sscnnFigure01

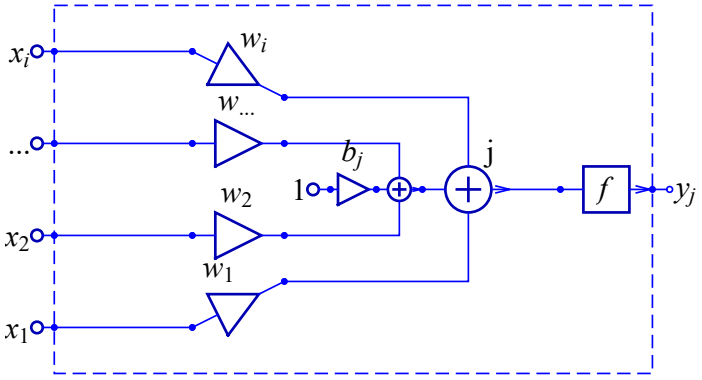
In[ ]:=
```
SmainDir = SetDirectory[NotebookDirectory[]];
Get["SchematicSolver`"]
<< SSknowledgeImp00`
```

In[ ]:=
```
SetOptions[DrawElement, {BaseStyle → {FontFamily → "Times", FontSize → 16}}];
plotStyleDD = {{Dashing[.01], GrayLevel[.5]}, {Dashing[.01], GrayLevel[.5]}};
SetOptions[DrawElement, {ElementSize → {1, 1},
   PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.01]},
     {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.01]}},
   ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
   BaseStyle → {FontFamily → "Times", FontSize → 16}}];
```

In[ ]:=
```
artNeuronInp04 = {
   {"Input", {0, 0}, XX1, " ", TextOffset → {1, 0}},
   {"Input", {0, 2}, XX2, " ", TextOffset → {1, 0}}, {"Input", {0, 4}, XX3, " ",
    TextOffset → {1, 0}}, {"Input", {0, 6}, XX4, " ", TextOffset → {1, 0}}};
artNeuronOut04 = {
   {"Output", {13, 3}, YY1, " ", TextOffset → {-1, 0}}};
artNeuronBody04 = {
   {"Adder", {{7, 3}, {5, 2}, {8, 3}, {5, 4}}, {1, 1, 2, 1}, " "},
   {"Adder", {{8, 3}, {5, 1}, {11, 3}, {5, 5}}, {1, 1, 2, 1}, "j"},
   {"Function", {{11, 3}, {13, 3}}, f, " "},
   {"Input", {6, 3}, 1, " ", TextOffset → {1, 0}},
   {"Line", {{0, 0}, {3, 0}}}, {"Line", {{0, 2}, {3, 2}}},
   {"Line", {{0, 4}, {3, 4}}}, {"Line", {{0, 6}, {3, 6}}},
   {"Multiplier", {{3, 0}, {5, 1}}, W11, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 2}, {5, 2}}, W21, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 4}, {5, 4}}, W31, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 6}, {5, 5}}, W41, " ", ElementSize → {1, 1}},
   {"Multiplier", {{6, 3}, {7, 3}}, b01, " "},
   {"Polyline", {{0, -1}, {13, -1}, {13, 7}, {0, 7}, {0, -1}}}};
```

```
In[•]:= subsSymbW = {W11 → w1, W21 → w2, W31 → w"..", W41 → wi};
        subsSymbB = {b01 → bj};
        subsSymbX = {XX1 → x1, XX2 → x2, XX3 → "...", XX4 → xi, YY1 → yj};
        mySchematic = artNeuronInp04 ~ Join ~ artNeuronBody04 ~ Join ~ artNeuronOut04;
        figure001 = Show[
          {DrawElement /@ (mySchematic /. subsSymbX /. subsSymbB /. subsSymbW)}, Frame → False]
```

Out[•]=



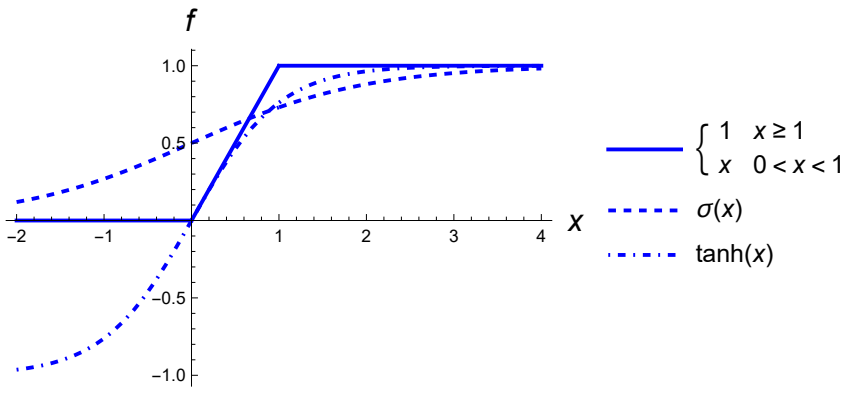

```
In[•]:= (* Export["figure001.tif",figure001,ImageResolution→300] *)
```

sscnnFigure02

```
In[•]:= mainDir = SetDirectory[NotebookDirectory[]];
```

```
In[•]:= Clear[b, softmax, W, x, yRamp]
        Piecewise[{{Piecewise[{{x, x < 1}}, 1], x > 0}}, 0];
        yRamp = PiecewiseExpand[%];
        figure002 = Plot[{yRamp, LogisticSigmoid[x], Tanh[x]}, {x, -2, 4},
          ImageSize → {Automatic, 200},
          PlotStyle → {{Thick, Blue}, {Thick, Dashed, Blue}, {Thick, DotDashed, Blue},
            {Thick, DotDashed, Orange}}, PlotLegends → {PiecewiseExpand[
              Piecewise[{{Piecewise[{{x, x < 1}}, 1], x > 0}}, 0]] // TraditionalForm,
            LogisticSigmoid[x] // TraditionalForm, Tanh[x] // TraditionalForm},
          AxesLabel → {Style["x", Italic, FontFamily → "Arial", FontSize → 16],
            Style["f", Italic, FontFamily → "Arial", FontSize → 16]}]
```

Out[•]=




```
In[•]:= (* Export["figure002.tif",figure002,ImageResolution→300] *)
```

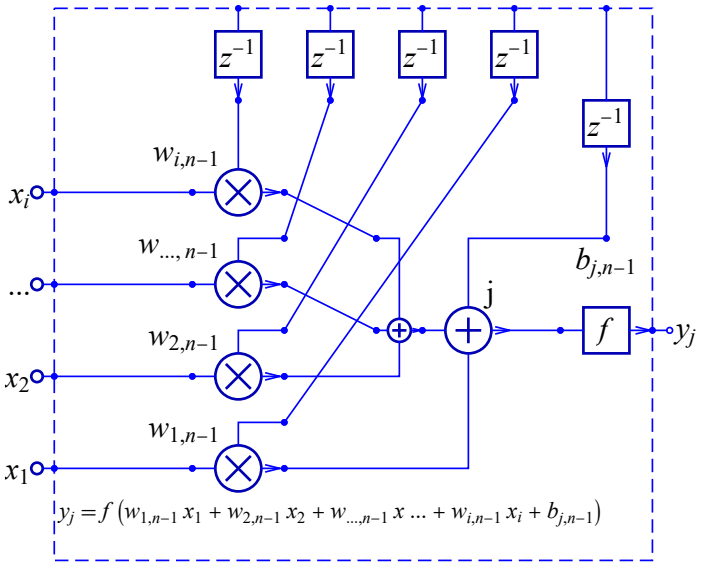
sscnnFigure03

```
In[•]:= SmainDir = SetDirectory[NotebookDirectory[]];
        Get["SchematicSolver`"]
        << SSknowledgeImp00`

In[•]:= SetOptions[DrawElement, {BaseStyle → {FontFamily → "Times", FontSize → 16}}];
        plotStyleDD = {{Dashing[.01], GrayLevel[.5]}, {Dashing[.01], GrayLevel[.5]}};
        SetOptions[DrawElement, {ElementSize → {1, 1},
            PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.01]},
              {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.01]}},
            ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
            BaseStyle → {FontFamily → "Times", FontSize → 16}}];

In[•]:= artNeuronInp04 = {
            {"Input", {0, 0}, XX1, " ", TextOffset → {1, 0}},
            {"Input", {0, 2}, XX2, " ", TextOffset → {1, 0}}, {"Input", {0, 4}, XX3, " ",
             TextOffset → {1, 0}}, {"Input", {0, 6}, XX4, " ", TextOffset → {1, 0}}};
        artNeuronOut04 = {
            {"Output", {13, 3}, YY1, " ", TextOffset → {-1, 0}}};
        artNeuronBody04 = {
            {"Adder", {{7, 3}, {5, 2}, {8, 3}, {7, 5}}, {1, 1, 2, 1}, " "},
            {"Adder", {{8, 3}, {5, 0}, {11, 3}, {12, 5}}, {1, 1, 2, 1}, "j"},
            {"Function", {{11, 3}, {13, 3}}, f, " "},
            {"Line", {{0, 0}, {3, 0}}},
            {"Line", {{0, 2}, {3, 2}}},
            {"Line", {{0, 4}, {3, 4}}},
            {"Line", {{0, 6}, {3, 6}}},
            {"Line", {{5, 6}, {7, 5}}},

            {"Line", {{5, 5}, {6, 8}}},
            {"Line", {{5, 4}, {7, 3}}},

            {"Line", {{5, 3}, {8, 8}}},
            {"Line", {{5, 1}, {10, 8}}},
            {"Modulator", {{3, 0}, {4, -1}, {5, 0}, {5, 1}},
             {1, 0, 2, 1}, W11, TextOffset → {1, -1}},
            {"Modulator", {{3, 2}, {4, -1}, {5, 2}, {5, 3}},
             {1, 0, 2, 1}, W21, TextOffset → {1, -1}},
            {"Modulator", {{3, 4}, {4, -1}, {5, 4}, {5, 5}},
             {1, 0, 2, 1}, W31, TextOffset → {1, -1}},
            {"Modulator", {{3, 6}, {4, 5}, {5, 6}, {4, 8}},
             {1, 0, 2, 1}, W41, TextOffset → {1, -1}},
            {"Polyline", {{0, -2}, {13, -2}, {13, 10}, {0, 10}, {0, -2}}}};
```

In[•]:=
```
artNeuronDelay04 = {
    {"Text", {6, -1}, "y_j = f (w_{1,n-1} x_1 + w_{2,n-1} x_2 + w_{...,n-1} x ... + w_{i,n-1} x_i + b_{j,n-1})",
     BaseStyle → {FontFamily → "Times", FontSize → 12}},
    {"Text", {12, 4.5}, b_{j,n-1}},
    {"Delay", {{4, 10}, {4, 8}}, 1, " "},
    {"Delay", {{6, 10}, {6, 8}}, 1, " "},
    {"Delay", {{8, 10}, {8, 8}}, 1, " "},
    {"Delay", {{10, 10}, {10, 8}}, 1, " "},
    {"Delay", {{12, 10}, {12, 5}}, 1, " "}};
```

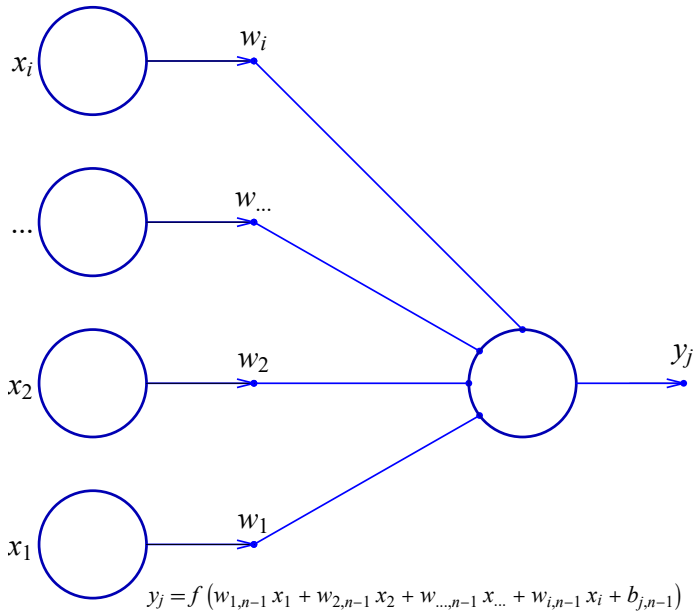
In[•]:=
```
subsSymbW = {W11 → w_{1,n-1}, W21 → w_{2,n-1}, W31 → w_{"...,"n-1}, W41 → w_{i,n-1}};
subsSymbB = {b01 → b_{j,n-1}};
subsSymbX = {XX1 → x_1, XX2 → x_2, XX3 → "...", XX4 → x_i, YY1 → y_j};
mySchematic =
  artNeuronInp04 ~ Join ~ artNeuronBody04 ~ Join ~ artNeuronOut04 ~ Join ~ artNeuronDelay04;
figure003a = Show[
  {DrawElement /@ (mySchematic /. subsSymbX /. subsSymbB /. subsSymbW)}, Frame → False]
```

Out[•]=



In[•]:=
```
artNeuronLin004 = {
    {"Input", {2, 12}, XX4, "", ElementSize → 8, TextOffset → {1, 0}},
    {"Input", {2, 9}, XX3, "", ElementSize → 8, TextOffset → {1, 0}},
    {"Input", {2, 6}, XX2, "", ElementSize → 8, TextOffset → {1, 0}},
    {"Input", {2, 3}, XX1, "", ElementSize → 8, TextOffset → {1, 0}},
    {"Line", {{2, 12}, {7, 7}}},
    {"Line", {{2, 6}, {6, 6}}},
    {"Line", {{2, 9}, {6.2, 6.6}}},
    {"Line", {{2, 3}, {6.2, 5.4}}},
    {"Text", {5, 2}, "yj = f (w1,n-1 x1 + w2,n-1 x2 + w...,n-1 x... + wi,n-1 xi + bj,n-1)",
     BaseStyle → {FontFamily → "Times", FontSize → 12}},
    {"Text", {10, 6.6}, "yj"},
    {"Text", {2, 12.5}, "wi"},
    {"Text", {2, 9.5}, "w..."},
    {"Text", {2, 6.5}, "w2"},
    {"Text", {2, 3.5}, "w1"},
    {"Arrow", {{2, 12}, {0, 12}}},
    {"Arrow", {{2, 9}, {0, 9}}},
    {"Arrow", {{2, 6}, {0, 6}}},
    {"Arrow", {{2, 3}, {0, 3}}},
    {"Arrow", {{10, 6}, {9, 6}}},
    {"Input", {10, 6}, "", "", ElementSize → 8, TextOffset → {1, 0}}};
figure003b =
 Show[{DrawElement /@ (artNeuronLin004 /. subsSymbX /. subsSymbB /. subsSymbW)},
  Frame → False]
```

