## 8-bit Addition

Write a program to add 8 bit variables A and B together, and storing the sum into 8 bit variable C. For this programming problem you may assume that the sum is less than 255 if A and B are unsigned and between -128 and 127 if signed.

C = A + B

Simulation of the problem C = 170 + 39, where the answer C should equal 209 (0xD1), or the signed problem C = -86 + 39, where the answer should equal -47 (0xD1). Neither solution results in a carry (unsigned) overflow (signed) condition.

			Watch						
0. b4	Adder 88		Name	Value	Туре	Location			
¢	lds	r0,A r2,B r0,r2 C,r0 Adder88	A	OxAA '-'	SRAM Locat:	0x0100	[SRAM]		
	lds		В	0x27 '''	SRAM Locat:	0x0101	[SRAM]		
	add		С	0x00 ''	SRAM Locat:	0x0102	[SRAM]		
	rjmp								

Figure 1 Start of 8-bit Addition program with variable A initialized to 0xAA (170<sub>10</sub> unsigned or  $-86_{10}$  signed) and B initialized to 0x27 (39<sub>10</sub> signed or unsigned).

			Watch					
.CSEG			Name	Value	Туре	Location		
	lds lds add sts rjmp	r0,A r2,B r0,r2 C,r0 Adder88	A	OxAA '*'	SRAM Locat:	0x0100	[SRAM]	
			B	0x27 '''	SRAM Locat:	0x0101	[SRAM]	
			с	0xD1 'Ñ'	SRAM Locat:	0x0102	[SRAM]	
<b>→</b> :								

Figure 2 End of Addition program with variable C containing 0xD1 (20910 unsigned or -4710)