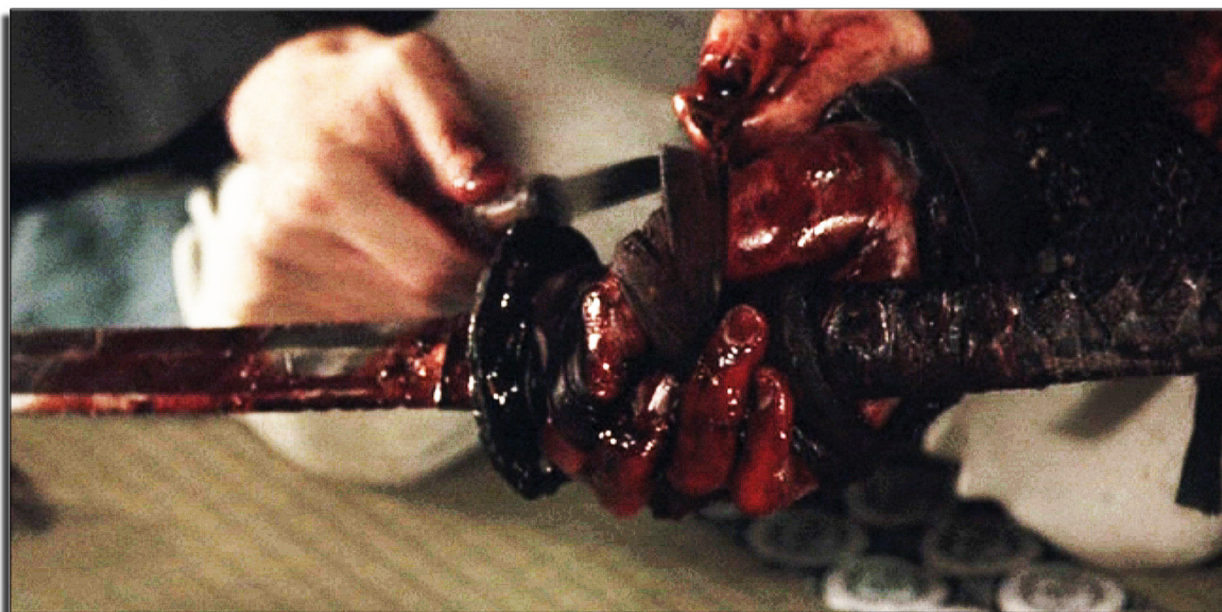




If I only had had Yoshimura at my side... the 301 would have held the "HOT GATES" for one more precious day, alas, I disregarded Sanmayce's advice to invite the "NORTH STAR"... I should not have killed that wolf back then... KARMA.

Why I didn't pay the Shogun those heaps of gems for borrowing that hatamoto (personal guard) of his, Sanmayce was right, that Mibu Wolf would have prevented the shameful delay and casualties... unforgivable mistake on my part.



@King\_of\_Sparta\_Leonidas @King\_of\_Persia\_Xerxes

Kings, don't be too harsh on yourselves, my words "*In order to sustain/inflict heavy damage on the battlefield you ought to implement the secret alchemical Interleaving & Interlacing techniques.*" were not easy to decipher.

One DVD per second as 32bit code, on Core 2 T7500 2200MHz the DIAMANTINE FNV1A\_Yoshimura has slashed one 64MB block at 4488MB/s, 'Everest' says 5395MB/s Memory Read, i.e. 17% below the MAX, the FASTEST! The homepage of 'Yoshimura' is [http://www.sanmayce.com/Fastest\\_Hash](http://www.sanmayce.com/Fastest_Hash)

