

Name : Ghanashyam Mahesh Bhat

SRN : PES1UG20CS153

SECTION : C



Department of Computer Science & Engineering
Microprocessor & Computer Architecture
MPCA-Laboratory/Assignment/Hands-on/Project
UE20CS252

Sl. No	Programs
Week cNo.3	<p>3. Write a program in ARM7TDMI-ISA to find the sum of N data items at alternate [odd or even positions] locations in the memory. Store the result in the memory location.</p> <p>a. Use Pre-indexing addressing mode</p> <p>;Preindexing Mode - Sum of number at odd position</p> <p>.DATA</p> <p>A:WORD 10,20,30,40,50</p> <p>SUM_ALT:WORD 0</p> <p>.TEXT</p> <p>MOV R1,#0 ; sum of alternate numbers</p> <p>MOV R4,#0 ; temporary register</p> <p>MOV R6,#0 ; loop count</p> <p>LDR R2,=A</p> <p>LDR R3,=SUM_ALT</p> <p>SUB R2,R2,#8</p> <p>LOOP:</p> <p>LDR R4,[R2,#8]</p> <p>ADD R2,R2,#8</p> <p>ADD R1,R1,R4</p> <p>ADD R6,R6,#1</p> <p>CMP R6,#3</p> <p>BNE LOOP</p> <p>STR R1,[R3]</p>

.END

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 0

R2 : 0

R3 : 0

R4 : 0

R5 : 0

R6 : 0

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 4096

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Carry (C) : 0

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

0x000000df

Antoindexing.s Postindexing.s Preindexing.s

:Preindexing Mode - Sum of number at odd position

.DATA

000010D8: A: .WORD 10,20,30,40,50

000010EC: SUM_ALT: .WORD 0

.TEXT

0000106C:E3A01000 MOV R1,#0 ; sum of alternate numbers

00001070:E3A04000 MOV R4,#0 ; temporary register

00001074:E3A06000 MOV R6,#0 ; loop count

00001078:E59F2020 LDR R2,=A

0000107C:E59F3020 LDR R3,=SUM_ALT

00001080:E2422008 SUB R2,R2,#8

00001084: LOOP:

00001084:E5924008 LDR R4,[R2,#8]

00001088:E2822008 ADD R2,R2,#8

0000108C:E0811004 ADD R1,R1,R4

00001090:E2866001 ADD R6,R6,#1

00001094:E3560003 CMP R6,#3

00001098:1AFFFFF9 BNE LOOP

MemoryView1

Word Size

8bit 16bit 32bit

000010EC

000010EC 00000000 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001118 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001144 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001170 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

0000119C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000011C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000011F4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001220 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

0000124C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001278 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0

R1 : 0

R2 : 4328

R3 : 4332

R4 : 50

R5 : 0

R6 : 3

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 21504

R14 (lr) : 0

R15 (pc) : 70656

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable : 1

FIQ Disable : 1

Thumb (T) : 0

CPU Mode : System

0x600000df

Antoindexing.s Postindexing.s Preindexing.s

:Preindexing Mode - Sum of number at odd position

.DATA

000010D8: A: .WORD 10,20,30,40,50

000010EC: SUM_ALT: .WORD 0

.TEXT

0000106C:E3A01000 MOV R1,#0 ; sum of alternate numbers

00001070:E3A04000 MOV R4,#0 ; temporary register

00001074:E3A06000 MOV R6,#0 ; loop count

00001078:E59F2020 LDR R2,=A

0000107C:E59F3020 LDR R3,=SUM_ALT

00001080:E2422008 SUB R2,R2,#8

00001084: LOOP:

00001084:E5924008 LDR R4,[R2,#8]

00001088:E2822008 ADD R2,R2,#8

0000108C:E0811004 ADD R1,R1,R4

00001090:E2866001 ADD R6,R6,#1

00001094:E3560003 CMP R6,#3

00001098:1AFFFFF9 BNE LOOP

MemoryView1

Word Size

8bit 16bit 32bit

000010EC

000010EC 0000005A 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001118 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001144 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001170 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

