Representing Negative Numbers

In computers, and generally

Ancient History

There was an (excellent) early computer game that gave different leaders aggression scores of 0, 1 or 2. These were stored in 8 bit unsigned integers

	Aggression
Genghis Khan	00000010
Julius Ceaser	00000001
Ghandi	00000000

Ancient History

There was an (excellent) early computer game that gave different leaders aggression scores of 0, 1 or 2. These were stored in 8 bit unsigned integers

	Aggression	Dec
Ghandi	00000000	0
Democracy Mod	(00000010)	2
Democracy Mod (2s Comp)	11111110	-2
Sum	11111110	-2

(These values are added to two more 0, 1 or 2 values for a total threat.)

However late game - there was a modifier for the Indian leader - that he took Democracy his score was adjusted -2

Ancient History

There was an early (great) computer game that gave different leaders aggression scores of 0, 1 or 2. These were stored in 8 bit unsigned integers

	Aggression	Dec
Ghandi	00000000	0
Democracy Mod	(00000010)	2
Democracy Mod (2s Comp)	11111110	-2
Sum	11111110	-2

These values are added to two more 0, 1 or 2 values for a total threat.

However late game - there was a modifier for the Indian leader - that he took Democracy his score was adjusted -2

This is no problem - if this value is read as a signed number -2 is very low

Ancient History

There was an early (great) computer game that gave different leaders aggression scores of 0, 1 or 2. These were stored in 8 bit unsigned integers

	Aggression	Dec
Ghandi	00000000	0
Democracy Mod	(00000010)	2
Democracy Mod (2s Comp)	11111110	-2
Sum	11111110	254

These values are added to two more 0, 1 or 2 values for a total threat.

However late game - there was a modifier for the Indian leader - that he took Democracy his score was adjusted -2

This is no problem - if this value is read as a signed number -2 is very low

Greetings from M.Gandhi, ruler
and King of the Indians...
Our words are backed
with NUCLEAR WEAPONS!

(Not actually Civ 1)

So why is this -2?

11111110

The Challenge

Represent and use negative numbers in a digital computer

High and Low voltage interpreted as 1 and 0

Don't want special purpose sign bits - no help to arithmetic

So challenge is really representing negatives in just digits, base 2 onl
somewhat relevant

The Challenge

Represent and use negative numbers in a digital computer

High and Low voltage interpreted as 1 and 0

Don't want special purpose sign bits - no help to arithmetic

So challenge is really representing negatives in just digits, base 2 onl
somewhat relevant

The Challenge

Represent and use negative numbers in a digital computer

High and Low voltage interpreted as 1 and 0

Don't want special purpose sign bits - no help to arithmetic

So challenge is really representing negatives in just digits, base 2 onl
somewhat relevant

The Challenge

Represent and use negative numbers in a digital computer

High and Low voltage interpreted as 1 and 0

Don't want special purpose sign bits - no help to arithmetic

So challenge is really representing negatives in just digits, base 2 onl
somewhat relevant

The Answer(s)

< many attempts to numerous to mention>

Eventual solution originally proposed by Von Neumann 1945

Systems Called 2s complement

Based of method of complements in use for base 10 calculation

The Answer(s)

< many attempts to numerous to mention>

Eventual solution originally proposed by Von Neumann 1945

Systems Called 2s complement

Based of method of complements in use for base 10 calculation

The Answer(s)

< many attempts to numerous to mention>

Eventual solution originally proposed by Von Neumann 1945

Systems Called 2s complement

Based of method of complements in use for base 10 calculation

The Answer(s)

