Name: Gia Dao
ID: 1001747062

Home work 2

3. For each of these C functions, specify the ARM 7 register(s) in which each argument is passed and the result is returned

a) uint32_t fn4(uint16_t a, uint32_t b, int8_t c, uint32_t d)

a is passed in: R0 [bits: 0-15]

b is passed in: R1 [bits: 0-31]

c is passed in: R2 [bits: 0-7]

d is passed in: R3 [bits: 0-31]

The result is returned in: R0

b) uint64_t fn2 (uint64_t a, uint64_t b)

a is passed in: R0:R1 [R0 bits 0-31, bits 32-63 in R1]

b is passed in: R2:R3 [R2 bits 0-31, bits 32-63 in R3]

The result is stored in: R0:R1 [R0 bits 0-31, R1 bits 32-63]

4. Determine the value of the 12-bit value stored as operand2 for the 32 bit immediate value of the following instructions:

a) MOV R0, #0x8300 000

N = #0x8300 000 (hexa)

in 32 bit: 0000 1000 0011 0000 0000 0000 0000 0000

n = #0x83 = 1000 0011 (8 bit)

N = 0000 0000 0000 0000 0000 0000 1000 0011

→ ROR = 12 → s = 6

→ n = #0x83 = 1000 0011, s = 6

b) MOV, R0, #135168

N = #135168 (dec) = 0000 0000 0000 0010 0001 0000 0000 0000

Take n as 8-bit → n = 0010 0001 = #0x21 (hex) 8 bit

N = 0000 0000 0000 0000 0000 0000 0010 0001

→ ROR = 20 → s = 10

n = #0x21 = 0010 0001, s = 10

c) Move a value of -23 to R0

MVN R0, #22 (R0 = -23)

22 = 0000 0000 0000 0000 0000 0000 0001 0110

→ ~22 (Not 22) = 1111 1111 1111 1111 1111 1111 1110 1001

Add1 = 1111 1111 1111 1111 1111 1111 1110 1010 (-23)

→ n = 22 = $0001\ 0110_2$

s = 0

1 Assuming R0 contains 0x40000000, R1 contains 0x12345678, and R2 contains 0x00000002, write the contents of the memory allocations below after the STR instruction writes to memory

a) STR R_1, [R0], little-endian

Value at address 0x40 000 000 is 0x78

Value at address 0x40 000 001 is 0x56

Value at address 0x40 000 002 is 0x34

Value at address 0x40 000 003 is 0x12

b) STRH R_1, [R0]; assuming little-endian convention

Value at address 0x40 000 000 is 0x78

value at address 0x40000 001 is 0x56

value at address 0x40 000 002 is X

value at address 0x40 000 003 is X

c) STRB R_2, [R0], assuming little-endian convention

value at address 0x40 000 000 is 0x78

value at address 0x40 000 001 is X

value at address 0x40 000 002 is X

value at address 0x40 000 003 is X

d) STR R_1, [R0]; assuming big-endian convention

value at address 0x40 000 000 is 0x12

value at address 0x40 000 001 is 0x34

value at address 0x40 000 002 is 0x56

value at address 0x40 000 003 is 0x78

e) STRH R1, [R0] ; assuming big-endian convention:

value at address	0x40000000	0x12
value at address	0x40000001	0x34
value at address	0x40000002	X
value at address	0x40000003	X

f) STRB R1, [R0], assuming big-endian convention

value at address	0x40000000	0x12
value at address	0x40000001	X
value at address	0x40000002	X
value at address	0x40000003	X

g) STRB R_1, [R0, R_2] ; assuming big-endian convention:

value at address	0x40000000	X
value at address	0x40000001	X
value at address	0x40000002	0x12
value at address	0x40000003	X

h) STRH R_1, [R0, R2] ; assuming big-endian convention

value at address	0x40000000	X
value at address	0x40000001	X
value at address	0x40000002	0x12
value at address	0x40000003	0x34

2. Assuming the memory allocations contain data below what is the value of R0 (all 32-bits hex)

a) LDR R0, [R1] assuming R1 = 0x50000000

⇒ **R0 = 0x8A335712**

b) LDRH R0, [R1, R2] assuming R1 = 0x50000000 and R2 = 4

⇒ **R0 = 0x0000C09A**

c) LDRSH R0, [R1] assuming R1 = 0x50000002

⇒ **R0 = 0xFFFF8A33**

d) LDRB R0, [R1] assuming R1 = 0x50000005

⇒ **R0 = 0x000000C0**

e) LDRSB R0, [R1] assuming R1 = 0x50000005

⇒ **R0 = 0xFFFFFFC0**

f) LDRSB R0, [R_1, R_2] assuming R1 = 0x50000000 and R2 = 3

R0 = **0xFFFFFF8A**