Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and **Valerio Selis**

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

UNIVERSITY OF LIVERPOOL

Week 03 – Lecture 07
Microprocessor Systems

Question

What values are held in r4, r5 and r15 after the execution of the following?

Memory Address	Instruction
0x00008000	SUBS r4, r7, r7
0x00008002	BEQ 0x00008006
0x00008004	MOVS r4, #17
0x00008006	ADDS r5, r4, #6

UNIVERSITY OF LIVERPOOL

Answer

Register bank

r4	0x00112233
r5	0x22CC33DD
r7	0x00AA11BB
r15 (PC)	0x00008000

SUBS r4, r7, r7

r4	**0x00000000**
r5	0x22CC33DD
r7	0x00AA11BB
r15 (PC)	**0x00008002**

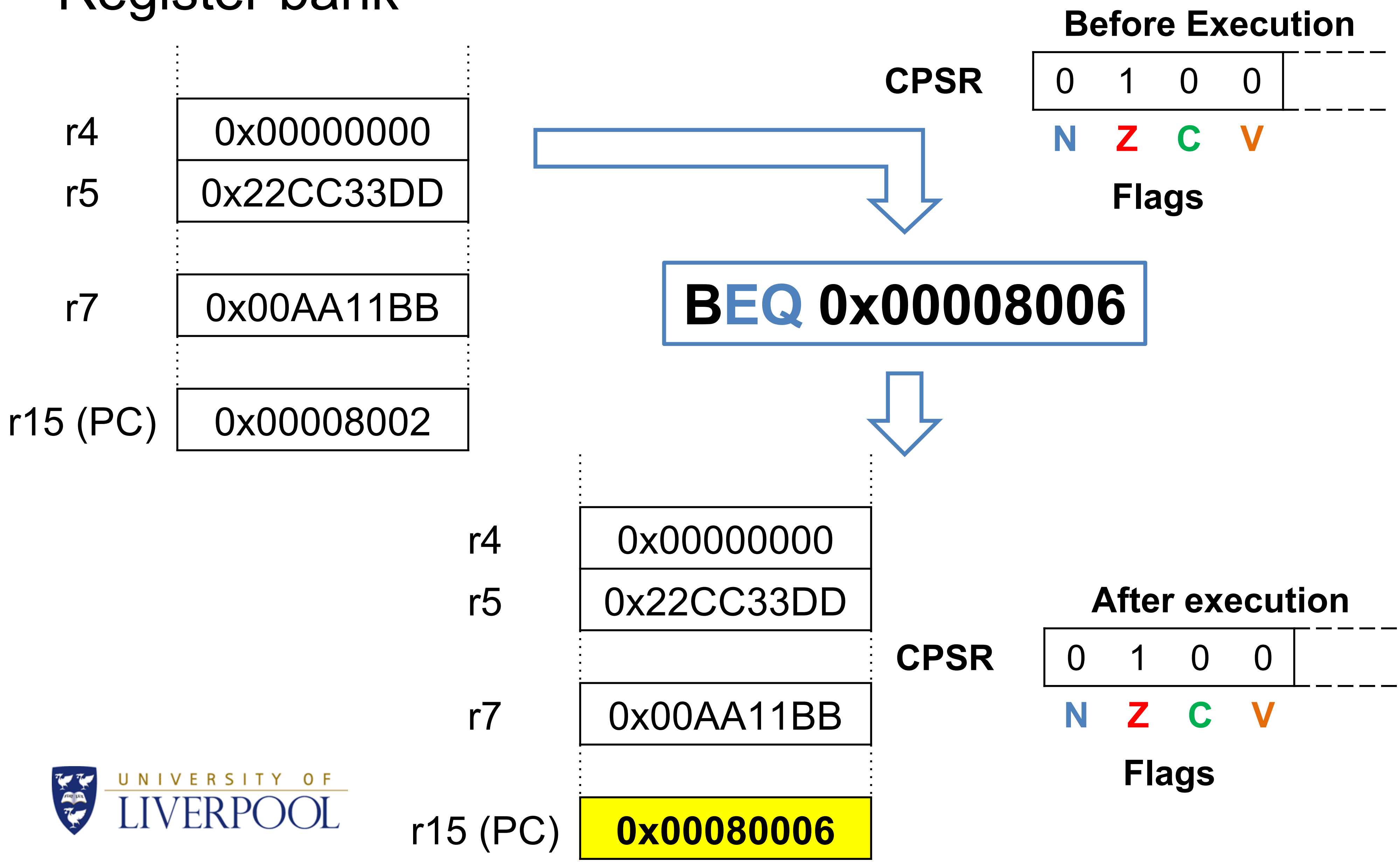
After execution

	N	Z	C	V
CPSR	**0**	**1**	**0**	**0**

Flags

Answer

Register bank



Answer

Register bank

r4	0x00000000
r5	0x22CC33DD
r7	0x00AA11BB
r15 (PC)	0x00008006

Before Execution

CPSR

N	Z	C	V
0	1	0	0

Flags

ADDS r5, r4, #6

r4	0x00000000
r5	**0x00000006**
r7	0x00AA11BB
r15 (PC)	**0x00080008**

After execution

CPSR

N	Z	C	V
0	**0**	**0**	**0**

Flags

UNIVERSITY OF LIVERPOOL

Answer

SUBS r4, r7, r7

- r4 holds the value 0x00000000 and Z is set.

BEQ 0x00008006

- zero flag is set therefore load program counter with new value 0x00008006 i.e. skip the instruction at address 0x00008004

ADDS r5, r4, #6

- r5 will hold 0x00000006. ($6_{10} + 0 = 6_{10} = 6_{16}$)

The program counter, r15, will hold 0x00008008.

Question

If r5 holds the value 0xFFFFFFFF, what happens to the zero and carry flags after each addition in the following programme?

```
ADDS r4, r5, #0      ;add 0 to r5
ADDS r4, r5, #1      ;add 1 to r5
ADDS r4, r5, #2      ;add 2 to r5
```