# Yoshimura

## Fastest 32bit hash table function

```
uint32_t FNV1A_Hash_Yoshimura(const char *str, uint32_t wrdlen)
{
    const uint32_t PRIME = 709607;
    uint32_t hash32 = 2166136261;
    uint32_t hash32B = 2166136261;
    const char *p = str;
    uint32_t Loop_Counter;
    uint32_t Second_Line_Offset;

    if (wrdlen >= 2*2*sizeof(uint32_t)) {
        Second_Line_Offset = wrdlen-((wrdlen>>4)+1)*(2*4); // ((wrdlen>>1)>>3)
        Loop_Counter = (wrdlen>>4);
        //if (wrdlen%16) Loop_Counter++;
        Loop_Counter++;
        for(; Loop_Counter; Loop_Counter--, p += 2*sizeof(uint32_t)) {
            hash32 = (hash32 ^ (_rotl(*(uint32_t *) (p+0),5) ^ *(uint32_t *) (p+0+Second_Line_Offset))) * PRIME;
            hash32B = (hash32B ^ (_rotl(*(uint32_t *) (p+4+Second_Line_Offset),5) ^ *(uint32_t *) (p+4))) * PRIME;
        }
    } else {
        // Cases: 0,1,2,3,4,5,6,7,...,15
        if (wrdlen & 2*sizeof(uint32_t)) {
            hash32 = (hash32 ^ *(uint32_t *) (p+0)) * PRIME;
            hash32B = (hash32B ^ *(uint32_t *) (p+4)) * PRIME;
            p += 4*sizeof(uint16_t);
        }
        // Cases: 0,1,2,3,4,5,6,7
        if (wrdlen & sizeof(uint32_t)) {
            hash32 = (hash32 ^ *(uint16_t *) (p+0)) * PRIME;
            hash32B = (hash32B ^ *(uint16_t *) (p+2)) * PRIME;
            p += 2*sizeof(uint16_t);
        }
        if (wrdlen & sizeof(uint16_t)) {
            hash32 = (hash32 ^ *(uint16_t *) p) * PRIME;
            p += sizeof(uint16_t);
        }
        if (wrdlen & 1)
            hash32 = (hash32 ^ *p) * PRIME;
    }
    hash32 = (hash32 ^ _rotl(hash32B,5) ) * PRIME;
    return hash32 ^ (hash32 >> 16);
}
```



Below, the results after running 32bit code by *Intel 12.1 compiler (/Ox used)*:

Linear speed on *Fantasy's Black-and-Red Rig (i7-3930K, 4500MHz, CPU bus: 125MHz, RAM bus: 2400MHz Quad Channel)*:

Fetching/Hashing a **64MB** block 1024 times i.e. 64GB ...

BURST_Read_4DWORDS:	(64MB block); 65536MB fetched in 4584 clocks or	14.297MB per clock
FNV1A_YoshimitsuTRIAD:	(64MB block); 65536MB hashed in 5623 clocks or	11.655MB per clock
FNV1A_Yorikke:	(64MB block); 65536MB hashed in 6212 clocks or	10.550MB per clock
<b>FNV1A_Yoshimura:</b>	(64MB block); 65536MB hashed in 5329 clocks or	<b>12.298MB per clock</b>
CRC32_SlicingBy8:	(64MB block); 65536MB hashed in 37555 clocks or	1.745MB per clock

Fetching/Hashing a **16KB** block 4\*1024\*1024 times ...

BURST_Read_4DWORDS:	(16KB block); 65536MB fetched in 1968 clocks or	33.301MB per clock
FNV1A_YoshimitsuTRIAD:	(16KB block); 65536MB hashed in 4393 clocks or	14.918MB per clock
FNV1A_Yorikke:	(16KB block); 65536MB hashed in 5126 clocks or	12.785MB per clock
<b>FNV1A_Yoshimura:</b>	(16KB block); 65536MB hashed in 4551 clocks or	<b>14.400MB per clock</b>
CRC32_SlicingBy8:	(16KB block); 65536MB hashed in 36227 clocks or	1.809MB per clock

*For reference: [www.sanmayce.com/Fastest\\_Hash/index.html#Yoshimura](http://www.sanmayce.com/Fastest_Hash/index.html#Yoshimura)*