



**COMPUTER ORGANIZATION AND MICROPROCESSOR
ENCS2380**

Assembly Assignment

Name: Faris Abufarha

Student ID: 1200546

Instructor: Dr. Abualseoud Hanani

Section: 1

Date: 23/6/2022

CODE:

AREA RESET, DATA, READONLY

EXPORT __Vectors

__Vectors DCD 0x20001000

DCD Reset_Handler

ALIGN

ARRAY DCB 34, 56, 27, 156, 200, 68, 128, 235, 17, 45

AREA MYRAM, DATA, READWRITE

;initializing these two in the memory of size 32 bit

SUM DCD 0

EVEN DCD 0

LP2_ARRAY DCB 1,1,1,1,1,1,1,1,1,1 ; LARGEST POWER ARRAY OF 2, INITIALLY ONE
THIS WILL CHANGE RELYING ON CORROSPONDING INDEX IN THE ORIGINAL ARRAY

AREA MYCODE, CODE, READONLY

ENTRY

EXPORT Reset_Handler

;IMPLEMENTING THE POW2 PROC

POW2 PROC

LOOP3

TST R3, #1 ; THIS METHOD IS USED TO CHECK IF THE NUMBER IS EVEN OR
ODD, BY MAKING 'TST' (AND WITHOUT CHANGING THE ORIGINAL VALUE, IF IT GIVES
ZERO THEN IT'S ODD OTHERWISE EVEN

LSREQ R3, #1 ;LOGICAL SHIFT RIGHT (DIVIDING BY 2)

LSLEQ R4, #1;LOGICAL SHIFT LEFT (MULTIPLYING BY 2)

BEQ LOOP3 ; IF Z FLAG 1 BRANCH

STRB R4, [R1] ; STORES THE VALUE IN MEMORY

```
        BX LR ; BX : branch and exchange instruction set, LR : HOLDS THE RETURN  
ADDRESS  
        ENDP ; END OF PROCEDURE
```

Reset_Handler

```
;;;;;;;;;your assembly code here
```

```
;  
=====
```

```
;; QUESTION A & B  
LDR R1, =ARRAY ; GETS THE ADDRESS OF THE FIRST ELEMENT IN THE  
ARRAY
```

```
MOV R4, #10 ; 10 NUMBER OF ELEMENTS IN THE ARRAY
```

LOOP

```
LDRB R2, [R1] ; GETTS THE VALUE OF INDEX R1 AND STORES IT  
IN R2 B: BYTE  
ADD R1, R1, #1 ; INCREMENTING R1
```

```
ADD R3, R3, R2 ; R3 PRESENTS THE LOCAL SUM
```

```
SUBS R4, R4, #1 ;DECREMENT R4 SO WHEN IT'S ZERO WE END ,S FOR FLAG
```

```
BNE LOOP ; NOT EQUAL ZERO
```

```
LDR R5, =SUM ; GETS THE ADDRESS OF SUM AND STORES IT IN R5
```

```
STR R3, [R5] ; STORES LOCAL SUM (R3) INTO GLOBAL SUM WHICH ITS  
ADDRESS IN R5
```

```
;  
=====
```

```
; QUESTION C ::
```

```
LDR R1, =ARRAY
```

```
MOV R4, #10
LDR R6, =EVEN ; STORES ADDRESS OF 'EVEN' INTO THE REGISTER R6
```

LOOP2

```
        LDRB R2, [R1]      ; GETTS THE VALUE OF INDEX R1 AND STORES IT
IN R2 B: BYTE
        TST R2, #1         ; FOR CHECKING IF EVEN ... (EXPLAINED MORE ABOVE)
        ADDEQ R7,R7,R2     ; EVENSUM STORED IN R7
        ADD R1, R1, #1 ; ITERATE
        SUBS R4, R4, #1 ;DECREMENT S FOR FLAG
        BNE LOOP2 ; NOT EQUAL ZERO

        STR R7, [R6] ; STORES INTO MEMORY
```

```
;
=====
```

; QUESTION : D

```
;BIT CLEAR THESE REGISTERS
BIC R1, #0xFFFFFFFF
BIC R2, #0xFFFFFFFF
BIC R3, #0xFFFFFFFF
BIC R4, #0xFFFFFFFF
BIC R0, #0xFFFFFFFF
BIC R5, #0xFFFFFFFF
```

```
LDR R0, =ARRAY
LDR R1, =LP2_ARRAY ; GETS THE ADDRESS OF FIRST ELEMENT IN THE
ARRAY
```

```
MOV R2, #10 ; SIZE OF ARRAY AND = NUMBER OF ITERATIONS
```

; ITERATING OVER ALL ELEMENTS IN THE ARRAY

LOOP4

```
LDRB R3, [R0] ; GETS THE VALUE IN INDEX R0 IN THE ARRAY
MOV R4, #1 ; USED AS FOR PARAMETR STUFF FOR POW2
BL POW2
ADD R0,R0,#1
ADD R1,R1,#1
```

```
SUBS R2,R2,#1  
BNE LOOP4
```

Stop B Stop

end

SNAP SHOTS:

```

*** Using Compiler 'V6.15', folder: 'C:\Keil_v5\ARM\ARMCLANG\Bin'
Build target 'Target 1'
"orga.s", line 136: Warning: A1581W: Added 2 bytes of padding at address 0x7a
    136 0000007c    end
0 Errors, 1 Warning
assembling orga.s...
linking...
Program Size: Code=140 RO-data=20 RW-data=20 ZI-data=0
".\Objects\ORGA_assignment.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:00

```

Initially :

Memory 1

Address: 0x20000000

0x20000000:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x20000017:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2000002E:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x20000045:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2000005C:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x20000073:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2000008A:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x200000A1:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x200000B8:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x200000CF:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x200000E6:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Call Stack + Locals | Memory 1

After execution:

Register	Value
Core	
R0	0x08000012
R1	0x20000012
R2	0x00000000
R3	0x0000002D
R4	0x00000001
R5	0x00000000
R6	0x20000004
R7	0x00000282
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20001000
R14 (LR)	0x08000081
R15 (PC)	0x0800008C
xPSR	0x61000000
+ Banked	
+ System	
- Internal	
Mode	Thread

Memory 1	
Address:	0x20000000
0x20000000:	C6 03 00 00 82 02 00 00 02 08 01 04 08 04 80 01 01 01 00 00 00 00 00 00
0x20000017:	00 00
0x2000002E:	00 00
0x20000045:	00 00
0x2000005C:	00 00
0x20000073:	00 00

P.S: including every needed thing, SUM(0X20000000), EVENSUM(0X20000004), ARRAY OF POWER2 FIRST INDEX : (0X20000008)