

**# === Initialize array values ===**

**SET R7, 0x0000 ; R7 = 0x00000000 → array base address**

**SET R9, 0x0004 ; R9 = 0x00000004 → number of elements (n = 4)**

**SET R10, 0x0005 ; R10 = 0x00000005**

**SW R10, 0(R7) ; MEM[0] = 5**

**SET R10, 0x0003 ; R10 = 0x00000003**

**SW R10, 1(R7) ; MEM[1] = 3**

**SET R10, 0x0008 ; R10 = 0x00000008**

**SW R10, 2(R7) ; MEM[2] = 8**

**SET R10, 0x0001 ; R10 = 0x00000001**

**SW R10, 3(R7) ; MEM[3] = 1**

**# === Call procedure to compute sum ===**

**JALR R29, R0, array\_sum ; R29 = return address, jump to label 'array\_sum'**

**# === Exit: infinite loop ===**

**halt:**

**JALR R0, R0, halt ; Infinite loop to stop execution**

**# === Procedure: Sum array ===**

**array\_sum:**

**SET R12, 0x0000 ; R12 = 0 → sum = 0**

**SET R14, 0x0000 ; R14 = 0 → index i = 0**

**sum\_loop:**

**ADD R15, R7, R14 ; R15 = R7 + R14 → address of a[i]**

**LW R16, 0(R15) ; R16 = MEM[a[i]]**

**ADD R12, R12, R16 ; R12 = R12 + R16 → accumulate sum**

**ADDI R14, R14, 1 ; R14 = R14 + 1 → i++**

**BNE R14, R9, sum\_loop ; if i != n, repeat**

**000001CD 0004024D 0005028D 000A3811 0003028D 000A3851 0008028D 000A3891**  
**0001028D 000A38D1 000C074F 000B000F 0000030D 0000038D 008E3BC0 00007C10**  
**00906300 00017385 FFE97713 0000E80F**

**# === Return from procedure ===**

**JALR R0, R29, 0 ; Jump back to caller (PC = R29)****Detailed Execution Trace**

PC (Hex)	Instruction	Register/Memory Values After Execution
0x0000	SET R7, 0x0000	R7 = 0x0000
0x0001	SET R9, 0x0004	R9 = 0x0004
0x0002	SET R10, 0x0005	R10 = 0x0005
0x0003	SW R10, 0(R7)	MEM[0] = 0x0005
0x0004	SET R10, 0x0003	R10 = 0x0003
0x0005	SW R10, 1(R7)	MEM[1] = 0x0003
0x0006	SET R10, 0x0008	R10 = 0x0008
0x0007	SW R10, 2(R7)	MEM[2] = 0x0008
0x0008	SET R10, 0x0001	R10 = 0x0001
0x0009	SW R10, 3(R7)	MEM[3] = 0x0001

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x000A	JALR R29, R0, array_sum	R29 = 0x000B, PC = 0x000C
--------	-------------------------	---------------------------

### 1st Loop Iteration

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x000C	SET R12, 0x0000	R12 = 0x0000
--------	-----------------	--------------

0x000D	SET R14, 0x0000	R14 = 0x0000
--------	-----------------	--------------

0x000E	ADD R15, R7, R14	R15 = 0x0000
--------	------------------	--------------

0x000F	LW R16, 0(R15)	R16 = 0x0005
--------	----------------	--------------

0x0010	ADD R12, R12, R16	R12 = 0x0005
--------	-------------------	--------------

0x0011	ADDI R14, R14, 1	R14 = 0x0001
--------	------------------	--------------

0x0012	BNE R14, R9, sum_loop	PC = 0x000E (branch taken)
--------	-----------------------	----------------------------

### 2nd Loop Iteration

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x000E	ADD R15, R7, R14	R15 = 0x0001
--------	------------------	--------------

0x000F	LW R16, 0(R15)	R16 = 0x0003
--------	----------------	--------------

0x0010	ADD R12, R12, R16	R12 = 0x0008
--------	-------------------	--------------

0x0011	ADDI R14, R14, 1	R14 = 0x0002
--------	------------------	--------------

0x0012	BNE R14, R9, sum_loop	PC = 0x000E (branch taken)
--------	-----------------------	----------------------------

### 3rd Loop Iteration

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x000E	ADD R15, R7, R14	R15 = 0x0002
--------	------------------	--------------

0x000F	LW R16, 0(R15)	R16 = 0x0008
--------	----------------	--------------

0x0010	ADD R12, R12, R16	R12 = 0x0010
--------	-------------------	--------------

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x0011	ADDI R14, R14, 1	R14 = 0x0003
0x0012	BNE R14, R9, sum_loop	PC = 0x000E (branch taken)

#### 4th Loop Iteration

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x000E	ADD R15, R7, R14	R15 = 0x0003
0x000F	LW R16, 0(R15)	R16 = 0x0001
0x0010	ADD R12, R12, R16	R12 = 0x0011
0x0011	ADDI R14, R14, 1	R14 = 0x0004
0x0012	BNE R14, R9, sum_loop	PC = 0x0013 (branch not taken)

#### Function Return and Program Termination

PC (Hex)	Instruction	Register/Memory Values After Execution
----------	-------------	--

0x0013	JALR R0, R29, 0	PC = 0x000B
0x000B	JALR R0, R0, halt	PC = 0x000B (infinite loop)

#### Final Register Values

Register	Final Value (Hex)	Description
----------	-------------------	-------------

R0	0x0000	Always 0 (hardwired)
R7	0x0000	Array base address
R9	0x0004	Array size (n = 4)
R10	0x0001	Last value stored
R12	0x0011	Sum result (17 in decimal)
R14	0x0004	Final loop counter value
R15	0x0003	Last array address accessed

**Register Final Value (Hex) Description**

R16	0x0001	Last array value loaded
R29	0x000B	Return address

**Final Memory State****Memory Address Value (Hex) Description**

MEM[0]	0x0005	Array element 0
MEM[1]	0x0003	Array element 1
MEM[2]	0x0008	Array element 2
MEM[3]	0x0001	Array element 3