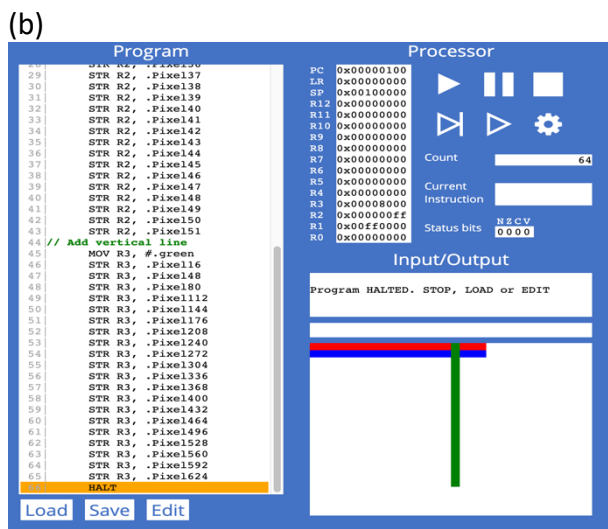
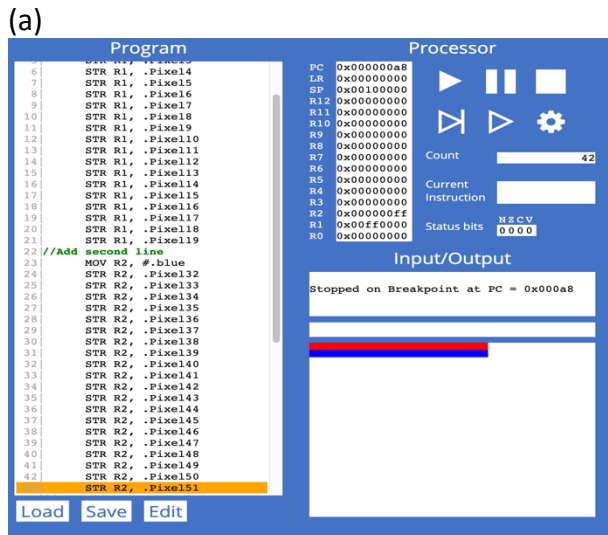


COMPUTER SYSTEMS

LAB 09



Exercise 9.1.3

- (a) The given code is indirect addressing as it can store the specific value of r2 in [r4] which has a memory address and in its own value. In the code it is adding four bytes every time to the pixel until it reaches 80 and moving data to the memory address 4 bytes, 32 bits. and R1 is the base address if we add value in R3 it will add to the R1 which stores the address and which can form the new pointer to the next pixel.

(b)

Program

```
1  MOV R1, #.PixelScreen // base address of the medium
2  MOV R2, #.red
3  MOV R3, #0
4  loop:
5      ADD R4, R1, R3 // calculate the byte offset (R1 + R3)
6      STR R2, [R4]
7      ADD R3, R3, #4
8      CMP R3, #80
9      BLT loop
10     MOV R3, #256
11 loop2:
12     ADD R4, R1, R3
13     STR R2, [R4]
14     ADD R3, R3, #4
15     CMP R3, #336
16     BLT loop2
17     HALT
```

Processor

PC	0x0000003c
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0xffff314c
R3	0x00000150
R2	0x00ff0000
R1	0xffff3000
R0	0x00000000

Count

Current Instruction

Status bits **N Z C V**
0 1 1 0

Input/Output

Program HALTED. STOP, LOAD or EDIT

(c)

Program

```
1  MOV R1, #.PixelScreen // base address of the medium
2  MOV R2, #.red
3  MOV R3, #0
4  loop:
5      ADD R4, R1, R3 // calculate the byte offset (R1 + R3)
6      STR R2, [R4]
7      ADD R3, R3, #4
8      CMP R3, #80
9      BLT loop
10     MOV R3, #256
11 loop2:
12     ADD R4, R1, R3
13     STR R2, [R4]
14     ADD R3, R3, #4
15     CMP R3, #336
16     BLT loop2
17     MOV R3, #128
18 loop3:
19     ADD R4, R1, R3
20     STR R2, [R4]
21     ADD R3, R3, #256
22     CMP R3, #5120
23     BLT loop3
24     HALT
```

Processor

PC	0x00000054
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0xffff4380
R3	0x00001480
R2	0x00ff0000
R1	0xffff3000
R0	0x00000000

Count

Current Instruction

Status bits **N Z C V**
0 0 1 0

Input/Output

Program HALTED. STOP, LOAD or EDIT

Load

Save

Edit

Exercise 9.2.1

Program

```
1  MOV R1, #.PixelScreen
2  MOV R2, #.red
3  MOV R3, #0
4  loop:
5      STR R2, [R1+R3]
6      ADD R3,R3,#4
7      CMP R3, #80
8      BLT loop
9  HALT
```

