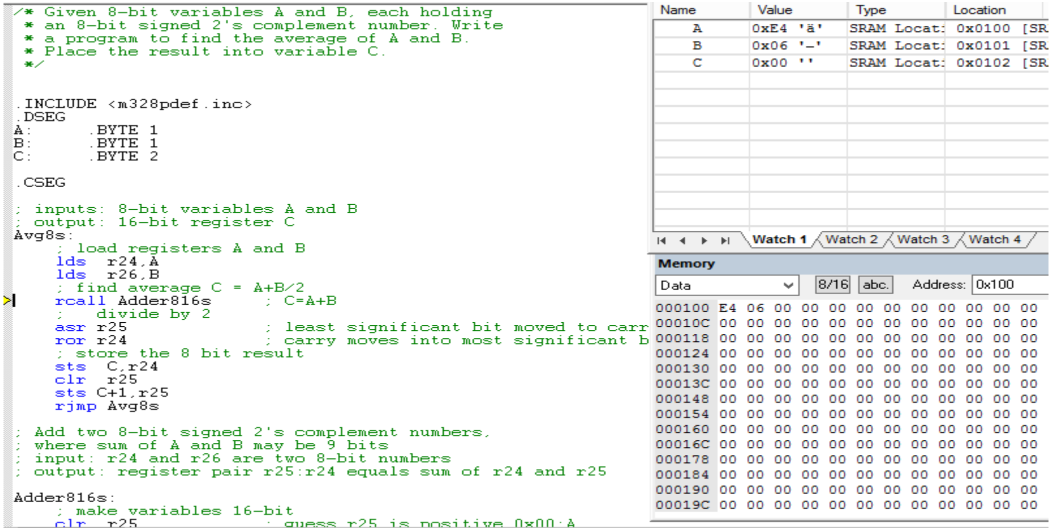
Avg8s: Simulation



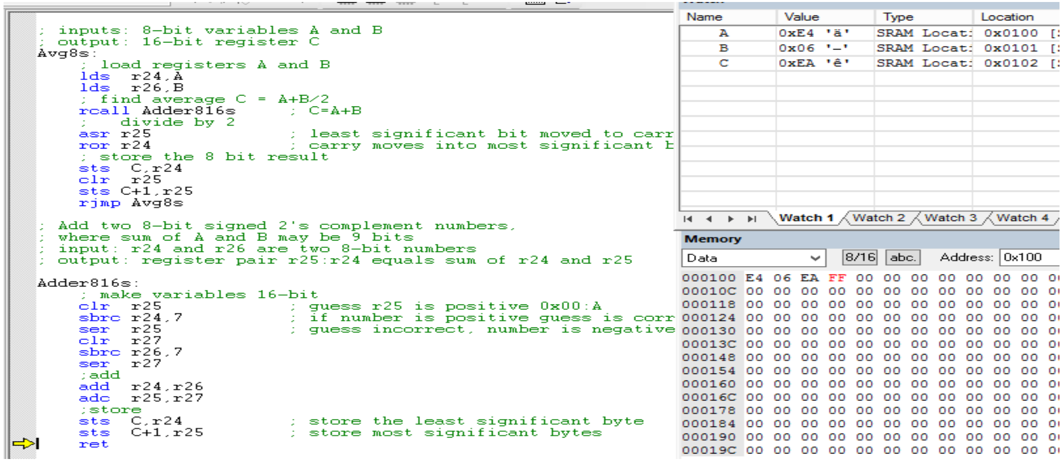
```
/* Given 8-bit variables A and B, each holding
 * an 8-bit signed 2's complement number. Write
 * a program to find the average of A and B.
 * Place the result into variable C.
 */


.INCLUDE <m328pdef.inc>
.DSEG
A:      .BYTE 1
B:      .BYTE 1
C:      .BYTE 2

.CSEG

; inputs: 8-bit variables A and B
; output: 16-bit register C
Avg8s:
    ; load registers A and B
    lds  r24,A
    lds  r26,B
    ; find average C = A+B/2
    rcall Adder816s      ; C=A+B
    ;   divide by 2
    asr r25              ; least significant bit moved to carr
    ror r24              ; carry moves into most significant b
    ; store the 8 bit result
    sts  C,r24
    clr  r25
    sts C+1,r25
    rjmp Avg8s

; Add two 8-bit signed 2's complement numbers,
; where sum of A and B may be 9 bits
; input: r24 and r26 are two 8-bit numbers
; output: register pair r25:r24 equals sum of r24 and r25

Adder816s:
    ; make variables 16-bit
    clr  r25             ; guess r25 is positive 0x00:A
```

