

I2BIT I/O

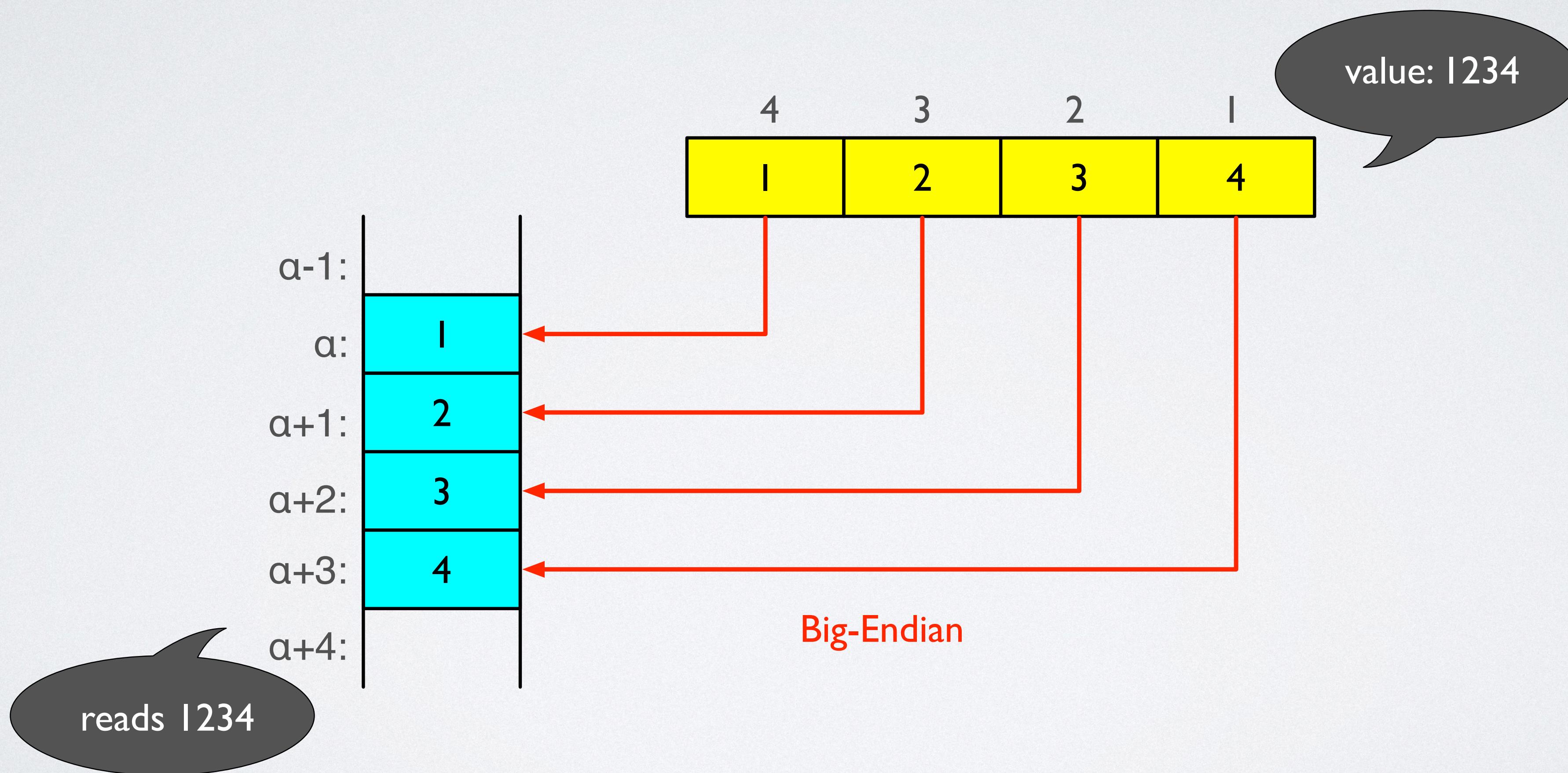
Design Pattern

ENDIANNESS

- When two parties wish to exchange information, then they need to agree on an ordering convention if the data being exchanged is too large to be sent in one piece.
- In computing, **endianness** refers to the byte or bit ordering of data stored in the computer memory or send over the network.
- We distinguish two orderings:
 - Big-endian order
 - Little-endian order

