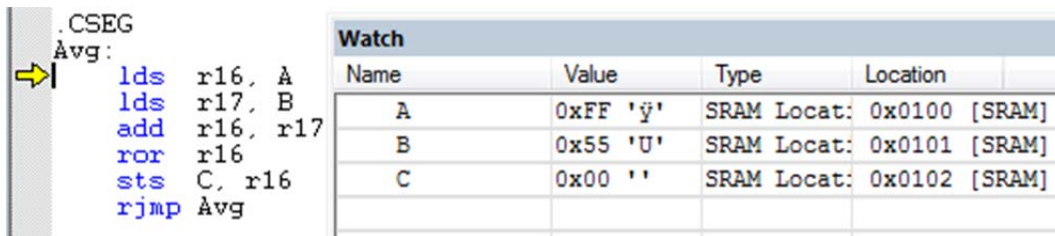


Unsigned 8-bit Average

Given 8-bit variables A and B, each holding an 8-bit unsigned 2's complement number, write a program to find the average of A and B and put the result into variable C.

Hint: Shifting (or rotating) a binary number to the left divides the number by 2.

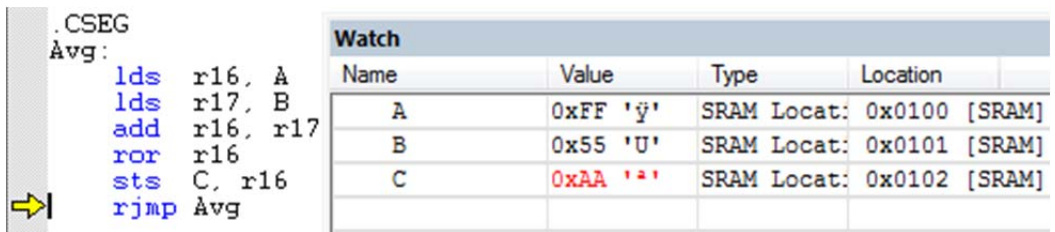
Simulation of the unsigned problem $C = 255 + 85$, where the answer should equal 170 ($0xAA$).



```
.CSEG
Avg:
→ lds r16, A
   lds r17, B
   add r16, r17
   ror r16
   sts C, r16
   rjmp Avg
```

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

Figure 1 Start of unsigned 8-bit Average program with variable A initialized to $0xFF$ (255_{10}) and B initialized to $0x55$ (85_{10}).



```
.CSEG
Avg:
   lds r16, A
   lds r17, B
   add r16, r17
   ror r16
   sts C, r16
→ rjmp Avg
```

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

Figure 2 End of unsigned 8-bit Average program with variable C containing $0xAA$ (170_{10}).