



Faculty of Engineering
Ain Shams University
CSE 211s:Introduction to Embedded
Systems
REPORT (2)

Name: Habiba El-Sayed Mowafy

Sec.: 1

Program: CSE

ID: 2100792

Question 3:

Q3. Write ARM assembly code to sum the array items of size 10 and store it in the memory. The array contains the following values:1, 2, 3, 4, 5, 6, 7,8, 9, 10.

Assembly Code:

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors

DCD 0x20008000
DCD Reset_Handler
ALIGN
AREA myCode, CODE, ReadOnly
ENTRY
EXPORT Reset_Handler

Reset_Handler

ARR_SIZE      EQU          10

MOV R0, #0 ; initialize array index -i- to 0
MOV R1, #0 ; initialize sum to 0
LDR R2, =Arr ; load the base address of the array in R2
LDR R3, =0x20000000 ; storing the address sum will be stored in it in R3

LOOP          CMP R0, #ARR_SIZE ; check loop condition i < array size
              BEQ STORING
              LDR R4, [R2, R0, LSL #2] ; load data of Arr[i]
              ADD R1, R1, R4 ; sum = sum + Arr[i]
              ADD R0, R0, #1 ; increment of i
              B     LOOP
```

STR **R1, [R3]**; storing sum in the memory address stored in R3

ALIGN

AREA myData, DATA, READONLY

Arr DCD 1,2,3,4,5,6,7,8,9,10
END

Registers:

General Registers

Register	Value
Core	
R0	0x0000000A
R1	0x00000037
R2	0x0000002C
R3	0x20000000
R4	0x0000000A
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000

Special Registers

R13 (SP)	0x20008000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00040000
+ xPSR	0x21000000

Memory:

Question 4:

Assembly Code:

```
AREA RESET, DATA, READONLY

EXPORT __Vectors

__Vectors

DCD 0x20008000

DCD Reset_Handler

ALIGN

AREA myCode, CODE, ReadOnly

ENTRY

EXPORT Reset_Handler

Reset_Handler

MOV R0, #6 ; R0 elem count
MOV R1, #-14
MOV R2, #5
MOV R3, #32
MOV R4, #-7
MOV R5, #0
MOV R6, #-5
PUSH {R4}
PUSH {R5}
PUSH {R6}
BL MAXMIN
B FINISH

MAXMIN
POP {R6}
POP {R5}
POP {R4}
PUSH {LR}
```

	LDR R7 , =ARR
	STR R1 , [R7]
	STR R2 , [R7 , #4]
	STR R3 , [R7 , #8]
	STR R4 , [R7 , #12]
	STR R5 , [R7 , #16]
	STR R6 , [R7 , #20]
	MOV R8 , #0 ;INDEX COUNT
	LDR R9 , [R7] ;TEMP REG. WITH FIRST VALUE IN TEMP ARR. MAX TEMP
	LDR R12 , [R7] ;TEMP REG. WITH FIRST VALUE IN TEMP ARR MIN TEMP
LOOP	SUBS R0 , #1
	BEQ RETURN
	ADD R8 , R8 , #1
	LDR R10 , [R7 , R8 , LSL #2]
	CMP R10 , R9 ;R10 – R9
	BGT GT
	CMP R10 , R12 ;R10 – R12
	BLT LT
DONE	B LOOP
RETURN	POP {LR}
	MOV R0 , R9 ;MAX
	MOV R1 , R12 ;MIN
	BX LR
GT	MOV R9 , R10 ; UPDATE THE VALUE TO THE BIGGER ONE , R9 CONTAIN MAX
	B DONE
LT	MOV R12 , R10 ; UPDATE THE VALUE TO THE BIGGER ONE , R11 CONTAIN MIN
	B DONE
FINISH	STR R0 , [R7 , #24]
	STR R1 , [R7 , #28]

ALIGN

AREA myVarArr ,DATA ,READWRITE

ARR

SPACE

24

END

Registers:

General Registers

Registers	
Register	Value
Core	
R0	0x00000020
R1	0xFFFFFFFF2
R2	0x00000005
R3	0x00000020
R4	0xFFFFFFFF9
R5	0x00000000
R6	0xFFFFFFFFB
R7	0x20000000
R8	0x00000005
R9	0x00000020
R10	0xFFFFFFFFB
R11	0x00000000
R12	0xFFFFFFFF2

Special Registers

R13 (SP)	0x20008000
R14 (LR)	0x0000002F
R15 (PC)	0x0000007C
xPSR	0x61000000

Memory:

Memory 1	
Address:	0x20000000
0x20000000:	F2 FF FF FF 05 00 00 00 20 00 00 00 F9 FF FF FF 00 00 00 00 FB FF FF
0x20000017:	FF 20 00 00 00 F2 FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2000002E:	00 00
0x20000045:	00 00
Call Stack + Locals Memory 1	

>>> The max. value in R0 is stored in memory in address 0x20000018

>>> The min. value in R1 is stored in memory in address 0x2000001C