

## Re: Fastcode : function won't work with length>65535 ?

---

*Source:*

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.language.basm/2006-10/msg00109.html>

---

- *From:* Nibbler <[no@spam](mailto:no@spam)>
  - *Date:* Tue, 10 Oct 2006 11:57:57 +0200
- 

Hi, thank you very much Davy !

Unfortunately, I need a function that can restart it's job from an older value of the crc ... do you know how to change the function to have an old crc value as a parameter ?

Thanks a Lot !

Davy Landman a écrit :

I too was searching for the fastest CRC function and found the folowing to be faster than your function :)

have fun!

```
function CalculateCRC32(var ABuffer; const BufferSize:LongWord): LongWord;
```

```
asm
```

```
push esi
```

```
push edi
```

```
push ebx
```

```
mov edi,edx
```

```
mov esi,eax
```

```
xor ebx,ebx
```

```
mov eax,$ffffff
```

```
mov ecx,edi
```

```
shr ecx,2
```

```
jecxz @Rest
```

```
@Loop:
```

```
mov edx,[esi]
```

```
mov bl,al
```

```
xor bl,dl
```

```
shr eax,8
```

```
xor eax,dword ptr [CRC32table+ebx*4]
```

Re: Fastcode : function won't work with length>65535 ?

```
mov bl,al
xor bl,dh
shr eax,8
xor eax,dword ptr [CRC32table+ebx*4]

shr edx,16

mov bl,al
xor bl,dl
shr eax,8
xor eax,dword ptr [CRC32table+ebx*4]

mov bl,al
xor bl,dh
shr eax,8
xor eax,dword ptr [CRC32table+ebx*4]

add esi,4

loop @Loop

@Rest:
mov ecx,edi
and ecx,3
jecxz @End

@Loop_Rest:
mov bl,al
xor bl,[esi]
shr eax,8
inc esi
xor eax,dword ptr [CRC32table+ebx*4]
loop @Loop_Rest

@End:

xor eax,$ffffff

pop ebx
pop edi
pop esi
end;
"Nibbler" <no@spam> wrote in message news:451b8148@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Hello everyone,

Searching for a fast CRC function I found on google groups the Aleksandr Sharahov optimized one :

```
function GetShaCRC11(OldCRC: cardinal; StPtr: pointer; StLen: integer):
cardinal;
```

Re: Fastcode : function won't work with length>65535 ?

Re: Fastcode : function won't work with length>65535 ?

```
asm
jcxz @ret
push ebx
xor ebx, ebx
@loop:
movzx ebx, byte ptr [edx]
inc edx
xor bl, al
// shl ebx, 02
shr eax, 08
xor eax, dword ptr [CRCtable+ebx*4]
dec ecx
jne @loop
pop ebx
@ret:
ret
end;
```

It works great and faster than the previous one I was using, but it seems to be limited to a StLen<=\$FFFF

Using a length > 65535 and an OldCRC=\$FFFFFFFF (to be compatible with the zip CRC) will result in an immediate exit.

I really don't have any asm knowledge, would you please help me to make it work with a StLen>\$FFFF ?

Thank you in advance,  
Nibbler.