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. */ Name Value Туре Location 0xE4 'ä' SRAM Locat: 0x0100 [SR SRAM Locat: 0x0101 [SR A B 0x06 '-' 0x00 '' C SRAM Locat: 0x0102 [SR .INCLUDE <m328pdef.inc> DSEG BYTE 1 BYTE 1 BYTE 2 A: B: C: CSEG .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 rccall Adder816s : C=A+B : divide by 2 : divide by 2 : sorre the 8 bit result : sts C,r24 clr r25 clr r25 clr r25 clr r25 cli r25 IA A D N Watch 1 Watch 2 Watch 3 Watch 4 Memory ✓ 8/16 abc. Address: 0x100 Data
 Cold
 Early
 Carcy
 <thC >
 001148
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 <th 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 ; and r25; and

Figure 1.Variable A is initialized to $0 \times E4$ (or -28) and variable B initialized to 0x06. The program calls the Adder816s subroutine to convert A and B to 16bit signed numbers and then add them, since (2C=A+B).

	Name		Valu	e		Туре	e		Loc	ation	
; inputs: 8-bit variables A and B	A		0xE4	'ä	•	SRAN	1 Lo	cat:	0x	0100	[:
; output: 16-bit register C	В		0x06	• •	•	SRAN	1 Lo	cat:	0x	0101	E:
Avgos: . load registers A and B	с		OXEZ	•ê	•	SRAN	1 Lo	cat	0x	0102	Ē.
lds r24.A											
lds r26,B											
; find average $C = A+B/2$											
rcall Adder816s ; C=A+B											
; divide by 2											
ror r24 carry moves into most significant h											
; store the 8 bit result											
sts C,r24											
clr r25											
sts C+1, r25											
tjmb wodos	L						1		- 1-		
: Add two 8-bit signed 2's complement numbers.	14 4 F	M 7	Wate	<u>n 1</u>	(Wat	tch 2	<u>∧</u> <u>w</u>	atch :	<u>3 ⁄ v</u>	Vatch	4
whome own of A and P naw be 9 bits											
, where sum of a and b may be ? Dits	Memory										
; input: r24 and r26 are two 8-bit numbers	Memory				9/16			Adde		0100	
; input: r24 and r26 are two 8-bits ; output: r24 and r26 are two 8-bit numbers ; output: register pair r25:r24 equals sum of r24 and r25	Data			- [8/16	abo	∍.	Addr	ess: [0x100	1
, where sufficient and the measure bits numbers ; input: r24 and r26 are two 8-bit numbers ; output: register pair r25:r24 equals sum of r24 and r25 Adder816s:	Data 000100	E4	06 E.	✓ [A FF	8/16 00	abo	2. 00	Addr 00 0	ess: [00 0	0×100	01
; where sum of A and 26 are two 8-bit numbers ; output: r24 and r26 are two 8-bit numbers ; output: register pair r25:r24 equals sum of r24 and r25 Adder816s: ; make variables 16-bit	Data 000100 00010C	E4 00	06 E. 00 0	- [8/16 00	00 00	2. 00 00	Addr 00 0	ess: [00 0	0×100 0 00 0 00	01
, winput: 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	Data 000100 00010C 000118	E4 00 00	06 E 00 0 00 0	- [8/16 00 00	00 00 00	00 00 00	Addr 00 0 00 0	ess: [00 0 00 0	0×100 0 00 0 00 0 00	01
, where sum of A and D may be, 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	Data 000100 00010C 000118 000124	E4 00 00	06 E 00 0 00 0	- [8/16 00 00 00	00 00 00 00	00 00 00 00	Addr 00 0 00 0 00 0	ess: [00 0 00 0 00 0	0×100 0 00 0 00 0 00	0100
, "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	Data 000100 00010C 000118 000124 000130	E4 00 00 00	06 E 00 0 00 0 00 0	- [8/16 00 00 00	00 00 00 00 00	00 00 00 00	Addr 00 0 00 0 00 0	ess: [00 0 00 0 00 0 00 0	0x100 0 00 0 00 0 00 0 00	0100000
, where sum of A and D mat De 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 ;	Data 000100 00010C 000118 000124 000130 00013C	E4 00 00 00 00	06 E 00 0 00 0 00 0 00 0	FF 0 00 0 00 0 00 0 00 0 00	8/16 00 00 00 00	00 00 00 00 00 00		Addr 00 0 00 0 00 0 00 0	ess: [00 0 00 0 00 0 00 0 00 0	0×100 0 00 0 00 0 00 0 00 0 00	
, "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 obr r26.7 ser r26.7	Data 000100 00010C 000118 000124 000130 00013C 000148	E4 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0	FF 0 00 0 00 0 00 0 00 0 00 0 00 0 00	8/16 00 00 00 00 00	00 00 00 00 00 00	2. 00 00 00 00 00 00	Addr 00 0 00 0 00 0 00 0	ess: [00 0 00 0 00 0 00 0 00 0	0×100 0 00 0 00 0 00 0 00 0 00 0 00	0000000
<pre>, where subject a full a manufacture bit numbers ; input: r24 and r26 are two 8-bit numbers ; output: register pair r25:r24 equals sum of r24 and r25 Adder816s: inake variables 16-bit clr r25</pre>	Data 000100 00010C 000118 000124 000130 00013C 000148 000154	E4 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8/16 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	Addr 00 0 00 0 00 0 00 0 00 0	ess: [)0 0)0 0)0 0)0 0)0 0)0 0)0 0	0×100 0 00 0 00 0 00 0 00 0 00 0 00 0 00	000000000
<pre>, input r24 and r26 are two 8-bit numbers ; output: register pair r25:r24 equals sum of r24 and r25 Adder816s:</pre>	Data 000100 00010C 000118 000124 000130 00013C 000148 000154 000160	E4 00 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0 00 0	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8/16 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00	Addr 00 0 00 0 00 0 00 0 00 0	ess: [00 0 00 0 00 0 00 0 00 0 00 0 00 0	0×100 0 00 0 00 0 00 0 00 0 00 0 00 0 00	0000000000
, where r24 and r26 are two 8-bit numbers ; 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 shrc r24,7 ; if number is positive guess is corr ser r25 ; guess incorrect, number is negative clr r27 shrc r26,7 ser r27; ; add add r24,r26 add r24,r26 add r24,r26	Data 000100 000100 000124 000130 000124 000130 000148 000154 000160	E4 00 00 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0 00 0 00 0	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8/16 00 00 00 00 00 00 00	abo 00 00 00 00 00 00 00 00	2. 00 00 00 00 00 00 00 00 00 00	Addr 00 0 00 0 00 0 00 0 00 0 00 0	ess: [00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0		000000000000000000000000000000000000000
<pre>; 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 str r25 ; guess incorrect, number is negative sbrc r26.7 ser r27 ;add r24.r26 add r24.r26 add r24.r26 adc r25.r27 ;store r24 ; is term the least significant bate</pre>	Memory Data 000100 000124 000130 000132 000148 000154 000156 000166 000178	E4 00 00 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0 00 0 00 0	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8/16 00 00 00 00 00 00 00	abo 00 00 00 00 00 00 00 00 00		Addr 00 0 00 0 00 0 00 0 00 0 00 0 00 0 0	ess: [00 0 00 0 0 00 0 0		000000000000000000000000000000000000000
, where sr24 and r26 are two 8-bit numbers 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 add r24,r26 add r25,r27 ; store the least significant byte sts Cr24 ; store most significant bytes	Data 000100 00010C 000124 000124 00013C 000148 000154 00016C 00016C 000178 000184	E4 00 00 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0 00 0 00 0	- FF 0 00 0 00 0 00 0 00 0 00 0 00 0 00	8/16 00 00 00 00 00 00 00 00			Addr 00000000000000000000000000000000000	ess: [000000000000000000000000000000000000000
<pre>; input r24 and r24 are two 8-bit sumbers ; output: register pair r25:r24 equals sum of r24 and r25 Adder816s: ; make variables 16-bit clr r25 shrc r24.7 ; guess r25 is positive 0x00:A shrc r24.7 ; if number is positive guess is corr clr r27 shrc r26.7 shrc r26.7 shrc r27 ;add add r24.r26 add r24.r26 adc r25.r27 ;store the least significant byte sts C+1.r25 ; store most significant bytes ret</pre>	Data 000100 00010C 000118 000124 000130 00013C 000148 000160 00016C 000162 000164 000162 000184 000190	E4 00 00 00 00 00 00 00 00 00 00 00 00	06 E 00 0 00 0 00 0 00 0 00 0 00 0 00 0	- FF 0 00 0 00 0 00 0 00 0 00 0 00 0 00	8/16 00 00 00 00 00 00 00 00 00 00			Addr 00 0 00 0 00 0 00 0 00 0 00 0 00 0 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. Value lype RAM L

