Code Gremlins

The SD-8516 Gremlin strikes again

Gremlin Catalog

Gremlin	Symptom	Fix
CABC Gremlin	JC/JNC backwards everywhere	Remember: CF = (A >= B), JC = jump if >=
AH Garbage Gremlin	Numbers like 58922 instead of 42	Always `LDAH #0` before 8-bit accumulation
# vs @ Gremlin	Tokens = 0 or 351 instead of $99	Use `@LABEL` not `#LABEL` for constants
Register Overlap Gremlin	ELM and ELD share EL! GLD and FLD share D!	Use independent pairs (GLK vs FLD)
Stack Imbalance Gremlin	Returns to $292929	Match every PUSH with POP on ALL paths
POP Clobbers Flags Gremlin	Carry flag lost after function call	Use F/B/U flags instead of CF for returns
Reversal Loop Gremlin	“42” prints as blank or “24”	JNC → JC in pointer comparison

Meet the Gremlins

When the VC-3 portal opened between dimensions, something else came through...

The CABC Gremlin

Carry or No Carry? JC? JNC? Left or Right, up or down, cats and dogs living together... Chaos!

This mischievous creature inverts your conditional jumps when you least expect it. It whispers “JNC” when you mean “JC” and cackles as your loops run upside down, back to front, forever or never at all.

Weakness: The mantra “If A >= B, set the C!” (If A shoots B… like an arrow, points to C)

The "High on Garbage" Gremlin

“What's in the high byte? Surprise!”

Lurks in the shadows of 8-bit operations, leaving mysterious garbage in AH. Your number 42 becomes 58922. Your line number 10 becomes 25610.

Weakness: The cleansing spell <pre>LDAH #0</pre>. Always initialize your variables!

The Sigil Gremlin

“Is it a number? Is it a name? Who knows!”

Confuses immediate values with label references. #TOK_PRINT becomes zero, labels vanish into the void.

Weakness: Remember the ancient law: # for decimals, $ for hex, @ for labels.

Doppelganger Gremlins

**“We are many, yet we are one...”*

ELM and ELD look different but share a soul (the EL register). Modify one, corrupt the other. FLD and GLD both claim the D register. More subtle, AL and AB. Modify one, modify the other.

Weakness: The separation ritual: use independent pairs (GLK > GLD when paired with FLD).

Stack Goblins

“What goes up must come down... eventually...” Aaand... it's GONE!

There are two forms of this goblin. They look exactly the same except one of them has a beard and a red hat, while all the other goblins have white hats. These goblins are relentless, and they are constantly trying to fuck with the stack.

The first one looks for functions with unbalanced PUSH/POP. Returns to address $292929. Endless calls to LDA #0000. Corrupts everything in its wake.

But the POPA goblin is the worst. He lies to you and tells you this is better but behind your back he is stealing our return values. POPA over-writes a return value, or you accidentally PUSH/POP a register that is used as a return value.

Weakness: Just like you never count your chickens before they hatch, never count your pushes before your pops! Pushes and pops have to match – and don't PUSH/POP return values. PUSHA/POPA must be carefully considered. With great power comes great responsibility.

Flag Goblins

“Your carry flag? I'll take that.”

This little green monster is technically a goblin – he's always tryina' gobblin' ur flags. He steals your carefully-set carry flag during innocent POP instructions. Your success becomes failure, your failure becomes chaos. Well, at least you THINK it was a POP operation.

Weakness: Use non-reserved flags or put return values in registers. Declare your spec on every function: NAME, PURPOSE, IN, OUT, NOTES. Also avoid using protected flags like CARRY to return data. Use the user-facing flags F, B and U as they are immune to this goblin's trickery.

Mirror Gremlins (Reversal Bug)

“42? Not so fast! You mean... 24! Hahahaha!”

Inverts your strings, reverses your numbers, makes everything backwards.

Weakness: His true name. You must call him by his true name: The carry gremlin. Call him by his name and, his true form revealed, he will disappear in a puff of smoke.

Zero gremlins

This is really a goblin, but it is wearing a mask. Remove the mask and you see.. a goblin! Hes gobblin your CPU time. But wait.. he's wearing another mask! Remove the mask and what do you find?

```
NOP
```

Great scott! We're being invaded by NOPs!

To get rid of this Goblin, remember which ops set cpu flags. Before you flagrantly issue a CMP against zero, ask yourself; is it already set?

Here is the magic scroll to find and eradicate these gremlins; but beware, they are wearing masks, you must thoroughly investigate each case.

```
[^;]CMP [A-Za-z]+,\s*[#$]?0\b
```

or maybe

```
^\s+(LD|ADD|SUB|AND|OR|XOR|SHR|SHL|INC|DEC|MOV)\w*.*\n\s+CMP\s+[A-Za-z]+,\s*[#$]?0\b\n\s+JZ
```

Surefire cases look like this:

```
LDA [$C000]
CMP A, #0
JZ @label
```

Here, the CMP is always redundant as LD sets Z. Other common ops that set Z; ST, AND, DIV, MOV, INC, DEC...

Zero gremlins II: The Revenge

The Zero gremlin never really went away, he just changed masks.

```
^\s+(LD|ST)\w+.*\n\s+(INC|DEC)\s+
^\s+(INC|DEC)\s+.*\n\s+(LD|ST)\w+
```

The above will unmask these goblins right away!

History

The Stellar Dynamics engineers learned to respect these creatures. For every gremlin caught, the VC-3 grew stronger...

Lessons Learned

When something weird happens, check:

1. **Is AH/BH cleared?** (8-bit ops leave high byte dirty) 2. **Is it # vs @?** (immediates vs labels) 3. **Is the carry logic inverted?** (CABC strikes again!) 4. **Do pointers share registers?** (ELM/ELD/FLD all overlap!) 5. **Is stack balanced on ALL paths?** (especially error paths)

Always remember: Gremlins are just part of the retro experience. They never really went away, but always remember, the Gremlins never win. They always blow up in a microwave, fall into a blender, or catch fire or something. You just have to keep trying.