UNIVERSITY OF LIVERPOOL

Answer

Register bank

r4	0x00112233
r5	0xFFFFFFFF

ADDS r4, r5, #0

r4	**0xFFFFFFFF**
r5	0xFFFFFFFF

After execution

CPSR

N	Z	C	V
1	**0**	**0**	**0**

Flags

UNIVERSITY OF LIVERPOOL

Answer

Register bank

Before Execution

r4	0xFFFFFFFF
r5	0xFFFFFFFF

CPSR

N	Z	C	V
1	0	0	0

Flags

ADDS r4, r5, #1

After execution

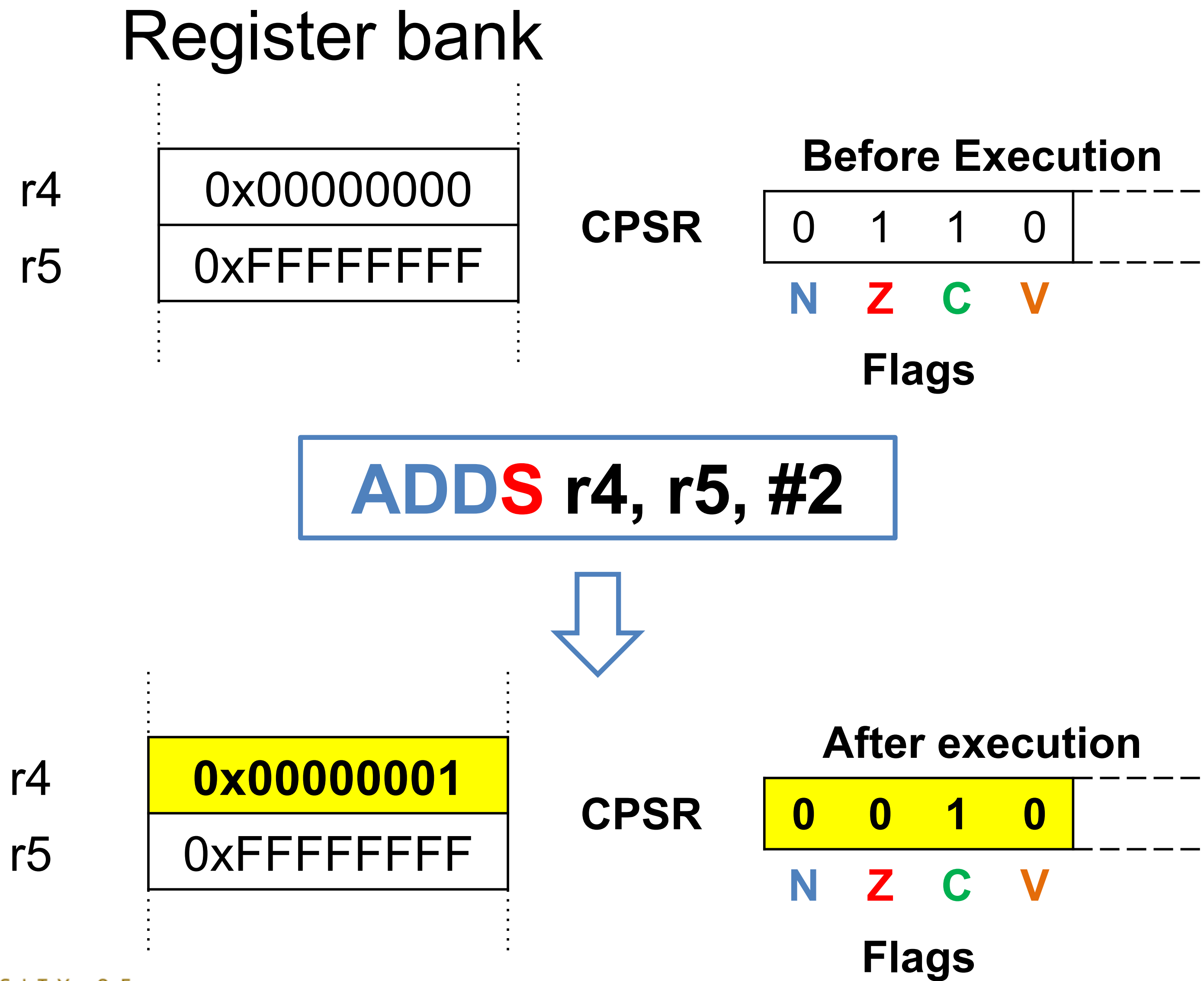
r4	**0x00000000**
r5	0xFFFFFFFF

CPSR

N	Z	C	V
0	**1**	**1**	**0**

Flags

Answer



UNIVERSITY OF LIVERPOOL

Answer

ADDS r4, r5, #0

- r4 holds the value 0xFFFFFFFF and both the zero flag, Z, and the carry flag, C, are cleared.

ADDS r4, r5, #1

- r4 holds the value 0x00000000 and both the zero flag, Z, and the carry flag, C, are set.

ADDS r4, r5, #2

- r4 holds the value 0x00000001, the zero flag, Z, is cleared and the carry flag, C, is set.

Week 03 – Lecture 08
Microprocessor Systems

UNIVERSITY OF LIVERPOOL

Question

What is the two’s complement of the following numbers in 32 bits?

$-1{,}500{,}000{,}000_{10}$ ($1{,}500{,}000{,}000_{10}$ = 0x59682F00)
-211_{10} (211_{10} = 0x000000D3)
-2017_{10} (2017_{10} = 0x000007E1)

Answer

Original No:	0 1 2 3 4 5 6 7 8 9 A B C D E F
Inverted No:	F E D C B A 9 8 7 6 5 4 3 2 1 0

1st : the positive value is: $1{,}500{,}000{,}000_{10}$ or 0x59682F00

2nd : invert all bits 0x59682F00 ➔ 0xA697D0FF

3rd : add 1 to the result:
0xA697D0FF + 1 = 0xA697D100

Result: 0xA697D100 is the 2's complement representation of $-1{,}500{,}000{,}000_{10}$