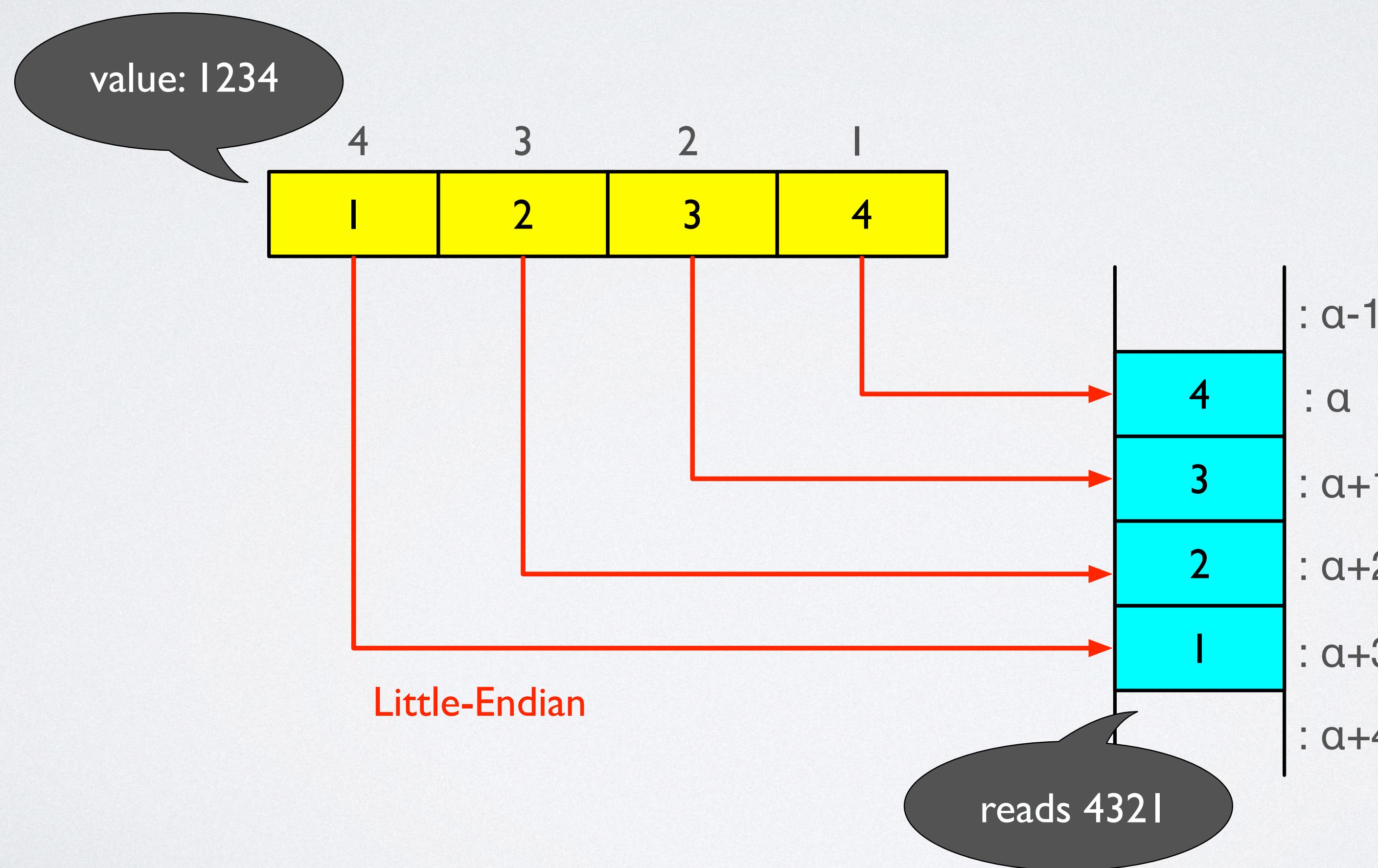
BIG-ENDIAN

- The most significant byte or bit (MSB) is stored at the memory location with the lowest address:



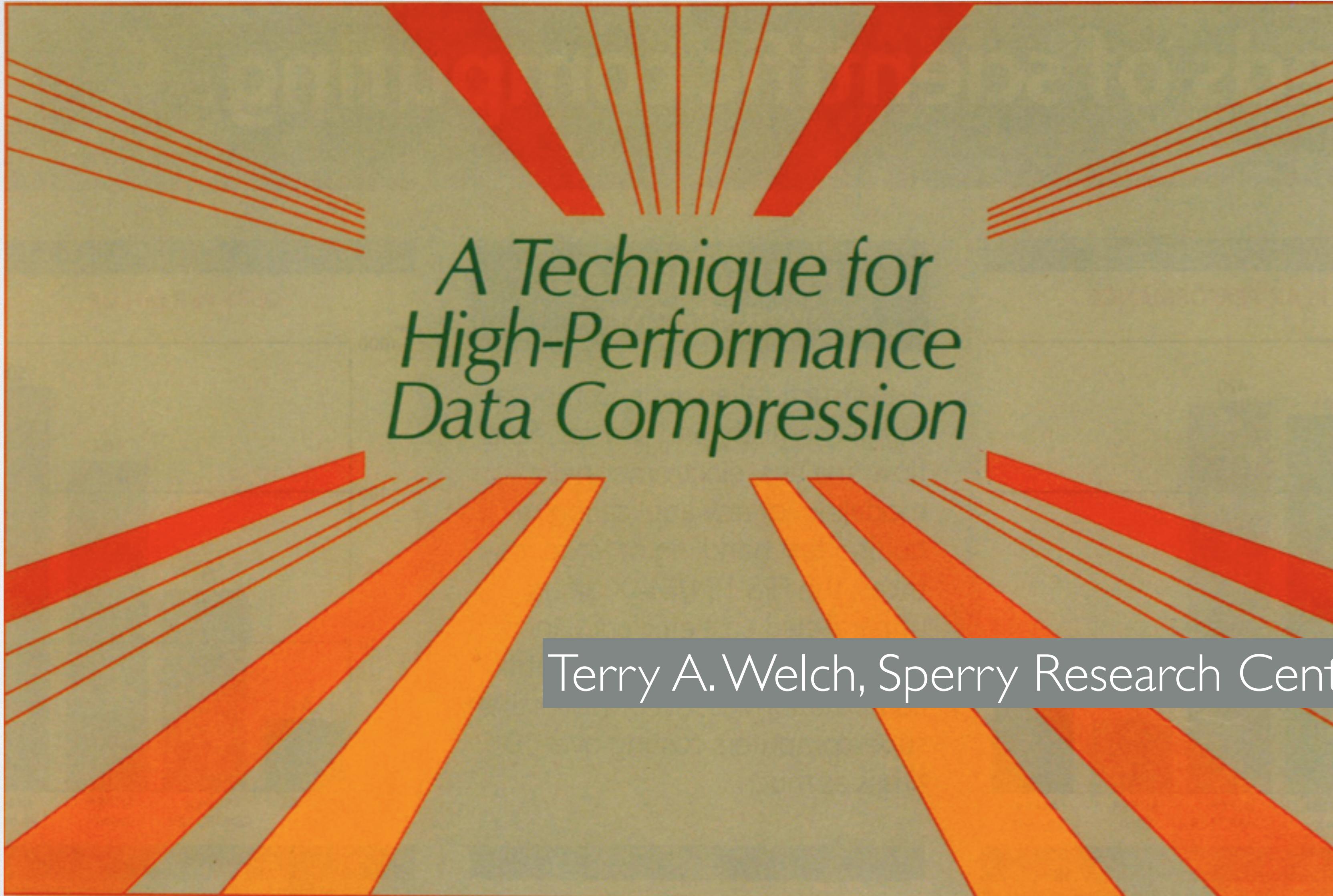
LITTLE ENDIAN

- The least significant byte or bit (LSB) is stored at the memory location with the lowest address:



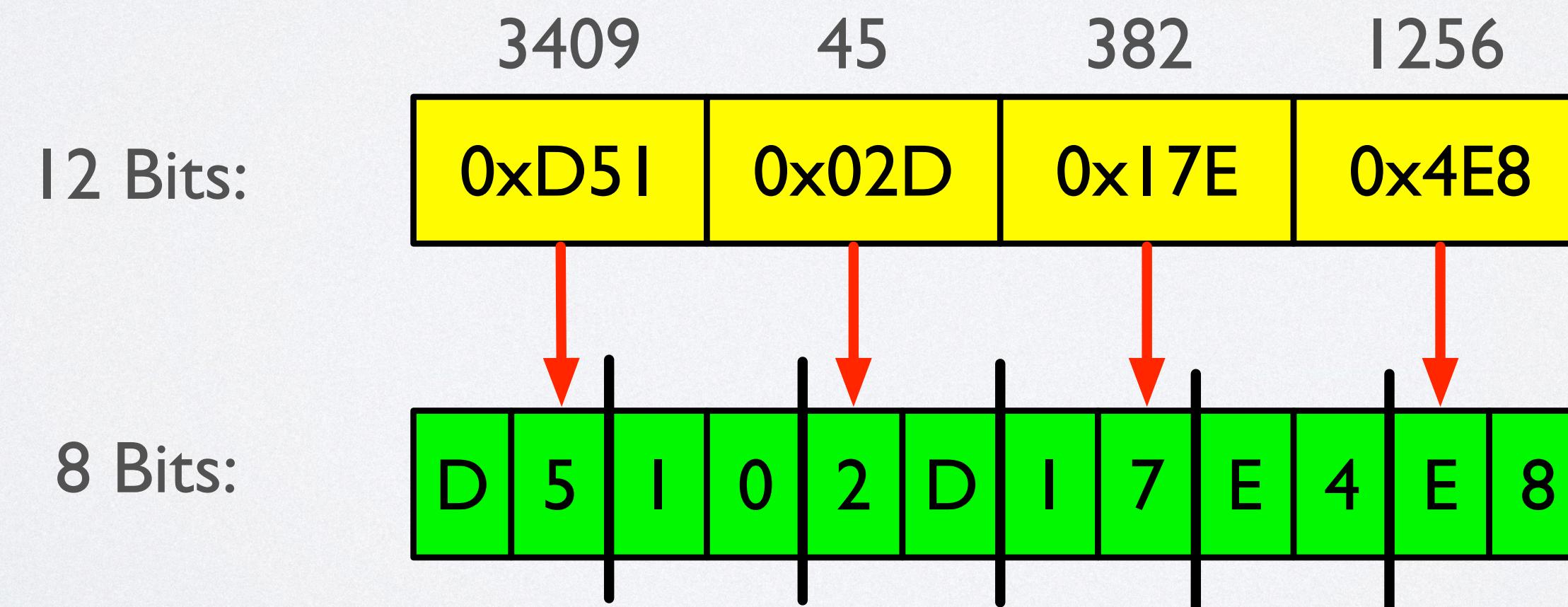
12 BIT VALUES

LZW COMPRESSION

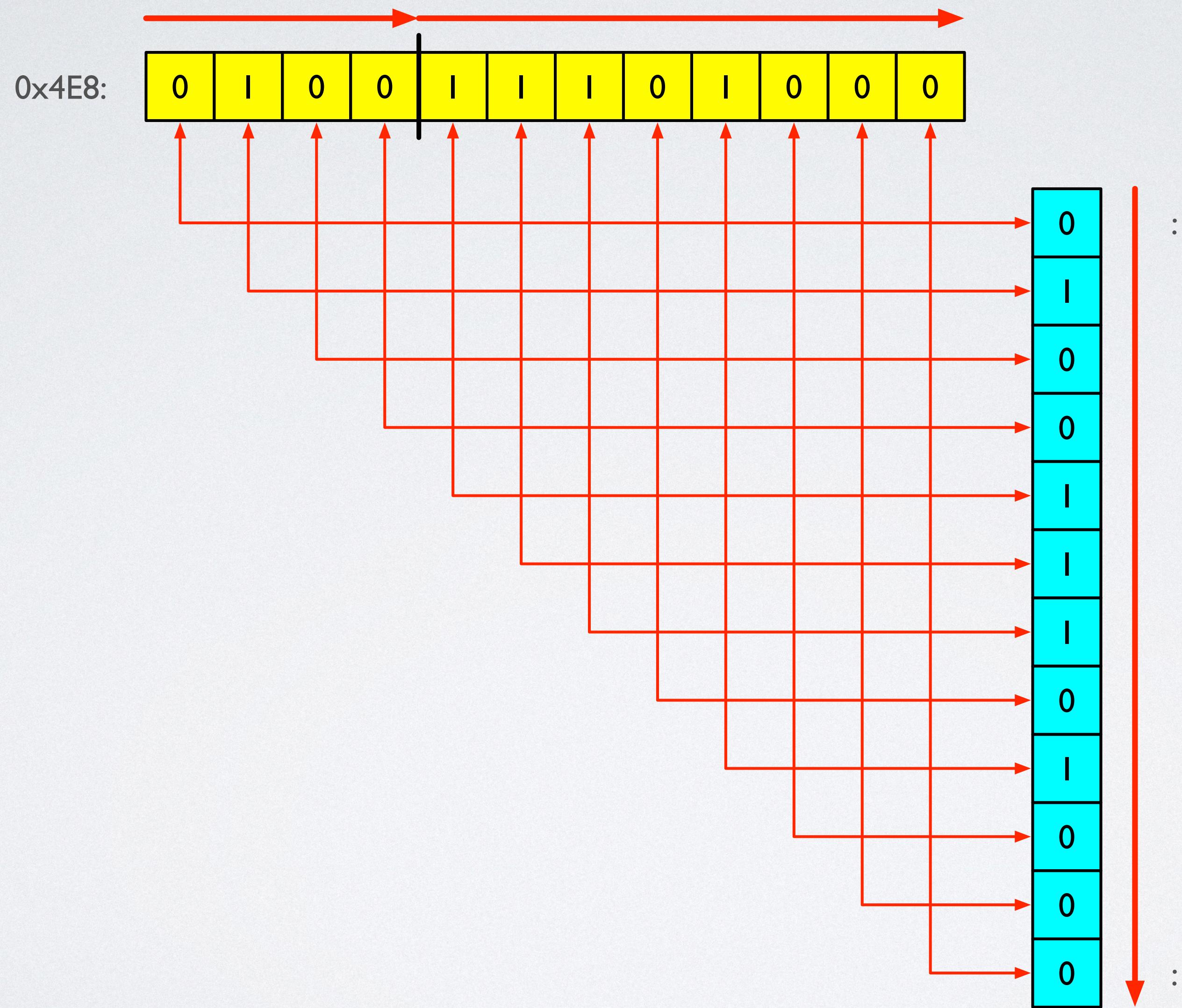


WHICH ORDERING WORKS BEST FOR 12 BIT I/O

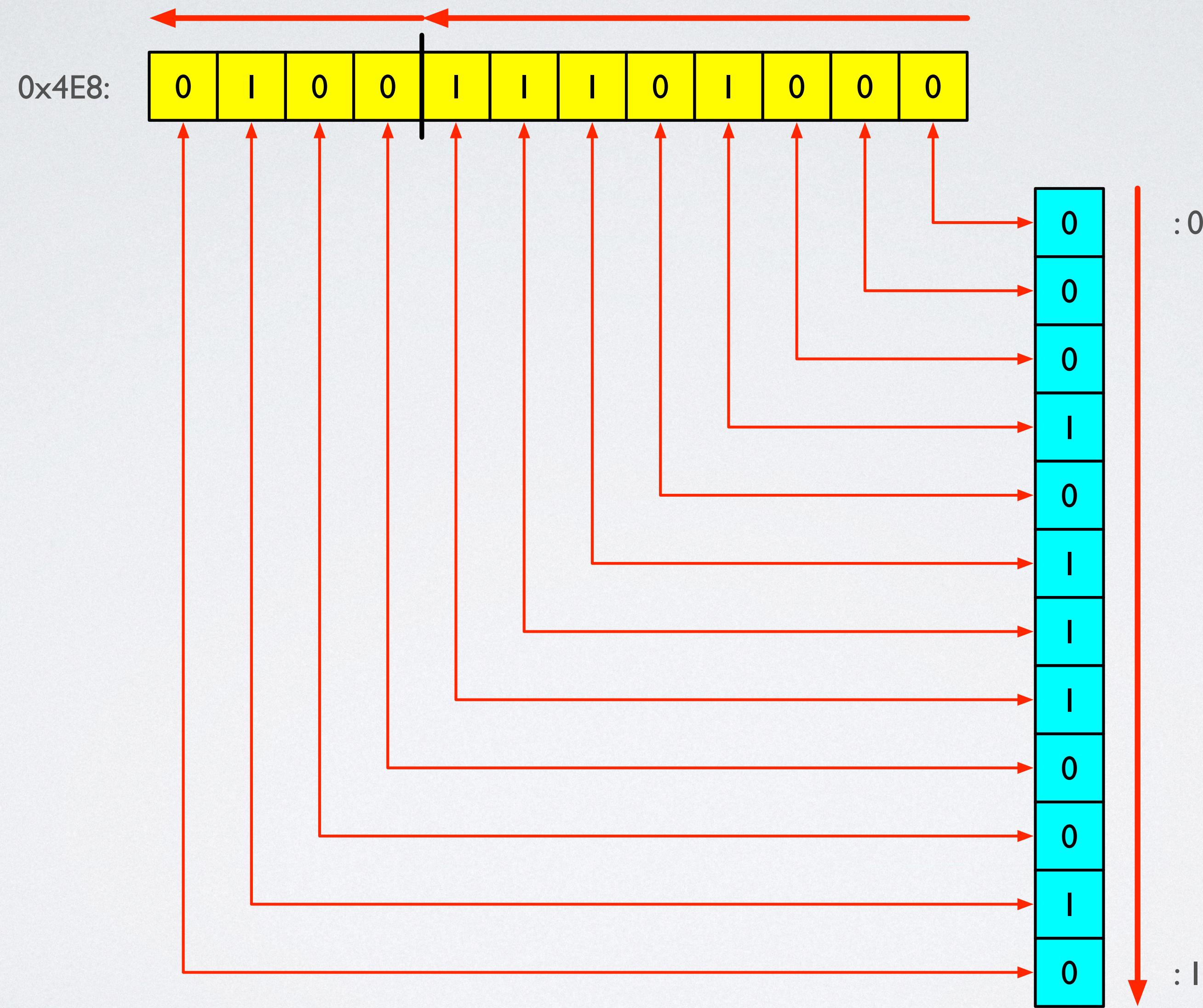
- Consider four 12-bit numbers: 1256, 382, 45, and 3409:
 - 12-bit values cannot be stored in a byte.
 - We could use a 16-bit word, but this would waste 4 bits per value.
 - We need to devise an approach where we store the 12-bit values as a consecutive bitstream, each value requiring 12 bits.
 - Which ordering works better: little endian or big endian?



