Multiple GROUP declarations in ALF

Context: In an ALF timing specification, many-to-many timing arcs may be defined as in the following example, adapted from Fig. 28 of *Library Harmonization for Timing*, Draft v. 0.3 of 2004-02-23:

```
CELL ManyMany1
  {
   GROUP AddressBit { 0 : 2 }
   GROUP DataBit    { 1 : 4 }
   //
   PIN [2:0] Abus { DIRECTION = input;  }
   PIN [1:4] Dbus { DIRECTION = output; }
   //
   VECTOR ( 01 Abus[AddressBit] -> 01 Dbus[DataBit] )
     {
      DELAY = 1.0
        {
         FROM { PIN = Abus[AddressBit]; }
         TO   { PIN = Dbus[DataBit];    }
        }
     }
  }
```

The arcs in cell *ManyMany1* are assigned a rise delay of 1.0 as shown in Fig. 1, with coloring to clarify the internal "fanouts" affected.

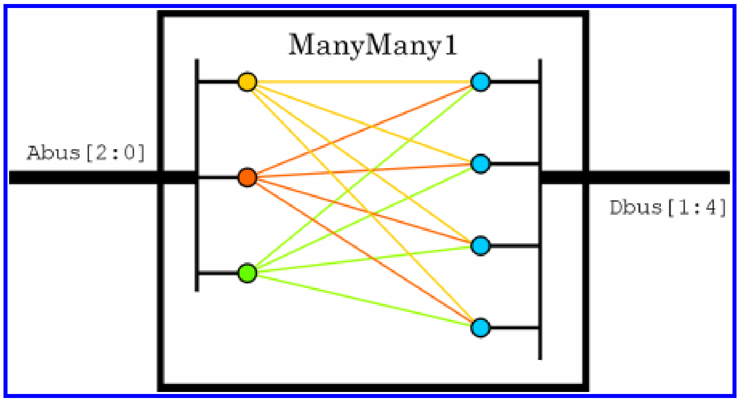

Figure 1. ALF many-to-many timing with two GROUPs declared. The arcs from different inputs are color-coded.

Problems:

A. What would be the use of three (or more) GROUP declarations in this context?

B. What might be the meaning, if any, of a "many-many-many" (many^3) timing specification?

Solution A: Three or more GROUPs merely mean that there are three or more busses in the cell to be iterated; no special relationship need be implied among the different delays. The example cell *ManyMany2* (possibly a sort of demultiplexor) shows the use of three GROUP declarations:

```
CELL ManyMany2
  {
   GROUP DataInBit     { 0 : 3 }
   GROUP DataOutBit1   { 1 : 4 }
   GROUP DataOutBit2   { 0 : 1 }
   //
   PIN [3:0] Din   { DIRECTION = input;  }
   PIN [1:4] Dout1 { DIRECTION = output; }
   PIN [1:0] Dout2 { DIRECTION = output; }
   //
   VECTOR ( 01 Din[DataInBit] -> 01 Dout1[DataOutBit1] )
     {
      DELAY = 2.0
        {
         FROM { PIN =   Din[DataInBit];   }
         TO   { PIN = Dout1[DataOutBit1]; }
        }
     }
   VECTOR ( 01 Din[DataInBit] -> 01 Dout2[DataOutBit2] )
     {
      DELAY = 1.0
        {
         FROM { PIN =   Din[DataInBit];   }
         TO   { PIN = Dout2[DataOutBit2]; }
        }
     }
  }
```

The arcs in cell *ManyMany2* are assigned a rise delay either of 1.0 or 2.0, as shown in Fig. 2.

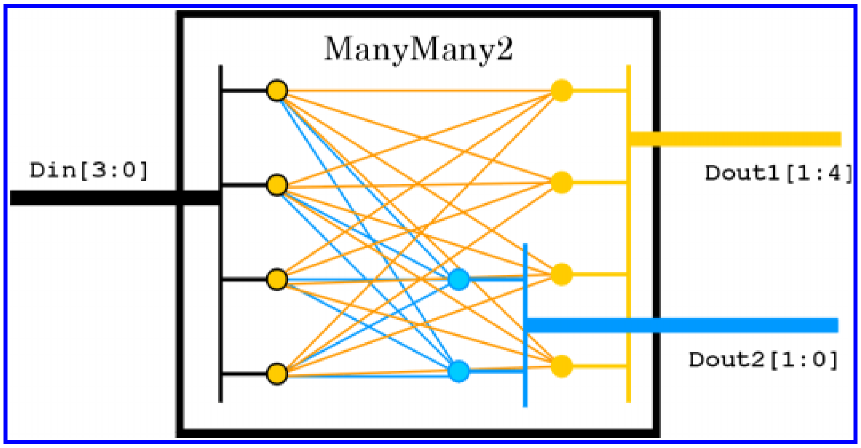

Figure 2. ALF many-to-many timing with three GROUPs declared. The two different delays are shown in different colors.