< many attempts to numerous to mention>

Eventual solution originally proposed by Von Neumann 1945

Systems Called 2s complement

Based of method of complements in use for base 10 calculation

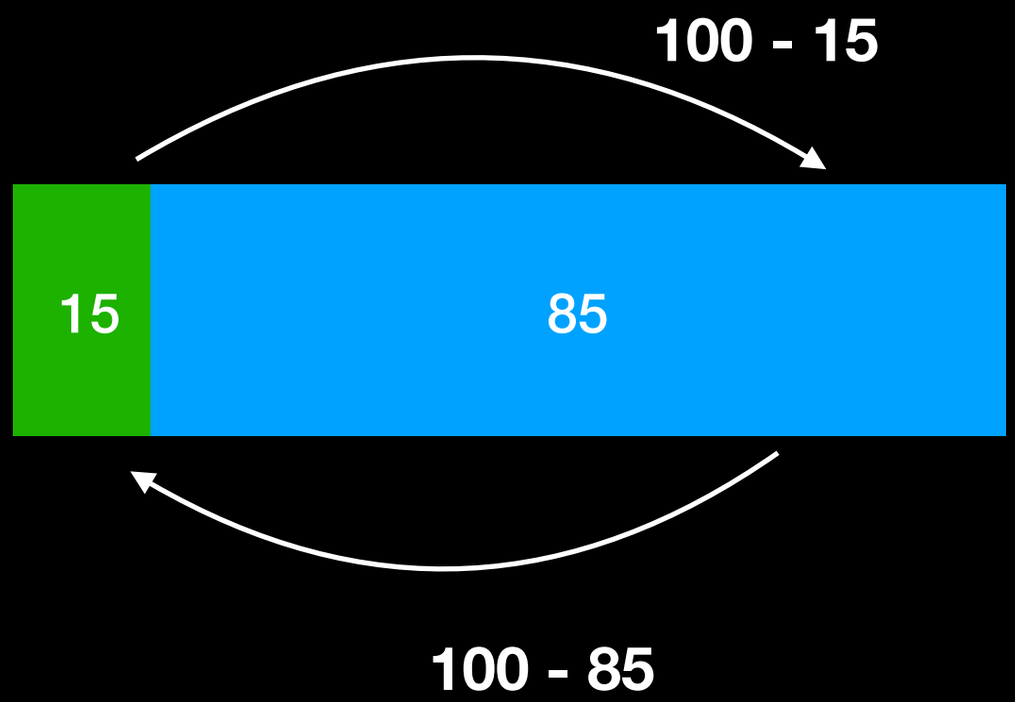
- **(9s Complement)**
- **10s Complement**

10s Complement

Represents k in n digits as the difference between k and 1$\underbrace{00...0}_{\text{n zeros}}$

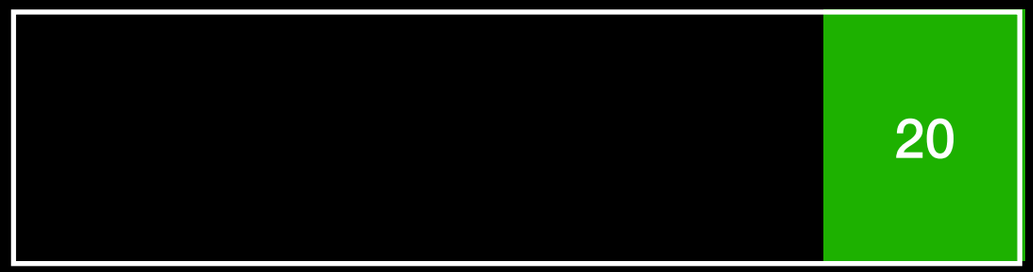
Bottom half represents positive numbers
Top half represents negative numbers

As the Base-complement operation (10s) is difference between number and base^n it is reversible



10s Complement - Example in 2 digits

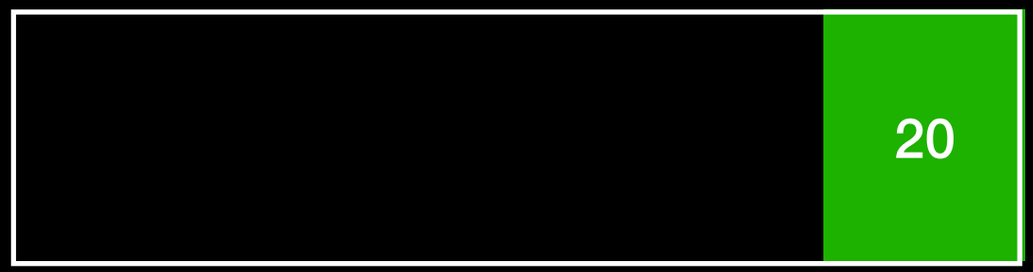
Makes Subtraction just addition of complement



