Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and **Valerio Selis**

dmc@liverpool.ac.uk
V.Selis@liverpool.ac.uk

UNIVERSITY OF LIVERPOOL

Week 02 – Lecture 04
Microprocessor Systems

UNIVERSITY OF
LIVERPOOL

Question

Registers r1, r3 and r15 hold the values 0xCCDDEEFF, 0xFEDCBA98 and 0x00000108 respectively.

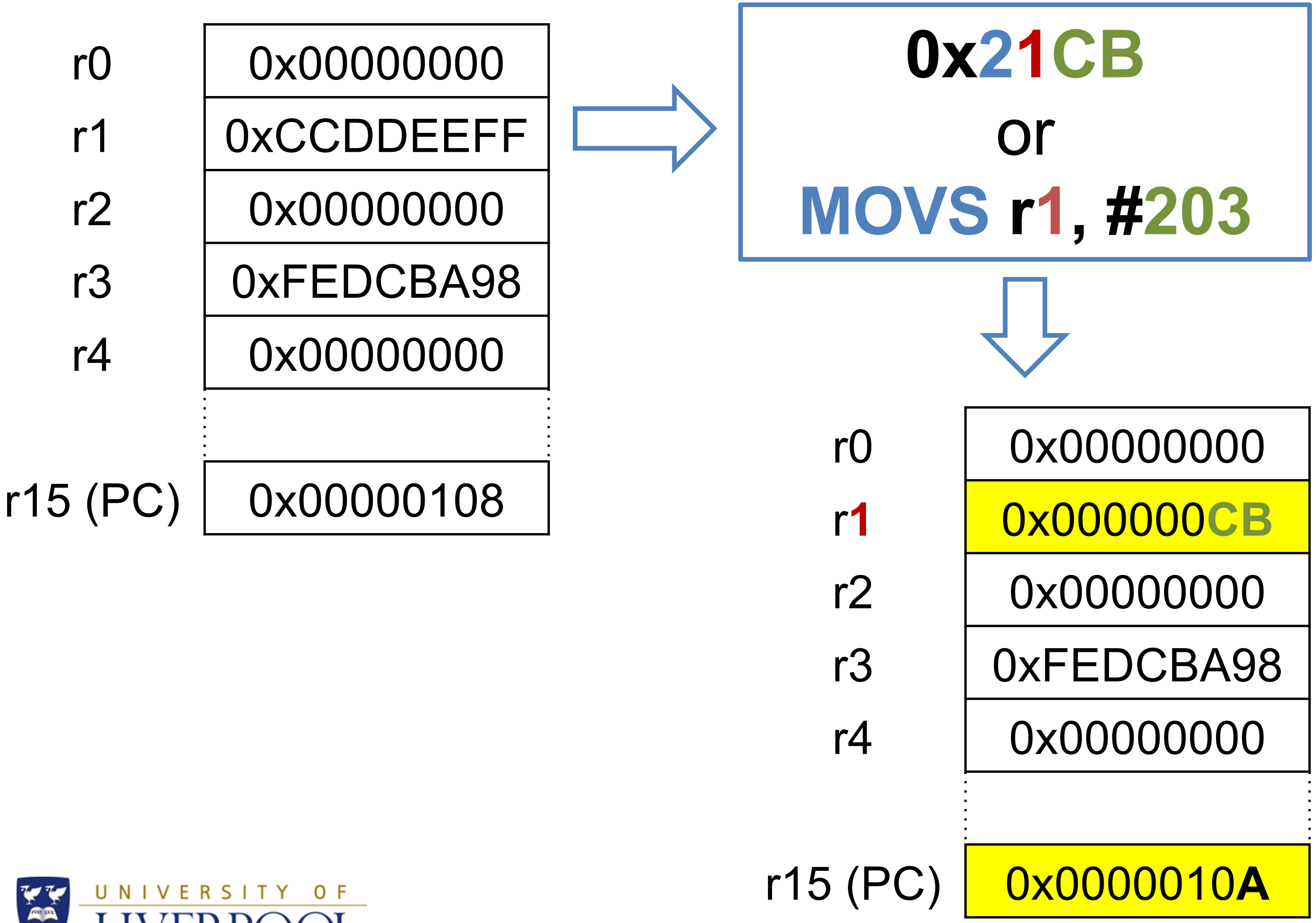
What values are held by registers r1, r3 and r15 after the execution of the instruction with machine code 0x21CB?

Answer

- The machine code 0x21CB means:
  - move the value CB_{16} into register r1
- So after the instruction is executed:
  - register r1 holds the value 0x000000CB,
  - register r3 is unchanged (0xFEDCBA98) and
  - register r15 is incremented by 2 and holds the value 0x0000010A ($A_{16} = 10_{10}$)

Answer

Register bank



r0	0x00000000
r1	0xCCDDEEFF
r2	0x00000000
r3	0xFEDCBA98
r4	0x00000000
r15 (PC)	0x00000108

0x21CB

or

MOVS r1, #203

r0	0x00000000
r1	0x000000CB
r2	0x00000000
r3	0xFEDCBA98
r4	0x00000000
r15 (PC)	0x0000010A

