

## Homework 2

### Q1. Register Transfer Language (RTL)

LDR R2, [R1]  
 ↓  
 load register      ↓ where      get data from memory address  
 store data

MAR ← PC  
 MDR ← MAR  
 IR ← MDR  
 PC ← PC + 4

Instructions:

- ① Fetch: LDR, R2, [R1] from memory
- ② Decode: load data into R2 through memory address of R1
- ③ Address Calculation: Get memory address to R1
- ④ Read value: stored in memory address
- ⑤ Load Value: store value into R2

### Q2. Application Program Status Register (APSR's) flags

MOV R0, #0x80000000  
 MOV R1, #0x1  
 SUBS R2, R1, R0

	Flag	Value	
(Negative)	N	1	R0
(Zero)	Z	0	R1
(Carry)	C	0	
(Overflow)	V	0	R2 = R1 - R0

### Q3 Memory Endianess and Alignment

1)  $\#1234567890 \rightarrow 0x499602D2$

Bigendian:

Address	Data Contents
0x20000000	0x49 (49)
0x20000001	0x96 (96)
0x20000002	0x02 (02)
0x20000003	0xD2 (D2)

$0x\overbrace{49}^{\leftarrow}\overbrace{96}^{\leftarrow}\overbrace{02}^{\leftarrow}\overbrace{D2}^{\leftarrow}$

Little endian:

Address	Data Contents
0x20000000	0xD2 (D2)
0x20000001	0x02 (02)
0x20000002	0x96 (96)
0x20000003	0x49 (49)

$0x\overbrace{49}^{\leftarrow}\overbrace{96}^{\leftarrow}\overbrace{02}^{\leftarrow}\overbrace{D2}^{\leftarrow}$

### Q3. Memory Endianness and Alignment

2)

char a - 1 byte

long long int b; - 8 byte

double c; - 8 byte

Short d; - 2 byte

Char \*e; - 8 byte → 14

float f; - 4 byte

	+0th	+1st	+2nd	+3rd
0th byte	a			
4th	b	b	b	b
8th	b	b	b	b
12th	c	c	c	c
16th	c	c	c	c
20th	d	d	e	
24th	f	f	f	f
28th				
32th				

Improved  
version

	+0th	1st	2nd	3rd
0th	a			
4th				
8th	b	b	b	b
12th	b	b	b	b
16th	c	c	c	c
20th	c	c	c	c
24th	d	d	e	e
28th	e	e	f	f
32th	f	f	f	f

Q4:

1

both have 2 registers → ADDS R7, R5, #7  
ADDs R,D,RS, imm3

ADD\$	II	00	imm3	RS	Rd
000	11	00	111	10	1

$$\begin{aligned}
 &= 0001100111101111 \\
 &= 0001 \ 1001 \ 1111 \ 0111 \\
 &= 19F7 \\
 &= 0 \times 19F7
 \end{aligned}$$

MOVW Rd, #imm16

(2) MOVW R10, #0x1234

11110	i	10010	0 imm4	0 imm3	Rd	imm8
1 1 1	1 0 0	1 0 0	1 0 0 0 0 0	1 0	0 1 0	1 0 1 0 0 0 1 0 1 0 0

$$\begin{aligned}
 &= 1111001001000010100101000110100 \\
 &= 1111 \quad 0010 \quad 0100 \quad 0001 \quad 0010 \quad 1010 \quad 0011 \quad 0100
 \end{aligned}$$

= F2412A34

$$= 0xF2412A34$$

Reset to continue editing code

```

1 src    DCB    'a','b','c','d','e','f','g','h','l','j','k','l',0
2 dst    DCB    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3
4 begin
5   LDR    R0, =src
6   LDR    R1, =dst
7
8 loop   LDRB   R2, [R0], #1
9   CMP    R2, #0
10  BEQ   done
11  SUB    R2, R2, #0x20
12  STRB   R2, [R1], #1
13  B     loop
14
15 done  B     done
16 END
17

```

Branch

**View Memory Contents**

Start address:  End address:

Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x100	0x64	0x63	0x62	0x61	0x64636261
0x104	0x68	0x67	0x66	0x65	0x68676665
0x108	0x6C	0x6B	0x6A	0x69	0x6C6B6A69
0x10C	0x0	0x0	0x0	0x0	0x0
0x110	0x44	0x43	0x42	0x41	0x44434241
0x114	0x48	0x47	0x46	0x45	0x48474645
0x118	0x4C	0x4B	0x4A	0x49	0x4C4B4A49
0x11C	0x0	0x0	0x0	0x0	0x0

Word Value Format:

Memory Map Key:

Memory Map

R0	0x10D	Dec	Bin	Hex
R1	0x11C	Dec	Bin	Hex
R2	0x0	Dec	Bin	Hex
R3	0x0	Dec	Bin	Hex
R4	0x0	Dec	Bin	Hex
R5	0x0	Dec	Bin	Hex
R6	0x0	Dec	Bin	Hex
R7	0x0	Dec	Bin	Hex
R8	0x0	Dec	Bin	Hex
R9	0x0	Dec	Bin	Hex
R10	0x0	Dec	Bin	Hex
R11	0x0	Dec	Bin	Hex
R12	0x0	Dec	Bin	Hex
R13	0xFF000000	Dec	Bin	Hex
LR	0x0	Dec	Bin	Hex
PC	0x2C	Dec	Bin	Hex

Clock Cycles: 🕒 Current Instruction: 3 Total: 131

CSPR Status Bits (NZCV) 0 | 1 | 1 | 0