Answer

Original No:	0 1 2 3 4 5 6 7 8 9 A B C D E F
Inverted No:	F E D C B A 9 8 7 6 5 4 3 2 1 0

1st : the positive value is: 211_{10} or 0x000000D3

2nd : invert all bits 0x000000D3 ➔ 0xFFFFFF2C

3rd : add 1 to the result:

0xFFFFFF2C + 1 = 0xFFFFFF2D

Result: 0xFFFFFF2D is the 2's complement representation of -211_{10}

Answer

Original No:	0 1 2 3 4 5 6 7 8 9 A B C D E F
Inverted No:	F E D C B A 9 8 7 6 5 4 3 2 1 0

1st : the positive value is: 2017_{10} or 0x000007E1

2nd : invert all bits 0x000007E1 ➔ 0xFFFFF81E

3rd : add 1 to the result:
0xFFFFF81E + 1 = 0xFFFFF81F

Result: 0xFFFFF81F is the 2's complement representation of -2017_{10}

Week 03 – Lecture 09
Microprocessor Systems

UNIVERSITY OF
LIVERPOOL

Question

What value is held in register r4 after the 'add with carry' instruction, ADCS r4, r5, is executed assuming the initial value in register r4 is 0x000000FF (= $+255_{10}$) and the value in r5 is 0xFFFFFF00 (= -256_{10}) when

(i) the carry flag is clear and

(ii) the carry flag is set?

UNIVERSITY OF LIVERPOOL

Answer – carry flag clear

Register bank

r4	0x000000FF
r5	0xFFFFFF00

CPSR

Before Execution

N	**Z**	**C**	**V**
0	0	**0**	0

Flags

ADCS r4, r5

$$
\begin{array}{rl}
 & 0x\ 00\ 00\ 00\ FF \\
+ & 0x\ FF\ FF\ FF\ 00 \\
+ & \underline{0} \\
\hline
 & 0x\ FF\ FF\ FF\ FF
\end{array}
$$

r4	**0xFFFFFFFF**
r5	0xFFFFFF00

UNIVERSITY OF LIVERPOOL

Answer – carry flag set

Register bank

r4	0x000000FF
r5	0xFFFFFF00

Before Execution

CPSR

N	Z	C	V
0	0	1	0

Flags

ADCS r4, r5

```
  0x 00 00 00 FF
+ 0x FF FF FF 00
+              1
----------------
  0x 00 00 00 00
```

