

FURTHER ADVENTURES OF



THE PEARL SPECTRUM EMULATOR

Jon Mitchell

What is it?

The Spectrum: Popular home computer in the early 1980s

Created by Clive Sinclair

8bit Z80 processor, 48k RAM

The Emulator: The above, but in Perl

Some of you know this already...



Flashback - YAPC::EU 2004

Flashback - YAPC::EU 2004



Flashback - YAPC: :EU 2004



ALL WAS NOT AS IT SEEMED

Demo was genuine, but just not complete

Miner Willy couldn't jump

Could only run 2 games (Manic Miner and JSW)

Everything else crashed

But I didn't tell people that

Fast Forward to Today

Did nothing until two weeks ago

Some progress

- Jet Set Willy
- He can jump
 - He can collect objects!
 - He can die!

Lets see some code...

No! The Perl code!

At YAPC::EU I said that drawing the screen is the slowest operation. It isn't.

Drawing the screen takes just 0.02 secs – but it does get drawn 4 times a second

This is good news – there are still speed ups to be had

So what is it doing that is so slow?

CALLING SUBROUTINES

Originally all code was object orientated. So setting a register would be `$r->bc()`;

Registers can are 8 bit, but can be grouped into a 16 bit value

Still need a function call:

my (\$b, \$c);

```
sub b {
  @_ and $b = $_[0];
  $b;
}

sub c {
  @_ and $c = $_[0];
  $c;
}

sub bc {
  if (@_) {
    $b = ($_[0] >> 8);
    $c = ($_[0] & 0xff);
  } else {
    return ($b << 8) + $c;
  }
}
```

Example Z80 Opcode
- 0x03 INC BC

```
$instr_table[0x03] = # INC BC  
sub { $tstates += 6; r16_inc('bc'); };
```

```
sub r16_inc {  
  $old_value = &{$_[0]}();  
  &{$_[0]}(($old_value + 1) & 0xffff);  
  return $old_value;  
}
```

```
sub bc {  
  if (@_) {  
    $b = ($_[0] >> 8);  
    $c = ($_[0] & 0xff);  
  } else {  
    return ($b << 8) + $c;  
  }  
}
```

bc() is called twice

Over 700 instructions
all like this

```
$instr_table[0x03]();
```

Need inline

I don't think there is one – at least not what I think of as inline

INLINE:: on CPAN gives you some very strange things

There is an Acme::INLINE::Perl

But that just gives more subroutine calls

Can create functions at
runtime though

Create a parser that takes something like

0x03, BC = INC BC

And generates a string

```
$instr[0x03] = sub {  
    $tmp = ((( $b << 8) + $c) + 1) & 0xffff;  
    $b = $tmp >> 8;  
    $c = $tmp & 0xff;  
}
```

Which we eval to execute

This is 239% faster! - But I haven't done it yet

What's left?

I thought I had finished!

Reduce number of subroutine calls

Instruction Compiler - Help!

<http://sourceforge.net/projects/perl-spectrum/>

<http://www.worldofspectrum.org>

Worth perservering