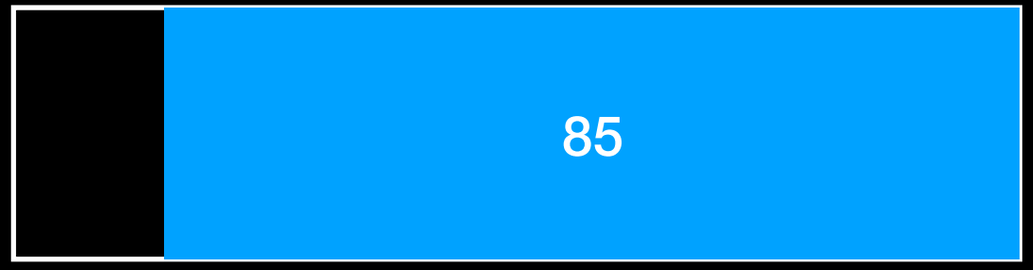
+

10s Complement - Example in 2 digits

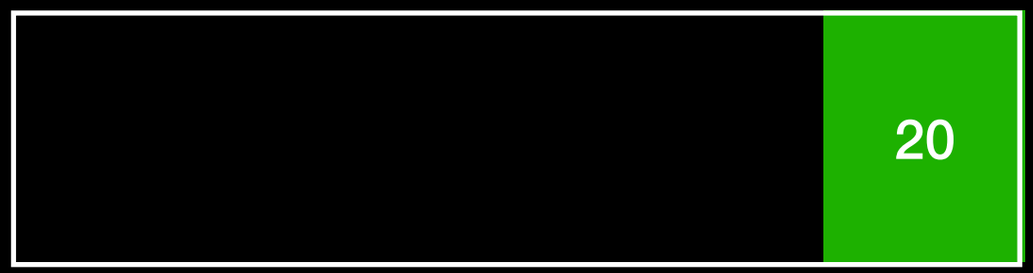
Taking the 10s complement of 15 this becomes