Name	Value	Type	Location
A	0xE4 'ä'	SRAM Locat:	0x0100 [SR
B	0x06 '–'	SRAM Locat:	0x0101 [SR
C	0x00 ''	SRAM Locat:	0x0102 [SR

Address: 0x100

```
000100 E4 06 00 00 00 00 00 00 00 00 00 00
00010C 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
000124 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
00013C 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
000154 00 00 00 00 00 00 00 00 00 00 00 00
000160 00 00 00 00 00 00 00 00 00 00 00 00
00016C 00 00 00 00 00 00 00 00 00 00 00 00
000178 00 00 00 00 00 00 00 00 00 00 00 00
000184 00 00 00 00 00 00 00 00 00 00 00 00
000190 00 00 00 00 00 00 00 00 00 00 00 00
00019C 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 1.Variable A is initialized to 0xE4 (or -28) and variable B initialized to 0x06. The program calls the Adder816s subroutine to convert A and B to 16-bit signed numbers and then add them, since (2C=A+B).



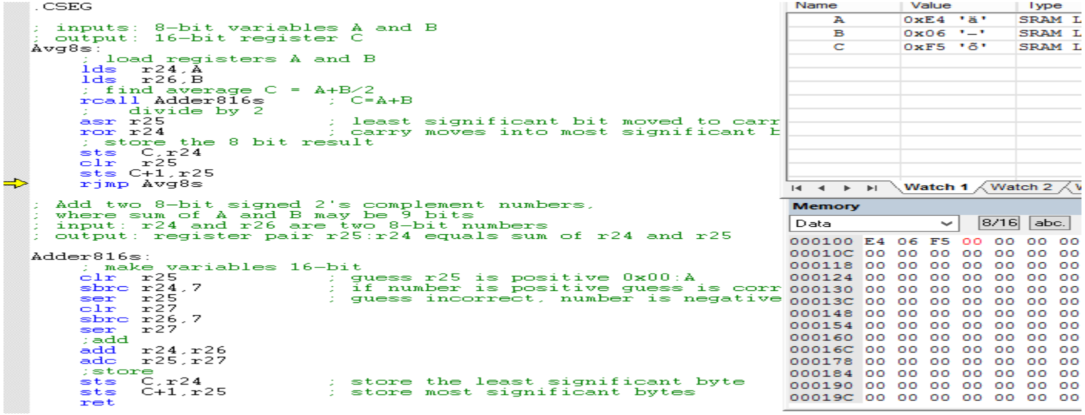
```
; inputs: 8-bit variables A and B
; output: 16-bit register C
Avg8s:
    ; load registers A and B
    lds  r24,A
    lds  r26,B
    ; find average C = A+B/2
    rcall Adder816s      ; C=A+B
    ;   divide by 2
    asr r25              ; least significant bit moved to carr
    ror r24              ; carry moves into most significant b
    ; store the 8 bit result
    sts  C,r24
    clr  r25
    sts C+1,r25
    rjmp Avg8s

; Add two 8-bit signed 2's complement numbers,
; where sum of A and B may be 9 bits
; input: r24 and r26 are two 8-bit numbers
; output: register pair r25:r24 equals sum of r24 and r25

