

I-WAR 2

High-Clock CPU Speed Fix

Simple HxD instructions for the verified one-byte flux.dll repair

About this fix

I-War 2 can run at extreme speed when the processor's invariant Time Stamp Counter (TSC) advances above **4,294,967,295 ticks per second** (approximately 4.295 GHz). The original 32-bit calculation wraps and the game uses a much smaller, incorrect timing value. This one-byte edit restores that single overflow.

The issue is most likely on newer high-clock desktop processors, including some AMD Ryzen and Intel Core systems whose invariant TSC reference exceeds the limit. CPU family and advertised boost clock alone do not decide it. Other processors are unaffected when their TSC stays below the limit, even if one core can briefly boost above 4.3 GHz.

Use this guide only for the verified DLL

The target **flux.dll** must be exactly **1,392,640 bytes** and must match the SHA-256 value below. If either value differs, stop. The offset in another DLL version may contain unrelated code.

File	flux.dll
Size	1,392,640 bytes
SHA-256	f5ceddfbebd4c23fe510d033918ccc1306eb02306157c3acf5d06151a5fcd39b
Offset	0x00018EF1
Change	00 to 01

Before editing

1. Close the game

Make sure I-War 2 is not running.

2. Back up flux.dll

Copy the original DLL to a safe folder. A backup named **flux.dll.original** beside the active file is also useful.

3. Verify the file

Confirm the exact file size above. To verify the hash, open PowerShell in the DLL folder and run:

```
Get-FileHash .\flux.dll -Algorithm SHA256
```

The result must match the SHA-256 value printed above. Then continue to the editing steps.

Apply the fix in HxD

1. Open the DLL

Start HxD and select **File > Open**. Open the verified **flux.dll**. HxD should remain in normal overwrite mode; do not insert or delete bytes.

2. Open Go To

Press **Ctrl+G**. Enter **18EF1**, select **hex**, select **begin**, and click **OK**.

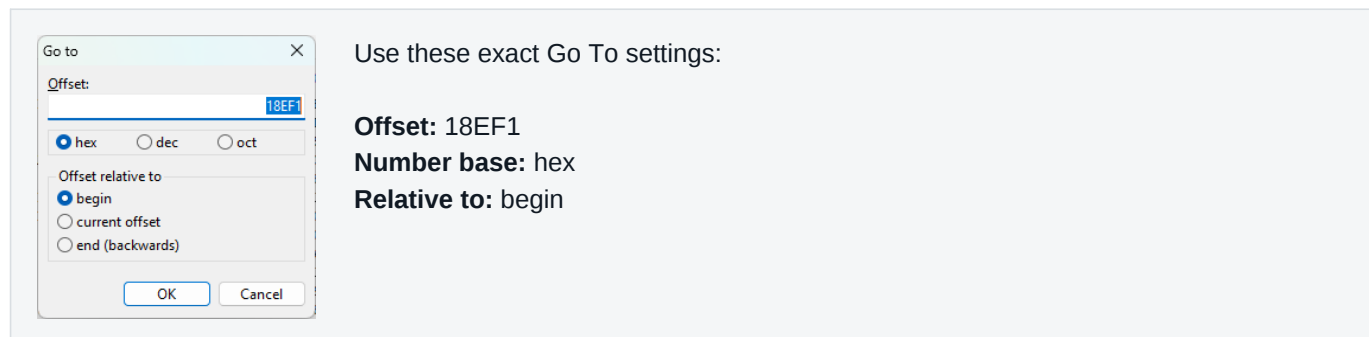


Figure 1. HxD Go To settings

3. Confirm the original byte

The byte at offset **00018EF1** must be **00**. The surrounding bytes must read:

89 75 F0 C7 45 F4 00 00 00 00 DF 6D F0

Stop if the byte or surrounding sequence differs

Do not make the change if offset 18EF1 is not 00 or if the surrounding sequence does not match. Restore the original DLL and confirm its size and SHA-256 hash.

4. Change one byte

Click the **00** at offset **00018EF1** and type **01**. Only that single byte should change. Do not add a byte or move the remaining data.

Confirm, save, and test

Before making the edit, HxD should place the cursor on the first **00** immediately after **F4**. The screenshot shows the verified location and surrounding bytes.