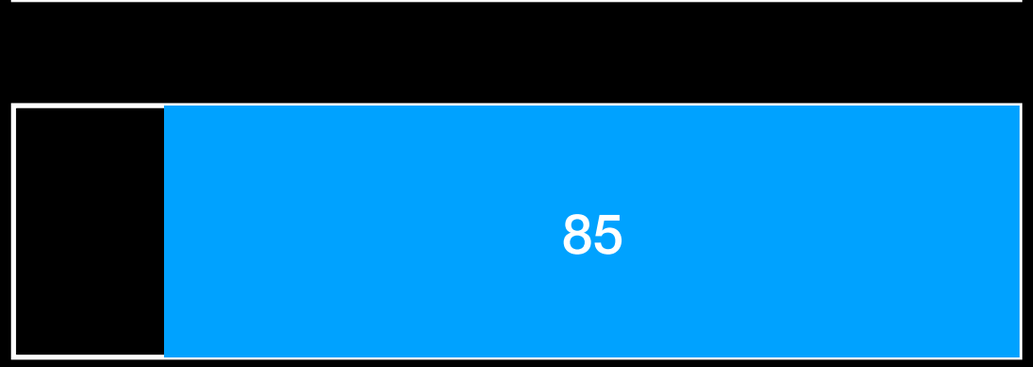
+

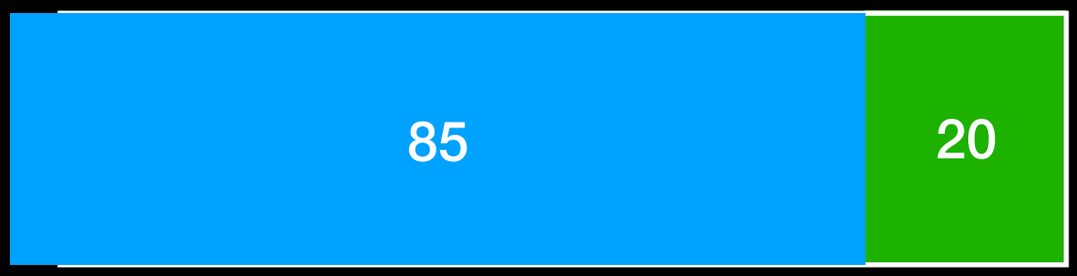


10s Complement - Example in 2 digits



+

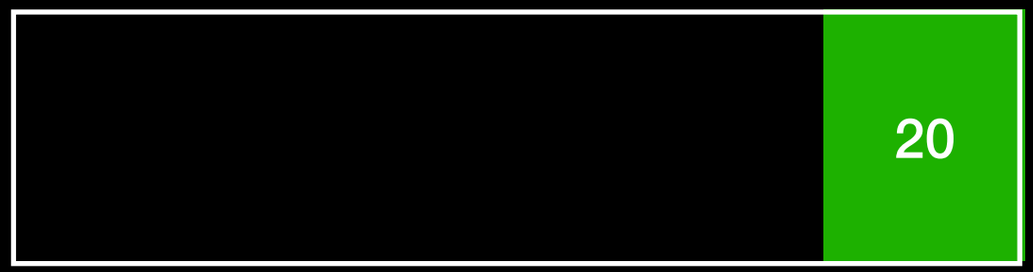




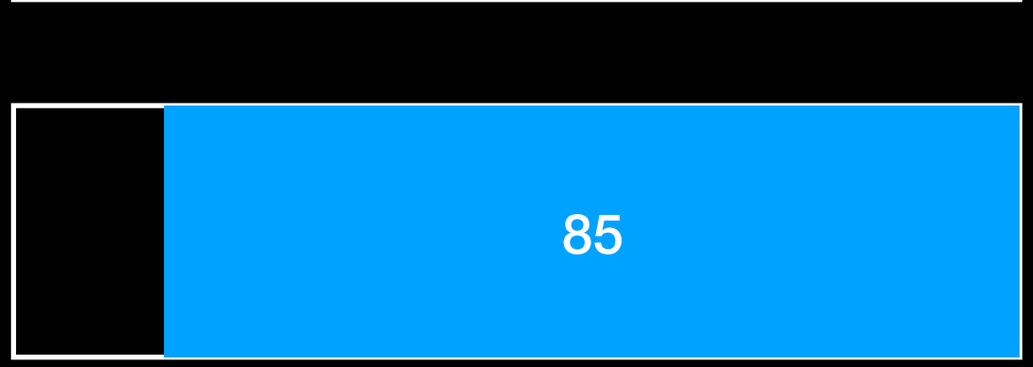
= 105

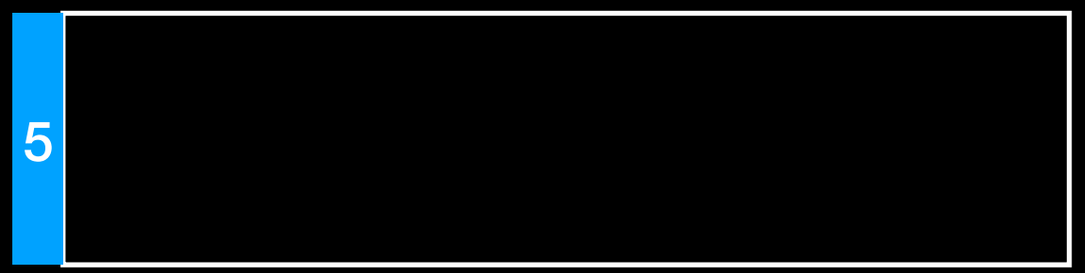
10s Complement - Example in 2 digits

But this exceeds our 2 digits - so we drop the 100



+





= 05

2s Complement

Represents binary k in n digits as the difference between k and 100...0 = 2^n

n zeros

2s Complement

Represents binary k in n digits as the difference between k and 100...0 = 2^n

n zeros

Bottom half represents positive numbers 0xxxxxx
Top half represents negative numbers 1xxxxxx

