32-bit Binary Calculator

User Guide

Dr. Bhargav Bhatkalkar

KFSCIS, Florida International University

Contents

1 Introduction **2**

2 Getting Started **2**

3 The Display **3**

4 Input Bases **3**

5 Arithmetic and Bitwise Operations **3**

6 Binary Column View **4**

7 Shift Operations **4**

8 Multiplying and Dividing with Shifts **5**

9 One-Value Operations **5**

10 Editing and Control Keys **6**

11 Keyboard Shortcuts **6**

12 Worked Examples **7**

13 Troubleshooting and FAQ **7**

1 Introduction

The 32-bit Binary Calculator is a browser-based tool for working with 32-bit integers (the size of a MIPS register). It shows every value in five formats at once:

- **HEX** (hexadecimal)
- **DEC signed** (two's complement)
- **DEC unsigned**
- **OCT** (octal)
- **BIN** (all 32 bits, with bit numbers 31 to 0)

It also performs bitwise logic, shifts, and sign or zero extension, and it shows binary addition and subtraction step by step in columns. The shift keys use MIPS names (sll, srl, sra), but all results follow standard 32-bit integer rules and apply to any processor.

2 Getting Started

Open the 32-bit Binary Calculator from its web link in any modern web browser (Chrome, Edge, Safari or Firefox). There is nothing to install.

It works on phones, tablets, and computers and adapts to any screen size. Tap the keys, click them with a mouse, or use a keyboard (see the Keyboard Shortcuts section). Press **? Help** at the top right to see help inside the calculator.

Quick start

1. Choose an input base: tap **HEX**, **DEC**, **OCT**, or **BIN**.
2. Type a value, press an operation key (for example, **AND** or **+**), and type a second value.
3. Press **=** (or Enter). The result appears in all five formats at once.

Figure 1 shows the main screen.



Figure 1: Main screen showing `0x0000FFFF AND 0x00000FF0 = 0x00000FF0`.

3 The Display

Part	What it shows
Top line	The input base (for example "Input: HEX"), the red note "Signed overflow" when it happens, and the **?** **Help** button.
Expression line	The calculation, for example `0x0000FFFF AND 0x00000FF0 =`
Result line	A small base tag (HEX, DEC, OCT, or BIN) and the digits of the result. BIN digits are grouped by 4.
Rows	The same value in HEX, DEC signed, DEC unsigned, and OCT.
BIN area	All 32 bits with bit numbers 31 to 0. Tap a bit to flip it.

4 Input Bases

Tap **HEX**, **DEC**, **OCT**, or **BIN**, or tap a row in the display. The selected base key turns yellow. Only the digits of that base can be typed. When you change the base, the value stays the same.

Base	Largest value you can type
HEX	8 digits (`FFFFFFFF`)
DEC signed	`2147483647`. To type a negative number, type the number and press **NEG**.
DEC unsigned	`4294967295` (tap the DEC unsigned row)
OCT	`37777777777`
BIN	32 digits

> **Tip:** to get −2147483648 (the most negative 32-bit value), type `80000000` in HEX, or tap the DEC unsigned row and type `2147483648`.

5 Arithmetic and Bitwise Operations

Type the first value, press the operation, type the second value, then press **=**. If you press **=** without a second value, the first value is used again ($5 + =$ gives 10). You can chain operations: $1 + 2 + 3 =$ gives 6.

Key	What it does
AND **OR** **XOR**	Bitwise AND, OR, XOR
NAND **NOR** **XNOR**	NOT of AND, OR, XOR
+ −	Add and subtract. The result wraps to 32 bits. The red note "Signed overflow" shows when the signed result of the last operation does not fit in 32 bits.

6 Binary Column View

In **BIN**, after **+** or − and **=**, the display shows the work in columns, like on paper: the first number, the second number, and the result, with bit numbers (31, 27, . . . , 3, 0) on top. For +, the **carry** row shows the carry into each bit, and the line under the result shows the carry into bit 31 and the carry out of bit 31.



Figure 2: Binary column view of `0x7FFFFFFF + 1`, with carries.

Figure 2 shows `0x7FFFFFFF + 1`. The carry into bit 31 is 1 and the carry out of bit 31 is 0. **Signed overflow happens when these two carries are different.**

7 Shift Operations

Type the value, press the shift, type the shift amount, then press **=**. The shift amount is **always typed in decimal (0 to 31)**, as in MIPS: `sll $t0, $t1, 4`. While you type the shift amount, the base keys, the display rows, the bits, the A–F keys, and the one-value keys are turned off until you press **=**.

Key	What it does
sll	Shift left logical. 0s fill from the right.
srl	Shift right logical. 0s fill from the left.
sra	Shift right arithmetic. Copies of bit 31 (the sign bit) fill from the left.

8 Multiplying and Dividing with Shifts

Shifting by n bits multiplies or divides by 2, 4, 8, 16, ... (2 to the power n). Example: $n = 3$ means $\times 8$ or $\div 8$. There is no separate arithmetic left shift (MIPS has no sla): **sll** works for both signed and unsigned values.