Out[•]=



In[•]:=
```
(*
Export["figure003a.tif",figure003a,ImageResolution→300];
Export["figure003b.tif",figure003b,ImageResolution→300];
 *)
```

sscnnFigure04 and sscnnFigure05

```
In[•]:= SmainDir = SetDirectory[NotebookDirectory[]];
        Get["SchematicSolver`"]
        << SSknowledgeImp00`

In[•]:= subsSymb1 = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
        subsSymb2 =
          {W11 → w11, W12 → w12, W13 → w13, W14 → w14, W21 → w21, W22 → w22, W23 → w23, W24 → w24,
           W31 → w31, W32 → w32, W33 → w33, W34 → w34, W41 → w41, W42 → w42, W43 → w43, W44 → w44};
        subsSymb3 = {b01 → b01, b02 → b02, b03 → b03, b04 → b04, b11 → b11,
           b12 → b12, b13 → b13, b14 → b14, b21 → b21, b22 → b22, b23 → b23, b24 → b24};
        SetOptions[DrawElement, {BaseStyle → {FontFamily → "Times", FontSize → 8}}];
        SetOptions[DrawElement, {ElementSize → {1, 1}}];
        plotStyleDD = {{Dashing[.01], GrayLevel[.5]}, {Dashing[.01], GrayLevel[.5]}};
        SetOptions[DrawElement, {ElementSize → {1, 1},
           PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
             {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
           ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
           BaseStyle → {FontFamily → "Times", FontSize → 8}}];

In[•]:= artNeuronInp04 = {
           {"Input", {0, 3}, XX1, " ", TextOffset → {1, 0}},
           {"Input", {0, 10}, XX2, " ", TextOffset → {1, 0}}, {"Input", {0, 17}, XX3, " ",
            TextOffset → {1, 0}}, {"Input", {0, 24}, XX4, " ", TextOffset → {1, 0}}};
        Show[{DrawElement /@ (artNeuronInp04 /. subsSymb1 /. subsSymb2)}, Frame → False];
        artNeuronOut04 = {
           {"Output", {13, 3}, YY1, " ", TextOffset → {-1, 0}},
           {"Output", {13, 10}, YY2, " ", TextOffset → {-1, 0}},
           {"Output", {13, 17}, YY3, " ", TextOffset → {-1, 0}},
           {"Output", {13, 24}, YY4, " ", TextOffset → {-1, 0}}};
        artNeuronBody04 = {
           {"Adder", {{7, 3}, {5, 2}, {8, 3}, {5, 4}}, {1, 1, 2, 1}, " "},
           {"Adder", {{8, 3}, {5, 1}, {11, 3}, {5, 5}}, {1, 1, 2, 1}, "1"},
           {"Adder", {{7, 10}, {5, 9}, {8, 10}, {5, 11}}, {1, 1, 2, 1}, " "},
           {"Adder", {{8, 10}, {5, 8}, {11, 10}, {5, 12}}, {1, 1, 2, 1}, "2"},
           {"Adder", {{7, 17}, {5, 16}, {8, 17}, {5, 18}}, {1, 1, 2, 1}, " "},
           {"Adder", {{8, 17}, {5, 15}, {11, 17}, {5, 19}}, {1, 1, 2, 1}, "3"},
           {"Adder", {{7, 24}, {5, 23}, {8, 24}, {5, 25}}, {1, 1, 2, 1}, " "},
           {"Adder", {{8, 24}, {5, 22}, {11, 24}, {5, 26}}, {1, 1, 2, 1}, "4"},
           {"Function", {{11, 24}, {13, 24}}, f, " "}, {"Function", {{11, 17}, {13, 17}}, f, " "},
           {"Function", {{11, 10}, {13, 10}}, f, " "}, {"Function", {{11, 3}, {13, 3}}, f, " "},
           {"Input", {6, 24}, 1, " ", TextOffset → {1, 0}},
           {"Input", {6, 17}, 1, " ", TextOffset → {1, 0}},
           {"Input", {6, 10}, 1, " ", TextOffset → {1, 0}},
           {"Input", {6, 3}, 1, " ", TextOffset → {1, 0}},
           {"Line", {{0, 3}, {3, 0}}}, {"Line", {{0, 10}, {3, 2}}},
           {"Line", {{0, 3}, {3, 7}}}, {"Line", {{0, 3}, {3, 14}}}, {"Line", {{0, 3}, {3, 21}}},
           {"Line", {{0, 17}, {3, 4}}}, {"Line", {{0, 10}, {3, 9}}}, {"Line", {{0, 10}, {3, 16}}},
           {"Line", {{0, 10}, {3, 23}}}, {"Line", {{0, 17}, {3, 11}}},
```

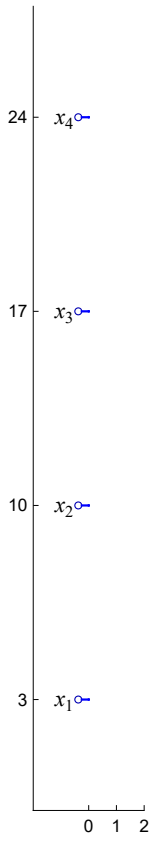
```
    {"Line", {{0, 17}, {3, 18}}}, {"Line", {{0, 17}, {3, 25}}},
    {"Line", {{0, 24}, {3, 27}}}, {"Line", {{0, 24}, {3, 20}}},
    {"Line", {{0, 24}, {3, 13}}}, {"Line", {{0, 24}, {3, 6}}},
    {"Multiplier", {{3, 0}, {5, 1}}, W11, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 2}, {5, 2}}, W21, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 4}, {5, 4}}, W31, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 6}, {5, 5}}, W41, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 7}, {5, 8}}, W12, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 14}, {5, 15}}, W13, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 21}, {5, 22}}, W14, " ", ElementSize → {1, 1}},
    {"Multiplier", {{3, 9}, {5, 9}}, W22, " "}, {"Multiplier",
     {{3, 11}, {5, 11}}, W32, " "}, {"Multiplier", {{3, 16}, {5, 16}}, W23, " "},
    {"Multiplier", {{3, 18}, {5, 18}}, W33, " "}, {"Multiplier", {{3, 13}, {5, 12}},
     W42, " "}, {"Multiplier", {{3, 20}, {5, 19}}, W43, " "}, {"Multiplier",
     {{3, 23}, {5, 23}}, W24, " "}, {"Multiplier", {{3, 25}, {5, 25}}, W34, " "},
    {"Multiplier", {{3, 27}, {5, 26}}, W44, " "}, {"Multiplier", {{6, 3}, {7, 3}},
     b01, " "}, {"Multiplier", {{6, 10}, {7, 10}}, b02, " "},
    {"Multiplier", {{6, 17}, {7, 17}}, b03, " "}, {"Multiplier", {{6, 24}, {7, 24}},
     b04, " "}, {"Polyline", {{4.7`, 0}, {10, 0}, {10, 6}, {4.7`, 6}, {4.7`, 0}}},
    {"Polyline", {{4.7`, 7}, {10, 7}, {10, 13}, {4.7`, 13}, {4.7`, 7}}},
    {"Polyline", {{4.7`, 14}, {10, 14}, {10, 20}, {4.7`, 20}, {4.7`, 14}}},
    {"Polyline", {{4.7`, 21}, {10, 21}, {10, 27}, {4.7`, 27}, {4.7`, 21}}}};
```

In[•]:=

```
subsSymbW0 = {W011 → w1,1, W021 → w1,2, W031 → w1,3, W041 → w1,4,
    W012 → w1,5, W022 → w1,6, W032 → w1,7, W042 → w1,8, W013 → w1,9, W023 → w1,10,
    W033 → w1,11, W043 → w1,12, W014 → w1,13, W024 → w1,14, W034 → w1,15, W044 → w1,16};
subsSymbW1 = {W111 → w2,1, W121 → w2,2, W131 → w2,3, W141 → w2,4,
    W112 → w2,5, W122 → w2,6, W132 → w2,7, W142 → w2,8, W113 → w2,9, W123 → w2,10,
    W133 → w2,11, W143 → w2,12, W114 → w2,13, W124 → w2,14, W134 → w2,15, W144 → w2,16};
subsSymbW2 = {W211 → w3,1, W221 → w3,2, W231 → w3,3, W241 → w3,4,
    W212 → w3,5, W222 → w3,6, W232 → w3,7, W242 → w3,8, W213 → w3,9, W223 → w3,10,
    W233 → w3,11, W243 → w3,12, W214 → w3,13, W224 → w3,14, W234 → w3,15, W244 → w3,16};
subsSymbB = {b01 → b1,1, b02 → b1,2, b03 → b1,3, b04 → b1,4, b11 → b2,1,
    b12 → b2,2, b13 → b2,3, b14 → b2,4, b21 → b3,1, b22 → b3,2, b23 → b3,3, b24 → b3,4};
subsSymbX = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
subsALL1 = (subsSymb1 ~ Join ~ subsSymbW0 ~ Join ~ subsSymbW2 ~
     Join ~ subsSymbW1 ~ Join ~ subsSymbW2 ~ Join ~ subsSymbB ~ Join ~ subsSymbX);
```

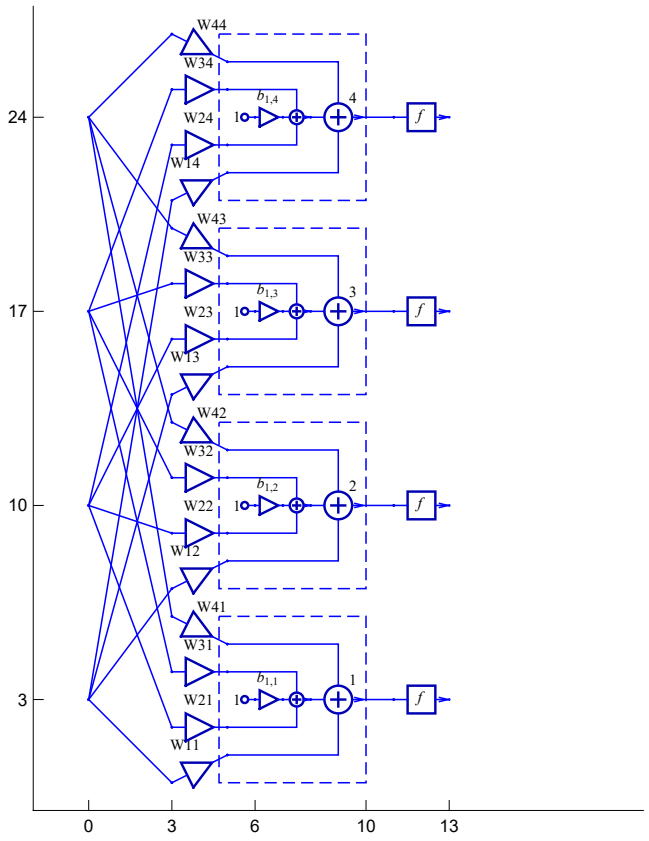
In[•]:=
```
SetOptions[DrawElement, {ElementSize → {1, 1},
    PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.01], PointSize[0.02]},
      {RGBColor[0, 0, 1], Thickness[0.02], PointSize[0.007]}},
    ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
    BaseStyle → {FontFamily → "Times", FontSize → 12}}];
figure004a = Show[{DrawElement /@ (artNeuronInp04 /. subsALL1)},
  Frame → {{True, False}, {True, False}},
  FrameTicks → {{0, 1, 2}, {3, 10, 17, 24}}, PlotRange → {{-2, 2}, {-1, 28}}]
```

Out[•]=



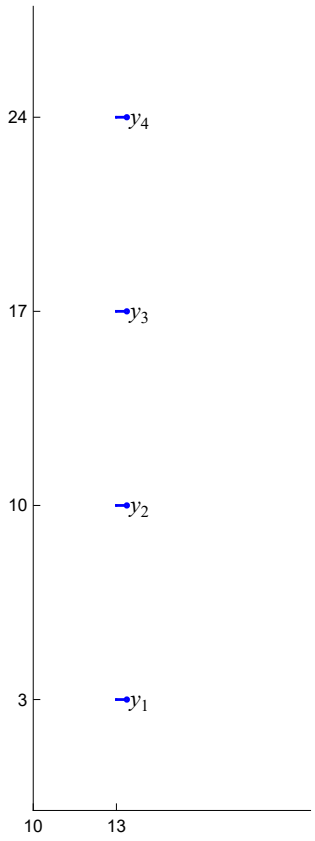
In[•]:=
```
SetOptions[DrawElement, {ElementSize → {1, 1},
    PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
      {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
    ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
    BaseStyle → {FontFamily → "Times", FontSize → 8}}];
figure004b = Show[{DrawElement /@ (artNeuronBody04 /. subsALL1)},
  Frame → {{True, False}, {True, False}},
  FrameTicks → {{0, 3, 6, 10, 13}, {3, 10, 17, 24}}, PlotRange → {{-2, 20}, {-1, 28}}]
```

Out[•]=



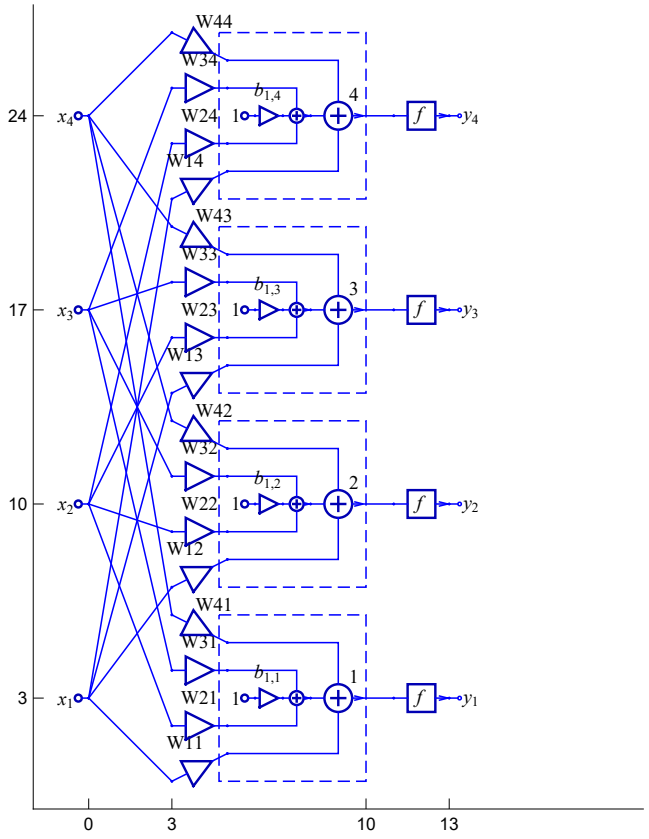
In[•]:=
```
SetOptions[DrawElement, {ElementSize → {1, 1},
    PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.01], PointSize[0.01]},
      {RGBColor[0, 0, 1], Thickness[0.01], PointSize[0.007]}},
    ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
    BaseStyle → {FontFamily → "Times", FontSize → 12}}];
figure004c = Show[{DrawElement /@ (artNeuronOut04 /. subsALL1)},
  Frame → {{True, False}, {True, False}},
  FrameTicks → {{10, 13}, {3, 10, 17, 24}}, PlotRange → {{10, 20}, {-1, 28}}]
```

Out[•]=



```
In[•]:= SetOptions[DrawElement, {ElementSize → {1, 1},
          PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
            {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
          ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
          BaseStyle → {FontFamily → "Times", FontSize → 10}}];
        mySchematic = artNeuronInp04 ~ Join ~ artNeuronOut04 ~ Join ~ artNeuronBody04;
        figure004d =
         Show[{DrawElement /@ (mySchematic /. subsALL1)}, Frame → {{True, False}, {True, False}},
          FrameTicks → {{0, 3, 10, 13}, {3, 10, 17, 24}}, PlotRange → {{-2, 20}, {-1, 28}}]
```

Out[•]=




```
(*
Export["figure004a.tif",figure004a,ImageResolution→300];
Export["figure004b.tif",figure004b,ImageResolution→300];
Export["figure004c.tif",figure004c,ImageResolution→300];
Export["figure004d.tif",figure004d,ImageResolution→300];
*)
```

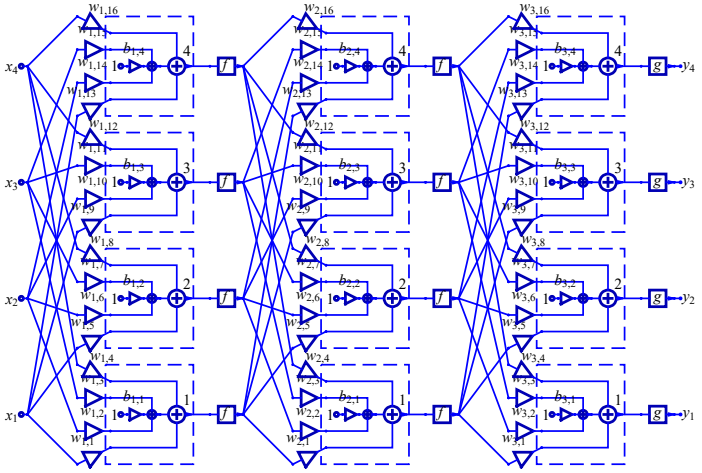
Figure 5

```
In[•]:= mySchematic = artNeuronInp04 ~ Join ~ artNeuronOut04 ~ Join ~ artNeuronBody04;
        SetOptions[DrawElement, {ElementSize → {1, 1},
          PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.002], PointSize[0.0012]},
            {RGBColor[0, 0, 1], Thickness[0.0012`], PointSize[0.0014]}},
          ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
          BaseStyle → {FontFamily → "Times", FontSize → 8}}];
```

```
In[•]:= numberOfStages = 3;
widthOfStages = 13;
systemBody = artNeuronBody04;
b1TOb1 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b01, b02, b03, b04}]];
b1TOb2 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b11, b12, b13, b14}]];
b1TOb3 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b21, b22, b23, b24}]];
b1TOb4 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b31, b32, b33, b34}]];
w1TOw1 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W011, W012, W013, W014, W021, W022, W023, W024,
     W031, W032, W033, W034, W041, W042, W043, W044}];
w1TOw2 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W111, W112, W113, W114, W121, W122, W123, W124,
     W131, W132, W133, W134, W141, W142, W143, W144}];
w1TOw3 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W211, W212, W213, W214, W221, W222, W223, W224,
     W231, W232, W233, W234, W241, W242, W243, W244}];
w1TOw4 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W311, W312, W313, W314, W321, W322, W323, W324,
     W331, W332, W333, W334, W341, W342, W343, W344}];
B1TObK = {b1TOb2, b1TOb3, b1TOb4};
W1TOWK = {w1TOw2, w1TOw3, w1TOw4};
f1TOfK = {f → f, f → f, f → g};
f1TOfK = {f → f, f → g, f → G};
Do[systemBody =
    Join[systemBody, (TranslateSchematic[artNeuronBody04, {k * widthOfStages, 0}] /.
        B1TObK[[k]] /. W1TOWK[[k]] /. f1TOfK[[k]])];
  , {k, numberOfStages - 1}];

In[•]:= systemInBodyOut = artNeuronInp04 ~ Join ~ systemBody ~ Join ~
   (TranslateSchematic[artNeuronOut04, {(numberOfStages - 1) * widthOfStages, 0}]);
```

```
In[•]:= mySchematic = systemInBodyOut;
        subsALL1 = (subsSymb1 ~Join~ subsSymbW0 ~Join~ subsSymbW2 ~
            Join~ subsSymbW1 ~Join~ subsSymbW2 ~Join~ subsSymbB ~Join~ subsSymbX);
        SetOptions[DrawElement, {ElementSize → {1, 1},
            PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
              {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
            ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
            BaseStyle → {FontFamily → "Times", FontSize → 8}}];
        figure005 = Show[{DrawElement /@ (mySchematic /. w1TOw1 /. subsALL1)}, Frame → False]
```

