

COMP I 22/L Lecture 4

Mahdi Ebrahimi

Slides adapted from Dr. Kyle Dewey

Outline

- Operations on binary values
 - Addition
 - Subtraction
- Floating point introduction

Addition

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

			$\begin{array}{r} 6 \\ +3 \\ \hline \end{array}$ <p>?</p>
--	--	--	---

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

		$\begin{array}{r} 8 \\ +2 \\ \hline \end{array}$	$\begin{array}{r} 6 \\ +3 \\ \hline \end{array}$
		?	9

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

Carry: 1

$$\begin{array}{r} 8 \\ +2 \\ -- \\ 0 \end{array}$$

$$\begin{array}{r} 6 \\ +3 \\ -- \\ 9 \end{array}$$

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

	$\begin{array}{r} 1 \\ 9 \\ +1 \\ \hline \end{array}$ <p>?</p>		
--	--	--	--

	$\begin{array}{r} 8 \\ +2 \\ \hline \end{array}$ <p>0</p>		
--	---	--	--

	$\begin{array}{r} 6 \\ +3 \\ \hline \end{array}$ <p>9</p>		
--	---	--	--

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

Carry: 1

$$\begin{array}{r} 1 \\ 9 \\ +1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 8 \\ +2 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 6 \\ +3 \\ \hline 9 \end{array}$$

Building Up Addition

- Question: how might we add the following, in decimal?

$$\begin{array}{r} 986 \\ +123 \\ \hline \end{array}$$

?

$$\begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ 9 \\ +1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 8 \\ +2 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 6 \\ +3 \\ \hline 9 \end{array}$$

Core Concepts

- We have a “primitive” notion of adding single digits, along with an idea of *carrying* digits
- We can build on this notion to add numbers together that are more than one digit long

Now in Binary

- Arguably simpler - fewer one-bit possibilities

0	0	1	1
+0	+1	+0	+1
--	--	--	--
?	?	?	?

Now in Binary

- Arguably simpler - fewer one-bit possibilities

0 +0 -- 0	0 +1 -- 1	1 +0 -- 1	1 +1 -- 0 Carry:1
--------------------	--------------------	--------------------	-------------------------------

Chaining the Carry

- Also need to account for any input carry

$\begin{array}{r} 0 \\ 0 \\ +0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ 0 \\ +1 \\ \hline 1 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ +0 \\ \hline 1 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ +1 \\ \hline 0 \end{array} \text{ Carry: } 1$
$\begin{array}{r} 1 \\ 0 \\ +0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ 0 \\ +1 \\ \hline 0 \end{array} \text{ Carry: } 1$	$\begin{array}{r} 1 \\ 1 \\ +0 \\ \hline 0 \end{array} \text{ Carry: } 1$	$\begin{array}{r} 1 \\ 1 \\ +1 \\ \hline 1 \end{array} \text{ Carry: } 1$

Adding Multiple Bits

- How might we add the numbers below?

```
  011
+001
-----
```

Adding Multiple Bits

- How might we add the numbers below?

$$\begin{array}{r} 0 \\ 011 \\ +001 \\ \hline \end{array}$$

-Need an initial carry-in of zero

Adding Multiple Bits

- How might we add the numbers below?

$$\begin{array}{r} 10 \\ 011 \\ +001 \\ \hline \end{array}$$

0

-Need an initial carry-in of zero

Adding Multiple Bits

- How might we add the numbers below?