Question

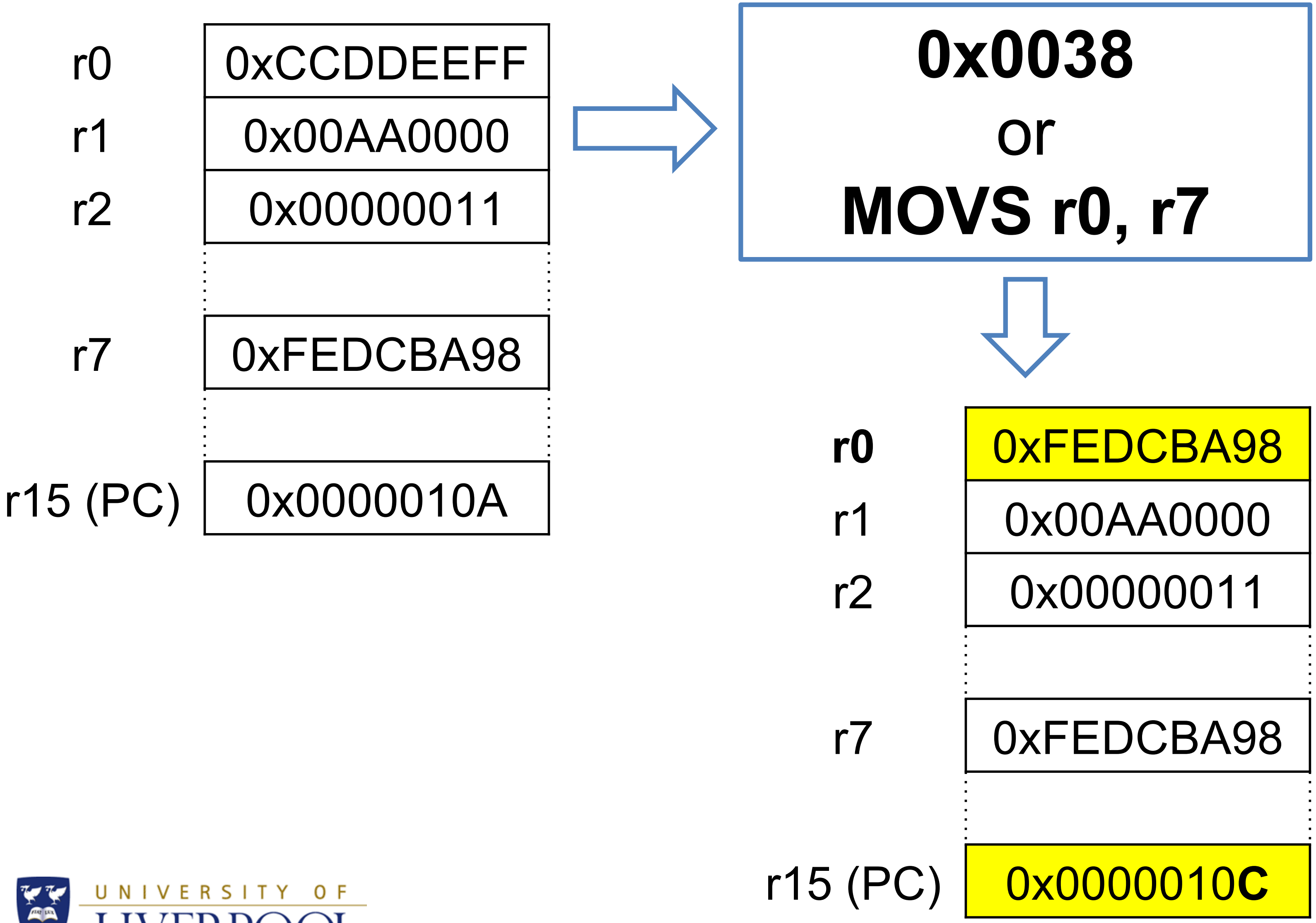
Registers r0, r7 and r15 hold the values 0xCCDDEEFF, 0xFEDCBA98 and 0x0000010A respectively.

What values are held by registers r0, r7 and r15 after the execution of the instruction with machine code 0x0038?

Answer

- The machine code 0x0038 or 0000 0000 0011 1000 means:
  - move into register r0 the value held in register r7
- So after the instruction is executed:
  - register r0 holds the value 0xFEDCBA98,
  - register r7 is unchanged (0xFEDCBA98) and
  - register r15 is incremented by 2 and holds the value 0x0000010C

Answer



Register bank

r0	0xCCDDEEFF
r1	0x00AA0000
r2	0x00000011
r7	0xFEDCBA98
r15 (PC)	0x0000010A

0x0038

or

MOVS r0, r7

r0	0xFEDCBA98
r1	0x00AA0000
r2	0x00000011
r7	0xFEDCBA98
r15 (PC)	0x0000010**C**

Question

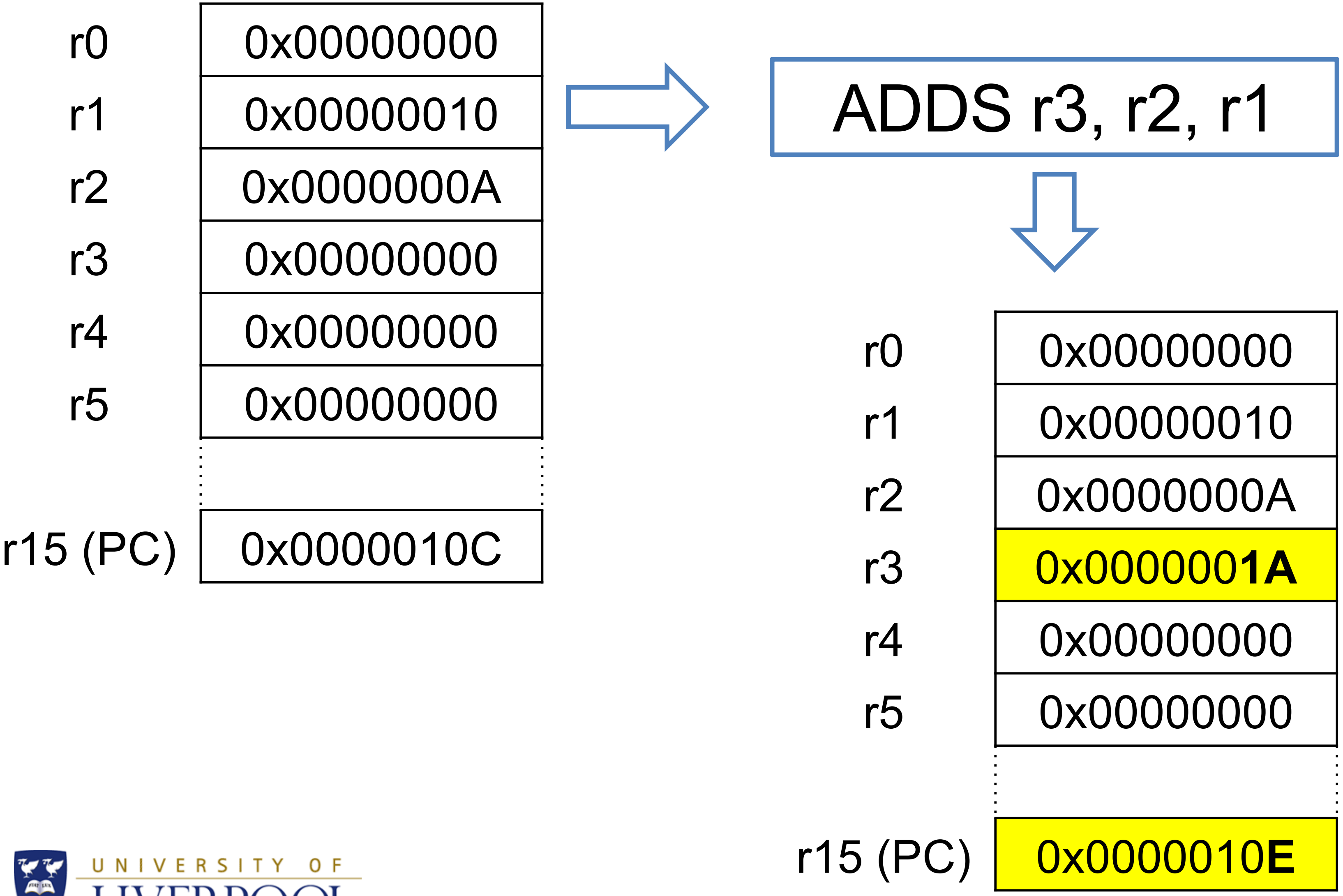
Registers r1, r2 and r15 hold the values 0x00000010, 0x0000000A and 0x0000010C respectively.
What values are held in r1, r2, r3, r4, r5 and r15 after the execution of the following?

```
    ADDS r3,r2,r1
    ADDS r4,r1,r2
    SUBS r5,r1,r2
```