Out[•]=



```
        (* Export["figure005.tif",figure005,ImageResolution→300] *)
```

sscnnFigure06

```
In[•]:= SmainDir = SetDirectory[NotebookDirectory[]];
        Get["SchematicSolver`"]
        << SSknowledgeImp00`
```

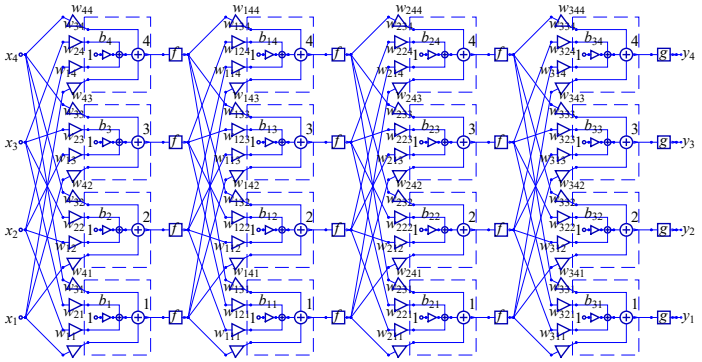
```
In[•]:= subsSymb1 = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
        subsSymb2 =
          {W11 → w11, W12 → w12, W13 → w13, W14 → w14, W21 → w21, W22 → w22, W23 → w23, W24 → w24,
           W31 → w31, W32 → w32, W33 → w33, W34 → w34, W41 → w41, W42 → w42, W43 → w43, W44 → w44};
        subsSymb3 = {b01 → b01, b02 → b02, b03 → b03, b04 → b04, b11 → b11,
            b12 → b12, b13 → b13, b14 → b14, b21 → b21, b22 → b22, b23 → b23, b24 → b24};
        SetOptions[DrawElement, {BaseStyle → {FontFamily → "Times", FontSize → 8}}];
        SetOptions[DrawElement, {ElementSize → {1, 1}}];
        plotStyleDD = {{Dashing[.01], GrayLevel[.5]}, {Dashing[.01], GrayLevel[.5]}};
        SetOptions[DrawElement, {ElementSize → {1, 1},
            PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
              {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
            ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
            BaseStyle → {FontFamily → "Times", FontSize → 8}}];
```

```
In[•]:= artNeuronInp04 = {
            {"Input", {0, 3}, XX1, " ", TextOffset → {1, 0}},
            {"Input", {0, 10}, XX2, " ", TextOffset → {1, 0}}, {"Input", {0, 17}, XX3, " ",
```

```
    TextOffset → {1, 0}}, {"Input", {0, 24}, XX4, " ", TextOffset → {1, 0}}};
Show[{DrawElement /@ (artNeuronInp04 /. subsSymb1 /. subsSymb2)}, Frame → False];
artNeuronOut04 = {
   {"Output", {13, 3}, YY1, " ", TextOffset → {-1, 0}},
   {"Output", {13, 10}, YY2, " ", TextOffset → {-1, 0}},
   {"Output", {13, 17}, YY3, " ", TextOffset → {-1, 0}},
   {"Output", {13, 24}, YY4, " ", TextOffset → {-1, 0}}};
artNeuronBody04 = {
   {"Adder", {{7, 3}, {5, 2}, {8, 3}, {5, 4}}, {1, 1, 2, 1}, " "},
   {"Adder", {{8, 3}, {5, 1}, {11, 3}, {5, 5}}, {1, 1, 2, 1}, "1"},
   {"Adder", {{7, 10}, {5, 9}, {8, 10}, {5, 11}}, {1, 1, 2, 1}, " "},
   {"Adder", {{8, 10}, {5, 8}, {11, 10}, {5, 12}}, {1, 1, 2, 1}, "2"},
   {"Adder", {{7, 17}, {5, 16}, {8, 17}, {5, 18}}, {1, 1, 2, 1}, " "},
   {"Adder", {{8, 17}, {5, 15}, {11, 17}, {5, 19}}, {1, 1, 2, 1}, "3"},
   {"Adder", {{7, 24}, {5, 23}, {8, 24}, {5, 25}}, {1, 1, 2, 1}, " "},
   {"Adder", {{8, 24}, {5, 22}, {11, 24}, {5, 26}}, {1, 1, 2, 1}, "4"},
   {"Function", {{11, 24}, {13, 24}}, f, " "}, {"Function", {{11, 17}, {13, 17}}, f, " "},
   {"Function", {{11, 10}, {13, 10}}, f, " "}, {"Function", {{11, 3}, {13, 3}}, f, " "},
   {"Input", {6, 24}, 1, " ", TextOffset → {1, 0}},
   {"Input", {6, 17}, 1, " ", TextOffset → {1, 0}},
   {"Input", {6, 10}, 1, " ", TextOffset → {1, 0}},
   {"Input", {6, 3}, 1, " ", TextOffset → {1, 0}},
   {"Line", {{0, 3}, {3, 0}}}, {"Line", {{0, 10}, {3, 2}}},
   {"Line", {{0, 3}, {3, 7}}}, {"Line", {{0, 3}, {3, 14}}}, {"Line", {{0, 3}, {3, 21}}},
   {"Line", {{0, 17}, {3, 4}}}, {"Line", {{0, 10}, {3, 9}}}, {"Line", {{0, 10}, {3, 16}}},
   {"Line", {{0, 10}, {3, 23}}}, {"Line", {{0, 17}, {3, 11}}},
   {"Line", {{0, 17}, {3, 18}}}, {"Line", {{0, 17}, {3, 25}}},
   {"Line", {{0, 24}, {3, 27}}}, {"Line", {{0, 24}, {3, 20}}},
   {"Line", {{0, 24}, {3, 13}}}, {"Line", {{0, 24}, {3, 6}}},
   {"Multiplier", {{3, 0}, {5, 1}}, W11, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 2}, {5, 2}}, W21, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 4}, {5, 4}}, W31, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 6}, {5, 5}}, W41, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 7}, {5, 8}}, W12, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 14}, {5, 15}}, W13, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 21}, {5, 22}}, W14, " ", ElementSize → {1, 1}},
   {"Multiplier", {{3, 9}, {5, 9}}, W22, " "}, {"Multiplier",
    {{3, 11}, {5, 11}}, W32, " "}, {"Multiplier", {{3, 16}, {5, 16}}, W23, " "},
   {"Multiplier", {{3, 18}, {5, 18}}, W33, " "}, {"Multiplier", {{3, 13}, {5, 12}},
    W42, " "}, {"Multiplier", {{3, 20}, {5, 19}}, W43, " "}, {"Multiplier",
    {{3, 23}, {5, 23}}, W24, " "}, {"Multiplier", {{3, 25}, {5, 25}}, W34, " "},
   {"Multiplier", {{3, 27}, {5, 26}}, W44, " "}, {"Multiplier", {{6, 3}, {7, 3}},
    b01, " "}, {"Multiplier", {{6, 10}, {7, 10}}, b02, " "},
   {"Multiplier", {{6, 17}, {7, 17}}, b03, " "}, {"Multiplier", {{6, 24}, {7, 24}},
    b04, " "}, {"Polyline", {{4.7`, 0}, {10, 0}, {10, 6}, {4.7`, 6}, {4.7`, 0}}},
   {"Polyline", {{4.7`, 7}, {10, 7}, {10, 13}, {4.7`, 13}, {4.7`, 7}}},
   {"Polyline", {{4.7`, 14}, {10, 14}, {10, 20}, {4.7`, 20}, {4.7`, 14}}},
   {"Polyline", {{4.7`, 21}, {10, 21}, {10, 27}, {4.7`, 27}, {4.7`, 21}}}};

mySchematic = artNeuronInp04 ~ Join ~ artNeuronOut04 ~ Join ~ artNeuronBody04;
```

```
In[•]:= SetOptions[DrawElement, {ElementSize → {1, 1},
    PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.002], PointSize[0.0012]},
      {RGBColor[0, 0, 1], Thickness[0.0012`], PointSize[0.0014]}},
    ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
    BaseStyle → {FontFamily → "Times", FontSize → 8}}];

In[•]:= numberOfStages = 4;
widthOfStages = 13;
systemBody = artNeuronBody04;
b1TOb1 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b01, b02, b03, b04}]];
b1TOb2 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b11, b12, b13, b14}]];
b1TOb3 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b21, b22, b23, b24}]];
b1TOb4 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b31, b32, b33, b34}]];
w1TOw1 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W011, W012, W013, W014, W021, W022, W023, W024,
     W031, W032, W033, W034, W041, W042, W043, W044}];
w1TOw2 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W111, W112, W113, W114, W121, W122, W123, W124,
     W131, W132, W133, W134, W141, W142, W143, W144}];
w1TOw3 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W211, W212, W213, W214, W221, W222, W223, W224,
     W231, W232, W233, W234, W241, W242, W243, W244}];
w1TOw4 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W311, W312, W313, W314, W321, W322, W323, W324,
     W331, W332, W333, W334, W341, W342, W343, W344}];
B1TObK = {b1TOb2, b1TOb3, b1TOb4};
W1TOWK = {w1TOw2, w1TOw3, w1TOw4};
f1TOfK = {f → f, f → f, f → g};
Do[systemBody =
    Join[systemBody, (TranslateSchematic[artNeuronBody04, {k * widthOfStages, 0}] /.
        B1TObK[[k]] /. W1TOWK[[k]] /. f1TOfK[[k]])];
  , {k, numberOfStages - 1}];

In[•]:= systemInBodyOut = artNeuronInp04 ~ Join ~ systemBody ~ Join ~
    (TranslateSchematic[artNeuronOut04, {(numberOfStages - 1) * widthOfStages, 0}]);
```

```
In[•]:= subsSymbW0 = {W011 → w011, W012 → w012, W013 → w013, W014 → w014,
          W021 → w021, W022 → w022, W023 → w023, W024 → w024, W031 → w031, W032 → w032,
          W033 → w033, W034 → w034, W041 → w041, W042 → w042, W043 → w043, W044 → w044};
        subsSymbW1 = {W111 → w111, W112 → w112, W113 → w113, W114 → w114,
          W121 → w121, W122 → w122, W123 → w123, W124 → w124, W131 → w131, W132 → w132,
          W133 → w133, W134 → w134, W141 → w141, W142 → w142, W143 → w143, W144 → w144};
        subsSymbW2 = {W211 → w211, W212 → w212, W213 → w213, W214 → w214,
          W221 → w221, W222 → w222, W223 → w223, W224 → w224, W231 → w231, W232 → w232,
          W233 → w233, W234 → w234, W241 → w241, W242 → w242, W243 → w243, W244 → w244};
        subsSymbW3 = {W311 → w311, W312 → w312, W313 → w313, W314 → w314,
          W321 → w321, W322 → w322, W323 → w323, W324 → w324, W331 → w331, W332 → w332,
          W333 → w333, W334 → w334, W341 → w341, W342 → w342, W343 → w343, W344 → w344};
        subsSymbB =
          {b01 → b01, b02 → b02, b03 → b03, b04 → b04, b11 → b11, b12 → b12, b13 → b13, b14 → b14,
           b21 → b21, b22 → b22, b23 → b23, b24 → b24, b31 → b31, b32 → b32, b33 → b33, b34 → b34};
        subsSymbX = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};

In[•]:= mySchematic = systemInBodyOut;
        subsALL1 = (subsSymb1 ~ Join ~ subsSymbW0 ~ Join ~ subsSymbW2 ~ Join ~ subsSymbW3 ~
            Join ~ subsSymbW1 ~ Join ~ subsSymbW2 ~ Join ~ subsSymbB ~ Join ~ subsSymbX);
        Show[{DrawElement /@ (mySchematic /. w1TOw1 /. subsALL1)}, Frame → False]

Out[•]=
```



```
In[•]:= procedureName = implementationProcedure;
        DiscreteSystemImplementation2[mySchematic, ToString[procedureName]];
        (* DiscreteSystemImplementationSummary2[mySchematic,Verbose→False]; *)
        eqns = DiscreteSystemImplementationEquations2[mySchematic, Verbose → False];
        numberOfInputs = Length[eqns[[1]]];
        inputSequence =
          MultiplexSequence @@ Table[UnitImpulseSequence[], {numberOfInputs}];
        inputSequence = {{XX1, XX2, XX3, XX4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
        initialConditions = 0 * eqns[[2]];
        systemParameters = eqns[[3]];
        {outputSequence, finalConditions} = DiscreteSystemImplementationProcessing2[
           inputSequence, initialConditions, systemParameters, procedureName];
        y1st = outputSequence[[1]];

        y1stIND = y1st /. subsSymb1 /. w1TOw1 /. subsALL1;
        % // StandardForm
```

```
Implementation procedure name: implementationProcedure

Implementation procedure usage:
```

Out[•]=

```
{{XX1, XX2, XX3, XX4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
```

Out[•]//StandardForm=

$$
\begin{aligned}
&\{g[b_{31} + f[b_{21} + \\
&\quad f[b_{11} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{111} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{121} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{131} + f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{141}] \\
&\quad w_{211} + f[b_{12} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{112} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{122} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{132} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{142}]\,w_{221} + \\
&\quad f[b_{13} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{113} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{123} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{133} + f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{143}] \\
&\quad w_{231} + f[b_{14} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{114} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{124} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{134} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{144}]\,w_{241}]\,w_{311} + \\
&\quad f[b_{22} + f[b_{11} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{111} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{121} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{131} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{141}]\,w_{212} + \\
&\quad f[b_{12} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{112} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{122} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{132} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{142}]\,w_{222} + \\
&\quad f[b_{13} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{113} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{123} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{133} + f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{143}] \\
&\quad w_{232} + f[b_{14} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{114} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{124} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{134} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{144}]\,w_{242}]\,w_{321} + \\
&\quad f[b_{23} + f[b_{11} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{111} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{121} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{131} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{141}]\,w_{213} + \\
&\quad f[b_{12} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{112} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{122} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{132} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{142}]\,w_{223} + \\
&\quad f[b_{13} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{113} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{123} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{133} + f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{143}] \\
&\quad w_{233} + f[b_{14} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{114} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{124} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{134} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{144}]\,w_{243}]\,w_{331} + \\
&\quad f[b_{24} + f[b_{11} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{111} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{121} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{131} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{141}]\,w_{214} + \\
&\quad f[b_{12} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{112} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{122} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{132} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{142}]\,w_{224} + \\
&\quad f[b_{13} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{113} + f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{123} + \\
&\qquad f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{133} + f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{143}] \\
&\quad w_{234} + f[b_{14} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{114} + \\
&\qquad f[b_2 + w_{12}\,x_1 + w_{22}\,x_2 + w_{32}\,x_3 + w_{42}\,x_4]\,w_{124} + f[b_3 + w_{13}\,x_1 + w_{23}\,x_2 + w_{33}\,x_3 + w_{43}\,x_4]\,w_{134} + \\
&\qquad f[b_4 + w_{14}\,x_1 + w_{24}\,x_2 + w_{34}\,x_3 + w_{44}\,x_4]\,w_{144}]\,w_{244}]\,w_{341}], \\
&g[b_{32} + f[b_{21} + f[b_{11} + f[b_1 + w_{11}\,x_1 + w_{21}\,x_2 + w_{31}\,x_3 + w_{41}\,x_4]\,w_{111} +
\end{aligned}
$$

```
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w312 +
f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w322 +
f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w332 +
f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w342],
g[b33 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
```

```
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w313 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w323 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w333 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w343],
  g[b34 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w314 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
```

```
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
      f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w324 +
  f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
      f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
      f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w334 +
  f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
      f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
      f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
      f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
      f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w344]}
```

