Today's learning goals

• Convert between positive integers written in any base
• Relate algorithms for integer operations to bitwise boolean operations
• Correctly use XOR and bit shifts
How many people were in your group for HW1?

A. I worked alone.
B. 2
C. 3
D. I joined this class late and didn’t submit HW1.

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

CENTR101: CA  PCYNH109: AB
Recall: Base expansion

Notation: for positive integer $n$

Write

$$(a_k a_{k-1} \ldots a_1 a_0)_b$$

when

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$$

Base $b$ expansion of $n$
What's the base 2 expansion of 42?

A. 111111
B. 100001
C. 101010
D. 110011
E. None of the above
What's the base 2 expansion of 42?

A. 111111
B. 100001
C. 101010
D. 110011
E. None of the above
What's the **biggest** integer value whose binary representation has 4 bits?

A. $2^4 = 16$
B. $2^3 = 8$
C. 4
D. 1000
E. None of the above
What's the **smallest** integer value whose binary representation has 4 bits?

A. 0
B. 1
C. 8
D. 1000
E. None of the above
**Fixed width** "binary expansions"

with 4 bits

(0000)$_2$ = 0
(0001)$_2$ = 1
(0010)$_2$ = 2
...
(1110)$_2$ = 14
(1111)$_2$ = 15

What about negative numbers?
From HW2... Two's complement

Use $n$ bits to represent integers in the range $[-2^{n-1}, 2^{n-1} - 1]$
From HW2... Two's complement

Use \( n \) bits to represent integers in the range \([-2^{n-1}, 2^{n-1}-1]\) e.g. for 4 bits the range is -8, -7, -6, ... , 5, 6, 7.

0000 \( \rightarrow \) 0

n-1 rather than \( n \) because use 1 bit to signal positive vs. negative
From HW2... Two's complement

Use \( n \) bits to represent integers in the range \([-2^{n-1}, 2^{n-1} - 1]\)
e.g. for 4 bits the range is -8, -7, -6, ..., 5, 6, 7.

.. 

0010 \rightarrow 2
0001 \rightarrow 1
0000 \rightarrow 0
From HW2… Two's complement

Use $n$ bits to represent integers in the range $[-2^{n-1}, 2^{n-1} - 1]$ e.g. for 4 bits the range is -8, -7, -6, …, 5, 6, 7.

Use leading 0 to indicate positive.

0010 $\rightarrow$ 2
0001 $\rightarrow$ 1
0000 $\rightarrow$ 0
From HW2... Two's complement

Use $n$ bits to represent integers in the range $[-2^{n-1}, 2^{n-1} -1]$

e.g. for 4 bits the range is -8, -7, -6, ... , 5, 6, 7.

..

0010 $\rightarrow$ 2
0001 $\rightarrow$ 1
0000 $\rightarrow$ 0

For negative integers $x$ the leftmost bit is 1 and the remaining
$n$-1 bits are the binary expansion of $2^{n-1} - |x|$.
From HW2... Two's complement

Use \( n \) bits to represent integers in the range \([-2^{n-1}, 2^{n-1} - 1]\). e.g. for 4 bits the range is -8, -7, -6, … , 5, 6, 7.

For negative integers \( x \) the leftmost bit is 1 and the remaining \( n-1 \) bits are the binary expansion of \( 2^{n-1} - |x| \).

What is the 4 bit two's complement representation of -1?

A. 0001  
B. 1001  
C. 1111  
D. 0111  
E. None of the above
From HW2... Two's complement

Use \( n \) bits to represent integers in the range \([-2^{n-1}, 2^{n-1} - 1]\)
e.g. for 4 bits the range is -8, -7, -6, … , 5, 6, 7.

For positive integers \( x \) the leftmost bit is 0 and the remaining
\( n-1 \) bits are the binary expansion of \( x \)

<table>
<thead>
<tr>
<th>Binary</th>
<th>2's Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>-1</td>
</tr>
<tr>
<td>1110</td>
<td>-2</td>
</tr>
</tbody>
</table>

For negative integers \( x \) the leftmost bit is 1 and the remaining
\( n-1 \) bits are the binary expansion of \( 2^{n-1} - |x| \)
Adding

What's the result of adding 2 and -2?

0010 → 2
0001 → 1
0000 → 0
1111 → -1
1110 → -2
...
Adding

What's the result of adding 2 and -2?

\[
\begin{array}{c c}
0010 & \rightarrow & 2 \\
0001 & \rightarrow & 1 \\
0000 & \rightarrow & 0 \\
1111 & \rightarrow & -1 \\
1110 & \rightarrow & -2 \\
\end{array}
\]

\[0010 + 1110 = 10000\]

Should equal 0
Adding

What's the result of adding 2 and -2?

0010 → 2
0001 → 1
0000 → 0
1111 → -1
1110 → -2

... Should equal 0

Key: to zero out bits, each column needs to cancel.
Arithmetic + Representations

In base b,

\[
\begin{align*}
    s_{k-1} \ldots s_1 & \; s_0 \\
+ t_{k-1} \ldots t_1 & \; t_0
\end{align*}
\]

Basic operations: one symbol addition, carry
### Arithmetic + Representations

For decimal

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Arithmetic + Representations

For binary

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Arithmetic + Representations

Alternatively,

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Arithmetic + Representations

Alternatively,

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Half adder logic circuit
## Computer bit operations

"x OR y is 1 if at least one of x or y is 1"

"x AND y is 1 if both x and y are 1"

"x XOR y is 1 if exactly one of x and y is 1"

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>x OR y</th>
<th>x AND y</th>
<th>x XOR y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>x OR y</td>
<td>x AND y</td>
<td>x XOR y</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

What is the **bitwise** AND of the strings 0011 and 0101?  
A. 0001  
B. 0111  
C. 0010  
D. 0100  
E. None of the above
### Computer bit operations

Rosen p. 11

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>x (\lor) y</th>
<th>x (\land) y</th>
<th>x (\oplus) y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

What is the **bitwise** XOR of the strings 0011 and 0101?

A. 0110  
B. 0000  
C. 0111  
D. 0100  
E. None of the above
Logic

• Use gates and circuits to express arithmetic.

• Precisely express theorems and invariant statements.

• Make valid arguments to prove theorems.
Definitions

- **Proposition**: declarative sentence that is T or F (not both)
- **Propositional variable**: variables that represent propositions.
- **Compound proposition**: new propositions formed from existing propositions using logical operators.
- **Truth table**: table with 1 row for each of the possible combinations of truth values of the input and an additional column that shows the truth value of the result of the operation corresponding to a particular row.
How many rows are in the truth table for $p \lor q$, the disjunction of $p$ and $q$?

A. 0110        D. 0100
B. 0000        E. None of the above
C. 0111
Truth tables

• Specify logical operator by truth table.
• Can use truth table to compute value of compound variable.

• Next time: how to prove two tables are equivalent?
Reminders

• Homework 2 due Friday
  • Integer representations
  • Algorithms

• Office hours