(all 16 bit instructions)

Answer

Register bank



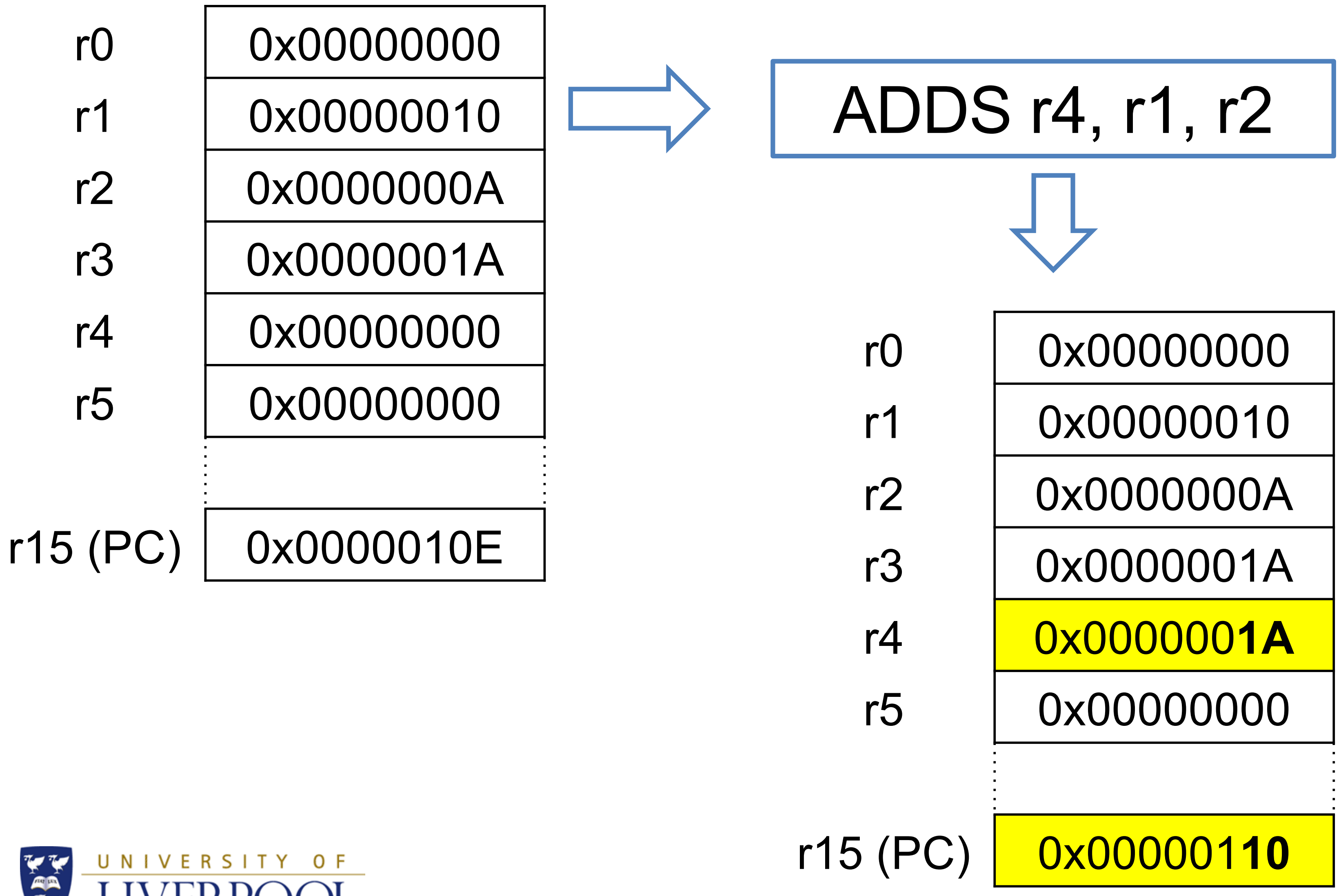
r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x00000000
r4	0x00000000
r5	0x00000000
	⋮
r15 (PC)	0x0000010C

ADDS r3, r2, r1

r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x000000**1A**
r4	0x00000000
r5	0x00000000
	⋮
r15 (PC)	0x0000010**E**

Answer

Register bank



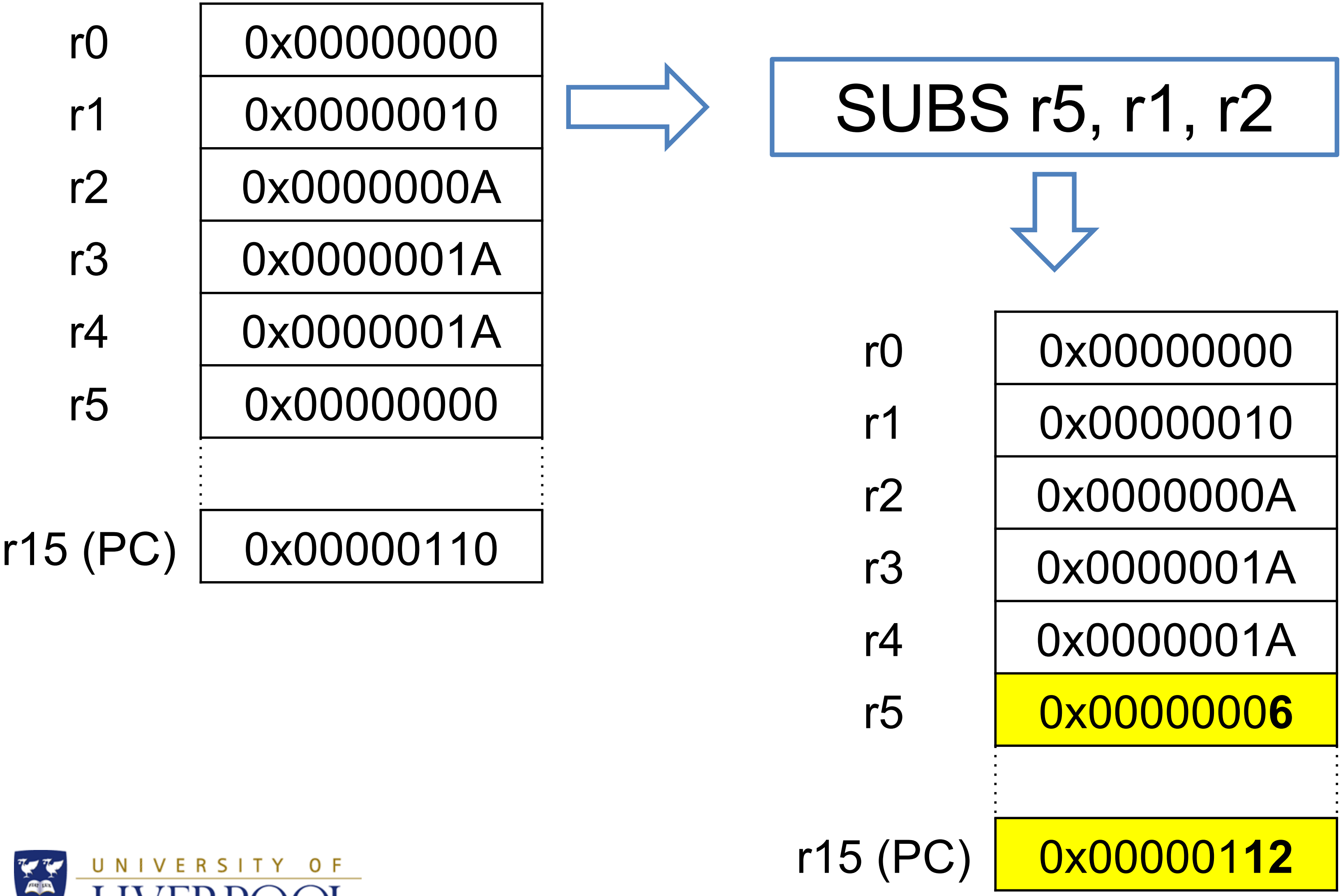
r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x0000001A
r4	0x00000000
r5	0x00000000
r15 (PC)	0x0000010E

