

Einbinden von in VHDL geschriebenen Komponenten in POL

Frederik Gröll

9. Juni 2008

Anforderungen an die Komponente

- Eingang `Strobe_In` vom Typ `std_logic`, der die Berechnung anstößt.
- Ausgang `Strobe_Out` vom Typ `std_logic`, der den Abschluss der Berechnung anzeigt.

Einbindung in POL

EBNF: component *component-name*((*vhdl-signal* \equiv *variable-name*)*)

Die VHDL-Datei wird über `import` in POL eingefügt und in `calc()` mit dem Schlüsselwort `component` verwendet. Daran schließen sich die VHDL-Signale an, jeweils gefolgt von “ \equiv ” und dem Variablennamen, mit dem die Signale verknüpft werden sollen. Ausgenommen sind die speziellen Signale `Strobe_In`, `Strobe_Out` und eventuell `CLK`, für die der Compiler selbst Code erzeugt. Für `CLK` wird der Systemtakt verwendet.

Beispiel

VHDL

Datei adder.vhd

```
COMPONENT adder
  PORT (a, b : IN std_logic_vector(15 DOWNTO 0);
        CLK, Strobe_In : IN std_logic;
        sum : OUT std_logic_vector(15 DOWNTO 0);
        carry : OUT std_logic;
        Strobe_Out : OUT std_logic
        );
END COMPONENT;

-- [...]
```

POL

```
import "adder.vhd"

class Adder extends ParObj {
  Slot a, b;
  Signal s;

  int in1, int in2;
  int result;
  bool carry;

  calc() {
    in1 = a.get();
    in2 = b.get();
    component adder(a => in1, b => in2,
                    sum => result, carry => carry);
    s.emit(result);
  }
}
```