µCore

Progress Report

Klaus Schleisiek
kschleisiek at freenet.de

Bytes

I implemented IP/UDP and (R)ARP on µCore. It worked pretty efficiently, although µCore is a cell addressed processor.

Looking at the amount of code needed for a full IP protocol stack implementation I concluded that it would be more efficient to realize byte addressing in µCore rather then re-coding the entire protocol stack for cell addressing.

Because a byte addressed processor can re-use most of MPE's IP protocol stack.

Byte Adressing

Realizing a byte addressed µCore turned out to take much less time then I wasted during the past 20 years explaining why byte addressing is not needed at all.

An new VHDL constant byte_addr_width has been introduced. It may take the following values:

0 - Cell addressed, no bytes, data_width may take any value.

1 - Byte addressed 16 bit machine, data_width = 16.

2 - Byte addressed 32 bit machine, data_width = 32.

A byte addressed machine uses about 10% more logic resources.

Division / Multiplication

In the past I used fuzzy tests for the signed/unsigned division and multiplication instructions. But it always gave me an uneasy feeling.

An exhaustive test routine that would test all possible numbers dividing a double integer dividend by a single integer divisor. It compiled into 1020 instructions and therefore, it would be executible by a 10 bit machine.

This reduced the time needed for a full test to about 5 hours.

Test Routine

Basically, the test routine works as follows:

```
Dividend 2@ Divisor @ m/mod   Divisor @ m* rot s>d d+
```

If the result equals the dividend, we have a correct result and should have no overflow. The following four cases may occur:

1. Correct result, overflow not set - ok
2. Correct result, overflow set - error
3. Incorrect result, overflow set - ok
4. Incorrect result, overflow not set - error

Several case 4 errors popped up and I was able to debug the overflow generation code.

Links

microCore is available on git:

https://github.com/microCore-VHDL

and here is documentation:

https://github.com/microCore-VHDL/microCore/tree/master/documents