ADDS r4, r1, r2

r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x0000001A
r4	0x000000**1A**
r5	0x00000000
r15 (PC)	0x000001**10**

Answer

Register bank



r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x0000001A
r4	0x0000001A
r5	0x00000000
r15 (PC)	0x00000110

SUBS r5, r1, r2

r0	0x00000000
r1	0x00000010
r2	0x0000000A
r3	0x0000001A
r4	0x0000001A
r5	0x0000000**6**
r15 (PC)	0x00000**112**

Answer

ADDS r3, r2, r1 means add value in r1 to value in r2 and put the sum in r3 so r3 holds 0x0000001A

ADDS r4, r1, r2 means add value in r2 to value in r1 and put the sum in r4 so r4 holds 0x0000001A

SUBS r5, r1, r2 means subtract value in r2 from value in r1 and put the difference in r5 so r5 holds 0x00000006

The program counter, r15, is incremented by 6 (3 instructions) and holds the value 0x00000112.

Registers r1 and r2 are unchanged.

Question

- Registers r1, r2, r3, r7 and r15 hold the values 0x00001020, 0x00000005, 0xABC00000, 0x00000010 and 0x00000112 respectively.
- What values are held in r1, r2, r3, r7 and r15 after the execution of the following?

```
        MULS r1, r2, r1
        MULS r3, r7, r3
```

(both 16 bit instructions)

Answer

MULS r1, r2, r1 means multiply value in r1 to value in r2 and put the product in r1 so r1 holds 0x000050A0

r1 = r2 · r1 = 0x00001020 · 0x00000005 = 0x000050A0

MULS r3, r7, r3 means multiply value in r3 to value in r7 and put the product (0xABC000000) in r3 so r3 holds 0xBC000000 - the lowest 32 bits.

r3 = r3 · r7; 0xABC00000 · 0x00000010 = 0xABC000000;
r3 = 0xBC000000

The program counter, r15, is incremented by 4 (two instructions) and holds the value 0x00000116.

Registers r2 and r7 are unchanged.

Week 02 – Lecture 05
Microprocessor Systems

UNIVERSITY OF LIVERPOOL

Question

- Register r1 holds the value 0x00000101 and registers r2, r3, r4, r5 and r6 all hold the value 0x00000011.
- What values are held in r1, r2, r3, r4, r5 and r6 after the execution of the following?

```
ANDS r2, r1
ORRS r3, r1
EORS r4, r1
BICS r5, r1
BICS r1, r6
```

Answer

ANDS r2, r1 means AND the value in r1, 0x00000101, with the value in r2, 0x00000011, and put the result in r2

```
     0000 0000 0000 0000 0000 0000 0001 0001
AND  0000 0000 0000 0000 0000 0001 0000 0001
     ---------------------------------------
=    0000 0000 0000 0000 0000 0000 0000 0001
```

So register r2 holds the value 0x00000001.

Answer

ORRS r3, r1 means OR the value in r3, 0x00000011, with the value in r1, 0x00000101, and put the result in r3

```
      0000 0000 0000 0000 0000 0000 0001 0001
OR    0000 0000 0000 0000 0000 0001 0000 0001
=     0000 0000 0000 0000 0000 0001 0001 0001
```

So register r3 holds the value 0x00000111.

Answer

EORS r4, r1 means XOR the value in r1, 0x00000101, with the value in r4, 0x00000011, and put the result in r4

```
       0000 0000 0000 0000 0000 0000 0001 0001
XOR    0000 0000 0000 0000 0000 0001 0000 0001
       ---------------------------------------
=      0000 0000 0000 0000 0000 0001 0001 0000
```

So register r4 holds the value 0x00000110.

Answer

BICS r5, r1 means ‘bit clear’ the value in r5, 0x00000011, with the value in r1, 0x00000101, and put the result in r5

0000 0000 0000 0000 0000 0000 0001 0001

BIC 0000 0000 0000 0000 0000 0001 0000 0001

= 0000 0000 0000 0000 0000 0000 0001 0000

So register r5 holds the value 0x00000010.

Answer

BICS r1, r6 means ‘bit clear’ the value in r1, 0x00000101, with the value in r6, 0x00000011, and put the result in r1

	0000 0000 0000 0000 0000 0001 0000 0001
BIC	0000 0000 0000 0000 0000 0000 0001 0001
=	0000 0000 0000 0000 0000 0001 0000 0000

So register r1 holds the value 0x00000100.

Question

What values are held in r4, r6 and r7 after the execution of the following?

```
MOVS r4, #17
ADDS r7, r4, #4
SUBS r6, r7, #7
ADDS r4, r4, #250
```

Answer

MOVS r4, #17

- so r4 holds the value 17_{10} or 0x00000011

ADDS r7, r4, #4 ⏩ 4 + 17 = 21

- so r7 holds the value 21_{10} or 0x00000015

SUBS r6, r7, #7 ⏩ 21 - 7 = 14

- so r6 holds the value 14 or 0x0000000E

ADDS r4, r4, #250 ⏩ 250 + 17 = 267

- so r4 holds the value 267 or 0x0000010B

Question

Which of the following instructions uses an immediate value that is allowed in the ARM Cortex M0?

MOVS r7, #0x10B ; 267_{10}
ADDS r3, r3, #0x0A4 ; 164_{10}
SUBS r0, r6, #0x03 ; 3_{10}
ADDS r5, r2, #0x0A4 ; 164_{10}

UNIVERSITY OF LIVERPOOL

Answer

No	0x10B is a 9 bit value, greater than 255_{10}
Yes	0x0A4 is an 8 bit value, less than 256_{10}
Yes	0x003 is a 2 bit value, less than 8_{10}
No	0x0A4 is an 8 bit value, greater than 7_{10}

Note that the last 2 instructions have different source and destination registers so that the immediate value is limited to 3 bits.