$$\begin{array}{r} 110 \\ 011 \\ +001 \\ \hline \end{array}$$

00

-Need an initial carry-in of zero

Adding Multiple Bits

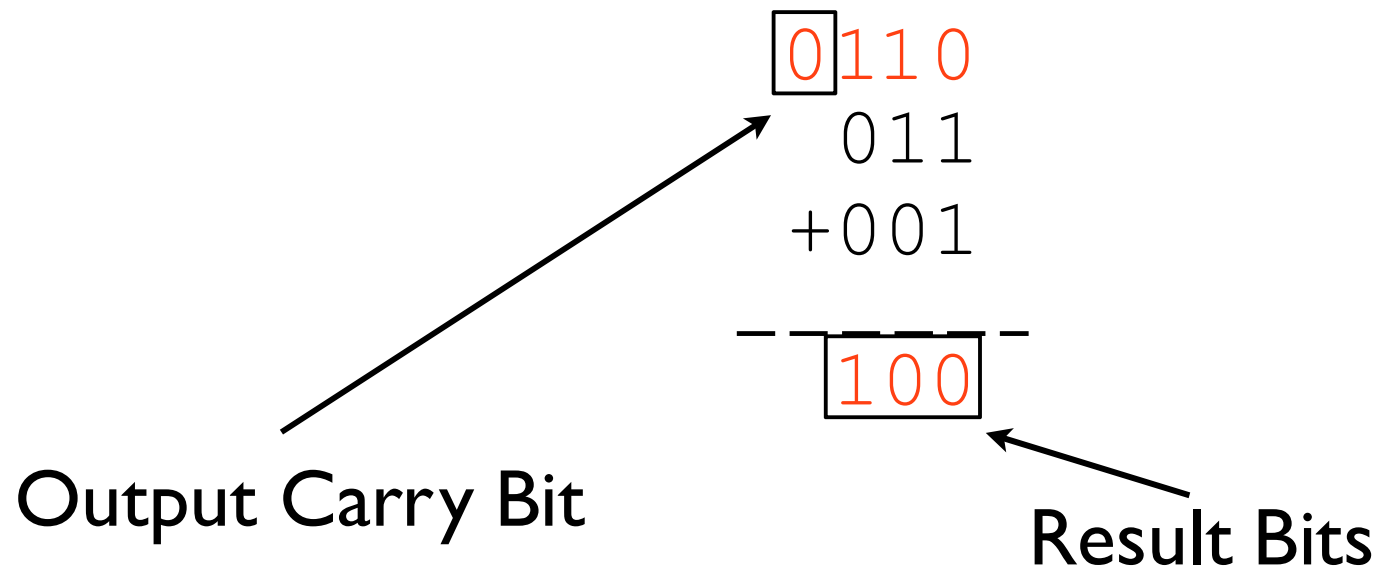
- How might we add the numbers below?

$$\begin{array}{r} 0110 \\ 011 \\ +001 \\ \hline 100 \end{array}$$

-Need an initial carry-in of zero

Adding Multiple Bits

- How might we add the numbers below?



-Need an initial carry-in of zero

Another Example

$$\begin{array}{r} 111 \\ +001 \\ \hline \end{array}$$

Another Example

$$\begin{array}{r} \\ 111 \\ +001 \\ \hline \end{array}$$

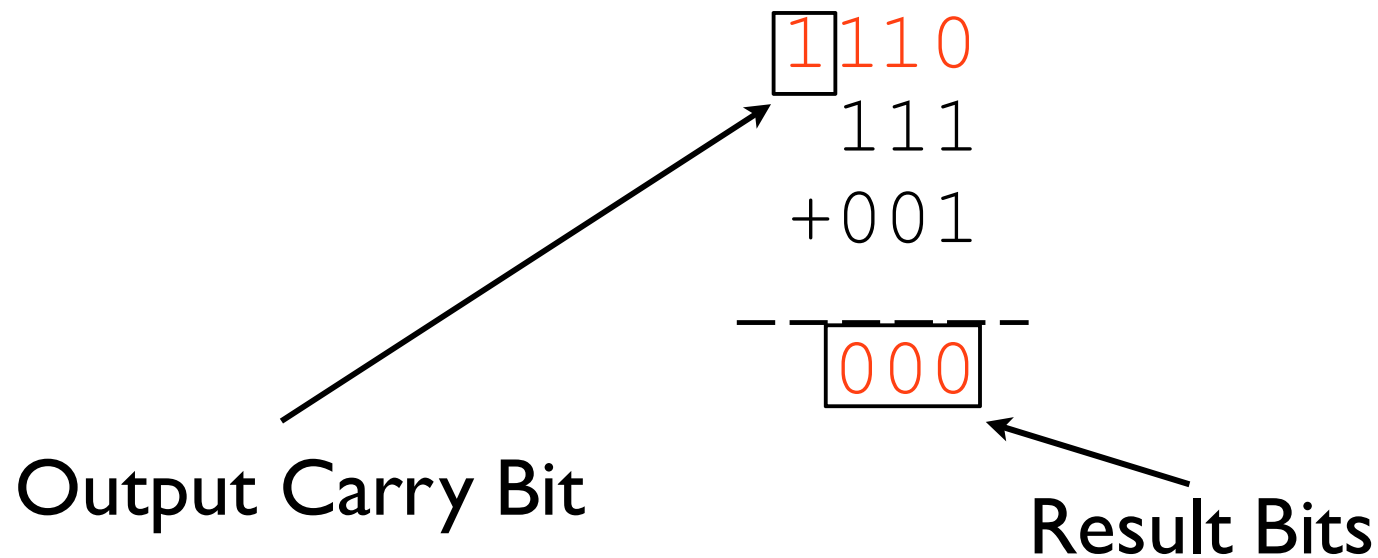
Another Example

$$\begin{array}{r} 10 \\ 111 \\ +001 \\ \hline 0 \end{array}$$

Another Example

$$\begin{array}{r} 110 \\ 111 \\ +001 \\ \hline 00 \end{array}$$

Another Example



-Now we have an output carry bit of 1. What does this mean?

Output Carry Bit Significance

- For unsigned numbers, it indicates if the result did not fit all the way into the number of bits allotted
- May be an error condition for software

Signed Addition

- Question: what is the result of the following operation?

$$\begin{array}{r} 011 \\ +011 \\ \hline \end{array}$$

?