In[•]:=

```
XXX =
{g[b31 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 + f[b2 + w12 x1 + w22 x2 + w32 x3 +
          w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
        w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w311 +
   f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
```

```
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
         w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w321 +
  f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
         w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w331 +
  f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
         w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w341],
g[b32 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
         w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w312 +
  f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
     f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
         w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w322 +
  f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
```

```
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
       f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w332 +
     f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
       f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w342],
   g[b33 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
       f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w313 +
     f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
       f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w323 +
     f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
```

```
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w333 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
           f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
           f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
         w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w343],
  g[b34 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
           f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
           f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
         w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w314 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
           f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
           f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
         w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w324 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
           f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
           f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
         w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
            w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w334 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
           f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
           f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
```

```
    f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
     f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
   w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
     f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
     f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
  f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
      w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
     f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w344]};
```

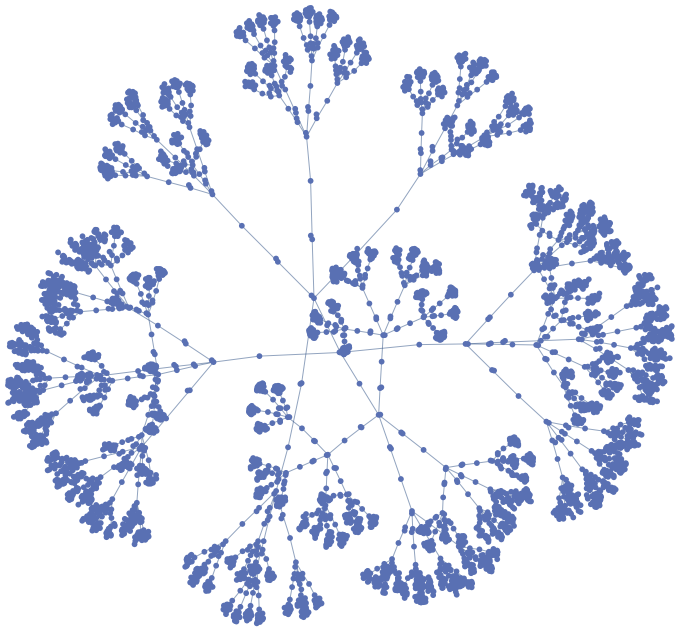
In[•]:= **y1stS = { ··· — };**

Head: Sequence
Length: 4
Byte count: 315880

Uniconize

In[•]:= **ExpressionGraph[y1st]**

Out[•]=



Implementation

```
In[•]:= SeedRandom[123];
        W1m = Round[1000 RandomReal[{-1, 1}, {4, 2}]] / 1000.;
        b1m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
        W2m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
        b2m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
        W3m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
        b3m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
        W4m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
        b4m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
        W1m4 = PadRight[W1m, {4, 4}]; x1m = {1.2, 4.5, 0, 0};
        subsSymbZ = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
        inputSequence /. subsSymbZ
        subsInp = {x1, x2, x3, x4} /. {x1 → 1.2, x2 → 4.5, x3 → 0, x4 → 0}
Out[•]=
        {{x1, x2, x3, x4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
Out[•]=
        {1.2, 4.5, 0, 0}

In[•]:= Clear[W11, W21, W31, W41, W12, W22, W32, W42, W13, W23, W33, W43, W14, W24, W34, W44]
        Wsubs =
          {{W11, W21, W31, W41}, {W12, W22, W32, W42}, {W13, W23, W33, W43}, {W14, W24, W34, W44}};
        subsW = Thread[Flatten[Wsubs] → Flatten[W1m4]]
        Bsubs = {b01, b02, b03, b04};
        subsB = Thread[Flatten[Bsubs] → Flatten[b1m]]
        subsW = Thread[Flatten[Wsubs] → Flatten[W1m4]]
        subsInp = Thread[Flatten[inputSequence][[1 ;; 4]] → Flatten[{1.2, 4.5, 0, 0}]]
        y1stN = N[y1st /. subsW /. subsB /. subsInp /. f → Tanh];
Out[•]=
        {W11 → -0.089, W21 → 0.956, W31 → 0, W41 → 0, W12 → 0.886, W22 → 0.924, W32 → 0, W42 → 0,
         W13 → -0.395, W23 → -0.067, W33 → 0, W43 → 0, W14 → -0.877, W24 → -0.229, W34 → 0, W44 → 0}
Out[•]=
        {b01 → -0.14, b02 → 0.557, b03 → -0.903, b04 → 0.257}
Out[•]=
        {W11 → -0.089, W21 → 0.956, W31 → 0, W41 → 0, W12 → 0.886, W22 → 0.924, W32 → 0, W42 → 0,
         W13 → -0.395, W23 → -0.067, W33 → 0, W43 → 0, W14 → -0.877, W24 → -0.229, W34 → 0, W44 → 0}
Out[•]=
        {XX1 → 1.2, XX2 → 4.5, XX3 → 0, XX4 → 0}
```

```
In[•]:= Wsubs1 = {{W111, W121, W131, W141}, {W112, W122, W132, W142},
          {W113, W123, W133, W143}, {W114, W124, W134, W144}};
        subsW1 = Thread[Flatten[Wsubs1] → Flatten[W2m]]
        Bsubs1 = {b11, b12, b13, b14};
        subsB1 = Thread[Flatten[Bsubs1] → Flatten[b2m]]
        y1stN = y1st /. subsW /. subsW1 /. subsB /. subsB1 /. subsInp /. f → Tanh;
Out[•]=
        {W111 → -0.444, W121 → -0.82, W131 → 0.753, W141 → -0.782, W112 → -0.468, W122 → 0.837,
         W132 → -0.66, W142 → -0.801, W113 → -0.06, W123 → -0.194, W133 → 0.943,
         W143 → -0.37, W114 → -0.748, W124 → -0.455, W134 → 0.211, W144 → 0.344}
Out[•]=
        {b11 → -0.515, b12 → -0.016, b13 → 0.126, b14 → 0.098}

In[•]:= Wsubs2 = {{W211, W221, W231, W241}, {W212, W222, W232, W242},
          {W213, W223, W233, W243}, {W214, W224, W234, W244}};
        subsW2 = Thread[Flatten[Wsubs2] → Flatten[W3m]]
        Bsubs2 = {b21, b22, b23, b24};
        subsB2 = Thread[Flatten[Bsubs2] → Flatten[b3m]]
Out[•]=
        {W211 → -0.037, W221 → 0.289, W231 → -0.165, W241 → 0.409, W212 → 0.923, W222 → 0.808,
         W232 → 0.741, W242 → 0.263, W213 → -0.215, W223 → -0.127, W233 → 0.137,
         W243 → -0.034, W214 → -0.679, W224 → 0.927, W234 → -0.908, W244 → -0.007}
Out[•]=
        {b21 → -0.008, b22 → -0.975, b23 → -0.023, b24 → -0.843}

In[•]:= Wsubs3 = {{W311, W321, W331, W341}, {W312, W322, W332, W342},
          {W313, W323, W333, W343}, {W314, W324, W334, W344}};
        subsW3 = Thread[Flatten[Wsubs3] → Flatten[W4m]]
        Bsubs3 = {b31, b32, b33, b34};
        subsB3 = Thread[Flatten[Bsubs3] → Flatten[b4m]]
        y1stN =
         y1st /. subsW /. subsW1 /. subsW2 /. subsW3 /. subsB /. subsB1 /. subsB2 /. subsB3 /.
           subsInp /. f → Tanh /. g → Exp
Out[•]=
        {W311 → -0.086, W321 → 0.266, W331 → 0.035, W341 → 0.334, W312 → 0.289,
         W322 → -0.174, W332 → 0.989, W342 → 0.578, W313 → -0.584, W323 → -0.469,
         W333 → 0.03, W343 → -0.275, W314 → 0.203, W324 → 0.009, W334 → 0.159, W344 → -0.213}
Out[•]=
        {b31 → 0.01, b32 → 0.891, b33 → -0.337, b34 → -0.087}
Out[•]=
        {1.03727, 4.72264, 0.876151, 0.765064}
```

```
In[•]:= yFinalN = y1stN / Total[y1stN]
        yMax = Max[yFinalN]
        yPos = (Position[yFinalN, yMax] // Flatten)[[1]]
Out[•]=
        {0.14015, 0.638097, 0.118381, 0.103371}
Out[•]=
        0.638097
Out[•]=
        2
```

Implementation symbolic

```
In[•]:= yFinalN =
         y1st / Total[y1st] /. subsW /. subsW1 /. subsW2 /. subsW3 /. subsB /. subsB1 /. subsB2 /.
           subsB3 /.  subsInp /. f → Tanh /. g → Exp
Out[•]=
        {0.14015, 0.638097, 0.118381, 0.103371}
```

---

sscnnFigure07

```
In[•]:= SmainDir = SetDirectory[NotebookDirectory[]];
        Get["SchematicSolver`"]
        << SSknowledgeImp00`
```

```
In[•]:= subsSymb1 = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
        subsSymb2 =
          {W11 → w11, W12 → w12, W13 → w13, W14 → w14, W21 → w21, W22 → w22, W23 → w23, W24 → w24,
           W31 → w31, W32 → w32, W33 → w33, W34 → w34, W41 → w41, W42 → w42, W43 → w43, W44 → w44};
        subsSymb3 = {b01 → b01, b02 → b02, b03 → b03, b04 → b04, b11 → b11,
           b12 → b12, b13 → b13, b14 → b14, b21 → b21, b22 → b22, b23 → b23, b24 → b24};
        SetOptions[DrawElement, {BaseStyle → {FontFamily → "Times", FontSize → 8}}];
        SetOptions[DrawElement, {ElementSize → {1, 1}}];
        plotStyleDD = {{Dashing[.01], GrayLevel[.5]}, {Dashing[.01], GrayLevel[.5]}};
        SetOptions[DrawElement, {ElementSize → {1, 1},
           PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.004], PointSize[0.0025]},
             {RGBColor[0, 0, 1], Thickness[0.0025`], PointSize[0.003]}},
           ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
           BaseStyle → {FontFamily → "Times", FontSize → 8}}];
```

```
In[•]:= artNeuronInp04 = {
           {"Input", {0, 3}, XX1, " ", TextOffset → {1, 0}},
           {"Input", {0, 10}, XX2, " ", TextOffset → {1, 0}}, {"Input", {0, 17}, XX3, " ",
            TextOffset → {1, 0}}, {"Input", {0, 24}, XX4, " ", TextOffset → {1, 0}}};
        Show[{DrawElement /@ (artNeuronInp04 /. subsSymb1 /. subsSymb2)}, Frame → False];
        artNeuronOut04 = {
           {"Output", {13, 3}, YY1, " ", TextOffset → {-1, 0}},
           {"Output", {13, 10}, YY2, " ", TextOffset → {-1, 0}},
           {"Output", {13, 17}, YY3, " ", TextOffset → {-1, 0}},
           {"Output", {13, 24}, YY4, " ", TextOffset → {-1, 0}}};
        artNeuronBody04 = {
```

```
{"Adder", {{7, 3}, {5, 2}, {8, 3}, {5, 4}}, {1, 1, 2, 1}, " "},
{"Adder", {{8, 3}, {5, 1}, {11, 3}, {5, 5}}, {1, 1, 2, 1}, "1"},
{"Adder", {{7, 10}, {5, 9}, {8, 10}, {5, 11}}, {1, 1, 2, 1}, " "},
{"Adder", {{8, 10}, {5, 8}, {11, 10}, {5, 12}}, {1, 1, 2, 1}, "2"},
{"Adder", {{7, 17}, {5, 16}, {8, 17}, {5, 18}}, {1, 1, 2, 1}, " "},
{"Adder", {{8, 17}, {5, 15}, {11, 17}, {5, 19}}, {1, 1, 2, 1}, "3"},
{"Adder", {{7, 24}, {5, 23}, {8, 24}, {5, 25}}, {1, 1, 2, 1}, " "},
{"Adder", {{8, 24}, {5, 22}, {11, 24}, {5, 26}}, {1, 1, 2, 1}, "4"},
{"Function", {{11, 24}, {13, 24}}, f, " "}, {"Function", {{11, 17}, {13, 17}}, f, " "},
{"Function", {{11, 10}, {13, 10}}, f, " "}, {"Function", {{11, 3}, {13, 3}}, f, " "},
{"Input", {6, 24}, 1, " ", TextOffset → {1, 0}},
{"Input", {6, 17}, 1, " ", TextOffset → {1, 0}},
{"Input", {6, 10}, 1, " ", TextOffset → {1, 0}},
{"Input", {6, 3}, 1, " ", TextOffset → {1, 0}},
{"Line", {{0, 3}, {3, 0}}}, {"Line", {{0, 10}, {3, 2}}},
{"Line", {{0, 3}, {3, 7}}}, {"Line", {{0, 3}, {3, 14}}}, {"Line", {{0, 3}, {3, 21}}},
{"Line", {{0, 17}, {3, 4}}}, {"Line", {{0, 10}, {3, 9}}}, {"Line", {{0, 10}, {3, 16}}},
{"Line", {{0, 10}, {3, 23}}}, {"Line", {{0, 17}, {3, 11}}},
{"Line", {{0, 17}, {3, 18}}}, {"Line", {{0, 17}, {3, 25}}},
{"Line", {{0, 24}, {3, 27}}}, {"Line", {{0, 24}, {3, 20}}},
{"Line", {{0, 24}, {3, 13}}}, {"Line", {{0, 24}, {3, 6}}},
{"Multiplier", {{3, 0}, {5, 1}}, W11, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 2}, {5, 2}}, W21, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 4}, {5, 4}}, W31, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 6}, {5, 5}}, W41, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 7}, {5, 8}}, W12, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 14}, {5, 15}}, W13, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 21}, {5, 22}}, W14, " ", ElementSize → {1, 1}},
{"Multiplier", {{3, 9}, {5, 9}}, W22, " "}, {"Multiplier",
 {{3, 11}, {5, 11}}, W32, " "}, {"Multiplier", {{3, 16}, {5, 16}}, W23, " "},
{"Multiplier", {{3, 18}, {5, 18}}, W33, " "}, {"Multiplier", {{3, 13}, {5, 12}},
 W42, " "}, {"Multiplier", {{3, 20}, {5, 19}}, W43, " "}, {"Multiplier",
 {{3, 23}, {5, 23}}, W24, " "}, {"Multiplier", {{3, 25}, {5, 25}}, W34, " "},
{"Multiplier", {{3, 27}, {5, 26}}, W44, " "}, {"Multiplier", {{6, 3}, {7, 3}},
 b01, " "}, {"Multiplier", {{6, 10}, {7, 10}}, b02, " "},
{"Multiplier", {{6, 17}, {7, 17}}, b03, " "}, {"Multiplier", {{6, 24}, {7, 24}},
 b04, " "}, {"Polyline", {{4.7`, 0}, {10, 0}, {10, 6}, {4.7`, 6}, {4.7`, 0}}},
{"Polyline", {{4.7`, 7}, {10, 7}, {10, 13}, {4.7`, 13}, {4.7`, 7}}},
{"Polyline", {{4.7`, 14}, {10, 14}, {10, 20}, {4.7`, 20}, {4.7`, 14}}},
{"Polyline", {{4.7`, 21}, {10, 21}, {10, 27}, {4.7`, 27}, {4.7`, 21}}}};
```

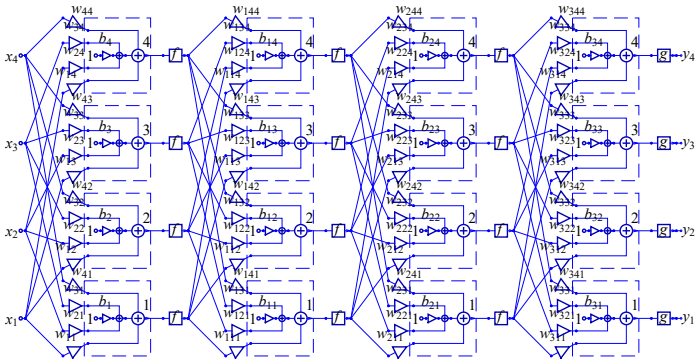
In[•]:=
```
mySchematic = artNeuronInp04 ~ Join ~ artNeuronOut04 ~ Join ~ artNeuronBody04;
```