Week 02 – Lecture 06
Microprocessor Systems

Question

0x4D is stored in memory at address 0x00006000,
0xA0 is stored at 0x00006001,
0x94 is stored at 0x00006002 and
0x0F is stored at 0x00006003.

r7 holds the value 0x00006000.

What value is held in r0 after the execution of

```
LDR r0, [r7]
```

for (a) little endian and (b) big endian?

Answer

a) little endian – little end at the lowest memory address so r0 holds (0x4D would be the least significant byte)

0x0F94A04D

b) big endian – big end at the lowest memory address so r0 holds (0x4D would be the most significant byte)

0x4DA0940F

Question

Registers r1, r2 and r3 hold the values 0xFFBEADDE, 0xE5AFDCBA and 0xE5A55ED1 respectively.

What values are held in r4, r5 and r6 after the execution of the following?

```
REV r6, r2
REV r4, r1
REV r5, r3
```

Answer

REV r6, r2

so r6 holds 0xBADCAFE5

REV r4, r1

so r4 holds 0xDEADBEFF

REV r5, r3

so r5 holds 0xD15EA5E5

Question

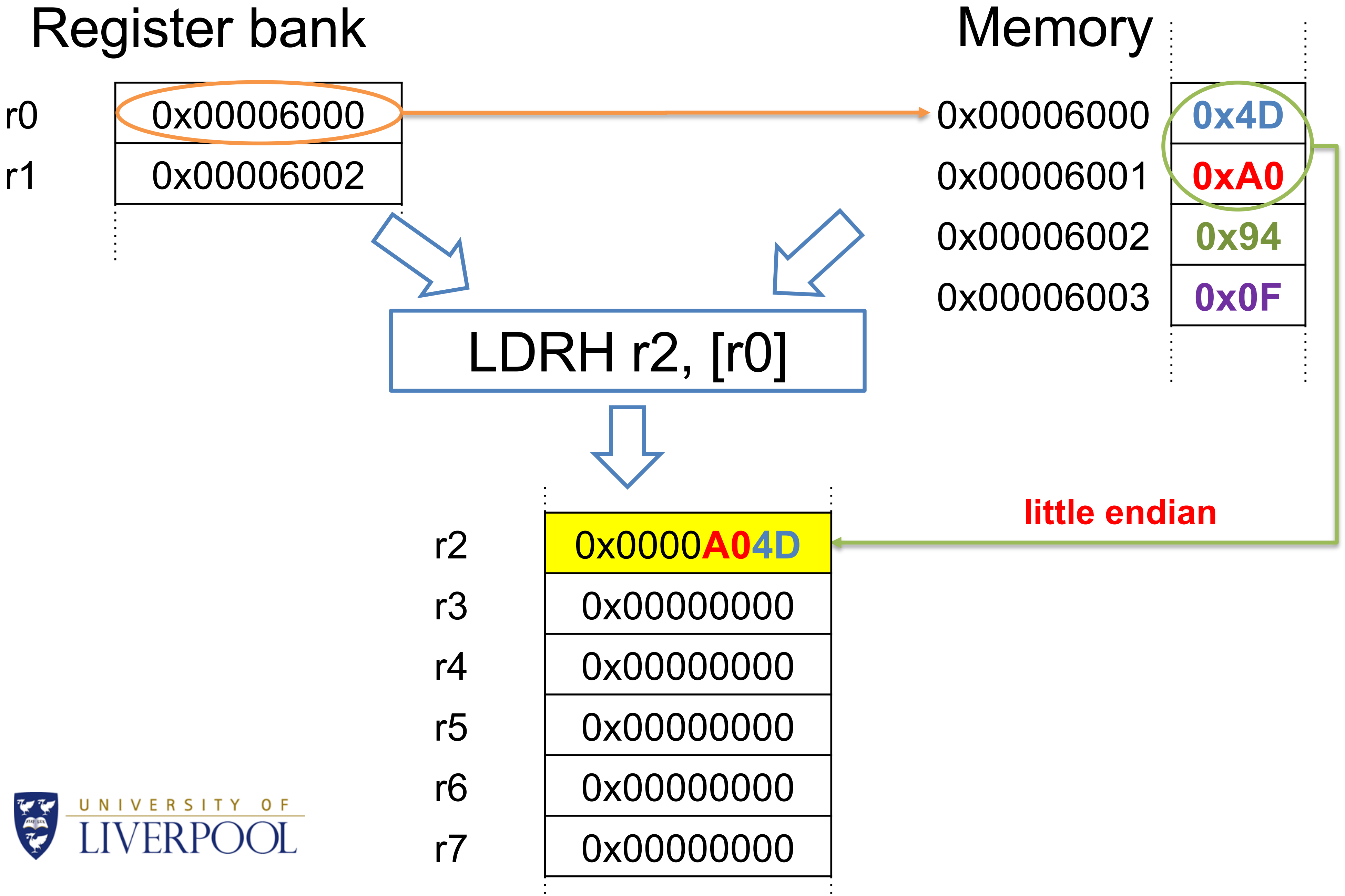
0x4D at 0x00006000
0xA0 at 0x00006001
0x94 at 0x00006002
0x0F at 0x00006003

r0 holds 0x00006000
r1 holds 0x00006002

What values are held in r2 - r7 after the execution of the following assuming little endian?

```
LDRH r2, [r0]        LDRSH r3, [r0]
LDRSH r4, [r1]       LDRB r5, [r1]
LDRSB r6, [r0]       LDRSB r7, [r1]
```

