

Unsigned 8-bit Addition with 16-bit result

Write a program to find the unsigned sum of 8 bit variables A and B. For this programming problem the sum may be greater than 255 if A and B. Store the sum into 16 bit variable C using little endian byte ordering.

$$C = A + B$$

Simulation of the unsigned problem $C = 255 + 85$, where the answer C should equal 340 (0x154). The solution results in a carry condition after the first addition.

The screenshot shows the start of the program. The assembly code is as follows:

```

C:      .BYTE 2
.CSEG
Adder816:
; load
clr r1
lds r0,A
clr r3
lds r2,B
; add
add r0,r2
adc r1,r3
; store
sts C,r0

```

The Watch window displays:

Name	Value	Type	Location
A	0xFF 'ÿ'	SRAM Locat:	0x0100 [SRAM]
B	0x55 'U'	SRAM Locat:	0x0101 [SRAM]
C	0x00 ''	SRAM Locat:	0x0102 [SRAM]

The Memory window shows the initial state of memory:

Address	Value
000100	FF 55 00 00 00 00 00 00 00 00 00 00 00 00 00
000118	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Figure 1 Start of 8-bit Addition program with variable A initialized to 0xFF (255₁₀ unsigned) and B initialized to 0x55 (85₁₀ signed).

The screenshot shows the end of the program. The assembly code is as follows:

```

C:      .BYTE 2
.CSEG
Adder816:
; load
clr r1
lds r0,A
clr r3
lds r2,B
; add
add r0,r2
adc r1,r3
; store
sts C,r0
sts C+1,r1
rjmp Adder816

```

The Watch window displays:

Name	Value	Type	Location
A	0xFF 'ÿ'	SRAM Locat:	0x0100 [SRAM]
B	0x55 'U'	SRAM Locat:	0x0101 [SRAM]
C	0x54 'T'	SRAM Locat:	0x0102 [SRAM]

The Memory window shows the final state of memory:

Address	Value
000100	FF 55 54 01 00 00 00 00 00 00 00 00 00 00 00
000118	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000148	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Figure 2 End of Addition program with variable C containing 0x0154 (340₁₀). Byte ordering is little endian.