r4	**0x00000000**
r5	0xFFFFFF00

UNIVERSITY OF LIVERPOOL

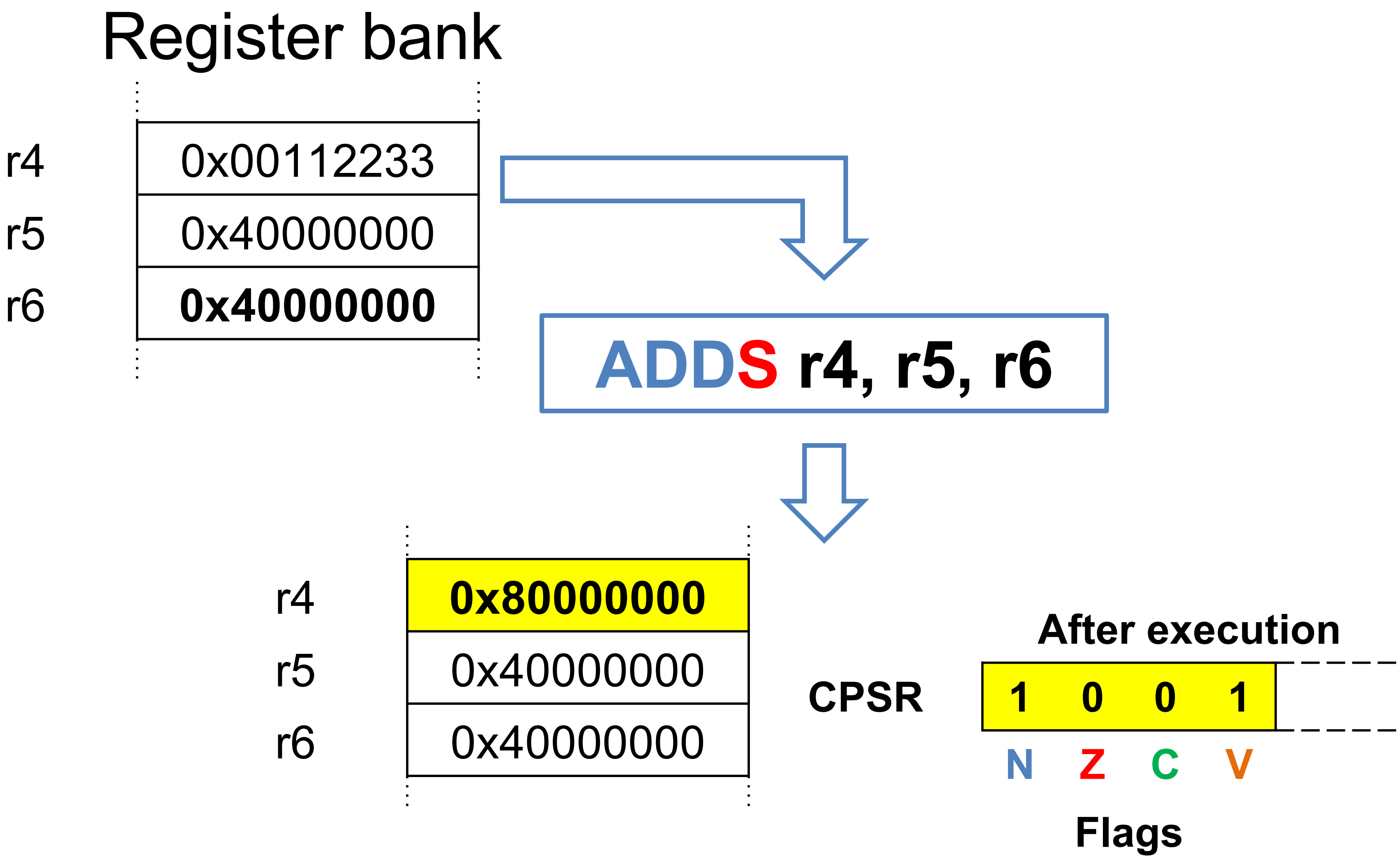
Question

What happens to the negative, overflow, zero and carry flags after each addition in the following programme when the value held in r5 is 0x40000000 (= 2^{30} = $1,073,741,824_{10}$)

```
ADDS r4, r5, r6        ; r6 := 0x40000000 = 2^30
ADDS r4, r5, r6        ; r6 := 0x80000000 = –2^31
ADDS r4, r5, r6        ; r6 := 0xC0000000 = –2^30
```

