Massimo Ginella

CS24 Elementary Computer Organization
Chapter 3 Exercises: 3.3, 3.20, 3.21, 3.22, 3.23

$\boxed{\textbf{3.3}}$ [10] <3.2> Convert 5ED4 into a binary number. What makes base 16 (hexadecimal) an attractive numbering system for representing values in computers?

Hexadecimal	Binary	Hexadecimal	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Binary Representation Based on the Chart:

F	E	D	4
0101	1110	1101	0100

Why is hexadecimal an attractive numbering system?

Hexadecimal is an attractive numbering systems for both humans and computers alike. Using base 16 numbering makes it much easier to condense long bit patterns into more concise and easier to read patterns made up of 16 different characters. Its much easier for a human to interpret the meaning of characters 1 - F than a long sequence of 1's and 0's. Converting back and fourth between hexadecimal and binary is also an easy task. Not only this, but memory addresses in computers are represented in hexadecimal which makes allocating memory much easier to navigate for programmers.

$\boxed{\textbf{3.20}}$ [5] <3.5> What decimal number does the bit pattern 0X0C000000 represent if it is a twos complement integer? An unsigned integer?

Convert hexadecimal into bits using above table:

0000 1100 0000 0000 0000 0000 0000 0000

Since the most significant bit in this binary sequence (which represents -1×2^{31}) is zero, we know that this bit pattern is representing a positive number and is not the 2s complement counterpart of another bit pattern. Therefore, this pattern represents the following decimal number:

$$\boxed{(1 \times 2^{27}) + (1 \times 2^{26}) = 201,326,592}$$

Because this bit pattern already represents a positive number, its unsigned counterpart will also be the same value.

$$\boxed{(1 \times 2^{27}) + (1 \times 2^{26}) = 201,326,592}$$

$\boxed{\textbf{3.21}}$ [10] <3.5>If the bit pattern 0X0000006F is placed into the Instruction Register, what RISC-V instruction will be executed?

Convert hexadecimal into bits using conversion table:

0000 0000 0000 0000 0000 0000 0110 1111

Looking at the last 7 bits tell us that the opcode is 1101111. We have a UJ instruction type.

UJ Instruction Structure:

20 bits	5 bits	7 bits
imm[20][10:1][11][19:12]	rd	opcode

Fill in with the relevant bits:

20 bits	5 bits	7 bits
00000000000000000000	00000	1101111

Rewrite in easily readable terms:

Immediate	Destination Register	Opcode
0x0000	x0	1101111

Finally, rewrite it in the instruction format:
jal rd, immediate

$\boxed{\text{jal x0, 0x00000}}$

$\boxed{\textbf{3.22}}$ [10] <3.5>What decimal number does the bit pattern 0X0C000000 represent if it is a floating point number? Use the IEEE 754 standard.

Convert hexadecimal into bits using conversion table:

0000 1100 0000 0000 0000 0000 0000 0000

Using the IEEE 754 standard method, we can retrieve the following information from this bit pattern:

Sign (MSB)	0
Exponent [30:23]	00011000 = 24
Significand [22:0]	00000000000000000000000

Plug these values into the IEEE 754 formula:

$$(-1)^{sign} \times (1 + significand) \times (2)^{exponent-127}$$

$$=$$

$$(-1)^0 \times (1.0) \times (2)^{24-127}$$

$$=$$

$$1.0 \times 2^{-103}$$

Converting 1.0 from bits into a decimal float still yields 1.0:

$$\boxed{1 \times 2^{-103} = 9.86 \times 10^{-32}}$$

$\boxed{\textbf{3.23}}$ [10] <3.5>Write down the binary representation of the decimal number 63.25 assuming the IEEE 754 single precision format.

63 in binary = 00111111
0.25 in binary = ?

Decimal Value	Decimal Value $\times$ 2	Binary Value
0.25	0.5	0
0.5	1.0	1
0	None	None

0.25 in binary = 01

63.25 in pseudo-binary = 00111111.01

Write in scientific notation: 1.1111101×2^5

Compare the bit pattern to the following standard formula:
$(-1)^{sign} \times (1 + significand) \times (2)^{exponent-127}$

Sign	0
Exponent	Exp - 127 = 5 : Exp = 132 = 10000100
Significand	1111101

Format the information in the table above to IEEE 754 Single Precision Format:

Sign Exponent Significand

0 10000100 11111010000

=

$\boxed{\text{0100 0010 0111 1101 0000 0000 0000 0000}}$