Answer



Register bank

r0	0x00006000
r1	0x00006002

Memory

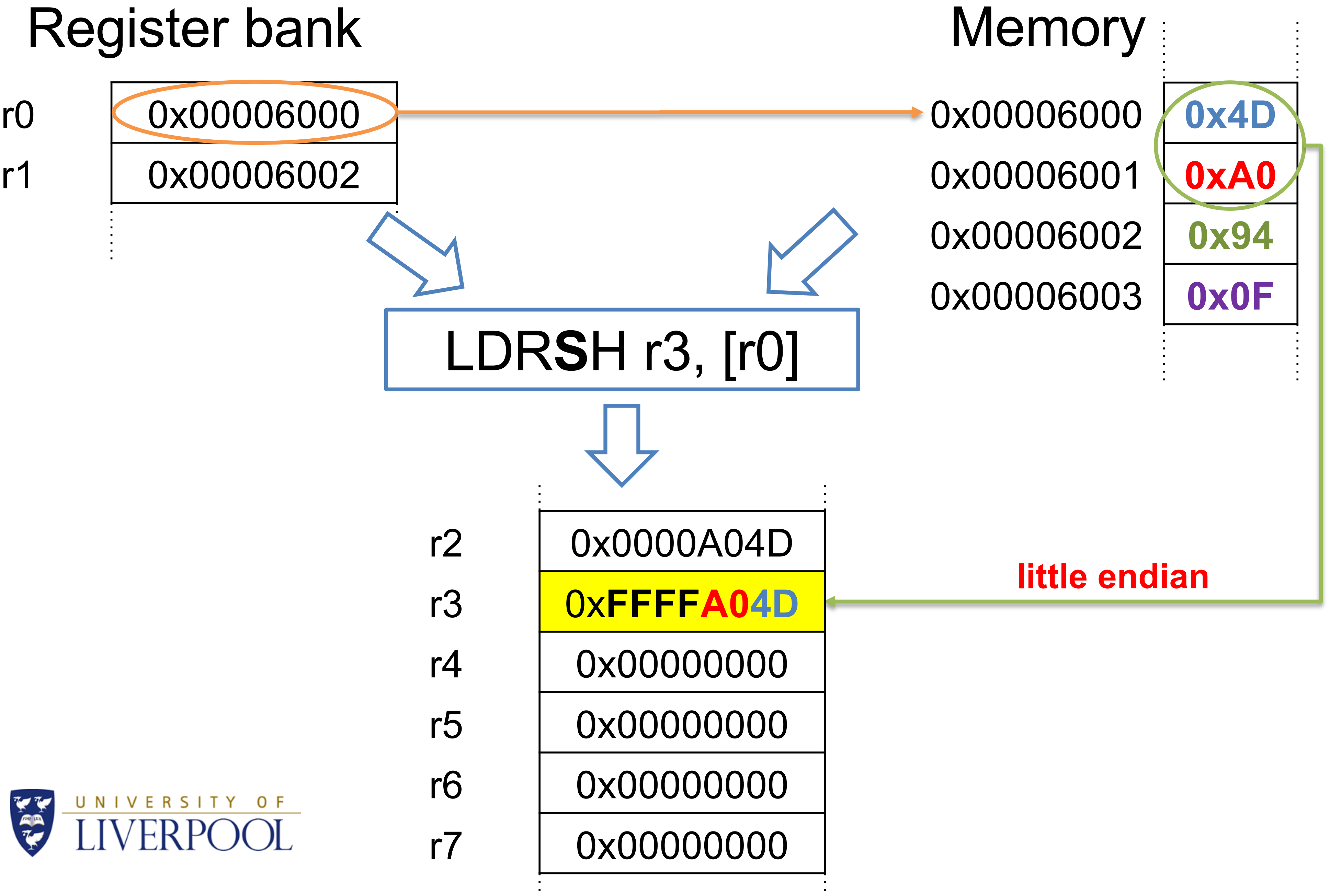
0x00006000	0x4D
0x00006001	0xA0
0x00006002	0x94
0x00006003	0x0F

LDRH r2, [r0]

r2	0x0000A04D
r3	0x00000000
r4	0x00000000
r5	0x00000000
r6	0x00000000
r7	0x00000000

little endian

Answer



Register bank

r0	0x00006000
r1	0x00006002

Memory

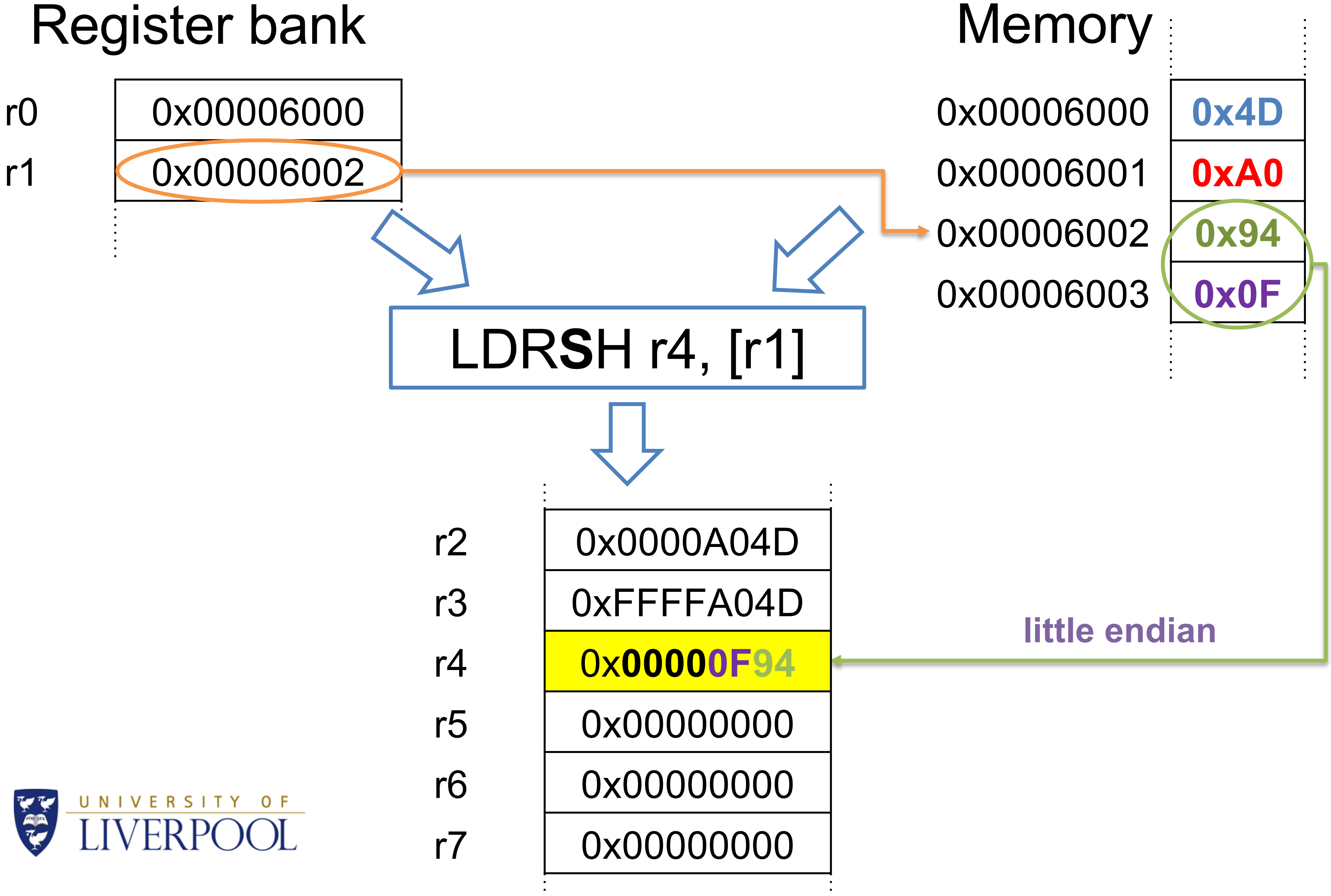
0x00006000	0x4D
0x00006001	0xA0
0x00006002	0x94
0x00006003	0x0F