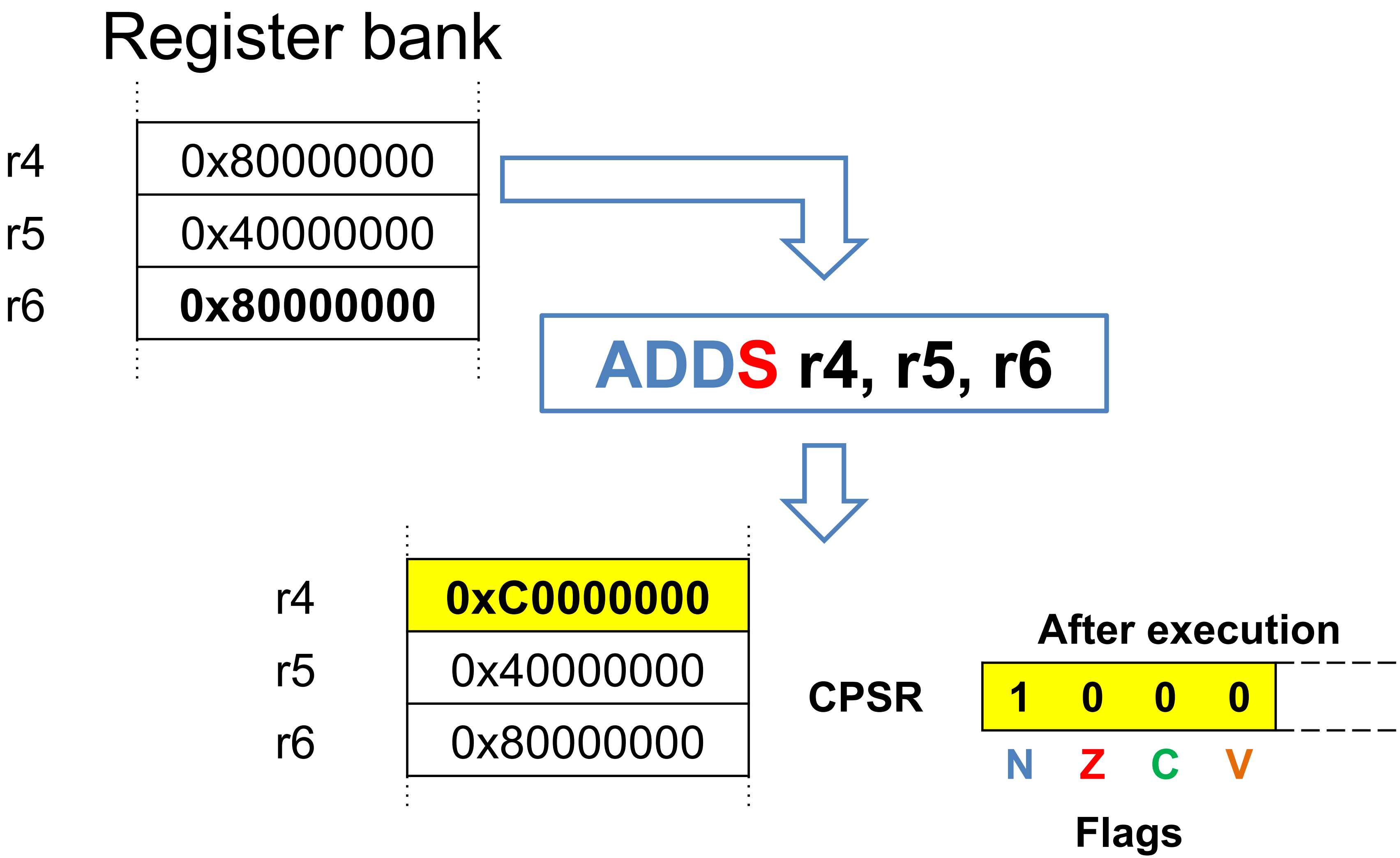
UNIVERSITY OF LIVERPOOL

Answer



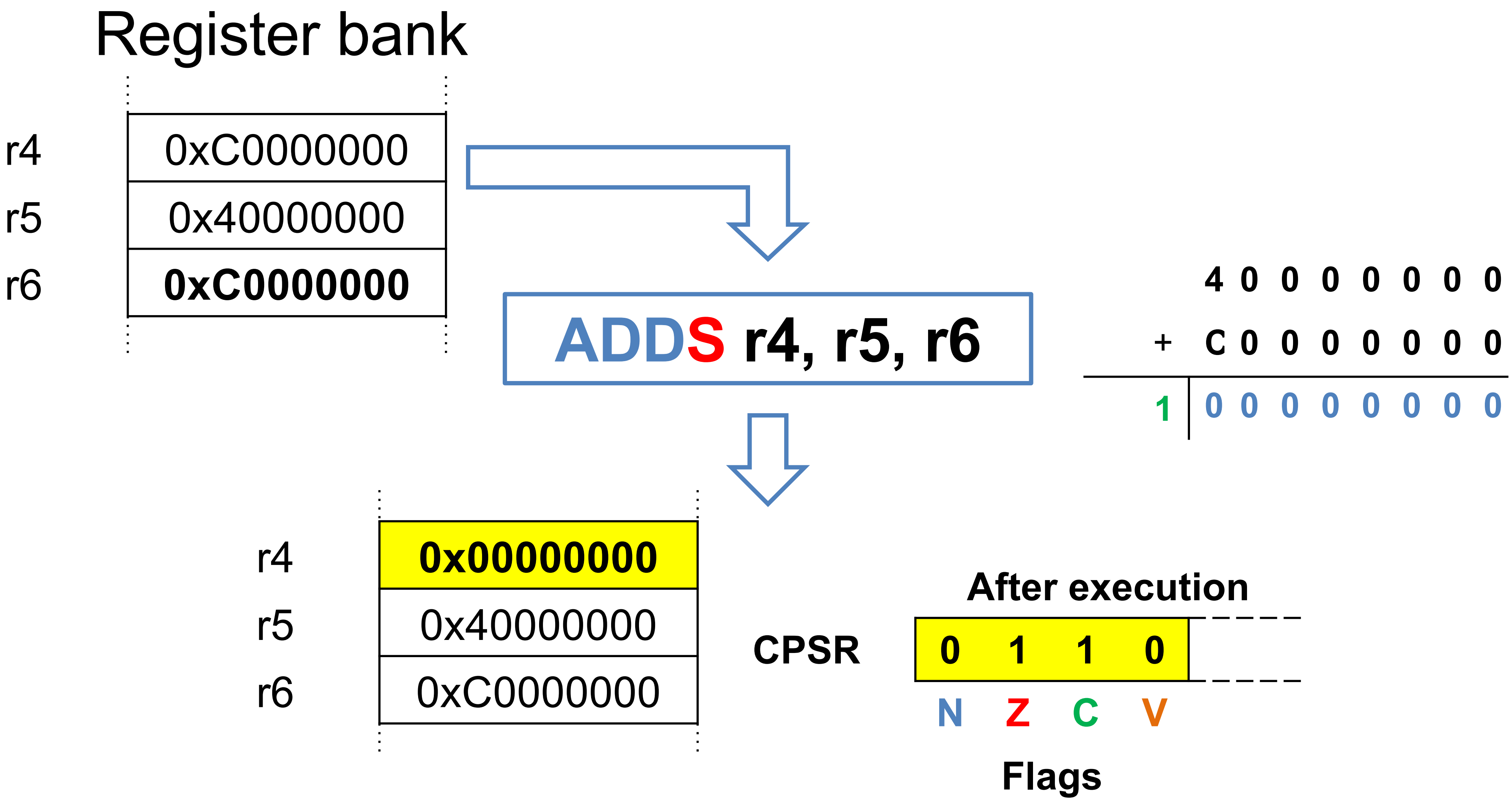
Adding two positive numbers: overflow into the sign bit and the result is negative.

Answer



Adding a positive and a negative number: the result is negative.

Answer



UNIVERSITY OF LIVERPOOL

Adding a positive and a negative number: the result (33 bits) is zero (lower 32 bits) with carry.

Question

What is the IEEE 754 format of the following numbers?

-75_{10} 0.75_{10}

Answer

-75_{10} in IEEE 754 format

1st : we need to normalize the number to obtain **a**

$-75_{10} = -1001011_2 \rightarrow -1.001011_2 \cdot 2^6$

2nd : -75_{10} is a negative number so **s** = 1

3rd : $\mathbf{n} = \mathbf{127_{10}} + 6_{10} = 133_{10} = 10000101_2$

Result : in 32 bit IEEE 754 single precision format

1 10000101 00101100000000000000000

Answer

0.75_{10} in IEEE 754 format

1st : we need to normalize the number to obtain **a**

$0.75_{10} = 0.11_2 \rightarrow 1.1_2 \cdot 2^{-1}$

2nd : 0.75_{10} is a positive number so **s** = 0

3rd : **n** = $\mathbf{127_{10}} + (-1_{10}) = 126_{10} = 01111110_2$

Result : in 32 bit IEEE 754 single precision format

0 01111110 10000000000000000000000

Binary conversion of 0.75		
0.75 × 2 =	0.5 +	1
0.5 × 2 =	0 +	1