0000119C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000011C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

000011F4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001220 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

0000124C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

00001278 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

b. Use Post- Indexing addressing mode

;Postindexing Mode - Sum of number at odd position

.DATA

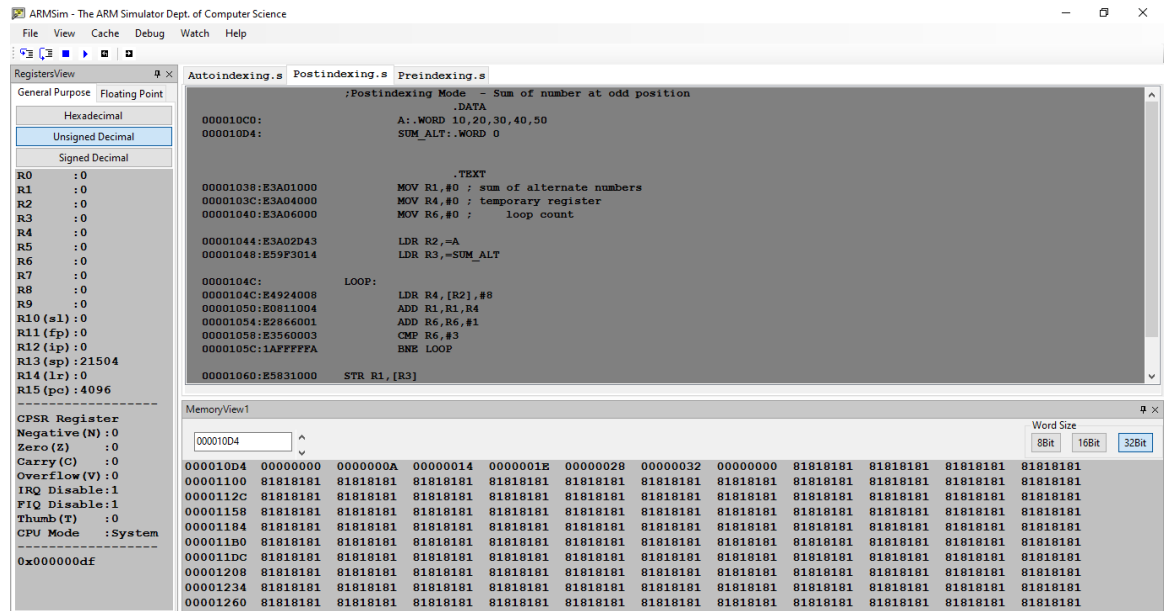
A: .WORD 10,20,30,40,50

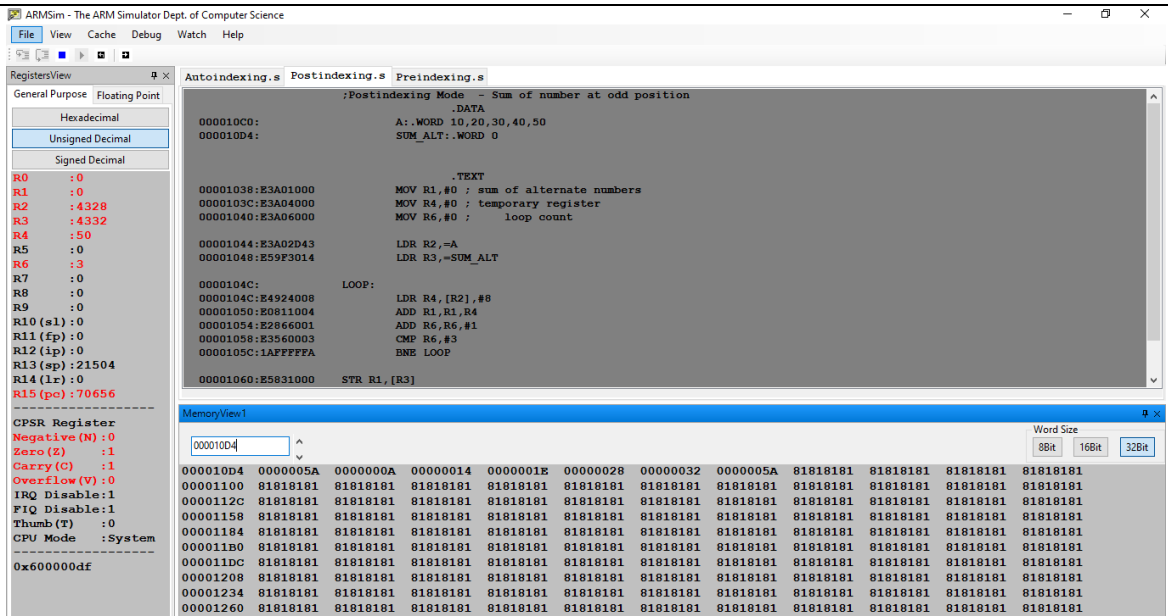
SUM_ALT: .WORD 0

.TEXT

MOV R1,#0 ; sum of alternate numbers

MOV R4,#0 ; temporary register





c. Use Auto-indexing addressing mode

;Autoindexing Mode - Sum of number at odd position

.DATA

A: .WORD 10,20,30,40,50

SUM_ALT: .WORD 0

.TEXT

MOV R1,#0 ; sum of alternate numbers

MOV R4,#0 ; temporary register

MOV R6,#0 ; loop count

LDR R2,=A

LDR R3,=SUM_ALT

SUB R2,R2,#8

LOOP:

LDR R4,[R2,#8]!

ADD R1,R1,R4

ADD R6,R6,#1

CMP R6,#3

BNE LOOP

STR R1,[R3]

The image displays two sequential screenshots of the ARMSim ARM Simulator interface, illustrating the execution of an assembly program.

Screenshot 1 (Top):

- Title Bar:** ARMSim - The ARM Simulator Dept. of Computer Science
- Menu Bar:** File View Cache Debug Watch Help
