Warmup

Write the following binary numbers in decimal:

1100

8 4 2 1

8 + 4 = 12

101010

32 16 8 4 2 1

32 + 8 + 2 = 42

1111_1111

Submit your answer on **pollev.com/stevenbell**

ES 4: Number systems (and more VHDL)

Steven Bell

4 October 2021

Tufts UNIVERSITY

By the end of class today, you should be able to:

- Convert between hexadecimal and binary
- Convert between 2's complement binary and decimal
- Explain why we prefer 2's complement to sign-magnitude
- Use the VHDL `process` structure for combinational logic and print-style debugging.

Hexadecimal

Is just a shorthand for binary numbers

0	0000	D	1101	32	dec 100000
1	0001	E	1110		
2	0010	F	1111	0x32	hex 00110010
3	0011				
4					
5					
6					
7					
8					
9	1001				
A	1010				
B	1011				
C	1100				

Practice!

Write the following binary numbers in hex:

0101_1010 1100_0011 1011_1110_1110_1111

Write these hexadecimal numbers in binary:

FF 80 DEAD

Submit binary → hex answers on **pollev.com/stevenbell**

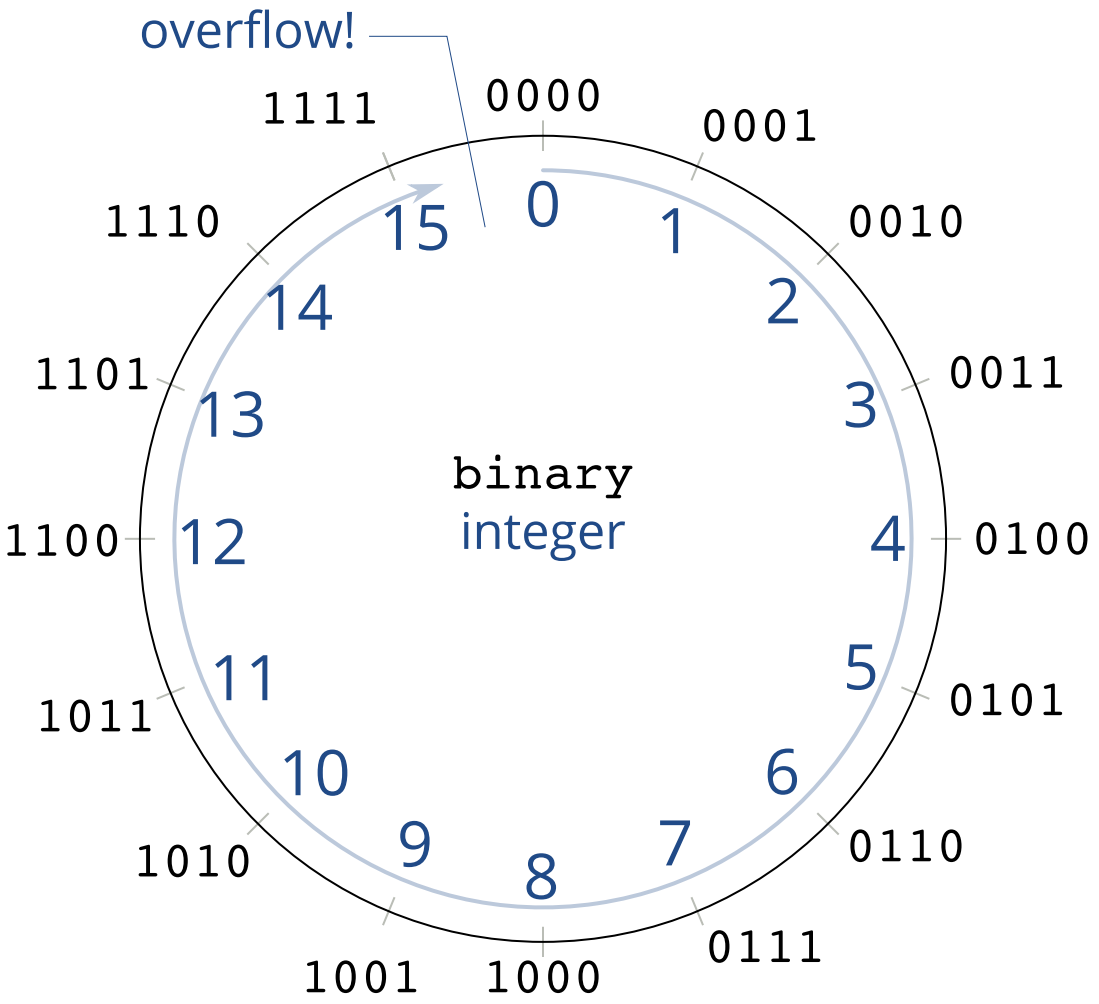
10

There are 10 kinds of people in the world.
Those who understand binary,
and those who don't.

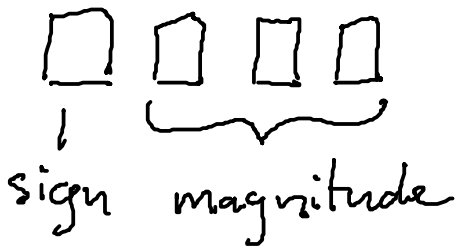
There are $\frac{10}{8}$ kinds of people in the world.

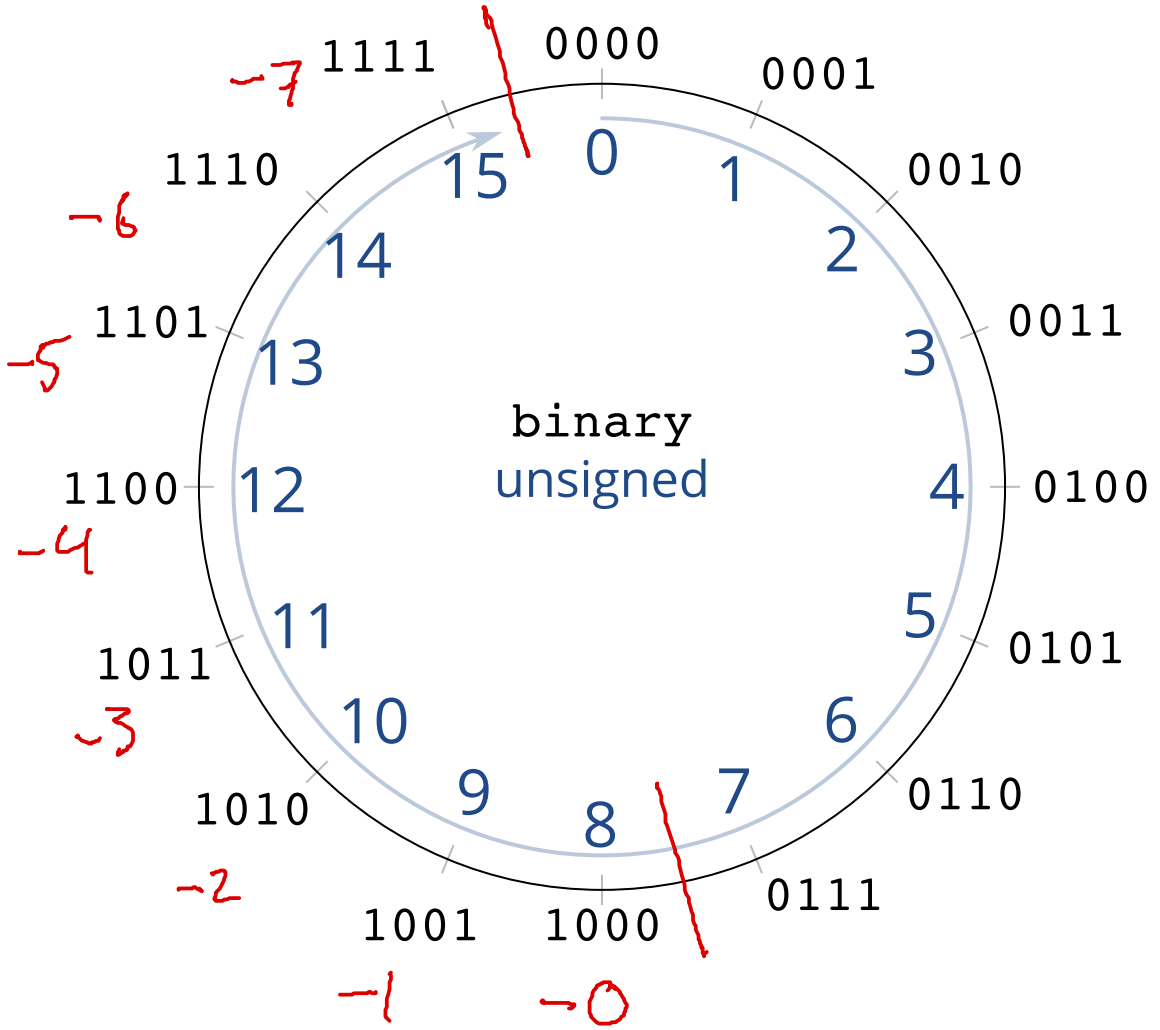
Those who understand hexadecimal,
and those who don't.