In[•]:=
```
SetOptions[DrawElement, {ElementSize → {1, 1},
    PlotStyle → {{RGBColor[0, 0, 0.7], Thickness[0.002], PointSize[0.0012]},
      {RGBColor[0, 0, 1], Thickness[0.0012`], PointSize[0.0014]}},
    ShowArrowTail → True, ShowNodes → True, TextOffset → Automatic,
    BaseStyle → {FontFamily → "Times", FontSize → 8}}];
```

```
In[•]:= numberOfStages = 4;
widthOfStages = 13;
systemBody = artNeuronBody04;
b1TOb1 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b01, b02, b03, b04}]];
b1TOb2 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b11, b12, b13, b14}]];
b1TOb3 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b21, b22, b23, b24}]];
b1TOb4 = Thread[Flatten[{b01, b02, b03, b04}] → Flatten[{b31, b32, b33, b34}]];
w1TOw1 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W011, W012, W013, W014, W021, W022, W023, W024,
     W031, W032, W033, W034, W041, W042, W043, W044}];
w1TOw2 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W111, W112, W113, W114, W121, W122, W123, W124,
     W131, W132, W133, W134, W141, W142, W143, W144}];
w1TOw3 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W211, W212, W213, W214, W221, W222, W223, W224,
     W231, W232, W233, W234, W241, W242, W243, W244}];
w1TOw4 =
  Thread[{W11, W12, W13, W14, W21, W22, W23, W24, W31, W32, W33, W34, W41, W42, W43, W44} →
    {W311, W312, W313, W314, W321, W322, W323, W324,
     W331, W332, W333, W334, W341, W342, W343, W344}];
B1TObK = {b1TOb2, b1TOb3, b1TOb4};
W1TOWK = {w1TOw2, w1TOw3, w1TOw4};
f1TOfK = {f → f, f → f, f → g};
Do[systemBody =
    Join[systemBody, (TranslateSchematic[artNeuronBody04, {k * widthOfStages, 0}] /.
        B1TObK[[k]] /. W1TOWK[[k]] /. f1TOfK[[k]])];
  , {k, numberOfStages - 1}];

In[•]:= systemInBodyOut = artNeuronInp04 ~ Join ~ systemBody ~ Join ~
    (TranslateSchematic[artNeuronOut04, {(numberOfStages - 1) * widthOfStages, 0}]);

In[•]:= subsSymbW0 = {W011 → w011, W012 → w012, W013 → w013, W014 → w014,
    W021 → w021, W022 → w022, W023 → w023, W024 → w024, W031 → w031, W032 → w032,
    W033 → w033, W034 → w034, W041 → w041, W042 → w042, W043 → w043, W044 → w044};
subsSymbW1 = {W111 → w111, W112 → w112, W113 → w113, W114 → w114,
    W121 → w121, W122 → w122, W123 → w123, W124 → w124, W131 → w131, W132 → w132,
    W133 → w133, W134 → w134, W141 → w141, W142 → w142, W143 → w143, W144 → w144};
subsSymbW2 = {W211 → w211, W212 → w212, W213 → w213, W214 → w214,
    W221 → w221, W222 → w222, W223 → w223, W224 → w224, W231 → w231, W232 → w232,
    W233 → w233, W234 → w234, W241 → w241, W242 → w242, W243 → w243, W244 → w244};
subsSymbW3 = {W311 → w311, W312 → w312, W313 → w313, W314 → w314,
    W321 → w321, W322 → w322, W323 → w323, W324 → w324, W331 → w331, W332 → w332,
    W333 → w333, W334 → w334, W341 → w341, W342 → w342, W343 → w343, W344 → w344};
subsSymbB =
  {b01 → b01, b02 → b02, b03 → b03, b04 → b04, b11 → b11, b12 → b12, b13 → b13, b14 → b14,
   b21 → b21, b22 → b22, b23 → b23, b24 → b24, b31 → b31, b32 → b32, b33 → b33, b34 → b34};
subsSymbX = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
```

In[•]:=
```
mySchematic = systemInBodyOut;
subsALL1 = (subsSymb1 ~ Join ~ subsSymbW0 ~ Join ~ subsSymbW2 ~ Join ~ subsSymbW3 ~
    Join ~ subsSymbW1 ~ Join ~ subsSymbW2 ~ Join ~ subsSymbB ~ Join ~ subsSymbX);
Show[{DrawElement /@ (mySchematic /. w1TOw1 /. subsALL1)}, Frame → False]
```

Out[•]=



In[•]:=
```
procedureName = implementationProcedure;
DiscreteSystemImplementation2[mySchematic, ToString[procedureName]];
(* DiscreteSystemImplementationSummary2[mySchematic,Verbose→False]; *)
eqns = DiscreteSystemImplementationEquations2[mySchematic, Verbose → False];
numberOfInputs = Length[eqns[[1]]];
inputSequence =
  MultiplexSequence @@ Table[UnitImpulseSequence[], {numberOfInputs}];
inputSequence = {{XX1, XX2, XX3, XX4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
initialConditions = 0 * eqns[[2]];
systemParameters = eqns[[3]];
{outputSequence, finalConditions} = DiscreteSystemImplementationProcessing2[
   inputSequence, initialConditions, systemParameters, procedureName];
y1st = outputSequence[[1]];

y1stIND = y1st /. subsSymb1 /. w1TOw1 /. subsALL1;
% // StandardForm
```

```
Implementation procedure name: implementationProcedure

Implementation procedure usage:
```

Out[•]=
```
{{XX1, XX2, XX3, XX4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
```

Out[•]//StandardForm=

$$\{g[b_{31} + f[b_{21} + f[b_{11} + f[b_1 + w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + w_{41} x_4] w_{111} + f[b_2 + w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + w_{42} x_4] w_{121} + f[b_3 + w_{13} x_1 + w_{23} x_2 + w_{33} x_3 + w_{43} x_4] w_{131} + f[b_4 + w_{14} x_1 + w_{24} x_2 + w_{34} x_3 + w_{44} x_4] w_{141}] w_{211} + f[b_{12} + f[b_1 + w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + w_{41} x_4] w_{112} + f[b_2 + w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + w_{42} x_4] w_{122} + f[b_3 + w_{13} x_1 + w_{23} x_2 + w_{33} x_3 + w_{43} x_4] w_{132} + f[b_4 + w_{14} x_1 + w_{24} x_2 + w_{34} x_3 + w_{44} x_4] w_{142}] w_{221} + f[b_{13} + f[b_1 + w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + w_{41} x_4] w_{113} + f[b_2 + w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + w_{42} x_4] w_{123} + f[b_3 + w_{13} x_1 + w_{23} x_2 + w_{33} x_3 + w_{43} x_4] w_{133} + f[b_4 + w_{14} x_1 + w_{24} x_2 + w_{34} x_3 + w_{44} x_4] w_{143}] w_{231} + f[b_{14} + f[b_1 + w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + w_{41} x_4] w_{114} + f[b_2 + w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + w_{42} x_4] w_{124} + f[b_3 + w_{13} x_1 + w_{23} x_2 + w_{33} x_3 + w_{43} x_4] w_{134} + f[b_4 + w_{14} x_1 + w_{24} x_2 + w_{34} x_3 + w_{44} x_4] w_{144}] w_{241}] w_{311} + f[b_{22} + f[b_{11} + f[b_1 + w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + w_{41} x_4] w_{111} +$$

```
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w321 +
  f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w331 +
  f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w341],
g[b32 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w312 +
  f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
```

```
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w322 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w332 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w342],
  g[b33 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w313 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
      f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
      f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
      w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w323 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
```

```
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w333 +
f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w343],
g[b34 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w221 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w231 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w314 +
f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w222 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w232 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w324 +
f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w223 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
```

```
    w233 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
       f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
       f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w334 +
  f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
       f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
       f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 +
    f[b12 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
       f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 +
       f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142] w224 +
    f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 +
       f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143]
     w234 + f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 +
       f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
       f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w344]}
```

In[•]:= **XXX =**

```
{g[b31 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 + f[b2 + w12 x1 + w22 x2 + w32 x3 +
                  w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
               f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
               f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
            w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
            f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
                  w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w311 +
         f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
               f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
               f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
            w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
            f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
                  w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w321 +
         f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
               f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
               f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
            w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
            f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
                  w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
               f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w331 +
         f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
               f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
```

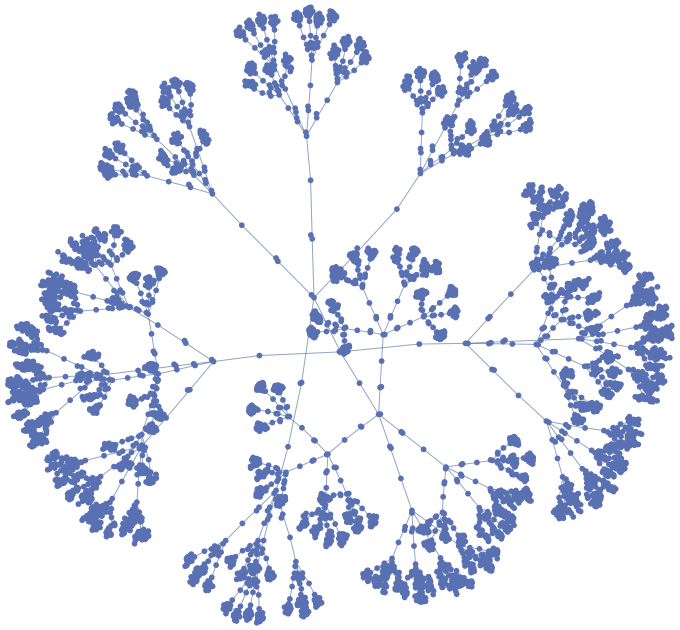
```
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w341],
  g[b32 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w312 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w322 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w332 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
        f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
        f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
      w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
        f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
        f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
```

```
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w342],
  g[b33 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w313 +
    f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w323 +
    f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w333 +
    f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
          f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
          f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
        w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
      f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
          w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w343],
  g[b34 + f[b21 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
          f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
          f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w211 + f[b12 +
```

```
f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
   f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
 w221 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w231 +
 f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
     w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w241] w314 +
f[b22 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w212 + f[b12 +
   f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
   f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
 w222 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w232 +
 f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
     w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w242] w324 +
f[b23 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w213 + f[b12 +
   f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
   f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
 w223 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w233 +
 f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
     w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w243] w334 +
f[b24 + f[b11 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w111 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w121 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w131 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w141] w214 + f[b12 +
   f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w112 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w122 +
   f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w132 + f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w142]
 w224 + f[b13 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w113 +
   f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4] w123 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w133 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w143] w234 +
 f[b14 + f[b1 + w11 x1 + w21 x2 + w31 x3 + w41 x4] w114 + f[b2 + w12 x1 + w22 x2 + w32 x3 + w42 x4]
     w124 + f[b3 + w13 x1 + w23 x2 + w33 x3 + w43 x4] w134 +
   f[b4 + w14 x1 + w24 x2 + w34 x3 + w44 x4] w144] w244] w344]};
```

In[•]:= `slika007b = ExpressionGraph[y1st]`

Out[•]=



```
(*
slika007a=Import["slika007a.bmp"];
slika007c=Import["slika007c.png"];
Export["slika007a.tif",slika007a,ImageResolution→300];
Export["slika007b.tif",slika007b,ImageResolution→300];
Export["slika007c.tif",slika007c,ImageResolution→300];
*)
(*
Export["slika007b.tif",slika007b,ImageResolution→300];
*)
```

Implementation

In[•]:=
```
SeedRandom[123];
W1m = Round[1000 RandomReal[{-1, 1}, {4, 2}]] / 1000.;
b1m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
W2m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
b2m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
W3m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
b3m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
W4m = Round[1000 RandomReal[{-1, 1}, {4, 4}]] / 1000.;
b4m = Round[1000 RandomReal[{-1, 1}, 4]] / 1000.;
W1m4 = PadRight[W1m, {4, 4}]; x1m = {1.2, 4.5, 0, 0};
subsSymbZ = {XX1 → x1, XX2 → x2, XX3 → x3, XX4 → x4, YY1 → y1, YY2 → y2, YY3 → y3, YY4 → y4};
inputSequence /. subsSymbZ
subsInp = {x1, x2, x3, x4} /. {x1 → 1.2, x2 → 4.5, x3 → 0, x4 → 0}
```

Out[•]=
```
{{x1, x2, x3, x4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}}
```

Out[•]=
```
{1.2, 4.5, 0, 0}
```

```
In[•]:= Clear[W11, W21, W31, W41, W12, W22, W32, W42, W13, W23, W33, W43, W14, W24, W34, W44]
        Wsubs =
          {{W11, W21, W31, W41}, {W12, W22, W32, W42}, {W13, W23, W33, W43}, {W14, W24, W34, W44}};
        subsW = Thread[Flatten[Wsubs] → Flatten[W1m4]]
        Bsubs = {b01, b02, b03, b04};
        subsB = Thread[Flatten[Bsubs] → Flatten[b1m]]
        subsW = Thread[Flatten[Wsubs] → Flatten[W1m4]]
        subsInp = Thread[Flatten[inputSequence][[1 ;; 4]] → Flatten[{1.2, 4.5, 0, 0}]]
        y1stN = N[y1st /. subsW /. subsB /. subsInp /. f → Tanh];
Out[•]=
        {W11 → -0.089, W21 → 0.956, W31 → 0, W41 → 0, W12 → 0.886, W22 → 0.924, W32 → 0, W42 → 0,
         W13 → -0.395, W23 → -0.067, W33 → 0, W43 → 0, W14 → -0.877, W24 → -0.229, W34 → 0, W44 → 0}
Out[•]=
        {b01 → -0.14, b02 → 0.557, b03 → -0.903, b04 → 0.257}
Out[•]=
        {W11 → -0.089, W21 → 0.956, W31 → 0, W41 → 0, W12 → 0.886, W22 → 0.924, W32 → 0, W42 → 0,
         W13 → -0.395, W23 → -0.067, W33 → 0, W43 → 0, W14 → -0.877, W24 → -0.229, W34 → 0, W44 → 0}
Out[•]=
        {XX1 → 1.2, XX2 → 4.5, XX3 → 0, XX4 → 0}

In[•]:= Wsubs1 = {{W111, W121, W131, W141}, {W112, W122, W132, W142},
            {W113, W123, W133, W143}, {W114, W124, W134, W144}};
        subsW1 = Thread[Flatten[Wsubs1] → Flatten[W2m]]
        Bsubs1 = {b11, b12, b13, b14};
        subsB1 = Thread[Flatten[Bsubs1] → Flatten[b2m]]
        y1stN = y1st /. subsW /. subsW1 /. subsB /. subsB1 /. subsInp /. f → Tanh;
Out[•]=
        {W111 → -0.444, W121 → -0.82, W131 → 0.753, W141 → -0.782, W112 → -0.468, W122 → 0.837,
         W132 → -0.66, W142 → -0.801, W113 → -0.06, W123 → -0.194, W133 → 0.943,
         W143 → -0.37, W114 → -0.748, W124 → -0.455, W134 → 0.211, W144 → 0.344}
Out[•]=
        {b11 → -0.515, b12 → -0.016, b13 → 0.126, b14 → 0.098}

In[•]:= Wsubs2 = {{W211, W221, W231, W241}, {W212, W222, W232, W242},
            {W213, W223, W233, W243}, {W214, W224, W234, W244}};
        subsW2 = Thread[Flatten[Wsubs2] → Flatten[W3m]]
        Bsubs2 = {b21, b22, b23, b24};
        subsB2 = Thread[Flatten[Bsubs2] → Flatten[b3m]]
Out[•]=
        {W211 → -0.037, W221 → 0.289, W231 → -0.165, W241 → 0.409, W212 → 0.923, W222 → 0.808,
         W232 → 0.741, W242 → 0.263, W213 → -0.215, W223 → -0.127, W233 → 0.137,
         W243 → -0.034, W214 → -0.679, W224 → 0.927, W234 → -0.908, W244 → -0.007}
Out[•]=
        {b21 → -0.008, b22 → -0.975, b23 → -0.023, b24 → -0.843}
```

```
In[•]:= Wsubs3 = {{W311, W321, W331, W341}, {W312, W322, W332, W342},
          {W313, W323, W333, W343}, {W314, W324, W334, W344}};
        subsW3 = Thread[Flatten[Wsubs3] → Flatten[W4m]]
        Bsubs3 = {b31, b32, b33, b34};
        subsB3 = Thread[Flatten[Bsubs3] → Flatten[b4m]]
        y1stN =
         y1st /. subsW /. subsW1 /. subsW2 /. subsW3 /. subsB /. subsB1 /. subsB2 /. subsB3 /.
           subsInp /. f → Tanh /. g → Exp
```

Out[•]=

```
{W311 → -0.086, W321 → 0.266, W331 → 0.035, W341 → 0.334, W312 → 0.289,
 W322 → -0.174, W332 → 0.989, W342 → 0.578, W313 → -0.584, W323 → -0.469,
 W333 → 0.03, W343 → -0.275, W314 → 0.203, W324 → 0.009, W334 → 0.159, W344 → -0.213}
```

Out[•]=

```
{b31 → 0.01, b32 → 0.891, b33 → -0.337, b34 → -0.087}
```

Out[•]=

```
{1.03727, 4.72264, 0.876151, 0.765064}
```

```
In[•]:= yFinalN = y1stN / Total[y1stN]
        yMax = Max[yFinalN]
        yPos = (Position[yFinalN, yMax] // Flatten)[[1]]
```

