

## Re: Disobeying jet engines – why?

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-01/msg04932.html>

---

- *From:* Didi <[didi@xxxxxxxx](mailto:didi@xxxxxxxx)>
  - *Date:* Wed, 30 Jan 2008 16:49:20 -0800 (PST)
- 

I am all for people sitting around and spending the time to write highly-optimized code in assembly when the job calls for it (and your code at <http://tgi-sci.com/vpaex/fcs.txt> looks pretty nice...).

[ BTW, the source is at <http://tgi-sci.com/vpaex/fcs.sa>, this is the assembler list with native opcodes. ]

Actually code optimization is not higher on my list than it is for a C programmer. I did this within an hour or so, as long as it took me to read the description of the algorithm in rfc1662 and this is the first time I look back at it since (last modified 08.18.2004.. :-).

You may be able to match this in C but you cannot do it any faster, and you need not to – this is fast enough, as fast as one moves from one thing to another.

What is key about using VPA (which originates from 68k assembly having evolved since) is the fact that you choose the language level, e.g. on one line you copy a register to another and on the next line you "do" something with an object, pointed to by a5... Language level variability is inherent to any language, human included.

High level programming languages put a lower limit on that which is too high – which is why programmers are slower using them. Similar to menu and command line driven systems: the former are great for first time users, the latter are a much faster way of doing work for those who do this all the time.

Dimiter

---

Dimiter Popoff Transgalactic Instruments

<http://www.tgi-sci.com>

---

[http://www.flickr.com/photos/didi\\_tgi/sets/72157600228621276/](http://www.flickr.com/photos/didi_tgi/sets/72157600228621276/)

Joel Koltner wrote:



Re: Disobeying jet engines – why?

```
\ 0000000C 912C LD R18, X
103 UCHAR i = (rc & 0xff);
\ 0000000E 0180 MOVW R17:R16, R1:R0
104 i ^= nb;
\ 00000010 2702 EOR R16, R18
105 rc = rc >> 8;
\ 00000012 2C01 MOV R0, R1
\ 00000014 2C12 MOV R1, R2
\ 00000016 2C23 MOV R2, R3
\ 00000018 2433 CLR R3
106 rc ^= CRCTable [i];
\ 0000001A .... LDI R30, LOW(CRCTable)
\ 0000001C .... LDI R31, HIGH(CRCTable)
\ 0000001E .... LDI R19, (CRCTable) >> 16
\ 00000020 E010 LDI R17, 0
\ 00000022 0F00 LSL R16
\ 00000024 1F11 ROL R17
\ 00000026 0F00 LSL R16
\ 00000028 1F11 ROL R17
\ 0000002A 0FE0 ADD R30, R16
\ 0000002C 1FF1 ADC R31, R17
\ 0000002E BF3B OUT 0x3B, R19
\ 00000030 9047 ELPM R4, Z+
\ 00000032 9057 ELPM R5, Z+
\ 00000034 9067 ELPM R6, Z+
\ 00000036 9076 ELPM R7, Z
\ 00000038 2404 EOR R0, R4
\ 0000003A 2415 EOR R1, R5
\ 0000003C 2426 EOR R2, R6
\ 0000003E 2437 EOR R3, R7
107 buf++;
\ 00000040 9611 ADIW R27:R26, 1
108 len--;
\ 00000042 5061 SUBI R22, 1
\ 00000044 4070 SBCI R23, 0
109 }
\ ??CalcCRC_0:
\ 00000046 2F06 MOV R16, R22
\ 00000048 2B07 OR R16, R23
\ 0000004A F701 BRNE ??CalcCRC_1
110
111 return rc;
\ 0000004C 0180 MOVW R17:R16, R1:R0
\ 0000004E 0191 MOVW R19:R18, R3:R2
\ 00000050 E0E8 LDI R30, 8
\ 00000052 ..... JMP ?EPILOGUE_B8_L09
112 }
```

"I leave it to the rest of the group to say which piece of code is easier to read and understand – and thus keep under control."

## Re: Disobeying jet engines – why?

I'd vote for neither one, since for something like a CRC understanding how a CRC really works (that you're more or less computing the remainder of one verrryyyy long division using almost-but-not-quite modulo-n arithmetic, but, oh, wait... sometimes people do things bit-reversed and use various starting and ending XORs for various purposes... and then to do things efficiently you can start using table lookup to process multiple bits at once...) is a lot harder than understanding either of the pieces of code.

---Joel