	COLG							1.2.000	-
		A		0x	E4	'ä'	S	RAM	I L
	; inputs: 8-bit variables A and B ; output: 16-bit register C	B		0x	06	·-·	S	RAM	L
	Avg8s:	C		0x	F5	•õ•	S	RAM	I L
-	<pre>iload registers A and B ids r24.A id ind 26.B id addrage C = A+B/2 i clude by 2 is r24 i clude by 2 is created in the second secon</pre>					•		h 3	
-			•	VVa	ten	1	vvarc	an 2	~
	; Add two 8-bit signed 2's complement numbers,	Memory	·						
	; input: r24 and r26 are two 8-bit numbers	Data			~	8.	/16	abo	>.
	; output: register pair r25:r24 equals sum of r24 and r25	000100	E4	06	FS	00	00	00	00
	Adder 816s	00010C	00	00	00	00	00	00	00
	; make variables 16-bit	000118	00	00	00	00	00	00	00
	clr r25 ; guess r25 is positive 0x00:A	000124	00	00	00	00	00	00	00
	<pre>sbrc r24,7 ; if number is positive guess is corr</pre>	000130	00	00	00	00	00	00	00
	ser r25 ; guess incorrect, number is negative	00013C	00	00	00	00	00	00	00
	sbrc r26 7	000148	00	00	00	00	00	00	00
	ser r27	000154	00	00	00	00	00	00	00
	add	000160	00	00	00	00	00	00	00
	add r24,r26	00016C	00	00	00	00	00	00	00
	adc r25,r27	000178	00	00	00	00	00	00	00
	ete C r24 · etore the least significant bute	000184	00	00	00	00	00	00	00
	sts C+1,r25 ; store most significant bytes	000190	00	00	00	00	00	00	00
		000190	00	00	00	00	00 1	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).