

Re: Painless micro program

Re: Painless micro program

Source: <http://sci.tech--archive.net/Archive/sci.electronics.design/2007-05/msg02929.html>

- *From:* krw <krw@xxxxxxxxxx>
 - *Date:* Tue, 22 May 2007 23:15:17 -0400
-

In article <czN4i.2144\$u56.1501@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, quiettechblue@xxxxxxxx says...

John Larkin wrote:

On Sun, 20 May 2007 19:22:13 -0700, joseph2k
<quiettechblue@xxxxxxxx>
wrote:

John Larkin wrote:

On 20 May 2007 07:42:49 -0700,
MooseFET <kensmith@xxxxxxxx> wrote:

On May 20, 7:03 am, krw
<k...@xxxxxxxx> wrote:

In article
<1179633027.463343.192...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
kensm...@xxxxxxxx
says...

On
May
19,
5:54
pm,
krw
<k...@xxxxxxxx>
wrote:

In
article
<1179616199.332790.162...@xxxxxxxxxxxxxxxxxxxxxxxxxx>

Re: Painless micro program

kensm...@xxxxxxxxxx
says...

[...
CDP1802
...]

Well,
the
8051,
at
least
in
its
original
incantation,
isn't
much
of
a
screamer
either
(12
clocks
per
op,
as
well).

There's
a
big
difference
between
12
cycles
of
a
12MHz
clock
and
12
cycles
of
a
2
MHZ
clock.
Besides:

Re: Painless micro program

Like I said,
there isn't
much of a
difference.

A
=
B
+
C

Compiles
to
something
like
16
instructions
if
A,
B
and
C
are
randomly
located
in
the
RAM
area.

The 8051 is
not so
different if
A, B, and C
are
randomly
located in
external
RAM. I'd
never use a
HLL on
such a
beast.

Re: Painless micro program

```
MOV DPTR,#B ; 1
MOVX A,@DPTR ; 2
MOV R5,A ; 3
MOV DPTR,#C ; 4
MOVX A,@DPTR ; 5
ADD A,R5 ; 6
MOV DPTR,#A ; 7
MOVX @DPTR,A ; 8
```

The 8051 took 8. Now lets see if I can remember the 1802 well enough:

```
LD #LOW(B) ; 1 Data =
low 8 of address
PLO R5 ; 2 Put to low 8
LD #HIGH(B) ; 3 Data =
high 8 of address
PHI R5 ; 4 Put to high 8
LD #LOW(C) ; 5
PLO R6 ; 6
LD #HIGH(C) ; 7
PHI R6 ; 8
LD #LOW(A) ; 9
PLO R7 ; 10
LD #HIGH(A) ; 11
PHI R7 ; 12
LDN R5 ; 13 Load what R5
points to
SEX R6 ; 14 Do math with
what R6 points to
ADD ; 15 Do the math
STD R7 ; 16 Store where
R7 points
```

The instruction count for the 1802 is double that of the 8051 and with the slower clock speed, the 1802 takes about a billion times longer to do the same operation.

Impressive! For a 6800-family gadget, it would be

```
LDAA A
```

Re: Painless micro program

ADDA B
STAA C

And on the 6803, using "LDAD..." (load double etc) would do it for word operands, using the A and B accumulators as a 16-bit register.

On a PDP-11, it would have been...

MOV.B A, C or just MOV for word data
ADD.B B, C

and the MSP430 is essentially the same, executing the sequence in about 400 ns. Both machines allow a choice of relative (relocatable) or absolute addressing. The rumor is that a few guys invented the PDP-11 architecture one night in Gordon Bell's basement.

On a 68K, it would be three ops, loadreg/add/store, but you'd have your choice of byte, word, or long.

There weren't many triple-operand machines around, ever.

John

I actually have worked with a machine that used up to 7 operands for a single instruction. IIRC the instruction was a repeated move selective replace using different source and destination indexes, different base address (with a special interpretation for one), a mask value, replace value, and iteration count.

Yikes!

Some of the CISC-ier machines, like HP 3000 and Vax and the nearly forgotten, failed 32-bit Intel chip (what was that called?) had outrageous instructions, like polynomial eval and hairy string ops, but that was just tons of microcode so doesn't really count, by my official decree. The 68332 even has a few, like the linear interpolation thing and hi/lo limit checks.

Re: Painless micro program

John

I think the old intel processor you are talking about the i432 "MicroFlame".

iAPX432.

IIRC it was a four or five chip group for the processor (kinda like a 2901 bit slice machine) and about a 11 by 14 board full of TTL logic (kinda like the old IBM PC or XT).

Three (monstrous) chips for the CPU, IIRC.

—

Keith

.