Adder816s:
    ; make variables 16-bit
    clr  r25             ; guess r25 is positive 0x00:A
    sbrc r24,7           ; if number is positive guess is corr
    ser  r25             ; guess incorrect, number is negative
    clr  r27
    sbrc r26,7
    ser  r27
    ;add
    add  r24,r26
    adc  r25,r27
    ;store
    sts  C,r24           ; store the least significant byte
    sts  C+1,r25         ; store most significant bytes
    ret
```

Name	Value	Type	Location
A	0xE4 'ä'	SRAM Locat:	0x0100 [:
B	0x06 '–'	SRAM Locat:	0x0101 [:
C	0xEA 'ê'	SRAM Locat:	0x0102 [:

Address: 0x100

```
000100 E4 06 EA FF 00 00 00 00 00 00 00 0
00010C 00 00 00 00 00 00 00 00 00 00 00 0
000118 00 00 00 00 00 00 00 00 00 00 00 0
000124 00 00 00 00 00 00 00 00 00 00 00 0
000130 00 00 00 00 00 00 00 00 00 00 00 0
00013C 00 00 00 00 00 00 00 00 00 00 00 0
000148 00 00 00 00 00 00 00 00 00 00 00 0
000154 00 00 00 00 00 00 00 00 00 00 00 0
000160 00 00 00 00 00 00 00 00 00 00 00 0
00016C 00 00 00 00 00 00 00 00 00 00 00 0
000178 00 00 00 00 00 00 00 00 00 00 00 0
000184 00 00 00 00 00 00 00 00 00 00 00 0
000190 00 00 00 00 00 00 00 00 00 00 00 0
00019C 00 00 00 00 00 00 00 00 00 00 00 0
```

Figure 2. The 16-bit sum is equal to 0xFFEA (or -22), which is stored in variable C, before returning to the main program.



```
.CSEG

; inputs: 8-bit variables A and B
; output: 16-bit register C
Avg8s:
    ; load registers A and B
    lds  r24,A
    lds  r26,B
    ; find average C = A+B/2
    rcall Adder816s      ; C=A+B
    ;   divide by 2
    asr r25              ; least significant bit moved to carr
    ror r24              ; carry moves into most significant b
    ; store the 8 bit result
    sts  C,r24
    clr  r25
    sts C+1,r25
    rjmp Avg8s

; Add two 8-bit signed 2's complement numbers,
; where sum of A and B may be 9 bits
; input: r24 and r26 are two 8-bit numbers
; output: register pair r25:r24 equals sum of r24 and r25

Adder816s:
    ; make variables 16-bit
    clr  r25             ; guess r25 is positive 0x00:A
    sbrc r24,7           ; if number is positive guess is corr
    ser  r25             ; guess incorrect, number is negative
    clr  r27
    sbrc r26,7
    ser  r27
    ;add
    add  r24,r26
    adc  r25,r27
    ;store
    sts  C,r24           ; store the least significant byte
    sts  C+1,r25         ; store most significant bytes
    ret
```

Name	Value	Type
A	0xE4 'ä'	SRAM L
B	0x06 '–'	SRAM L
C	0xF5 'õ'	SRAM L

```
000100 E4 06 F5 00 00 00 00
00010C 00 00 00 00 00 00 00
000118 00 00 00 00 00 00 00
000124 00 00 00 00 00 00 00
000130 00 00 00 00 00 00 00
00013C 00 00 00 00 00 00 00
000148 00 00 00 00 00 00 00
000154 00 00 00 00 00 00 00
000160 00 00 00 00 00 00 00
00016C 00 00 00 00 00 00 00
000178 00 00 00 00 00 00 00
000184 00 00 00 00 00 00 00
000190 00 00 00 00 00 00 00
00019C 00 00 00 00 00 00 00
```

Figure 3. Since C = (A + B)/2, the bits of the return variable C, are rotated to the right (which divides the number by two). When shifted, the least significant bit from the most significant byte falls off, and then rotates into the MSB of the 8-bit variable holding the result. The result is 0xF5 (or -11).