Out[•]=

```
{0.14015, 0.638097, 0.118381, 0.103371}
```

Out[•]=

```
0.638097
```

Out[•]=

```
2
```

Implementation symbolic

```
In[•]:= yFinalN =
         y1st / Total[y1st] /. subsW /. subsW1 /. subsW2 /. subsW3 /. subsB /. subsB1 /. subsB2 /.
           subsB3 /. subsInp /. f → Tanh /. g → Exp
```

Out[•]=

```
{0.14015, 0.638097, 0.118381, 0.103371}
```

```
In[•]:= yFinalN // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.14015 \\ 0.638097 \\ 0.118381 \\ 0.103371 \end{pmatrix}$$

sscnnFigure08

```
In[•]:= SmainDir = SetDirectory[NotebookDirectory[]];
        Clear[t0, t1, t2, t3, t4, x, y]
```

```
In[•]:= t0 = {{0.25`, 1.75`}, {0.26`, 10.5`}, {0.27`, 42.25`}, {0.28`, 87.75`}, {0.29`, 137.25`},
          {0.3`, 209.75`}, {0.31`, 295.75`}, {0.32`, 384.5`}, {0.33`, 474.75`},
          {0.34`, 535.5`}, {0.35`, 619.75`}, {0.36`, 652.75`}, {0.37`, 720.25`},
```

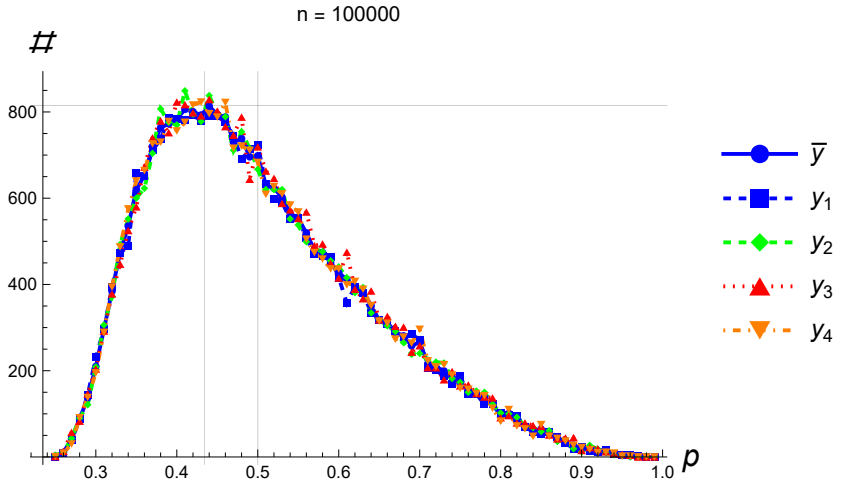
```
    {0.38`, 763.25`}, {0.39`, 772.75`}, {0.4`, 783.25`}, {0.41`, 806.`}, {0.42`, 800.5`},
    {0.43`, 792.75`}, {0.44`, 813.5`}, {0.45`, 794.`}, {0.46`, 788.25`},
    {0.47`, 729.5`}, {0.48`, 738.25`}, {0.49`, 696.`}, {0.5`, 698.25`}, {0.51`, 631.`},
    {0.52`, 623.`}, {0.53`, 602.25`}, {0.54`, 565.`}, {0.55`, 553.5`}, {0.56`, 519.75`},
    {0.57`, 479.5`}, {0.58`, 474.`}, {0.59`, 450.75`}, {0.6`, 427.`}, {0.61`, 411.75`},
    {0.62`, 393.`}, {0.63`, 382.5`}, {0.64`, 351.`}, {0.65`, 317.`}, {0.66`, 312.`},
    {0.67`, 289.5`}, {0.68`, 280.75`}, {0.69`, 257.75`}, {0.7`, 266.25`},
    {0.71`, 215.5`}, {0.72`, 209.25`}, {0.73`, 199.5`}, {0.74`, 185.`}, {0.75`, 171.`},
    {0.76`, 154.75`}, {0.77`, 148.25`}, {0.78`, 138.25`}, {0.79`, 123.5`},
    {0.8`, 94.75`}, {0.81`, 101.`}, {0.82`, 84.75`}, {0.83`, 71.`}, {0.84`, 62.`},
    {0.85`, 64.5`}, {0.86`, 54.75`}, {0.87`, 41.75`}, {0.88`, 38.5`}, {0.89`, 27.5`},
    {0.9`, 17.75`}, {0.91`, 19.75`}, {0.92`, 14.5`}, {0.93`, 12.25`}, {0.94`, 7.25`},
    {0.95`, 3.75`}, {0.96`, 2.5`}, {0.97`, 0.75`}, {0.98`, 0.`}, {0.99`, 0.`}};
t1 = {{0.25`, 1}, {0.26`, 10}, {0.27`, 39}, {0.28`, 87}, {0.29`, 144}, {0.3`, 231},
    {0.31`, 296}, {0.32`, 394}, {0.33`, 472}, {0.34`, 490}, {0.35`, 659}, {0.36`, 652},
    {0.37`, 711}, {0.38`, 737}, {0.39`, 785}, {0.4`, 784}, {0.41`, 782}, {0.42`, 793},
    {0.43`, 779}, {0.44`, 791}, {0.45`, 790}, {0.46`, 776}, {0.47`, 745},
    {0.48`, 690}, {0.49`, 713}, {0.5`, 724}, {0.51`, 632}, {0.52`, 599}, {0.53`, 591},
    {0.54`, 551}, {0.55`, 555}, {0.56`, 507}, {0.57`, 471}, {0.58`, 467},
    {0.59`, 463}, {0.6`, 416}, {0.61`, 357}, {0.62`, 395}, {0.63`, 381}, {0.64`, 334},
    {0.65`, 317}, {0.66`, 308}, {0.67`, 291}, {0.68`, 279}, {0.69`, 286}, {0.7`, 271},
    {0.71`, 207}, {0.72`, 201}, {0.73`, 187}, {0.74`, 170}, {0.75`, 187},
    {0.76`, 145}, {0.77`, 146}, {0.78`, 122}, {0.79`, 122}, {0.8`, 103}, {0.81`, 98},
    {0.82`, 96}, {0.83`, 69}, {0.84`, 59}, {0.85`, 53}, {0.86`, 59}, {0.87`, 46},
    {0.88`, 33}, {0.89`, 19}, {0.9`, 22}, {0.91`, 13}, {0.92`, 11}, {0.93`, 16},
    {0.94`, 7}, {0.95`, 4}, {0.96`, 4}, {0.97`, 2}, {0.98`, 0}, {0.99`, 0}};
t2 = {{0.25`, 4}, {0.26`, 10}, {0.27`, 43}, {0.28`, 90}, {0.29`, 121}, {0.3`, 209},
    {0.31`, 306}, {0.32`, 372}, {0.33`, 493}, {0.34`, 552}, {0.35`, 600}, {0.36`, 623},
    {0.37`, 704}, {0.38`, 807}, {0.39`, 776}, {0.4`, 770}, {0.41`, 849}, {0.42`, 795},
    {0.43`, 778}, {0.44`, 838}, {0.45`, 796}, {0.46`, 789}, {0.47`, 711},
    {0.48`, 754}, {0.49`, 716}, {0.5`, 667}, {0.51`, 620}, {0.52`, 620}, {0.53`, 620},
    {0.54`, 552}, {0.55`, 538}, {0.56`, 499}, {0.57`, 483}, {0.58`, 476},
    {0.59`, 455}, {0.6`, 441}, {0.61`, 416}, {0.62`, 381}, {0.63`, 393}, {0.64`, 334},
    {0.65`, 318}, {0.66`, 304}, {0.67`, 291}, {0.68`, 265}, {0.69`, 238}, {0.7`, 240},
    {0.71`, 225}, {0.72`, 220}, {0.73`, 218}, {0.74`, 182}, {0.75`, 173},
    {0.76`, 150}, {0.77`, 153}, {0.78`, 150}, {0.79`, 122}, {0.8`, 102}, {0.81`, 100},
    {0.82`, 92}, {0.83`, 71}, {0.84`, 68}, {0.85`, 59}, {0.86`, 61}, {0.87`, 36},
    {0.88`, 40}, {0.89`, 19}, {0.9`, 12}, {0.91`, 27}, {0.92`, 18}, {0.93`, 10},
    {0.94`, 5}, {0.95`, 3}, {0.96`, 0}, {0.97`, 1}, {0.98`, 0}, {0.99`, 0}};
t3 = {{0.25`, 1}, {0.26`, 13}, {0.27`, 57}, {0.28`, 83}, {0.29`, 146}, {0.3`, 204},
    {0.31`, 292}, {0.32`, 378}, {0.33`, 447}, {0.34`, 525}, {0.35`, 580}, {0.36`, 672},
    {0.37`, 740}, {0.38`, 779}, {0.39`, 752}, {0.4`, 823}, {0.41`, 817}, {0.42`, 798},
    {0.43`, 791}, {0.44`, 829}, {0.45`, 802}, {0.46`, 766}, {0.47`, 746},
    {0.48`, 788}, {0.49`, 644}, {0.5`, 719}, {0.51`, 663}, {0.52`, 646}, {0.53`, 588},
    {0.54`, 573}, {0.55`, 553}, {0.56`, 568}, {0.57`, 490}, {0.58`, 494},
    {0.59`, 449}, {0.6`, 415}, {0.61`, 475}, {0.62`, 389}, {0.63`, 367}, {0.64`, 385},
    {0.65`, 318}, {0.66`, 326}, {0.67`, 303}, {0.68`, 301}, {0.69`, 243}, {0.7`, 258},
    {0.71`, 207}, {0.72`, 207}, {0.73`, 179}, {0.74`, 198}, {0.75`, 165},
    {0.76`, 167}, {0.77`, 147}, {0.78`, 139}, {0.79`, 137}, {0.8`, 91}, {0.81`, 96},
    {0.82`, 79}, {0.83`, 79}, {0.84`, 73}, {0.85`, 71}, {0.86`, 51}, {0.87`, 42},
```

```
    {0.88`, 44}, {0.89`, 45}, {0.9`, 15}, {0.91`, 22}, {0.92`, 19}, {0.93`, 12},
    {0.94`, 8}, {0.95`, 6}, {0.96`, 4}, {0.97`, 0}, {0.98`, 0}, {0.99`, 0}};
t4 = {{0.25`, 1}, {0.26`, 9}, {0.27`, 30}, {0.28`, 91}, {0.29`, 138}, {0.3`, 195},
    {0.31`, 289}, {0.32`, 394}, {0.33`, 487}, {0.34`, 575}, {0.35`, 640}, {0.36`, 664},
    {0.37`, 726}, {0.38`, 730}, {0.39`, 778}, {0.4`, 756}, {0.41`, 776}, {0.42`, 816},
    {0.43`, 823}, {0.44`, 796}, {0.45`, 788}, {0.46`, 822}, {0.47`, 716},
    {0.48`, 721}, {0.49`, 711}, {0.5`, 683}, {0.51`, 609}, {0.52`, 627}, {0.53`, 610},
    {0.54`, 584}, {0.55`, 568}, {0.56`, 505}, {0.57`, 474}, {0.58`, 459},
    {0.59`, 436}, {0.6`, 436}, {0.61`, 399}, {0.62`, 407}, {0.63`, 389}, {0.64`, 351},
    {0.65`, 315}, {0.66`, 310}, {0.67`, 273}, {0.68`, 278}, {0.69`, 264}, {0.7`, 296},
    {0.71`, 223}, {0.72`, 209}, {0.73`, 214}, {0.74`, 190}, {0.75`, 159},
    {0.76`, 157}, {0.77`, 147}, {0.78`, 142}, {0.79`, 113}, {0.8`, 83}, {0.81`, 110},
    {0.82`, 72}, {0.83`, 65}, {0.84`, 48}, {0.85`, 75}, {0.86`, 48}, {0.87`, 43},
    {0.88`, 37}, {0.89`, 27}, {0.9`, 22}, {0.91`, 17}, {0.92`, 10}, {0.93`, 11},
    {0.94`, 9}, {0.95`, 2}, {0.96`, 2}, {0.97`, 0}, {0.98`, 0}, {0.99`, 0}};
sn = 100 000;
```

In[•]:=

```
yGridLines = {{0.43435463392312135, 0.5}, {815}};
slika008a = ListLinePlot[{t0, t1, t2, t3, t4},
  AxesLabel → {Style["p", Italic, FontFamily → "Arial", FontSize → 16],
    Style["#", Italic, FontFamily → "Arial", FontSize → 16]},
  PlotLabel → "n = " <> ToString[sn], PlotStyle → {{Thick, Blue}, {Thick, Dashed, Blue},
    {Thick, Dashed, Green}, {Thick, Dotted, Red}, {Thick, DotDashed, Orange}},
  PlotMarkers → {Automatic, Scaled[0.02]}, GridLines → yGridLines,
  Ticks → {Automatic, Automatic}, PlotLegends → {"y̅", "y₁", "y₂", "y₃", "y₄"}]
```

Out[•]=




```
(*
Export["slika008a.tif",slika008a,ImageResolution→300];
*)
```

In[•]:=
```
t0
```

Out[•]=
```
{{0.25, 1.75}, {0.26, 10.5}, {0.27, 42.25}, {0.28, 87.75}, {0.29, 137.25}, {0.3, 209.75},
 {0.31, 295.75}, {0.32, 384.5}, {0.33, 474.75}, {0.34, 535.5}, {0.35, 619.75},
 {0.36, 652.75}, {0.37, 720.25}, {0.38, 763.25}, {0.39, 772.75}, {0.4, 783.25},
 {0.41, 806.}, {0.42, 800.5}, {0.43, 792.75}, {0.44, 813.5}, {0.45, 794.},
 {0.46, 788.25}, {0.47, 729.5}, {0.48, 738.25}, {0.49, 696.}, {0.5, 698.25},
 {0.51, 631.}, {0.52, 623.}, {0.53, 602.25}, {0.54, 565.}, {0.55, 553.5},
 {0.56, 519.75}, {0.57, 479.5}, {0.58, 474.}, {0.59, 450.75}, {0.6, 427.},
 {0.61, 411.75}, {0.62, 393.}, {0.63, 382.5}, {0.64, 351.}, {0.65, 317.}, {0.66, 312.},
 {0.67, 289.5}, {0.68, 280.75}, {0.69, 257.75}, {0.7, 266.25}, {0.71, 215.5},
 {0.72, 209.25}, {0.73, 199.5}, {0.74, 185.}, {0.75, 171.}, {0.76, 154.75},
 {0.77, 148.25}, {0.78, 138.25}, {0.79, 123.5}, {0.8, 94.75}, {0.81, 101.},
 {0.82, 84.75}, {0.83, 71.}, {0.84, 62.}, {0.85, 64.5}, {0.86, 54.75}, {0.87, 41.75},
 {0.88, 38.5}, {0.89, 27.5}, {0.9, 17.75}, {0.91, 19.75}, {0.92, 14.5}, {0.93, 12.25},
 {0.94, 7.25}, {0.95, 3.75}, {0.96, 2.5}, {0.97, 0.75}, {0.98, 0.}, {0.99, 0.}}
```

In[•]:=
```
Clear[x, y]
y = FindFormula[t0, x, PerformanceGoal → "Quality"]
```

Out[•]=
```
96.2671 + 86.9355 Cos[16.728 x] - 344.777 Log[x] + 592.718 Log[x] Sin[9.2 x]
```

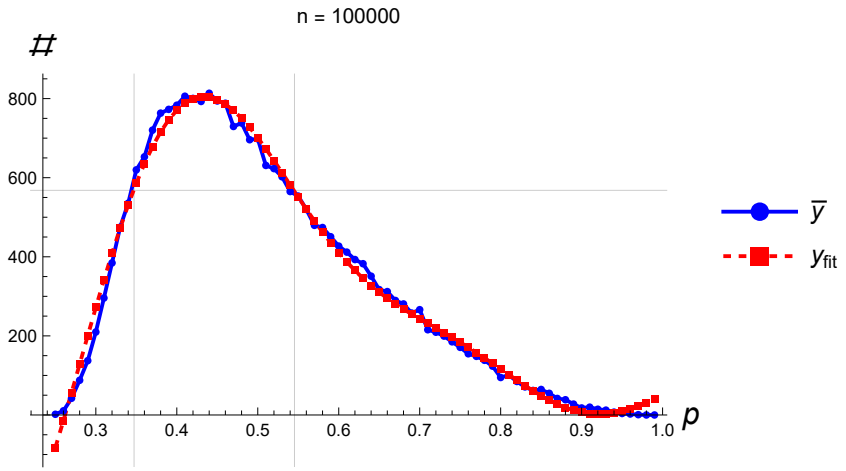
In[•]:=
```
y // TraditionalForm
```

Out[•]//TraditionalForm=

$$-344.777 \log(x) + 86.9355 \cos(16.728\, x) + 592.718 \log(x) \sin(9.2\, x) + 96.2671$$

In[•]:=
```
ys = Table[{x, y}, {x, 0.25, .99, 0.01}];
yGridLines = {{0.3472, 0.545357342}, {568}};
slika008b = ListLinePlot[{t0, ys},
  AxesLabel → {Style["p", Italic, FontFamily → "Arial", FontSize → 16],
    Style["#", Italic, FontFamily → "Arial", FontSize → 16]},
  PlotLabel → "n = " <> ToString[sn], PlotStyle → {{Thick, Blue}, {Thick, Dashed, Red},
    {Thick, Dashed, Green}, {Thick, Dotted, Red}, {Thick, DotDashed, Orange}},
  PlotMarkers → {Automatic, Scaled[0.02]}, GridLines → yGridLines,
  Ticks → {Automatic, Automatic}, PlotLegends → {"y̅", "y_fit"}]
```

Out[•]=