LDR**S**H r3, [r0]

r2	0x0000A04D
r3	0x**FFFFA04D**
r4	0x00000000
r5	0x00000000
r6	0x00000000
r7	0x00000000

little endian

Answer



Register bank

r0	0x00006000
r1	0x00006002

Memory

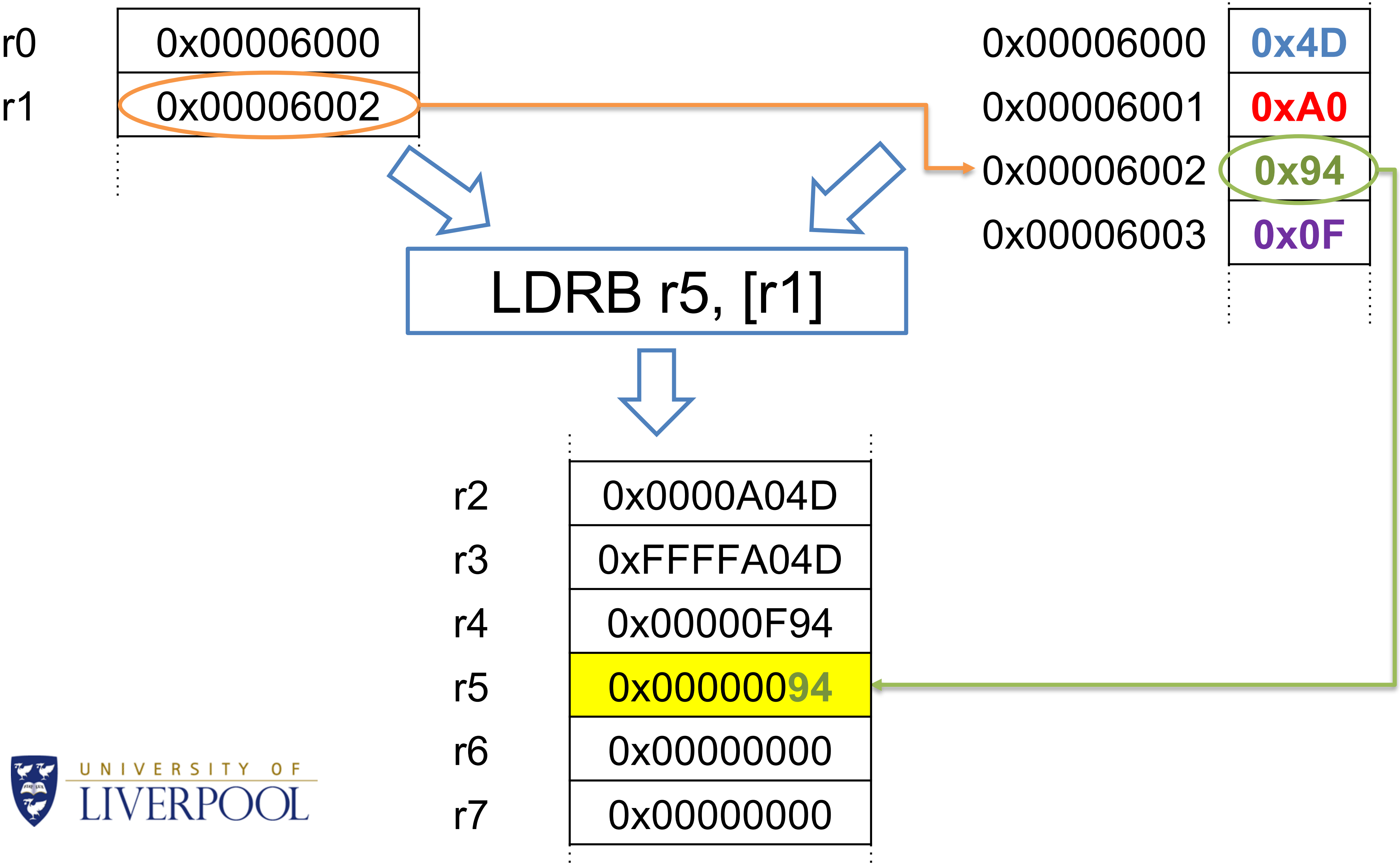
0x00006000	**0x4D**
0x00006001	**0xA0**
0x00006002	**0x94**
0x00006003	**0x0F**

LDR**S**H r4, [r1]

r2	0x0000A04D
r3	0xFFFFA04D
r4	0x**00000F94**
r5	0x00000000
r6	0x00000000
r7	0x00000000