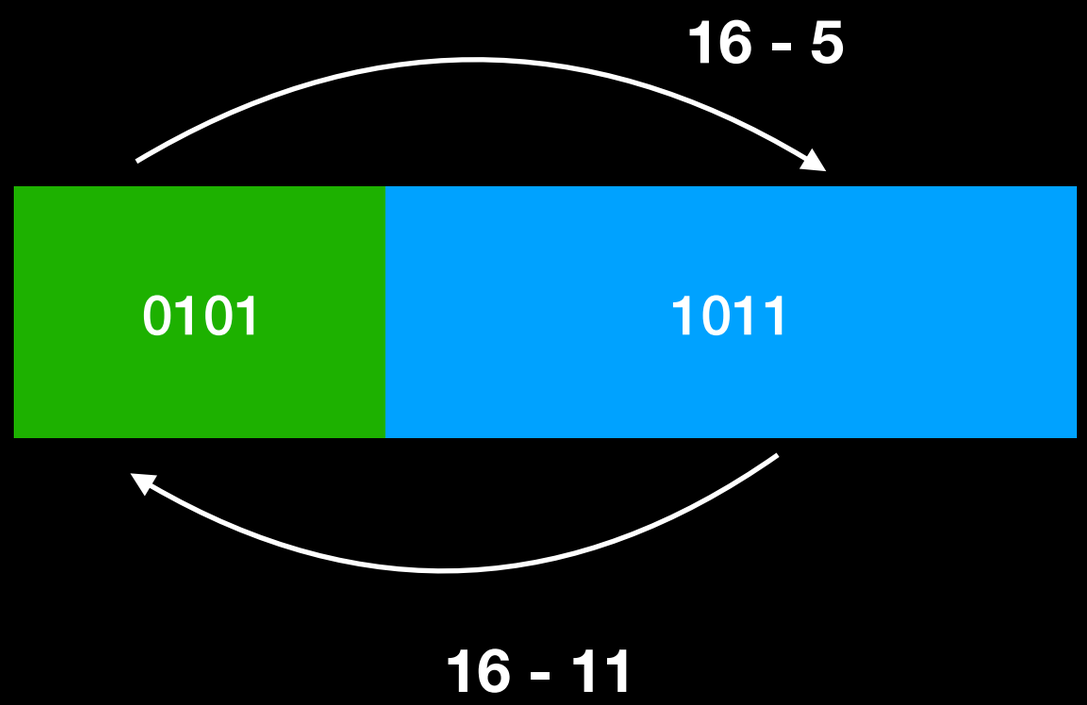
2s Complement

Represents binary k in n digits as the difference between k and 100...0 = 2^n

n zeros

Bottom half represents positive numbers 0xxxxxx
Top half represents negative numbers 1xxxxxx

Complement in 4 binary digits
2 ^ 4 = 16



Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp				
2s Comp				
Total				
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total				
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total			1	
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total			^{1}01	
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total			^{1}001	
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total			^{1}0001	
In 4 Digits				

But we are working in 4 digits - so we drop the 100 and get the answer

Example - 2s Complement Subtraction

Considering taking 6 - 5 in binary, just to keep things simple

We can work in 4 bits, and all positive number fit in the bottom 1/2

	Working	Dec	Sum	
LHS	0110	6	0110	6
RHS	0101	5		
1s Comp	1010	10		
2s Comp	1011	11	1011	11
Total			10001	17
In 4 Digits			**0001**	**1**

But we are working in 4 digits - so we drop the 10000 and get the answer

Adding machine for the Australian pound c.1910

9s Complements against each digit