Load

Save

Edit

Processor

PC 0x00000020
LR 0x00000000
SP 0x00100000
R12 0x00000000
R11 0x00000000
R10 0x00000000
R9 0x00000000
R8 0x00000000
R7 0x00000000
R6 0x00000000
R5 0x00000000
R4 0x00000000
R3 0x00000050
R2 0x00ff0000
R1 0xffff3000
R0 0x00000000

▶

⏸

◻

⏮

⏭

⚙

Count

84

Current Instruction

Status bits

NZCV
0110

Input/Output

Program HALTED. STOP, LOAD or EDIT

Exercise 9.2.2

Program

```
1  MOV R1, #.PixelScreen
2  MOV R2, #.red
3  MOV R3, #0
4  MOV R11, #80
5  MOV R12, #0
6  loop:
7      STR R2, [R1+R3]
8      ADD R3,R3,#4
9      CMP R3,R11
10     BLT loop
11     ADD R12,R12,#1
12     CMP R12,#10
13     BEQ stop
14     ADD R3,R3,#176
15     ADD R11,R11,#256
16     B loop
17 stop:
18 HALT
```

Load

Save

Edit

Processor

PC 0x00000040
LR 0x00000000
SP 0x00100000
R12 0x0000000a
R11 0x00000950
R10 0x00000000
R9 0x00000000
R8 0x00000000
R7 0x00000000
R6 0x00000000
R5 0x00000000
R4 0x00000000
R3 0x00000950
R2 0x00ff0000
R1 0xffff3000
R0 0x00000000

▶

⏸

◻

⏮

⏭

⚙

Count

863

Current Instruction

Status bits

NZCV
0110

Input/Output

Program HALTED. STOP, LOAD or EDIT

0x00024

Exercise 9.3.1

(a) The .Align 256 instruction ensures the next instruction is aligned with a word address divisible by 256.

(b)

The screenshot shows the ARMLite Simulator V1.2.4 interface. The **Program** window on the left contains the following assembly code:

```

1 | MOV R2, #arrayData
2 | LDR R0, [R2, #16]
3 | HALT
4 | .ALIGN 256
5 | arrayLength: 10
6 | arrayData: 9
7 | 8
8 | 7
9 | 6
10 | 5
11 | 4
12 | 3
13 | 2
14 | 1
15 | 0

```

The **Processor** window in the center shows the following registers and status:

- PC: 0x00000000
- LR: 0x00000000
- SP: 0x00100000
- R12: 0x00000000
- R11: 0x00000000
- R10: 0x00000000
- R9: 0x00000000
- R8: 0x00000000
- R7: 0x00000000
- R6: 0x00000000
- R5: 0x00000000
- R4: 0x00000000
- R3: 0x00000000
- R2: 0x00000104
- R1: 0x00000000
- R0: 0x00000005

The **Memory** window on the right shows the memory layout starting at address 0x00000000. The instruction at address 0x00000000 is `MOV R2, #arrayData` (0xe3a02f41). The instruction at address 0x00000004 is `LDR R0, [R2, #16]` (0xe5920010). The instruction at address 0x00000008 is `HALT` (0xe1000000). The instruction at address 0x0000000c is `.ALIGN 256` (0xe1000000).

The **Input/Output** window shows the program status: `Program HALTED. STOP, LOAD or EDIT`.

(c)

The screenshot shows the ARMLite Simulator V1.2.4 interface after the program has executed the `.ALIGN 256` instruction. The **Program** window on the left contains the same assembly code as in (b).

The **Processor** window in the center shows the following registers and status:

- PC: 0x00000010
- LR: 0x00000000
- SP: 0x00100000
- R12: 0x00000000
- R11: 0x00000000
- R10: 0x00000000
- R9: 0x00000000
- R8: 0x00000000
- R7: 0x00000000
- R6: 0x00000000
- R5: 0x00000000
- R4: 0x00000000
- R3: 0x00000000
- R2: 0x00000104
- R1: 0x00000010
- R0: 0x00000005

The **Memory** window on the right shows the memory layout starting at address 0x00000000. The instruction at address 0x00000000 is `MOV R2, #arrayData` (0xe3a02f41). The instruction at address 0x00000004 is `LDR R0, [R2, #16]` (0xe5920010). The instruction at address 0x00000008 is `HALT` (0xe1000000). The instruction at address 0x0000000c is `.ALIGN 256` (0xe1000000). The instruction at address 0x00000010 is `arrayLength: 10` (0xe1000000).

