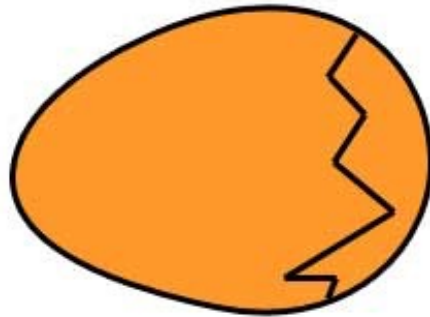


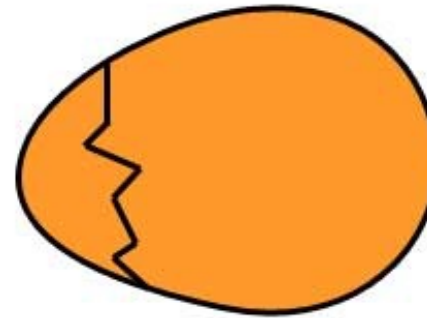
# Endianness

Lightning Talk — PDX-D2

## Big Endian, Little Endian – from Gulliver's Travels.



BIG ENDIAN - The way  
people always broke  
their eggs in the  
Lilliput land



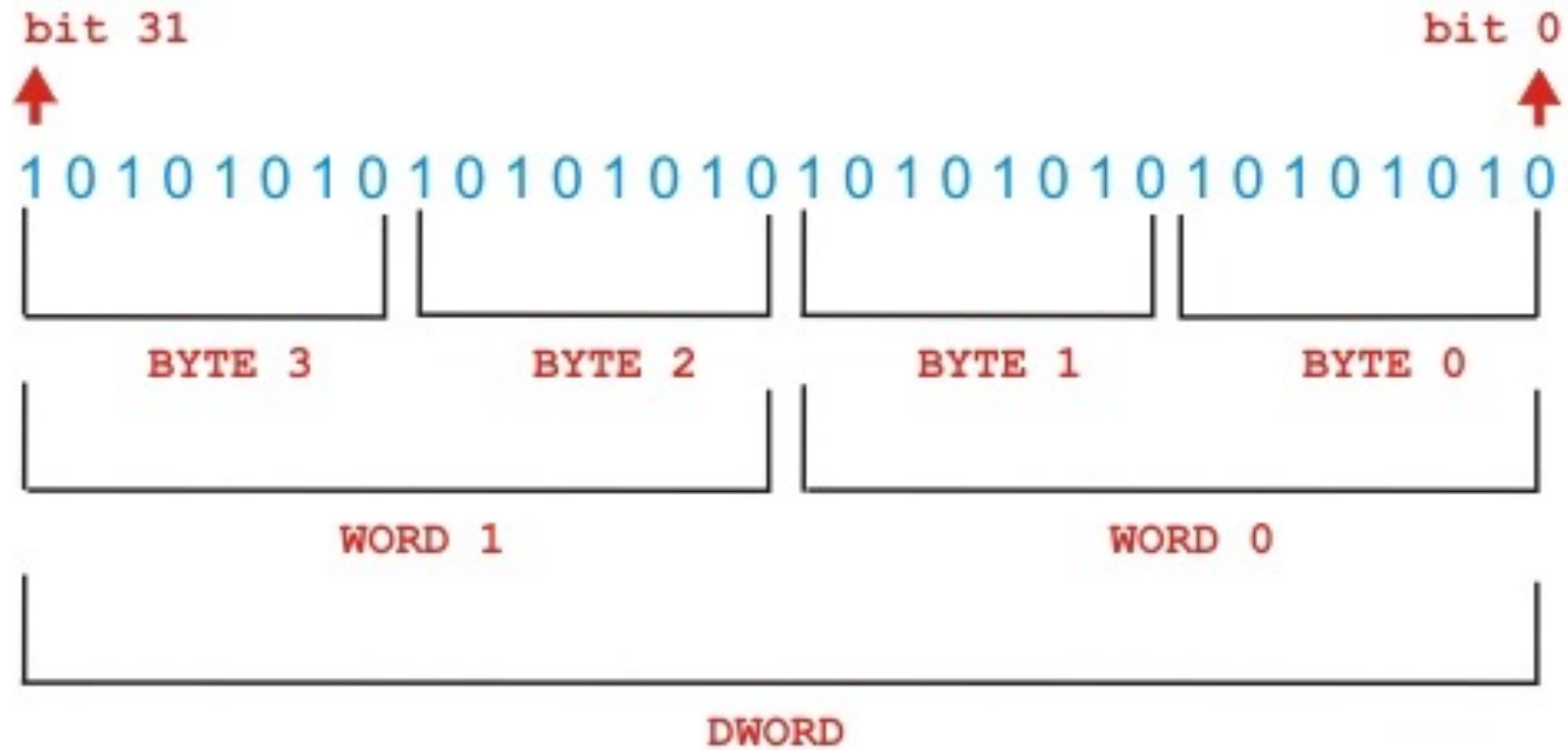
LITTLE ENDIAN - The  
way the king then  
ordered the people to  
break their eggs