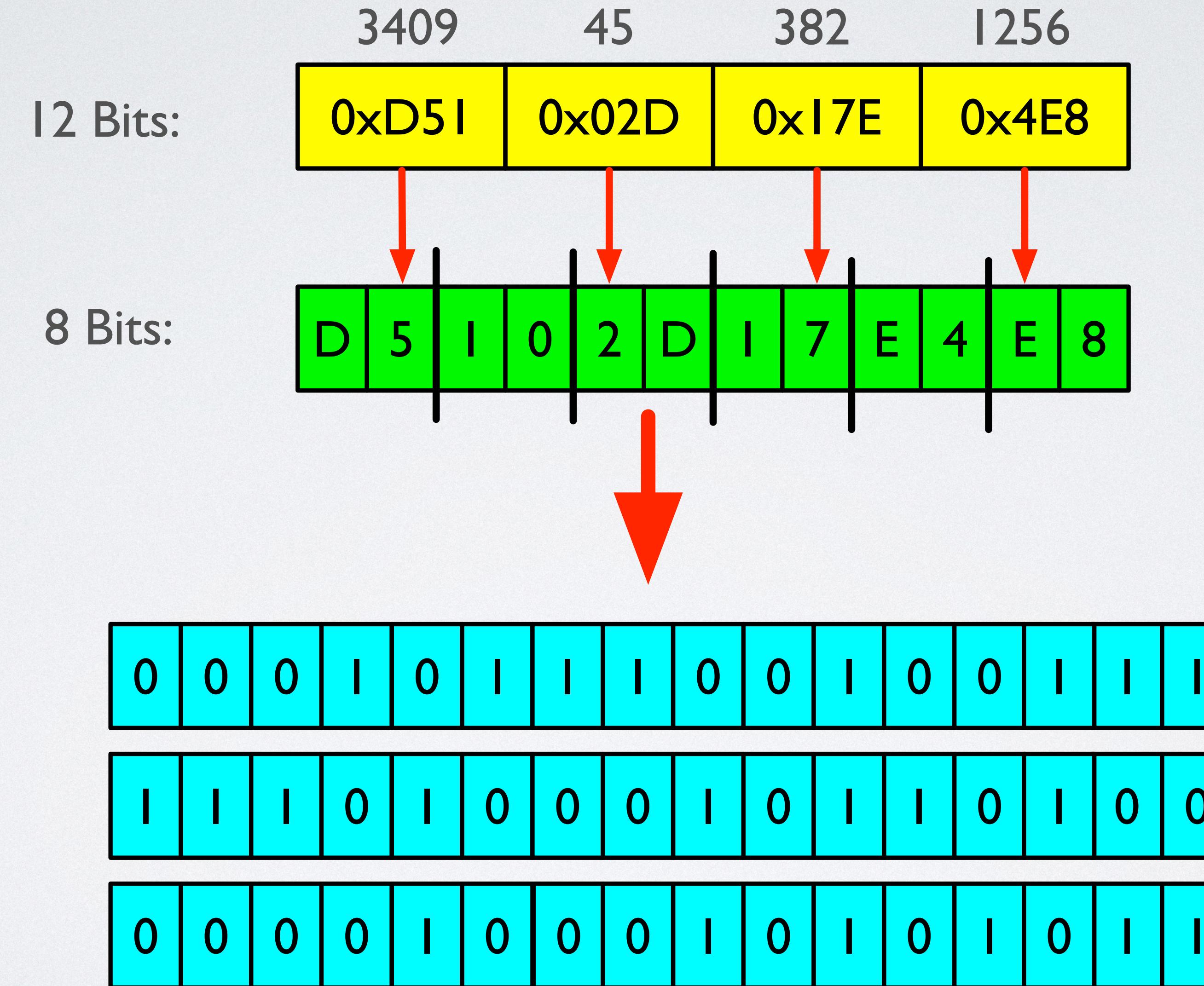
PROCESSING 0X4E8: BIG-ENDIAN



PROCESSING 0X4E8: LITTLE-ENDIAN



4E8 17E 02D D5I



WRITE 12 BIT VALUES (PSEUDOCODE)

```
write12Bits( aValue : 12Bit ) =  
  
    for i = 1 to 12  
  
        do  
  
            if (aValue & 0x1)      // fetch lowest bit  
  
                then send 1 to output;  
  
                else send 0 to output;  
  