The **Input/Output** window shows the program status: `Program HALTED. STOP, LOAD or EDIT`.

Exercise 9.3.2

Program

```
1 | MOV R1, #0
2 | MOV R2, #arrayData
3 | MOV R3, #0
4 | MOV R4, #arrayLength
5 | loop:
6 |   ADD R3,R3,#1
7 |   LDR R5,[R2+R1]
8 |   ADD R0,R0,R5
9 |   ADD R1,R1,#4
10 |  CMP R3,R4
11 |  BLT loop
12 |  HALT
13 | .ALIGN 256
14 | arrayLength: 10
15 | arrayData: 9
16 | 8
17 | 7
18 | 6
19 | 5
20 | 4
21 | 3
22 | 2
23 | 1
24 | 0
```

Load Save Edit

Processor

PC 0x0000002c
LR 0x00000000
SP 0x00100000
R12 0x00000000
R11 0x00000000
R10 0x00000000
R9 0x00000000
R8 0x00000000
R7 0x00000000
R6 0x00000000
R5 0x00000000
R4 0x00000100
R3 0x00000100
R2 0x00000104
R1 0x00000400
R0 0x0000002d

▶ || ◻

◀ ▶ ⚙

Count 1541

Current Instruction

Status bits NZCV 0110

Input/Output

Program HALTED. STOP, LOAD or EDIT

Memory

000	0x0	0x4	0x8	0xc
0x0000	0xe3a01000	0xe3a02f41	0xe3a03000	0xe3a04c01
0x0001	0xe2833001	0xe7925001	0xe0800005	0xe2811004
0x0002	0xe1530004	0xbafffff9	0xe1000070	0x00000000
0x0003	0x00000000	0x00000000	0x00000000	0x00000000
0x0004	0x00000000	0x00000000	0x00000000	0x00000000
0x0005	0x00000000	0x00000000	0x00000000	0x00000000
0x0006	0x00000000	0x00000000	0x00000000	0x00000000
0x0007	0x00000000	0x00000000	0x00000000	0x00000000
0x0008	0x00000000	0x00000000	0x00000000	0x00000000
0x0009	0x00000000	0x00000000	0x00000000	0x00000000
0x000a	0x00000000	0x00000000	0x00000000	0x00000000
0x000b	0x00000000	0x00000000	0x00000000	0x00000000
0x000c	0x00000000	0x00000000	0x00000000	0x00000000
0x000d	0x00000000	0x00000000	0x00000000	0x00000000
0x000e	0x00000000	0x00000000	0x00000000	0x00000000
0x000f	0x00000000	0x00000000	0x00000000	0x00000000
0x0010	0x0000000a	0x00000009	0x00000008	0x00000007
0x0011	0x00000006	0x00000005	0x00000004	0x00000003
0x0012	0x00000002	0x00000001	0x00000000	0x00000000
0x0013	0x00000000	0x00000000	0x00000000	0x00000000
0x0014	0x00000000	0x00000000	0x00000000	0x00000000
0x0015	0x00000000	0x00000000	0x00000000	0x00000000
0x0016	0x00000000	0x00000000	0x00000000	0x00000000
0x0017	0x00000000	0x00000000	0x00000000	0x00000000
0x0018	0x00000000	0x00000000	0x00000000	0x00000000
0x0019	0x00000000	0x00000000	0x00000000	0x00000000
0x001a	0x00000000	0x00000000	0x00000000	0x00000000
0x001b	0x00000000	0x00000000	0x00000000	0x00000000
0x001c	0x00000000	0x00000000	0x00000000	0x00000000
0x001d	0x00000000	0x00000000	0x00000000	0x00000000
0x001e	0x00000000	0x00000000	0x00000000	0x00000000
0x001f	0x00000000	0x00000000	0x00000000	0x00000000

Hex Clear

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23
[Documentation](#)

Exercise 9.4.1

Program

```
1 | MOV R1, #36
2 | MOV R2, #arrayData
3 | MOV R3, #0
4 | MOV R4, #arrayLength
5 | MOV R5, #0
6 | MOV R6, #invertedArray
7 | MOV R7, #0
8 | loop:
9 |   ADD R3,R3,#1
10 |  LDR R5,[R2+R1]
11 |  STR R5, [R6+R7]
12 |  SUB R1,R1,#4
13 |  ADD R7,R7,#4
14 |  CMP R3,#10
15 |  BLT loop
16 | stop:
17 | HALT
18 | .ALIGN 256
19 | arrayLength: 10
20 | arrayData: 9
21 | 8
22 | 7
23 | 6
24 | 5
25 | 4
26 | 3
27 | 2
28 | 1
29 | 0
30 | invertedArray: 0
31 | 0
32 | 0
33 | 0
34 | 0
35 | 0
36 | 0
37 | 0
38 | 0
39 | 0
```