Signed Addition

- Question: what is the result of the following operation?

$$\begin{array}{r} 011 \\ +011 \\ \hline 0110 \end{array}$$

-If these are treated as signed numbers in two's complement, then we need a leading 0 to indicate that this is a positive number
-Truncated to three bits, the result is a negative number!

Overflow

- In this situation, *overflow* occurred: this means that both the operands had the same sign, and the result's sign differed

$$\begin{array}{r} 011 \\ +011 \\ \hline 110 \end{array}$$

- Possibly a software error

Overflow vs. Carry

- These are **different ideas**
 - Carry is relevant to **unsigned** values
 - Overflow is relevant to **signed** values

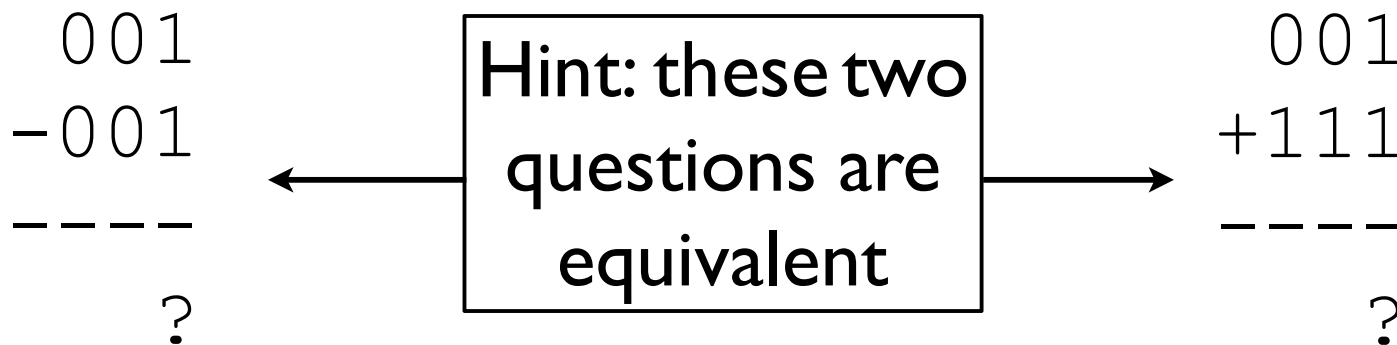
111 +001 ---- 000	011 +011 ---- 110	111 +100 ---- 011	001 +001 ---- 010
No Overflow; Carry	Overflow; No Carry	Overflow; Carry	No Overflow; No Carry

-As to when is it a problem, this all depends on exactly what it is you're doing

Subtraction

Subtraction

- Have been saying to invert bits and add one to second operand
- Could do it this way in hardware, but there is a trick



Subtraction Trick

- Assume we can cheaply invert bits, but we want to avoid adding twice (once to add 1 and once to add the other result)
- How can we do this easily?

Subtraction Trick

- Assume we can cheaply invert bits, but we want to avoid adding twice (once to add 1 and once to add the other result)
- How can we do this easily?
 - Set the initial carry to 1 instead of 0

Subtraction Example

```
  0101
- 0011
-----
```

Subtraction Example

$$\begin{array}{r} 0101 \\ -0011 \\ \hline \end{array} \quad \begin{array}{l} \text{Invert } 0011 \\ \hline \end{array}$$

Subtraction Example

$$\begin{array}{r} 0101 \\ -0011 \\ \hline \end{array} \quad \begin{array}{l} \text{Invert } 0011 \\ \hline \end{array} \rightarrow 1100$$

Subtraction Example

$$\begin{array}{r} 0101 \\ -0011 \\ \hline \end{array} \quad \begin{array}{c} \text{Invert } 0011 \\ \longrightarrow \end{array} \quad \begin{array}{c} 1100 \\ \longrightarrow \end{array} \quad \begin{array}{c} \text{Equivalent to} \\ \longrightarrow \end{array}$$

Subtraction Example

$$\begin{array}{r} 0101 \\ -0011 \\ \hline \end{array} \xrightarrow{\text{Invert } 0011} 1100 \xrightarrow{\text{Equivalent to}} \begin{array}{r} \overset{1}{0}101 \\ +1100 \\ \hline \end{array}$$

-An initial carry-in of 1 is equivalent to adding 1 and then adding the other operand

Subtraction Example

$$\begin{array}{r} 0101 \\ -0011 \\ \hline \end{array} \xrightarrow{\text{Invert } 0011} 1100 \xrightarrow{\text{Equivalent to}} \begin{array}{r} \boxed{1}1011 \\ 0101 \\ +1100 \\ \hline 0010 \end{array}$$

-An initial carry-in of 1 is equivalent to adding 1 and then adding the other operand