Solution B.1: A many^3 specification may be interpreted as use of three GROUP declarations in the context of a MATRIX pin. The cell *ManyMany3* shows an example of three GROUP declarations in this interpretation of a many^3 specification:

```
CELL ManyMany3
  {
   GROUP DataInBit     { 2 : 0 }
   GROUP DataOutRow1   { 2 : 1 }
   GROUP DataOutRow2   { 2 : 1 }
   //
   PIN [2:0] BitIn      { DIRECTION = input;  }
   PIN [2:1] MxOut[2:1] { DIRECTION = output; }
   //
   VECTOR ( 01 BitIn[DataInBit] -> 01 [DataOutRow1]MxOut[1] )
     {
      DELAY = 1.0
        {
         FROM { PIN = BitIn[DataInBit];      }
         TO   { PIN = [DataOutRow1]MxOut[1]; }
        }
     }
   VECTOR ( 01 BitIn[DataInBit] -> 01 [DataOutRow2]MxOut[2] )
     {
      DELAY = 2.0
        {
         FROM { PIN = BitIn[DataInBit];      }
         TO   { PIN = [DataOutRow2]MxOut[2]; }
        }
     }
  }
```

Fig. 3 shows the timing arcs in this case. The schematic arrangement of the output nodes is arbitrary.

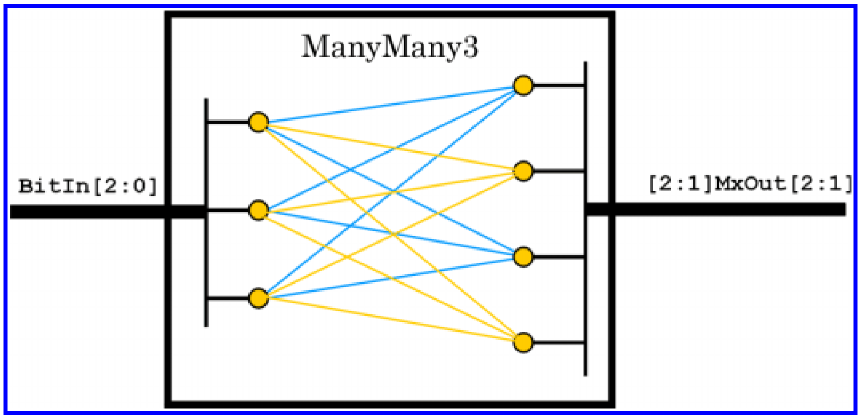

Figure 3. ALF many-to-many timing with three GROUPs and MATRIX pin declared. The two different delays are shown in different colors.

Solution B.2: A many^3 specification may be interpreted as use of three GROUP declarations in the context of a bidirectional pin. The cell

ManyMany4 shows an example of three GROUP declarations in this interpretation of a many^3 specification:

```
CELL ManyMany4
  {
   GROUP InBit    { 0 : 2 }
   GROUP BidirBit { 1 : 3 }
   GROUP OutBit   { 1 : 3 }
   //
   PIN [2:0] Din    { DIRECTION = input;  }
   PIN [1:3] Dbidir { DIRECTION = both;   }
   PIN [3:1] Dout   { DIRECTION = output; }
   //
   VECTOR ( 01 Din[InBit] -> 01 Dbidir[BidirBit] )
     {
      DELAY = 1.0
        {
         FROM { PIN =    Din[InBit];    }
         TO   { PIN = Dbidir[BidirBit]; }
        }
     }
   VECTOR ( 01 Dbidir[BidirBit] -> 01 Dout[OutBit] )
     {
      DELAY = 2.0
        {
         FROM { PIN = Dbidir[BidirBit]; }
         TO   { PIN =   Dout[OutBit];   }
        }
     }
  }
```

Fig. 4 shows the timing arcs in this case. The input and output pins imply the timing directions.

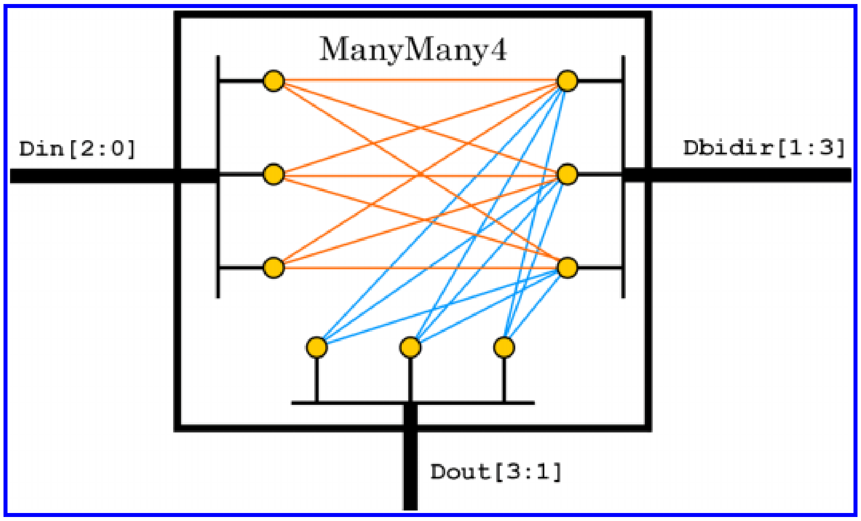

Figure 4. ALF many-to-many timing with three GROUPs and bidirectional intermediate pin declared. The two different delays are shown in different colors.