**BIT#:**     7     6     5     4     3     2     1     0

0	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---

**VALUE:**      $2^7$     $2^6$     $2^5$     $2^4$     $2^3$     $2^2$     $2^1$     $2^0$   
              128   64   32   16   8   4   2   1

# Bits, Bytes, Words ....



[http://www.intelliproject.net/articles/showArticle/index/bitwise\\_operators\\_cpp](http://www.intelliproject.net/articles/showArticle/index/bitwise_operators_cpp)

Decimal	Binary	Hex
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

0x3A



0	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---

## 32-bit architecture

**A  
d  
d  
r  
e  
s  
s  
e  
s**

0xFFFFFFFF

1000 0000

.....

.....

0x00000008

0100 1001

0x00000007

1100 1100

0x00000006

0110 1110

0x00000005

0110 1110

0x00000004

0000 0000

0x00000003

0110 1011

0x00000002

0101 0001

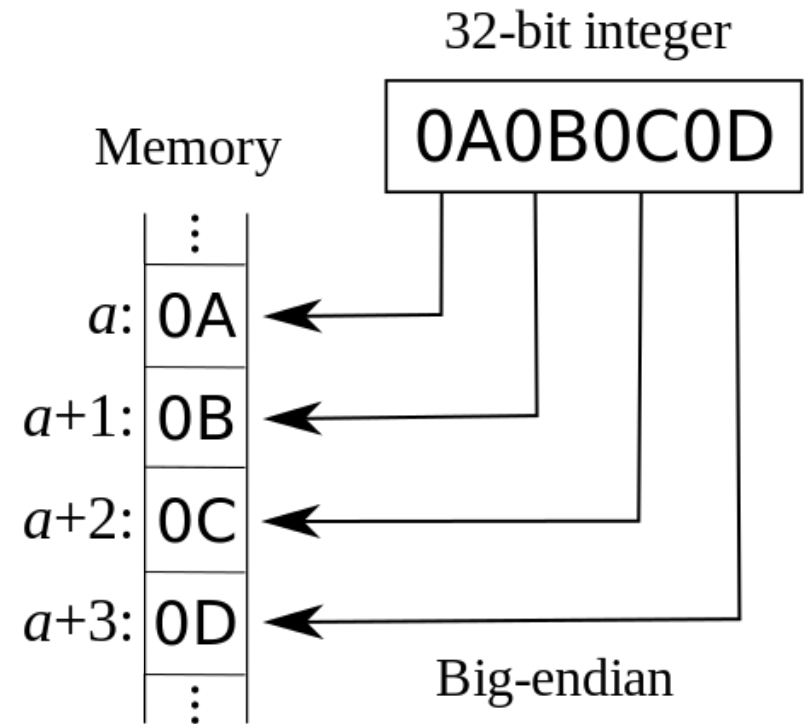
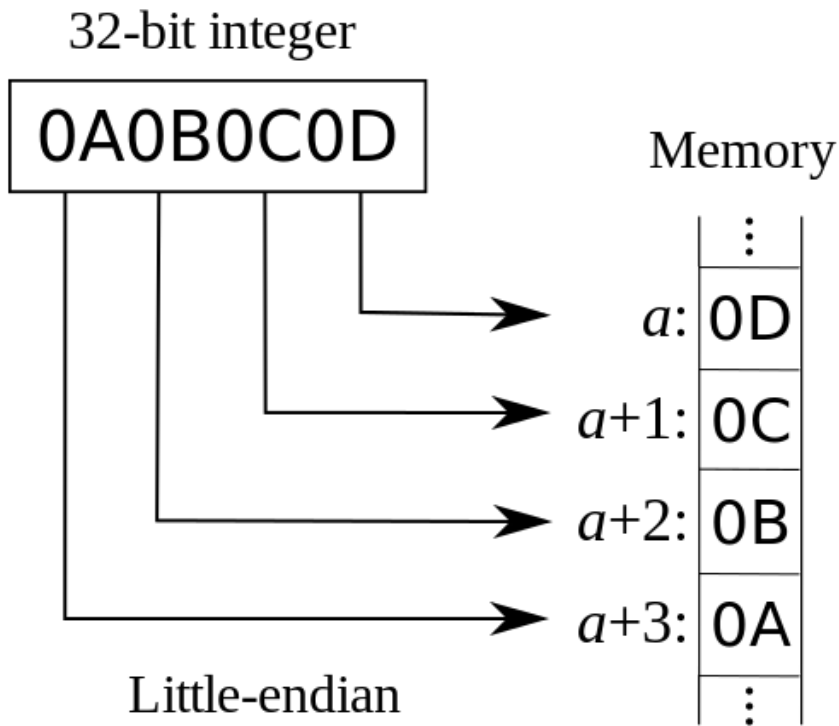
0x00000001