```
(*
Export["slika008b.tif",slika008b,ImageResolution→300];
*)
```

In[•]:=
```
Export["slika008b.tif", slika008b, ImageResolution → 300]
```

Out[•]=

slika008b.tif

In[•]:=
```
100 Integrate[y, {x, 0.25, 0.99}]
```

Out[•]=

25 025.6

In[•]:=
```
yd1 = D[y, x]
yd2 = D[yd1, x]
```

Out[•]=

$$-\frac{344.777}{x}+5453.\,\text{Cos}[9.2\,x]\,\text{Log}[x]+\frac{592.718\,\text{Sin}[9.2\,x]}{x}-1454.25\,\text{Sin}[16.728\,x]$$

Out[•]=

$$\frac{344.777}{x^2}+\frac{10\,906.\,\text{Cos}[9.2\,x]}{x}-24\,326.7\,\text{Cos}[16.728\,x]-\frac{592.718\,\text{Sin}[9.2\,x]}{x^2}-50\,167.6\,\text{Log}[x]\,\text{Sin}[9.2\,x]$$

In[•]:=
```
xZero = FindRoot[yd1, {x, 0.5}]
yMax = y /. xZero
eq = (y == 0.70710 yMax)
x09a = FindRoot[eq, {x, 0.4}];
x09b = FindRoot[eq, {x, 0.6}];
{x09a, x09b}
y /. {x09a, x09b}
100 Integrate[y, {x, (x /. x09a), (x /. x09b)}]
```

Out[•]=

{x → 0.433535}

Out[•]=

804.809

Out[•]=

96.2671 + 86.9355 Cos[16.728 x] − 344.777 Log[x] + 592.718 Log[x] Sin[9.2 x] == 569.081

Out[•]=

{{x → 0.346788}, {x → 0.544074}}

Out[•]=

{569.081, 569.081}

Out[•]=

14 213.

In[•]:=
```
{Total[t0][[2]], Total[t1][[2]], Total[t2][[2]], Total[t3][[2]], Total[t4][[2]],
 Total[t1][[2]] + Total[t2][[2]] + Total[t3][[2]] + Total[t4][[2]]}
```

Out[•]=

{25 000., 24 747, 24 974, 25 299, 24 980, 100 000}

In[•]:=
```
t0[[All, 2]]
```

Out[•]=
```
{1.75, 10.5, 42.25, 87.75, 137.25, 209.75, 295.75, 384.5, 474.75, 535.5, 619.75, 652.75,
 720.25, 763.25, 772.75, 783.25, 806., 800.5, 792.75, 813.5, 794., 788.25, 729.5, 738.25,
 696., 698.25, 631., 623., 602.25, 565., 553.5, 519.75, 479.5, 474., 450.75, 427.,
 411.75, 393., 382.5, 351., 317., 312., 289.5, 280.75, 257.75, 266.25, 215.5, 209.25,
 199.5, 185., 171., 154.75, 148.25, 138.25, 123.5, 94.75, 101., 84.75, 71., 62., 64.5,
 54.75, 41.75, 38.5, 27.5, 17.75, 19.75, 14.5, 12.25, 7.25, 3.75, 2.5, 0.75, 0., 0.}
```

sscnnFigureMatrix

Classic Neural Networks

In[•]:=
```
SeedRandom[123];
W1 = RandomReal[{-1, 1}, {4, 2}]; b1 = RandomReal[{-1, 1}, 4];
W2 = RandomReal[{-1, 1}, {4, 4}]; b2 = RandomReal[{-1, 1}, 4];
W3 = RandomReal[{-1, 1}, {4, 4}]; b3 = RandomReal[{-1, 1}, 4];
(NumberForm[#1, 2] &)[MatrixForm /@ Association[W1 → W1, W2 → W2, W3 → W3]]
(NumberForm[#1, 2] &)[MatrixForm /@ Association[b1 → b1, b2 → b2, b3 → b3]]
```

Out[•]//NumberForm=

$$\left\langle \left| W_1 \to \begin{pmatrix} -0.089 & 0.96 \\ 0.89 & 0.92 \\ -0.4 & -0.067 \\ -0.88 & -0.23 \end{pmatrix}, W_2 \to \begin{pmatrix} -0.44 & -0.82 & 0.75 & -0.78 \\ -0.47 & 0.84 & -0.66 & -0.8 \\ -0.06 & -0.19 & 0.94 & -0.37 \\ -0.75 & -0.46 & 0.21 & 0.34 \end{pmatrix}, \right.$$

$$\left. W_3 \to \begin{pmatrix} -0.037 & 0.29 & -0.16 & 0.41 \\ 0.92 & 0.81 & 0.74 & 0.26 \\ -0.22 & -0.13 & 0.14 & -0.034 \\ -0.68 & 0.93 & -0.91 & -0.0075 \end{pmatrix} \right| \right\rangle$$

Out[•]//NumberForm=

$$\left\langle \left| b_1 \to \begin{pmatrix} -0.14 \\ 0.56 \\ -0.9 \\ 0.26 \end{pmatrix}, b_2 \to \begin{pmatrix} -0.52 \\ -0.016 \\ 0.13 \\ 0.098 \end{pmatrix}, b_3 \to \begin{pmatrix} -0.0082 \\ -0.98 \\ -0.023 \\ -0.84 \end{pmatrix} \right| \right\rangle$$

In[•]:=
```
Tanh[x] // TraditionalForm
```

Out[•]//TraditionalForm=

$$\tanh(x)$$

```
In[•]:= X1 = {1.2, 4.5};
        % // MatrixForm
        W11 = W1;
        % // MatrixForm
        W11 . X1;
        % // MatrixForm
        b1 = b1;
        % // MatrixForm
        W11 . X1 + b1;
        % // MatrixForm
        X11 = Tanh[W11 . X1 + b1];
        % // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 1.2 \\ 4.5 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.0885617 & 0.955652 \\ 0.886429 & 0.924431 \\ -0.395304 & -0.0665829 \\ -0.876723 & -0.228711 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} 4.19416 \\ 5.22366 \\ -0.773988 \\ -2.08127 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.140323 \\ 0.557488 \\ -0.902819 \\ 0.256534 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} 4.05384 \\ 5.78114 \\ -1.67681 \\ -1.82473 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.999398 \\ 0.999981 \\ -0.932446 \\ -0.949308 \end{pmatrix}$$

In[•]:=
```
X1 = X11;
% // MatrixForm
W21 = W2;
% // MatrixForm
W21 . X11;
% // MatrixForm
b2 = b2;
% // MatrixForm
W21 . X11 + b2;
% // MatrixForm
X22 = Tanh[W21 . X11 + b2];
% // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.999398 \\ 0.999981 \\ -0.932446 \\ -0.949308 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.444026 & -0.819565 & 0.753173 & -0.781786 \\ -0.468485 & 0.837219 & -0.660168 & -0.800843 \\ -0.0596039 & -0.193521 & 0.943171 & -0.370142 \\ -0.74844 & -0.45548 & 0.211495 & 0.343527 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -1.22345 \\ 1.74482 \\ -0.781161 \\ -1.72678 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.515198 \\ -0.0163946 \\ 0.125886 \\ 0.0983918 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -1.73864 \\ 1.72842 \\ -0.655275 \\ -1.62839 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.940069 \\ 0.938869 \\ -0.57521 \\ -0.925832 \end{pmatrix}$$

```
In[•]:= X1 = X22;
        % // MatrixForm
        W31 = W3;
        % // MatrixForm
        W31 . X22;
        % // MatrixForm
        b3 = b3;
        % // MatrixForm
        W31 . X22 + b3;
        % // MatrixForm
        X33 = W31 . X22 + b3;
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.940069 \\ 0.938869 \\ -0.57521 \\ -0.925832 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.036781 & 0.288511 & -0.164898 & 0.409043 \\ 0.922663 & 0.807615 & 0.740989 & 0.263322 \\ -0.215215 & -0.127425 & 0.136959 & -0.0341827 \\ -0.67907 & 0.926666 & -0.907707 & -0.00747479 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.0215967 \\ -0.779139 \\ 0.035548 \\ 2.03743 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.00815844 \\ -0.975298 \\ -0.0227348 \\ -0.842961 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.0134383 \\ -1.75444 \\ 0.0128131 \\ 1.19447 \end{pmatrix}$$

```
In[•]:= Exp[X33] / Total[Exp[X33]]
        % // MatrixForm
```

Out[•]=

{0.184236, 0.0314483, 0.184121, 0.600195}

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.184236 \\ 0.0314483 \\ 0.184121 \\ 0.600195 \end{pmatrix}$$

```
In[•]:= softmax[x_] := Exp[x] / Total[Exp[x]];

In[•]:= softmax[W3.Tanh[W2.Tanh[W1.{1.2, 4.5} + b1] + b2] + b3]
```

Out[•]=

{0.184236, 0.0314483, 0.184121, 0.600195}

```
In[•]:= Exp[x] / Total[Exp[x]] // TraditionalForm
```

Out[•]//TraditionalForm=

$$\frac{e^{x}}{\text{Total}[e^{x}]}$$

$$\left\langle \left| \mathbf{b}_1 \to \begin{pmatrix} -0.14 \\ 0.56 \\ -0.9 \\ 0.26 \end{pmatrix}, \mathbf{b}_2 \to \begin{pmatrix} -0.52 \\ -0.016 \\ 0.13 \\ 0.098 \end{pmatrix}, \mathbf{b}_3 \to \begin{pmatrix} -0.0082 \\ -0.98 \\ -0.023 \\ -0.84 \end{pmatrix} \right| \right\rangle$$

```
In[•]:= {{-0.089, 0.96}, {0.89, 0.92}, {-0.4, -0.067}, {-0.88, -0.23}} // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.0885617 & 0.955652 \\ 0.886429 & 0.924431 \\ -0.395304 & -0.0665829 \\ -0.876723 & -0.228711 \end{pmatrix}$$

```
In[•]:= {1.2, 4.5, 0, 0} // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 1.2 \\ 4.5 \\ 0 \\ 0 \end{pmatrix}$$

Out[•]//NumberForm=

$$\left\langle \left| \mathbf{W}_1 \to \begin{pmatrix} -0.089 & 0.96 \\ 0.89 & 0.92 \\ -0.4 & -0.067 \\ -0.88 & -0.23 \end{pmatrix}, \mathbf{W}_2 \to \begin{pmatrix} -0.44 & -0.82 & 0.75 & -0.78 \\ -0.47 & 0.84 & -0.66 & -0.8 \\ -0.06 & -0.19 & 0.94 & -0.37 \\ -0.75 & -0.46 & 0.21 & 0.34 \end{pmatrix}, \right.$$

$$\left. \mathbf{W}_3 \to \begin{pmatrix} -0.037 & 0.29 & -0.16 & 0.41 \\ 0.92 & 0.81 & 0.74 & 0.26 \\ -0.22 & -0.13 & 0.14 & -0.034 \\ -0.68 & 0.93 & -0.91 & -0.0075 \end{pmatrix} \right| \right\rangle$$

In[•]:=
```
W1 // MatrixForm
W2 // MatrixForm
W3 // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.0885617 & 0.955652 \\ 0.886429 & 0.924431 \\ -0.395304 & -0.0665829 \\ -0.876723 & -0.228711 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.444026 & -0.819565 & 0.753173 & -0.781786 \\ -0.468485 & 0.837219 & -0.660168 & -0.800843 \\ -0.0596039 & -0.193521 & 0.943171 & -0.370142 \\ -0.74844 & -0.45548 & 0.211495 & 0.343527 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.036781 & 0.288511 & -0.164898 & 0.409043 \\ 0.922663 & 0.807615 & 0.740989 & 0.263322 \\ -0.215215 & -0.127425 & 0.136959 & -0.0341827 \\ -0.67907 & 0.926666 & -0.907707 & -0.00747479 \end{pmatrix}$$

In[•]:=
```
b1 // MatrixForm
b2 // MatrixForm
b3 // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.140323 \\ 0.557488 \\ -0.902819 \\ 0.256534 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.515198 \\ -0.0163946 \\ 0.125886 \\ 0.0983918 \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.00815844 \\ -0.975298 \\ -0.0227348 \\ -0.842961 \end{pmatrix}$$