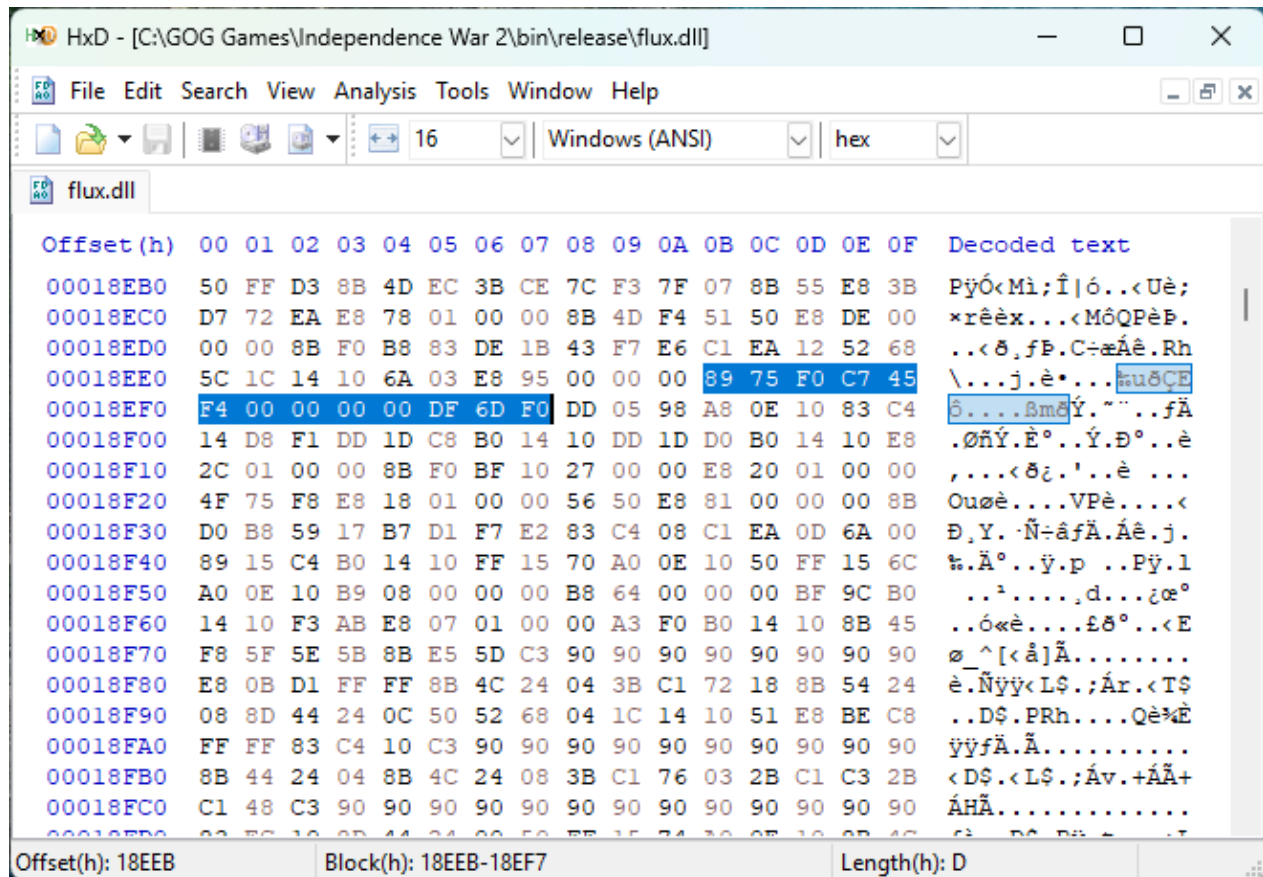


Figure 2. Verified HxD location before changing offset 18EF1 from 00 to 01

5. Verify the edited sequence

After the change, the bytes must read:

89 75 F0 C7 45 F4 01 00 00 00 DF 6D F0

6. Save

Select **File > Save**. Keep the filename **flux.dll**. Close and reopen the file, return to offset **18EF1**, and confirm the saved value is still **01**.

7. Check the size and test

Confirm that the DLL is still exactly **1,392,640 bytes**. Start I-War 2 normally. The game should now run at normal speed.

What the fix does

I-War 2 measures a processor timing counter during startup and stores the result as a 32-bit value. When the counter advances more than 4,294,967,295 times during the measurement period, the original calculation wraps around. The game then interprets the much smaller wrapped value as the processor timing rate, which makes simulation and animation run extremely fast.

The byte at offset **0x18EF1** is part of the instruction that initializes the upper half of the measured frequency. Changing it from **00** to **01** adds back one complete 32-bit range:

$$\text{corrected value} = \text{wrapped value} + 4,294,967,296$$

This corrects one overflow and is therefore appropriate for timing-counter rates from approximately **4.295 GHz** up to just under **8.590 GHz**. It is not a general performance enhancement and should not be applied to an unverified DLL.

The threshold is the processor's timing-counter rate

It is not necessarily the live frequency shown for an individual CPU core. Modern processors can boost and downclock while their invariant timing counter remains at a stable reference rate.

If something goes wrong

Problem	Action
Game will not start	Restore the untouched backup and verify that only one byte was overwritten.
Game runs too slowly	Restore the original DLL. The timing-counter rate may be below the range this edit corrects.
File size changed	Restore the backup. A byte was inserted or deleted instead of overwritten.
Hash or bytes differ	Do not patch that DLL at this offset. It requires separate verification.

Restoring the original DLL

Close the game, delete or rename the edited **flux.dll**, and copy the untouched backup into its place under the original filename **flux.dll**.

Revision: 2026-08-18 | Manual one-byte compatibility repair