1100 1001

0x00000000

0100 1111

**Main Memory**



<https://en.wikipedia.org/wiki/Endianness>

**Examples:**  
**Intel x86 and x86-64**  
**DEC Alpha**  
**PDP-11**

**Examples:**  
**Motorola 6800 and 68**  
**SUN Sparc**

# Bitmap Example

Offset hex	Offset dec	Size	Purpose
00	0	2 bytes	<p>The <b>header field</b> used to identify the BMP and <code>0x42 0x4D</code> in <b>hexadecimal</b>, same as <code>BM</code> in <b>ASCII</b>. The following entries are possible:</p> <ul style="list-style-type: none"> <li>• <b>BM</b> – Windows 3.1x, 95, NT, ... etc.</li> <li>• <b>BA</b> – OS/2 struct bitmap array</li> <li>• <b>CI</b> – OS/2 struct color icon</li> <li>• <b>CP</b> – OS/2 const color pointer</li> <li>• <b>IC</b> – OS/2 struct icon</li> <li>• <b>PT</b> – OS/2 pointer</li> </ul>
02	2	4 bytes	The size of the BMP file in bytes
06	6	2 bytes	Reserved; actual value depends on the application that creates the image
08	8	2 bytes	Reserved; actual value depends on the application that creates the image
0A	10	4 bytes	The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found.

offset → Integers stored in LE format

424d	c6b6	0400	0000	0000	3600	0000	2800
0000	6801	0000	e2fe	ffff	0100	1800	0000
0000	0000	0000	130b	0000	130b	0000	0000
0000	0000	0000	fdf3	e2fd	f3e2	fdf3	e2fd
f3e2	fdf3	e1fd	f3e2	fdf3	e2fc	f3e2	fbf2
e3f9	f3e5	f8f2	e7fb	f2e7	fcf5	e7fc	f4e4
faf3	e0fb	f2df	faf1	ddfa	f0dc	fbef	ddfc
eddd	fdec	dcfe	ecda	ffeb	d8fd	e9d8	fbe9
daf8	e9dd	f5ea	def9	e9df	fae8	dff7	e9e0
f5ea	dff3	eae0	f1e8	dff1	e8df	f3e8	e0f2
e9e0	ede9	e0ea	eade	e8ea	dce6	e7dc	e5e8
dee6	e9e0	e4e7	dee0	e4dc	dde1	d9e0	e4da
dce3	d6d6	ded4	d0da	d2c9	d9d0	bfd3	cdb6
cdc9	aec5	c4a2	bbbe	98b0	b78a	a5ad	7a96
a26a	8695	607a	8753	6773	4957	6141	4a4f
3738	3c2f	2b2c	2a27	2929	2b30	262f	392b
3a48	3347	5d39	506b	405b	794e	6281	696e
8a84	7d90	9a8d	97ae	9a97	c2a9	98d5	b596
dfbc	94e4	bf90	e8c1	8bec	c185	eec0	7ef0
bf7a	ecb9	75e2	b06b	ce9e	5fb5	8a50	926b
3669	4b1e	4937	1334	2b0e	2c24	0a28	2009
241c	0821	1908	201a	081e	1809	1d18	0b1a
1709	1a17	0917	1509	1614	0a15	140a	1616
0c14	150d	1113	0d0f	110e	0d10	110c	0f14
0c0f	140c	0f13	0c0f	140d	1015	0d10	150c



<b>Sample Length:</b>	4				4				4				4			
<b>Channel Membership:</b>	Alpha				Red				Green				Blue			
<b>Bit Number:</b>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>RGBAX</b>	R. G. B. A. X															
<b>Sample Length Notation:</b>	4.4.4.4.0															

[https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format)

```
var pixel = buffer.readUIntBE(offset, 2);
```

```
var green = ( pixel & 0x00F0 ) >>> 4;
```

Backup

AND



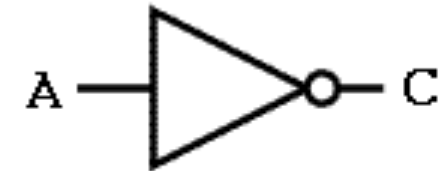
Inputs		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

OR



Inputs		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT



Input	Output
A	C
0	1
1	0

**AND**

1

0

1

1

0

0

0

1

0

0

0

1