- RegistersView Panel:**
 - General Purpose Floating Point
 - Hexadecimal
 - Unsigned Decimal
 - Signed Decimal
 - R0 : 0
 - R1 : 0
 - R2 : 0
 - R3 : 0
 - R4 : 0
 - R5 : 0
 - R6 : 0
 - R7 : 0
 - R8 : 0
 - R9 : 0
 - R10 (sl) : 0
 - R11 (fp) : 0
 - R12 (ip) : 0
 - R13 (sp) : 21504
 - R14 (lr) : 0
 - R15 (pc) : 4096
- CPSR Register:**
 - Negative (N) : 0
 - Zero (Z) : 0
 - Carry (C) : 0
 - Overflow (V) : 0
 - IRQ Disable : 1
 - FIQ Disable : 1
 - Thumb (T) : 0
 - CPU Mode : System
- MemoryView Panel:**
 - Word Size: 8Bit, 16Bit, 32Bit (selected)
 - Address: 000010B8
 - Content: 000010BC 00000000 0000000A 00000014 0000001E 00000028 00000032 00000000 0000000A 00000014 0000001E 00000028
- Main Assembly Window:**
 - Autoindexing.s Postindexing.s Preindexing.s
 - :Autoindexing Mode - Sum of number at odd position
 - .DATA
 - 000010A8: A: .WORD 10,20,30,40,50
 - 000010BC: SUM_ALT: .WORD 0
 - .TEXT
 - 00001000:E3A01000 MOV R1,#0 ; sum of alternate numbers
 - 00001004:E3A04000 MOV R4,#0 ; temporary register
 - 00001008:E3A06000 MOV R6,#0 ; loop count
 - 0000100C:E59F201C LDR R2,-A
 - 00001010:E59F301C LDR R3,-SUM_ALT
 - 00001014:E2422008 SUB R2,R2,#8
 - 00001018: LOOP:
 - 00001018:E5B24008 LDR R4,[R2,#8]!
 - 0000101C:E0811004 ADD R1,R1,R4
 - 00001020:E2866001 ADD R6,R6,#1
 - 00001024:E3560003 CMP R6,#3
 - 00001028:1AFFFFFA BNE LOOP