                aValue := aValue / 2;  // divide by 2  
  
        od;
```

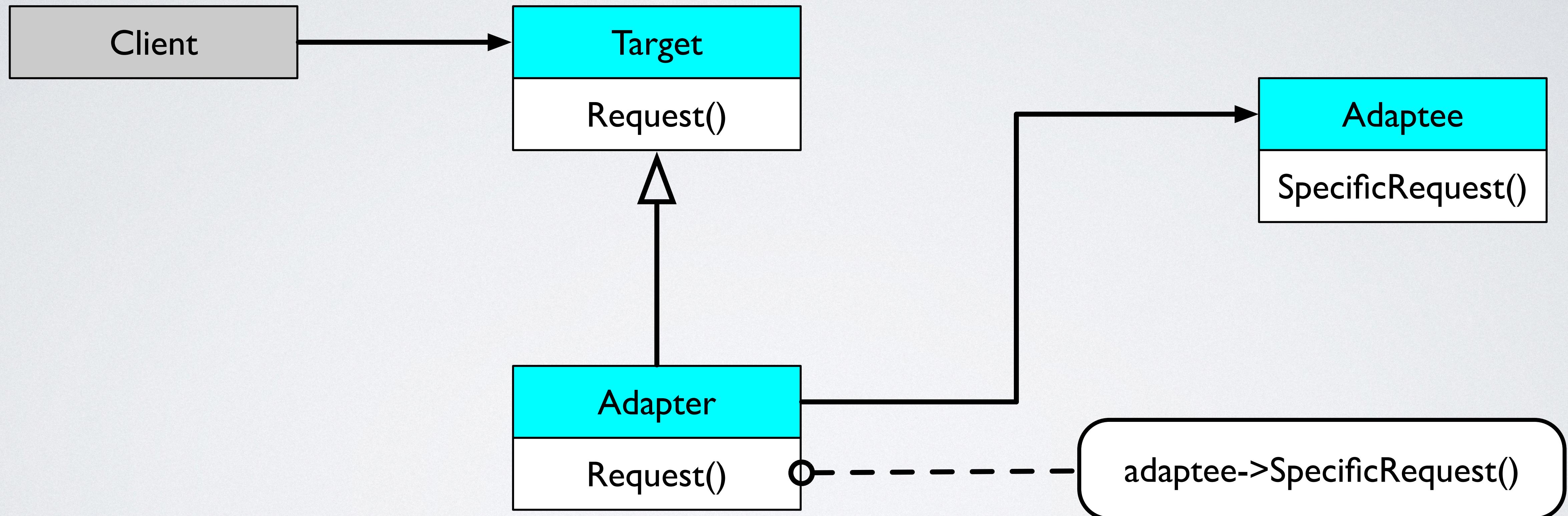
READ 12 BIT VALUES (PSEUDOCODE)

```
read12Bits() : I2Bit =  
  
declare Result : I2Bit = 0;  
  
for i = 1 to 12  
  
do  
  
declare IBit : Bit = input()           // get next bit  
  
if ( IBit == 1 )  
  
then Result = (I << (i-1)) + Result; // set bit at index i  
  
od;  
  
return Result;
```