Unsigned numbers

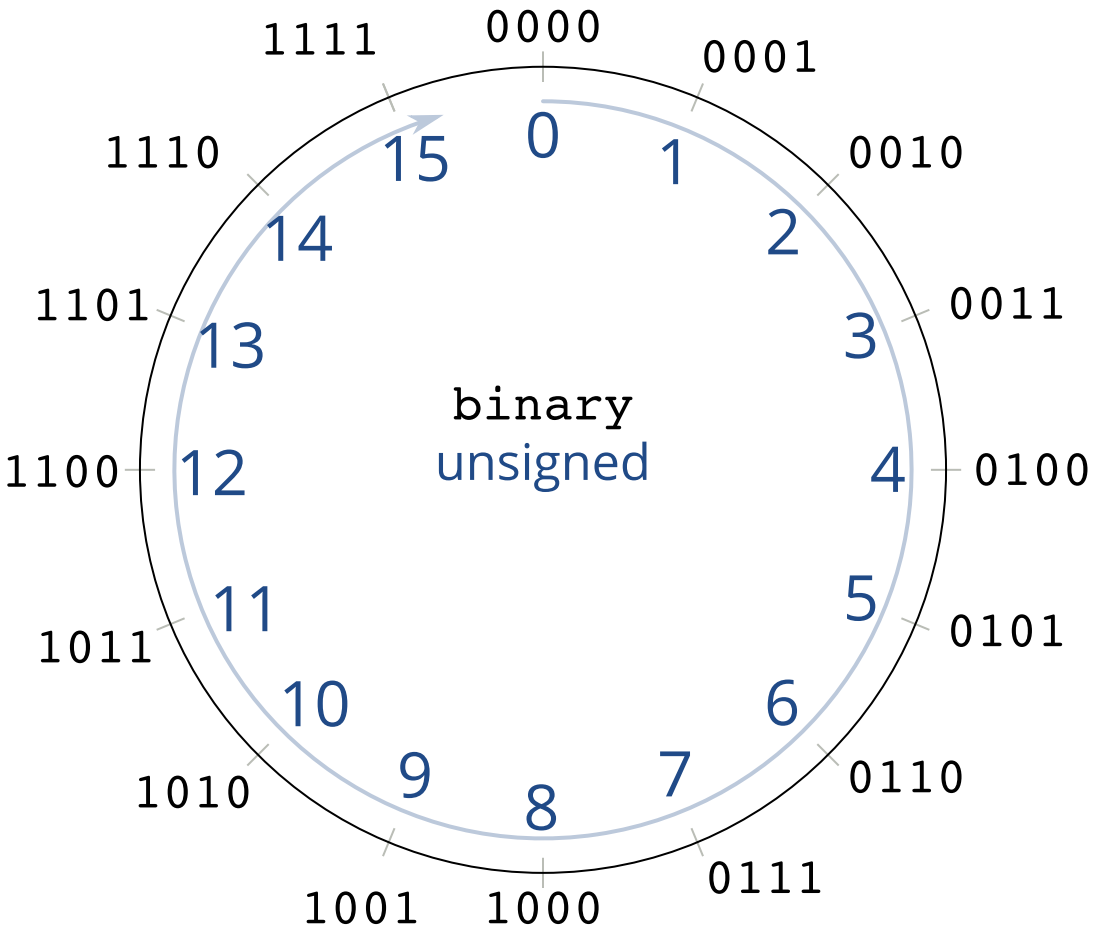


Sign-magnitude

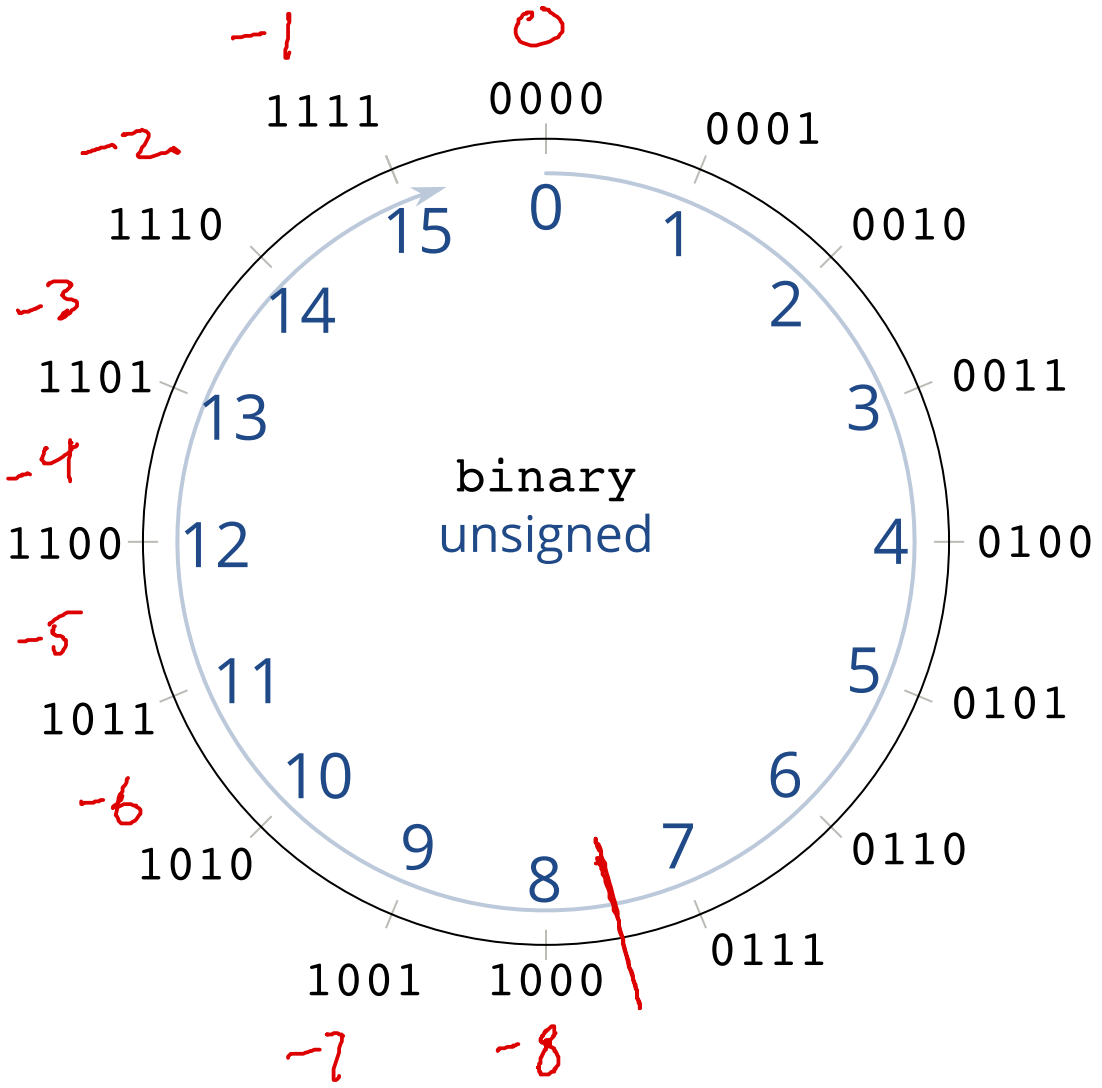


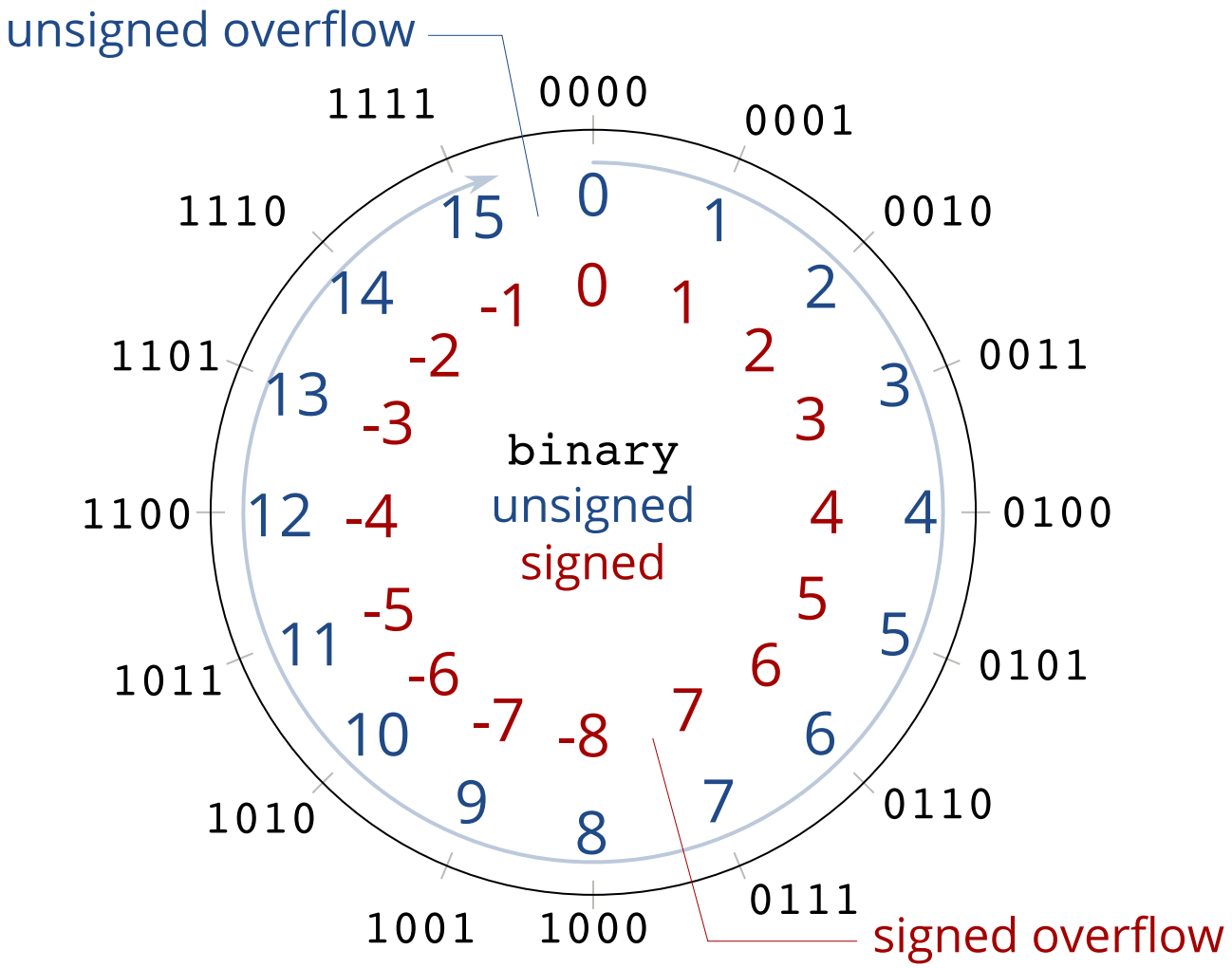


Sign-magnitude



Two's complement




unsigned overflow
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
2
3
4
5
6
7
-8
-7
-6
-5
-4
-3
-2
-1
binary
unsigned
signed
signed overflow

To write a negative number in 2's complement:

Write the positive number in binary

Flip all the bits (1→0, 0→1)

Add 1 (with all the appropriate carries)

3: 0011

1100

-3: 1101

To convert negative 2's complement to decimal,