little endian

Answer



Register bank

r0	0x00006000
r1	0x00006002

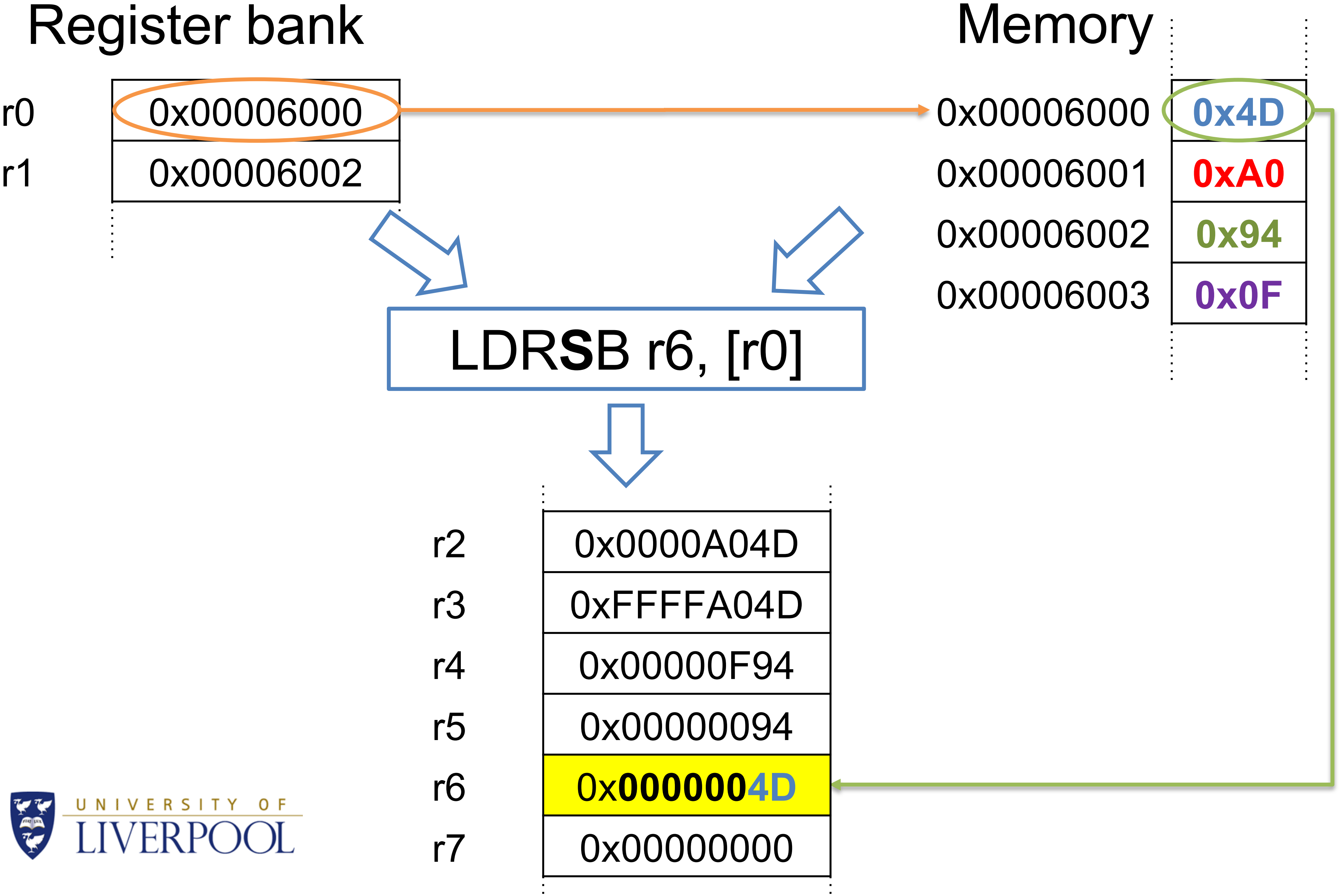
Memory

0x00006000	0x4D
0x00006001	0xA0
0x00006002	0x94
0x00006003	0x0F

LDRB r5, [r1]

r2	0x0000A04D
r3	0xFFFFA04D
r4	0x00000F94
r5	0x00000094
r6	0x00000000
r7	0x00000000

Answer



Register bank

r0	0x00006000
r1	0x00006002

Memory

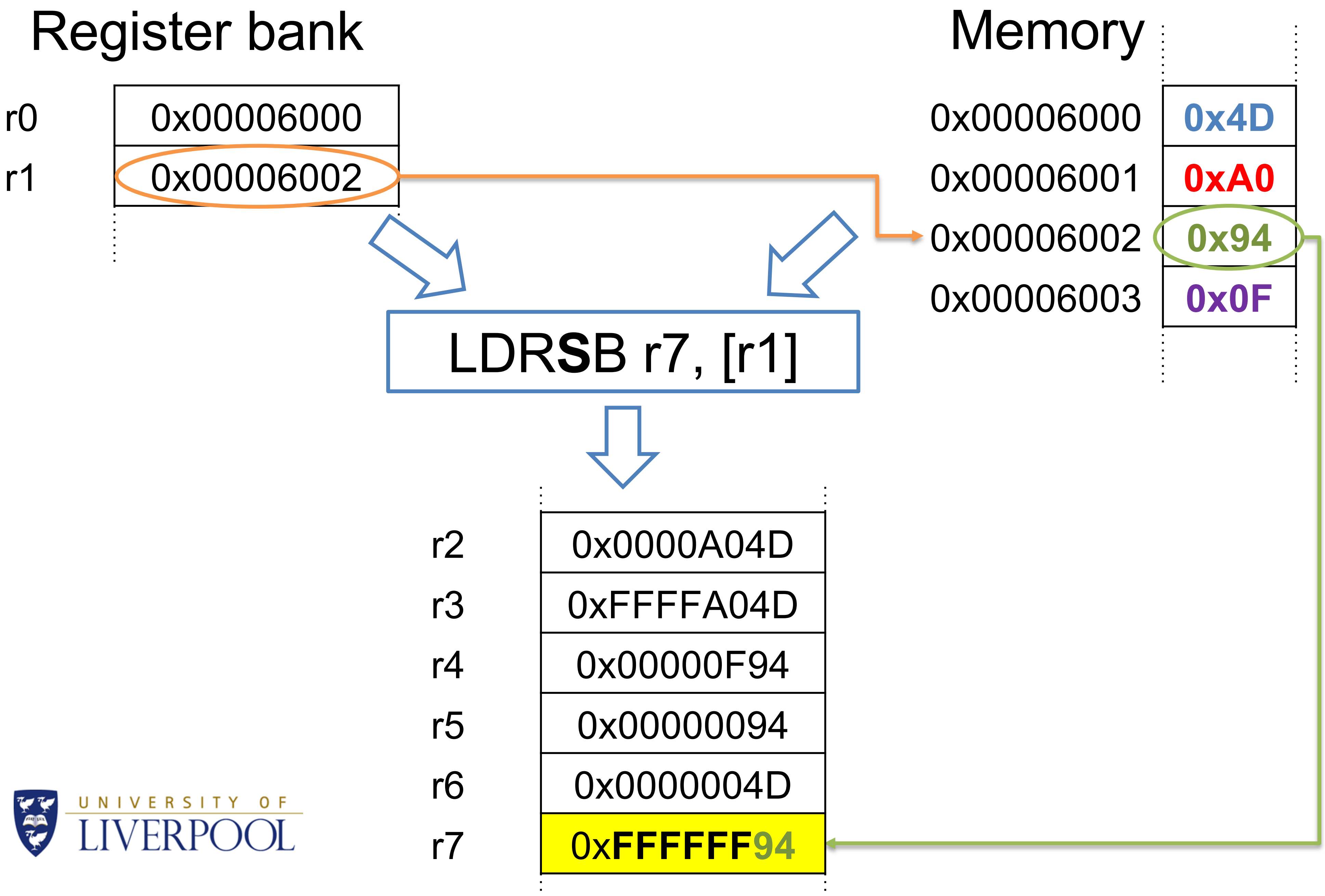
0x00006000	0x4D
0x00006001	0xA0
0x00006002	0x94
0x00006003	0x0F

LDR**S**B r6, [r0]

r2	0x0000A04D
r3	0xFFFFA04D
r4	0x00000F94
r5	0x00000094
r6	0x**000000**4D
r7	0x00000000

UNIVERSITY OF LIVERPOOL

Answer



Register bank

r0	0x00006000
r1	0x00006002

Memory

0x00006000	**0x4D**
0x00006001	**0xA0**
0x00006002	**0x94**
0x00006003	**0x0F**

LDR**S**B r7, [r1]

r2	0x0000A04D
r3	0xFFFFA04D
r4	0x00000F94
r5	0x00000094
r6	0x0000004D
r7	0x**FFFFFF94**

Answer

LDRH r2, [r0]	so r2 holds 0x0000A04D
LDRSH r3, [r0]	so r3 holds 0xFFFFA04D
LDRSH r4, [r1]	so r4 holds 0x00000F94
LDRB r5, [r1]	so r5 holds 0x00000094
LDRSB r6, [r0]	so r6 holds 0x0000004D
LDRSB r7, [r1]	so r7 holds 0xFFFFFF94