ADAPTER DESIGN PATTERN

- Intent:
 - Convert the interface of a class into another interface clients expect. Adapter lets classes work together that could not otherwise because of incompatible interfaces.
- Collaborations:
 - Clients call operations on an Adapter instance. In turn, the adapter calls Adaptee operations that carry out the request.

STRUCTURE OF AN OBJECT ADAPTER



ADAPTER FOR STD::OFSTREAM

cplusplus.com

TUTORIALS REFERENCE ARTICLES FORUM sign up log in [Legacy version]

C++

- Tutorials
- Reference
- Articles
- Forum

Reference

- C library:
- Containers:
- ▼ Input/Output:
 - <fstream>**
 - <iomanip>
 - <ios>
 - <iostfwd>
 - <iostream>
 - <istream>
 - <ostream>
 - <sstream>
 - <streambuf>
- Multi-threading:
- Other:

<fstream>

- ▼ class templates
 - basic_filebuf
 - basic_fstream
 - basic_ifstream
 - basic_ofstream
- ▼ classes
 - filebuf
 - fstream
 - ifstream
 - ofstream**
 - wfilebuf
 - wfstream
 - wifstream
 - wofstream
- ofstream
 - ofstream::ofstream
 - ▼ public member functions
 - ofstream::close
 - ofstream::is_open
 - ofstream::open
 - ofstream::operator=
 - ofstream::rdbuf
 - ofstream::swap
 - ▼ non-member overloads
 - swap (basic_ofstream)

Reference : <fstream> : ofstream

We're in the world's top 20 largest pension funds [Learn more](#)

class **std::ofstream**

Output file stream

```
ios_base ←--> ios ←--> ostream ←--> ofstream
```

Output stream class to operate on files.

Objects of this class maintain a [filebuf](#) object as their *internal stream buffer*, which performs input/output operations on the file they are associated with (if any).

File streams are associated with files either on [construction](#), or by calling member [open](#).

This is an instantiation of [basic_ofstream](#) with the following template parameters:

template parameter	definition	comments
charT	char	Aliased as member char_type
traits	char_traits<char>	Aliased as member traits_type

Apart from the internal [file stream buffer](#), objects of this class keep a set of internal fields inherited from [ios_base](#), [ios](#) and [istream](#):

	field	member functions	description
Formatting	format flags	flags setf unsetf	A set of internal flags that affect how certain input/output operations are interpreted or generated. See member type fmtflags .
	field width	width	Width of the next formatted element to insert.
	precision	precision	Decimal precision for the next floating-point value inserted.
	locale	getloc imbue	The locale object used by the function for formatted input/output operations affected by localization properties.
	fill character	fill	Character to pad a formatted field up to the field width (width).
State	error state	rdstate setstate clear	The current error state of the stream. Individual values may be obtained by calling good , eof , fail and bad . See member type iostate .
	exception mask	exceptions	The state flags for which a failure exception is thrown. See member type iostate .
	callback stack	register_callback	Stack of pointers to functions that are called when certain events occur.
Other	extensible arrays	iword pword xalloc	Internal arrays to store objects of type long and void*.
	tied stream	tie	Pointer to output stream that is flushed before each i/o operation on this stream.
	stream buffer	rdbuf	Pointer to the associated streambuf object, which is charge of all input/output operations.
	character count	gcount	Count of characters read by last unformatted input operation.

Member types