```
Context Module
      inv : self.out.size() <= 1

Context VCO
      inv : in->size() <= 1 and
            in->collect(p : Port | p.name = « pitch_in »)->size() <= 1 and
            out->collect(p : Port | p.name = « sig_out »)->size() = 1
      inv : parameters->collect(cp : ContinuousParameter | cp.name = « pitch »)->size() = 1 and
            parameters->collect(dp : DiscreteParameter | dp.name = « shape »)->size() = 1

Context VCF
      inv : in->size() = 2 and
            in->collect(p : Port | p.name = « sig_in »)->size() = 1 and
            in->collect(p : Port | p.name = « cut_off »)->size() = 1 and
            out->collect(p : Port | p.name = « sig_out »)->size() = 1
      inv : parameters->collect(cp : ContinuousParameter | cp.name = « cut_off »)->size() = 1 and
            parameters->collect(cp : ContinuousParameter | cp.name = « resonance »)->size() = 1

Context VCA
      inv : 1 <= self.in->size() <= 2 and
            in->collect(p : Port | p.name = « sig_in »)->size() = 1 and
            in->collect(p : Port | p.name = « gain »)->size() = 1 and
            out->collect(p : Port | p.name = « sig_out »)->size() = 1
      inv : parameters->collect(cp : ContinuousParameter | cp.name = « gain »)->size() = 1 and

Context ADSR
      inv : in->size() = 1 and
            in->collect(p : Port | p.name = « sig_in »)->size() = 1 and
            out->collect(p : Port | p.name = « sig_out »)->size() = 1
      inv : parameters->collect(cp : ContinuousParameter | cp.name = « attack »)->size() = 1 and
            parameters->collect(cp : ContinuousParameter | cp.name = « decay »)->size() = 1 and
            parameters->collect(cp : ContinuousParameter | cp.name = « sustain »)->size() = 1 and
            parameters->collect(cp : ContinuousParameter | cp.name = « release »)->size() = 1

Context OUT
      inv : in->size() = 1 and
            in->collect(p : Port | p.name = « sig_in »)->size() = 1

Context MIXER
      inv : in->size() >= 2 and
            in->forAll(p : Port | parameters->exists(cp : ContinuousParameter | cp.name = p.name)) and
            out->collect(p : Port | p.name = « sig_out »)->size() = 1
```