Load Save Edit

Processor

PC 60
LR 0
SP 1048576
R12 0
R11 0
R10 0
R9 0
R8 0
R7 40
R6 300
R5 9
R4 256
R3 10
R2 260
R1 4294967292
R0 0

▶ || ◻

◀ ▶ ⚙

Count 78

Current Instruction

Status bits NZCV 0110

Input/Output

Program HALTED. STOP, LOAD or EDIT

Memory

000	0x0	0x4	0x8	0xc
0x0000	3818917924	3818925889	3818926080	3818933249
0x0001	3818934272	3818942283	3818942464	3800248321
0x0002	3885125633	3884339207	3795914756	3800526852
0x0003	3813867530	3137339384	3774873712	0
0x0004	0	0	0	0
0x0005	0	0	0	0
0x0006	0	0	0	0
0x0007	0	0	0	0
0x0008	0	0	0	0
0x0009	0	0	0	0
0x000a	0	0	0	0
0x000b	0	0	0	0
0x000c	0	0	0	0
0x000d	0	0	0	0
0x000e	0	0	0	0
0x000f	0	0	0	0
0x0010	10	9	8	7
0x0011	6	5	4	3
0x0012	2	1	0	0
0x0013	1	2	3	4
0x0014	5	6	7	8
0x0015	9	0	0	0
0x0016	0	0	0	0
0x0017	0	0	0	0
0x0018	0	0	0	0
0x0019	0	0	0	0
0x001a	0	0	0	0
0x001b	0	0	0	0
0x001c	0	0	0	0
0x001d	0	0	0	0
0x001e	0	0	0	0
0x001f	0	0	0	0

Decimal (unsigned) Clear

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23
[Documentation](#)

Exercise 9.4.2

Program

```
1 | MOV R0,#0
2 | MOV R1,#36
3 | MOV R2,#0
4 | MOV R3,#arrayData
5 | loop:
6 |   ADD R2,R2,#1
7 |   LDR R4,[R3+R0]
8 |   LDR R5,[R3+R1]
9 |   MOV R6,R5
10 |  MOV R5,R4
11 |  MOV R4,R6
12 |  STR R4,[R3+R0]
13 |  STR R5,[R3+R1]
14 |  ADD R0,R0,#4
15 |  SUB R1,R1,#4
16 |  CMP R2,#5
17 |  BLT loop
18 | stop:
19 |   HALT
20 | .ALIGN 256
21 | arrayLength: 10
22 | arrayData: 9
23 |   8
24 |   7
25 |   6
26 |   5
27 |   4
28 |   3
29 |   2
30 |   1
31 |   0
```

Load Save Edit

Processor

PC 68
LR 0
SP 1048576
R12 0
R11 0
R10 0
R9 0
R8 0
R7 0
R6 4
R5 5
R4 4
R3 260
R2 5
R1 16
R0 20

▶ ⏸ ◻

◀ ▶ ⚙

Count 65

Current Instruction

Status bits NZCV 0110

Input/Output

Program HALTED. STOP, LOAD or EDIT

Memory

000	0x0	0x4	0x8	0xc
0x0000	3818913792	3818917924	3818921984	3818929985
0x0001	3800178689	3885187072	3885191169	3785383941
0x0002	3785379844	3785375750	3884138496	3884142593
0x0003	3800039428	3795914756	3813801989	3137339379
0x0004	3774873712	0	0	0
0x0005	0	0	0	0
0x0006	0	0	0	0
0x0007	0	0	0	0
0x0008	0	0	0	0
0x0009	0	0	0	0
0x000a	0	0	0	0
0x000b	0	0	0	0
0x000c	0	0	0	0
0x000d	0	0	0	0
0x000e	0	0	0	0
0x000f	0	0	0	0
0x0010	10	0	1	2
0x0011	3	4	5	6
0x0012	7	8	9	0
0x0013	0	0	0	0
0x0014	0	0	0	0
0x0015	0	0	0	0
0x0016	0	0	0	0
0x0017	0	0	0	0
0x0018	0	0	0	0
0x0019	0	0	0	0
0x001a	0	0	0	0
0x001b	0	0	0	0
0x001c	0	0	0	0
0x001d	0	0	0	0
0x001e	0	0	0	0
0x001f	0	0	0	0

Decimal (unsigned) Clear

ARMLite Simulator V1.2.4 © Peter Higginson 2020-23
[Documentation](#)