Key	Use it to
sll n	Multiply, signed or unsigned. 5 sll 3 = 40 (5×8). -5 sll 2 = -20 (-5×4). Bits shifted out of bit 31 are lost.
srl n	Divide an unsigned value. `0xF0000000` srl 4 = `0x0F000000` ($4026531840 \div 16 = 251658240$).
sra n	Divide a signed value. -64 sra 3 = -8 ($-64 \div 8$).

Warning 1: do not use srl on a negative value. -64 srl 3 = 536870904, not -8.
Warning 2: if a negative value does not divide exactly, sra rounds down, not toward zero. -7 sra 1 = -4, but MIPS div gives $-7 \div 2 = -3$. Word addresses and word offsets divide exactly by 4, so this does not affect them.

Tip: to type -5, tap the DEC signed row, type 5, then press **NEG**.

9 One-Value Operations

One-value keys act on the current value right away. You do not need to press **=**.

Key	What it does
NOT	Flip all 32 bits
NEG	Two's complement negate (0 − value)
SE16	Sign-extend the low 16 bits to 32 bits (MIPS: addi, lw, sw immediates; lh)
ZE16	Zero-extend the low 16 bits to 32 bits (MIPS: andi, ori, xori immediates; lhu)
SE8	Sign-extend the low 8 bits to 32 bits (MIPS: lb)
ZE8	Zero-extend the low 8 bits to 32 bits (MIPS: lbu)

Two's complement by hand: NEG is the same as NOT and then + 1. Try it: type 5, press **NOT**, then press **+** 1 **=**. The result is -5, the same as 5 **NEG**.

10 Editing and Control Keys

Key	What it does
AC	Clear everything
Delete key	Delete the last typed digit
Undo	Go back one step. Press it again to go back more steps (up to 50). It can also undo AC.
Redo	Go forward again after Undo, one step at a time. Use Undo and Redo to step back and forth through your work. A new change after Undo clears the Redo steps.
Bits	Tap any bit in the BIN area to flip it between 0 and 1
? Help	Open the help page

11 Keyboard Shortcuts

Key	What it does
0–9 **A–F**	Digits
Enter or **=**	Equals
Backspace	Delete the last digit
Esc	AC (when help is open, Esc closes help)
Ctrl+Z	Undo (Cmd+Z on a Mac)
Ctrl+Y	Redo (or Ctrl+Shift+Z; Cmd+Shift+Z on a Mac)
& **|** **^** **~**	AND, OR, XOR, NOT
+ **-**	Add, subtract
< **>**	sll, srl
Tab	Move between keys. Enter or Space presses the selected key. Type any shortcut key to go back to normal typing, where Enter is =.

Keys without a shortcut: sra, NEG, NAND, NOR, XNOR, SE16, ZE16, SE8, ZE8, and the base keys (HEX, DEC, OCT, BIN). Use them with a click, a tap, or Tab + Enter.

12 Worked Examples

Press **AC** before each example.

Base	Keys	Result
HEX	`FFFF AND FF0 =`	HEX `0x0000 0FF0`, DEC 4080
HEX	`F0F0 NOR F0F =`	HEX `0xFFFF 0000`, DEC signed -65536
HEX	`1 sll 31 =`	HEX `0x8000 0000`, DEC signed -2147483648
HEX	`80000000 srl 4 =`	HEX `0x0800 0000`, DEC 134217728
HEX	`80000000 sra 4 =`	HEX `0xF800 0000`, DEC signed -134217728
DEC	`100 NEG`	HEX `0xFFFF FF9C`, DEC signed -100
HEX	`8000 SE16`	HEX `0xFFFF 8000`, DEC signed -32768
HEX	`8000 ZE16`	HEX `0x0000 8000`, DEC 32768
HEX	`AB SE8`	HEX `0xFFFF FFAB`, DEC signed -85
HEX	`7FFFFFFF + 1 =`	HEX `0x8000 0000` and the note "Signed overflow"
BIN	`101 + 11 =`	BIN 1000 (5 + 3 = 8), shown in the column view with carries

13 Troubleshooting and FAQ

I can't type a digit.
Only digits that belong to the selected input base can be typed. For example, 8 and 9 are not valid in OCT, and A–F work only in HEX. Choose a different base, or tap a display row to switch.

Some keys are turned off.
While you type a shift amount, the base keys, display rows, bits, A–F keys, and one-value keys are turned off. Type the shift amount (0 to 31) and press **=** to finish the shift, or press **AC** to start again.

What does "Signed overflow" mean?
The signed result of the last addition or subtraction does not fit in 32 bits, so it wrapped around. In the BIN column view, overflow happens when the carry into bit 31 differs from the carry out of bit 31.

How do I enter a negative number?
Tap the DEC signed row, type the number, and press **NEG**. You can also type its two's-complement form directly in HEX (for example, `FFFFFFFF` for -1).

Why does srl give a large positive number for a negative value?
srl fills the left with 0s, so the sign is lost. Use sra to divide signed values. Note that sra rounds down (toward minus infinity), so -7 sra 1 gives -4.

How do I fix a mistake?
Press Delete (or Backspace) to remove the last digit, or press **Undo** (Ctrl+Z) to go back up to 50 steps. Undo can also reverse AC.