntax: **SJMP radd**; // Short Jump using the relative address

Range: **(-128 … +127) locations** because "radd" is an 8-bit signed number

Size of instruction: **2 Bytes** (Opcode of SJMP= 1Byte, radd = 1Byte)

New address calculation: **PC ← PC (address of next instruction) + radd**

Usage: **SJMP** (unconditional) and ALL Conditional jumps like JC, JNC, CJNE, DJNZ etc. ← Important for VIVA

Description: *Here the branch address (radd) is calculated as a relative distance from the next instruction to the branch location. In simple terms, instead of telling where we want to jump, we are telling how far we want to jump. This "radd" is then added to PC, which normally contains address of the next instruction.* ***For examples of SJMP, please refer #BharatSir Lecture Notes***

ABSOLUTE JUMP

Syntax: **AJMP sadd**;// Absolute Jump using the short address

Range: **max 2KB** as long as the Jump is within the **Same Page**

Size of instruction: **2 Bytes** (Opcode of AJMP= 1Byte, sadd = 1Byte)

New address calculation:

PC ← (16)	PC	Opcode of AJMP	Sadd
	5 bits Remains the same as branch is in the same page	3 bits Hence AJMP has 8 opcodes	8 bits Lower 8 bits of the jump location

Usage: **AJMP and ACALL**.

Description: *Here the entire program memory (64 KB), is divided into 32 pages, each page being of 2KB. We can jump to any location of the same page, giving a max range of 2 KB. As the jump is in the same page, only the lower 11 bits of the address will change. Out of them, lower 8 bits are given by "sadd" and the higher 3 bits are given by the opcode of AJMP. 3 bits have 8 combinations, hence AJMP has 8 opcodes.* ***For examples of AJMP, please refer #BharatSir Lecture Notes***

LONG JUMP

Syntax: **LJMP ladd**; // Long Jump using the long (full) address

Range: **64 KB** because "ladd" is a 16-bit address so can be any value from 0000H… FFFFH.

Size of instruction: **3 Bytes** (Opcode of LJMP= 1Byte, ladd = 2Bytes)

New address calculation: **PC ← ladd**

Usage: **LJMP, LCALL.**

Description: *This is the simplest type of Jump. Here we simply give the address where we wish to jump using "ladd". This "ladd" is then simply put into PC.* ***For examples of LJMP, please refer #BharatSir Lecture Notes***