Flip all the bits (1→0, 0→1)

Add 1 (with all the appropriate carries)

Write the number in decimal

-3: 1101

0010

3: 0011

00101100

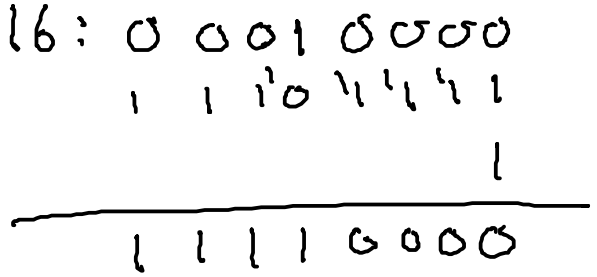
Wait, shouldn't this be reversed?

It turns out the result is the same either way. Try it!

Practice!

Write the following numbers in 8-bit 2's complement:

-16, -1, -127



Find the decimal value of these 2's complement numbers:

01001111, 11111110, 10000000

64 8 4 2 1 = 79

Submit your decimal answers on **pollev.com/stevenbell**

What's the point

Practically all real systems use 2's complement

So why did we waste all that time with sign-magnitude?

Because sign-magnitude seems "obvious"
and 2's complement is "weird"

Floating-point numbers do use sign-magnitude

Numbers in VHDL

Types

`std_logic` Basic logic type, can take values 0, 1, X, Z (and others)

`std_logic_vector (n downto m)` Ordered group of std_logic

`unsigned (n downto m)` Like std_logic_vector, but preferred for numerically meaningful signals

`signed (n downto m)`

`integer` Poor for synthesis, but constants are integers by default

if a is std_logic

a <= 0 a <= '0'

if b is std_logic_vector (7 downto 0)

b(3) <= '1'

b <= x"BF"

Literals

`'0', '1', 'X', 'Z'`

`"00001010", x"0c"` 8-bit binary, hex

`9x"101"` `3b"101"` `7d"101"`

9-bit hex 3-bit binary 7-bit decimal

`5, 38, 10000000`

0 0 0 1 0 0 0 0 0 0 0 1

When/else

```
RESULT_SIGNAL <= '1' when (SIGNAL1 = x"5") else '0';
```

Practice!

Multiplexer, thermometer decoder, 4-bit ALU

For Wednesday

1. Read the book (4.3, 4.9) and complete the pre-class quiz
2. Lab report 2 due this week
3. Homework 3 (VHDLweb) due Wednesday
4. Lab report 3 due next week