---

sscnnKnowledgeImp00

```
(* SSknowledgeImp00.m *)
(* << SSknowledgeImp00` *)

Clear[DiscreteSystemImplementationEquations2, DiscreteSystemImplementationSummary2, DiscreteSy

(*----------DiscreteSystemImplementationEquations2 --------*)
Options[DiscreteSystemImplementationEquations2] = {PrintFloatingPorts \
→ False, Verbose → False};
DiscreteSystemImplementationEquations2[
  schematicSpec : {{"Adder" | "Arrow" | "Delay" | "Function" |
       "Input" | "Line" | "Modulator" | "Multiplier" | "Node" |
```

```
      "Output" | "Polyline" |
      "Text", {_?NumberQ, _?
        NumberQ} | {{_?NumberQ, _?
         NumberQ}, {_?NumberQ, _?NumberQ} ..}, ___} ..},
 Y_Symbol : Y, opts___?OptionQ] :=
Module[{a, analysisElementList, countImplementationElements,
  duplicateCoords, duplicateFlag, elementDelay, elementDelayCoordsIn,
   elementDelayCoordsOut, elementDelayVectorIn,
  elementDelayVectorOut, elementInput, elementInputCoords,
  elementInputVector, elementOutput, elementOutputCoords,
  elementOutputVector, implementationElementList,
  implementationEquations, implementationVars, implementationVarsRaw,
   k, knownCoords, linePosition, lineRule, nodeCoords, nodeCoordsRaw,
   numberImplementationElements, outputCoords, outputCoordsRaw, pfp,
  q, response, signalCoords, signalVector, systemEquations, t,
  topologyElementList, topologyElementListRaw, v, verb, vGain,
  vInitial, V},
 {pfp, verb} = {PrintFloatingPorts, Verbose} /. Flatten[{opts}] /.
   Options[DiscreteSystemImplementationEquations2];
 topologyElementListRaw =
  Apply[NonlinearDiscreteElementTopology, schematicSpec, {1}];
 topologyElementList =
  Select[topologyElementListRaw, Length[#] ≠ 0 &];
 linePosition =
  Position[topologyElementList, {"Line", __}] // Flatten;
 Do[lineRule = {topologyElementList⟦linePosition⟦k⟧, 2⟧ →
     topologyElementList⟦linePosition⟦k⟧, 3⟧};
  t = topologyElementList //. lineRule;
  topologyElementList = t;, {k, Length[linePosition]}];
 elementInput = Select[topologyElementList, #⟦1⟧ == "Input" &];
 elementInputCoords = #⟦2⟧ & /@ elementInput;
 elementInputVector = Y /@ elementInputCoords;
 elementOutput = Select[topologyElementList, #⟦1⟧ == "Output" &];
 elementOutputCoords = #⟦2⟧ & /@ elementOutput;
 elementOutputVector = Y /@ elementOutputCoords;
 elementDelay = Select[topologyElementList, #⟦1⟧ == "Delay" &];
 elementDelayCoordsOut = #⟦2⟧ & /@ elementDelay;
 elementDelayVectorOut = Y /@ elementDelayCoordsOut;
 elementDelayCoordsIn = #⟦3⟧ & /@ elementDelay;
 elementDelayVectorIn = Y /@ elementDelayCoordsIn;
 analysisElementList =
  Select[topologyElementList, ((#⟦1⟧ ≠ "Line") && (#⟦1⟧ ≠
        "Output")) &];
 If[verb,
  Print["DiscreteSystemImplementationEquations ver. 2007.7.7"];
  Print["Schematic specification = ", schematicSpec];
  Print["Signals at nodes are designated by the symbol ", Y];
  Print["Topology element list (raw) = ", topologyElementListRaw];
  Print["Topology element list = ", topologyElementList];
  Print["Line-element position = ", linePosition];
  Print["Input elements = ", elementInput];
  Print["Input-element coordinates = ", elementInputCoords];
  Print["Input-element vector = ", elementInputVector];
  Print["Output elements = ", elementOutput];
  Print["Output-element coordinates = ", elementOutputCoords];
```

```
 Print["Output-element vector = ", elementOutputVector];
 Print["Delay elements = ", elementDelay];
 Print["Delay-element output coordinates = ", elementDelayCoordsOut];
 Print["Delay-element output vector = ", elementDelayVectorOut];
 Print["Delay-element input coordinates = ", elementDelayCoordsIn];
 Print["Delay-element input vector = ", elementDelayVectorIn];
 Print["Analysis elements = ", analysisElementList];];
If[analysisElementList == {},
 Message[DiscreteSystemImplementationEquations2::"noelems"];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[elementInput == {},
 Message[DiscreteSystemImplementationEquations2::"noinputs"];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[Length[elementInput] ≠ Length[elementInputCoords // Union],
 Message[DiscreteSystemImplementationEquations2::"duplin"];
 Print["Here are all Input elements: ",
  Sort[elementInput] // Column];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[elementOutput == {},
 Message[DiscreteSystemImplementationEquations2::"nooutputs"];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[Length[elementOutput] ≠ Length[elementOutputCoords // Union],
 Message[DiscreteSystemImplementationEquations2::"duplout"];
 Print["Here are all Output elements: ",
  Sort[elementOutput] // Column];
 Return[{{}, {}, {}, {}, {}, {}}]];
nodeCoordsRaw =
 If[#[[1]] === "Adder" || #[[1]] === "Modulator",
    Prepend[#[[3]], #[[2]], {#[[2]], #[[3]]}] & /@
  analysisElementList;
nodeCoords = Select[Flatten[nodeCoordsRaw, 1], Length[#] ≠ 0 &];
signalCoords = Sort[Union[nodeCoords]];
signalVector = Y /@ signalCoords;
outputCoordsRaw = #[[2]] & /@ analysisElementList;
outputCoords = Sort[Union[outputCoordsRaw]];
outputVector = Y /@ outputCoords;
duplicateCoords = Count[outputCoordsRaw, #] & /@ outputCoords;
duplicateFlag = (If[#1 > 1,
     Message[DiscreteSystemImplementationEquations2::"outputs",
      outputCoords[[First[Flatten[Position[duplicateCoords, #1]]]]]];
     True, False]) & /@ duplicateCoords;
If[verb,(*Print["Node coordinates (raw) = ",nodeCoordsRaw];*)
 Print["Node coordinates = ", nodeCoords];
 Print["Signal coordinates = ", signalCoords];
 Print["Signal vector = ", signalVector];
 Print["Output-port coordinates (raw) = ", outputCoordsRaw];
 Print["Output-port coordinates = ", outputCoords];
 Print["Output-port vector = ", outputVector];
 Print["Duplicate output count = ", duplicateCoords];
 Print["Duplicate output flag = ", duplicateFlag];
 Print["System variables = ", outputVector];];
If[Count[duplicateFlag, True] > 0,
 Message[DiscreteSystemImplementationEquations2::"intercon"];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[Not[signalVector === outputVector],
```

```
 Print["Unconnected nodes exist: ",
  Complement[signalVector, outputVector]];
 Message[DiscreteSystemImplementationEquations2::"intercon"];
 Return[{{}, {}, {}, {}, {}, {}}]];
If[Not[
  Intersection[signalVector, elementOutputVector] ==
   Sort[elementOutputVector]],
 Print["Unconnected system outputs exist: ",
  Complement[elementOutputVector, signalVector]];
 Message[DiscreteSystemImplementationEquations2::"intercon"];
 Return[{{}, {}, {}, {}, {}, {}}]];
systemEquations = {};
Do[a = analysisElementList[[k]]; v = Last[a];
 If[MatchQ[v, {_, _}] === True, {vGain, vInitial} =
   v, {vGain, vInitial} = {v, 0}];
 Switch[First[a], "Adder",
  AppendTo[systemEquations,
    Y[a[[2]]] ==
     Y[a[[3, 1]]]*v[[1]] + Y[a[[3, 2]]]*v[[2]] +
      Y[a[[3, 3]]]*v[[3]]];, "Delay",
  AppendTo[systemEquations,
    Y[a[[2]]] == previousSample[Y[a[[3]]]]];, "Function",
  AppendTo[systemEquations, Y[a[[2]]] == v[Y[a[[3]]]]];, "Input",
  AppendTo[systemEquations, Y[a[[2]]] == v];, "Modulator",
  AppendTo[systemEquations,
    Y[a[[2]]] ==
     If[v[[1]] == 0, 1, Y[a[[3, 1]]]]*
      If[v[[2]] == 0, 1, Y[a[[3, 2]]]]*
      If[v[[3]] == 0, 1, Y[a[[3, 3]]]]];, "Multiplier",
  AppendTo[systemEquations, Y[a[[2]]] == Y[a[[3]]]*v];, _,
  Message[DiscreteSystemImplementationEquations2::"unsup",
    a // InputForm];] (*end Switch*), {k,
  Length[analysisElementList]}];
If[verb, Print["System equations = ", systemEquations // Column];];
If[pfp || verb,
 If[signalVector === outputVector, Print["No floating ports."],
   Print["Floating ports = ",
     Complement[signalVector, outputVector]];];];
implementationElementList =
 Select[analysisElementList, ((#[[1]] ≠ "Delay") && (#[[1]] ≠
       "Input")) &];
implementationVarsRaw =
 Variables[Part[implementationElementList, All, 4]];
(*exclude names of symbols that have been defined*)
t = UndefinedSymbolQ[#] & /@ implementationVarsRaw;
implementationVars = Select[t, (# =!= False) &];
implementationEquations = {};
Do[a = elementInput[[k]]; v = Last[a];
 AppendTo[implementationEquations, Y[a[[2]]] == v];, {k,
  Length[elementInput]}];
Do[a = elementDelay[[k]]; v = Last[a];
 AppendTo[implementationEquations,
  Y[a[[2]]] == previousSample[Y[a[[3]]]]];, {k,
  Length[elementDelay]}];
knownCoords = Join[elementInputCoords, elementDelayCoordsOut];
```

```
  numberImplementationElements = Length[implementationElementList];
  Do[(*body Do*)
   countImplementationElements = Length[implementationElementList];
   Do[a = implementationElementList[[k]]; v = Last[a];
    Switch[First[a], "Adder",
     If[Complement[Select[a[[3]], # ≠ {} &],
         Intersection[Select[a[[3]], # ≠ {} &],
          knownCoords]] == {},(*then*)
       AppendTo[implementationEquations,
        Y[a[[2]]] ==
         Y[a[[3, 1]]]*v[[1]] + Y[a[[3, 2]]]*v[[2]] +
          Y[a[[3, 3]]]*v[[3]]];
       AppendTo[knownCoords, a[[2]]];
       t = Delete[implementationElementList, k];
       implementationElementList = t;
       Break[];,(*else*)False;];, "Function",
     If[Complement[{a[[3]]},
         Intersection[{a[[3]]}, knownCoords]] == {},(*then*)
       AppendTo[implementationEquations, Y[a[[2]]] == v[Y[a[[3]]]]];
       AppendTo[knownCoords, a[[2]]];
       t = Delete[implementationElementList, k];
       implementationElementList = t; Break[];,(*else*)False;];,
     "Modulator",
     If[Complement[Select[a[[3]], # ≠ {} &],
         Intersection[Select[a[[3]], # ≠ {} &],
          knownCoords]] == {},(*then*)
       AppendTo[implementationEquations,
        Y[a[[2]]] ==
         If[v[[1]] == 0, 1, Y[a[[3, 1]]]]*
          If[v[[2]] == 0, 1, Y[a[[3, 2]]]]*
          If[v[[3]] == 0, 1, Y[a[[3, 3]]]]];
       AppendTo[knownCoords, a[[2]]];
       t = Delete[implementationElementList, k];
       implementationElementList = t;
       Break[];,(*else*)False;];, "Multiplier",
     If[Complement[{a[[3]]},
         Intersection[{a[[3]]}, knownCoords]] == {},(*then*)
       AppendTo[implementationEquations, Y[a[[2]]] == Y[a[[3]]]*v];
       AppendTo[knownCoords, a[[2]]];
       t = Delete[implementationElementList, k];
       implementationElementList = t; Break[];,(*else*)False;];, _,
     Message[DiscreteSystemImplementationEquations2::"unsup",
       a // InputForm];] (*end Switch*), {k,
     countImplementationElements}];, \
{numberImplementationElements}];(*end Do*)
  If[verb,
   Print["Implementation elements = ", implementationElementList];
   Print["Number of implementation elements = ",
    numberImplementationElements];
   Print["Implementation parameters (raw) = ",
    implementationVarsRaw];
   Print["Implementation parameters = ", implementationVars];
   Print["Known coordinates = ", knownCoords];
   Print["Implementation equations = ",
    implementationEquations // Column];
```

```
   Print["Not-Implemented elements = ",
    implementationElementList // Column];];
  If[implementationElementList == {} (*then*),
   response = {elementInputVector, elementDelayVectorOut,
      implementationVars, implementationEquations,
      elementOutputVector, elementDelayVectorIn};,(*else*)
   response = {{}, {}, {}, {}, {}, {}};
   Message[DiscreteSystemImplementationEquations2::"algloop"];];
  response]

DiscreteSystemImplementationEquations2[___] :=
  Module[{},
   Message[DiscreteSystemImplementationEquations2::"invargs"];
   {{}, {}, {}, {}, {}, {}}];

(*----------DiscreteSystemImplementationSummary2----------*)
Options[DiscreteSystemImplementationSummary2] = {Verbose → False};
DiscreteSystemImplementationSummary2[schematicSpec_List,
   opts___?OptionQ] :=
  Module[{verb, equations},
   equations = DiscreteSystemImplementationEquations2[schematicSpec];
   If[Flatten[equations] == {}, Return[]];
   verb =
    Verbose /. Flatten[{opts}] /.
     Options[DiscreteSystemImplementationSummary2];
   Print["Input: ", equations // First];
   Print["Initial state: ", equations[[2]]];
   Print["Parameter: ", equations[[3]]];
   If[verb,(*then*)Print["Equations: ", equations[[4]] // Column];];
   Print["Output: ", equations[[5]]];
   Print["Final state: ", equations // Last];];

DiscreteSystemImplementationSummary2[_] :=
  Module[{},
   Message[DiscreteSystemImplementationSummary2::"invargs"];];

DiscreteSystemImplementationSummary2[a___] :=
  Module[{n}, n = Length[{a}];
   Message[DiscreteSystemImplementationSummary2::"argx", n, 1];];

(*----------DiscreteSystemImplementation2----------*)
DiscreteSystemImplementation2[schematicSpec_List,
   procedureName_String : "implementationProcedure"] :=
  Module[{schematicEquations, codeString},
   schematicEquations =
    DiscreteSystemImplementationEquations2[schematicSpec];
   If[Flatten[schematicEquations] == {}, Return[]];
   codeString =
    DiscreteSystemImplementationModule2[schematicEquations,
     procedureName, Verbose → True]];

DiscreteSystemImplementation2[___] :=
  Module[{}, Message[DiscreteSystemImplementation2::"invargs"];];

(*----------DiscreteSystemImplementationModule2----------*)
```

```
Options[DiscreteSystemImplementationModule2] = {Verbose → False};
DiscreteSystemImplementationModule2[
   r : {_List, _List, {_Symbol ...}, {_Equal ..}, _List, _List},
   procedureName_String : "implementationProcedure",
   opts___?OptionQ] :=
  Module[{eqnsL, negCoords, x, y, dims, dimsName, verb, signalName,
    signalNameStr, ruleYxpy, dataVars, dataVarsStr, dataEqns,
    initVars, initVarsStr, initEqns, localVars, localVarsStr, params,
    bodyEqns, bodyVars, bodyStr, bodyRepl, outArgs, outVars, outStr,
    inputArguments, comma, quote, usageString,
    implementationString},(*r is returned by DSIE,
   procedureName is a string*)
   If[Not[ValidImplementationModuleNameQ[procedureName]],
    Message[DiscreteSystemImplementationModule2::"invname",
     procedureName];
    Return[""]];
   (*check that procedureName is not a built-in function*)
   If[Names["System`"~StringJoin~procedureName] == {procedureName},
    Message[DiscreteSystemImplementationModule2::"sysname",
     procedureName];
    Return[""]];
   (*check for negative coords*)
   eqnsL = Part[r⟦4⟧, All, 1] /. h_[{x_, y_}] → {x, y};
   negCoords =
    Select[eqnsL, (Negative[First[#]] || Negative[Last[#]]) &];
   If[negCoords ≠ {},
    Message[DiscreteSystemImplementationModule2::"negcoords",
     negCoords];
    Return[""]];
   (*dimension info as procName[] call*)dims = Length[#] & /@ r;
   dimsName =
    StringJoin[procedureName, "[] := ", ToString[dims], "; "];
   ToExpression[dimsName];
   verb =
    Verbose /. Flatten[{opts}] /.
     Options[DiscreteSystemImplementationModule2];
   signalName = First[r] // First // Head;
   signalNameStr = signalName // ToString;
   ruleYxpy :=
    signalName[{x_, y_}] :↦
     ToExpression[
      StringJoin[signalNameStr, ToString[x], "p", ToString[y]]];
   dataVars = r⟦1⟧;
   dataVarsStr = ToString[dataVars /. ruleYxpy];
   dataEqns =
    StringJoin[" ", dataVarsStr, " = ", "dataSamples", ";"];
   initVars = r⟦2⟧;
   initVarsStr = ToString[initVars /. ruleYxpy];
   If[r⟦2⟧ == {}, initEqns = " "; (*else*),
    initEqns =
     StringJoin[" ", initVarsStr, " = ", "initialConditions",
      ";"];];
   localVars = Part[r⟦4⟧, All, 1]~Join~r⟦3⟧;
   localVarsStr = ToString[localVars /. ruleYxpy];
   If[r⟦3⟧ == {}, params = " "; (*else*),
```

```
    params =
      StringJoin[" ", ToString[r⟦3⟧], " = ",
       "systemParameters;"];];
   bodyEqns = Drop[r⟦4⟧, Length[r⟦1⟧] + Length[r⟦2⟧]];
   bodyVars = bodyEqns /. ruleYxpy;
   bodyStr = ToString[bodyVars // InputForm];
   bodyRepl =
    StringReplace[
     bodyStr, {"{" → " ", "==" → "=", "}" → ";", "," → ";"}];
   outArgs = r⟦{5, 6}⟧;
   outVars = outArgs /. ruleYxpy;
   outStr = StringJoin[" ", ToString[outVars]];
   inputArguments =
    "[dataSamples_List, initialConditions_List, \
systemParameters_List]";
   comma = ",";
   quote = "\"";
   usageString =
    StringJoin[procedureName, "::usage = ", quote, outStr, " = ",
     procedureName, "[", dataVarsStr, comma, initVarsStr, comma,
     ToString[r⟦3⟧], "]",
     " is the template for calling the procedure. ",
     " The general template is ",
     "{outputSamples, finalConditions} = ",
     "procedureName[inputSamples, initialConditions, \
systemParameters]. ",
     "See also: DiscreteSystemImplementationProcessing2", quote, ";",
     " "];
   implementationString =
    StringJoin[procedureName, inputArguments, " := ", "Module", "[ ",
     localVarsStr, comma, params, dataEqns, initEqns, bodyRepl,
     outStr, " ];"];
   (*activate implementation code*)
   ToExpression[usageString~StringJoin~implementationString];
   If[verb,(*then*)
    Print["Implementation procedure name: ", procedureName];
    Print["Implementation procedure usage: "];
    ToExpression["?"~StringJoin~procedureName];];
   StringJoin[usageString, dimsName, implementationString]];

DiscreteSystemImplementationModule2[x_] :=
  Module[{}, Message[DiscreteSystemImplementationModule2::"invargs"];
   ""];

DiscreteSystemImplementationModule2[x_, y_] :=
  Module[{}, Message[DiscreteSystemImplementationModule2::"invargs"];
   ""];

DiscreteSystemImplementationModule2[a___] :=
  Module[{n}, n = Length[{a}];
   Message[DiscreteSystemImplementationModule2::"argx", n];
   ""];

(*----------DiscreteSystemImplementationProcessing2----------*)
DiscreteSystemImplementationProcessing2[inputSequence_List?MatrixQ,
```

```
   initialConditions_List, systemParameters_List?VectorQ,
   procedureName_Symbol] :=
  Module[{dimsName, dimsValue, finalConditions = initialConditions, n,
    outputSequence = {},
   response},(*check that procedureName is not a built-in function*)
   If[Names[
      "System`"~StringJoin~ToString[procedureName]] == {ToString[
       procedureName]},
    Message[DiscreteSystemImplementationProcessing2::"sysname",
     procedureName];
    Return[{{}, {}}]];
   dimsName = ToString[procedureName]~StringJoin~"[]";
   dimsValue = ToExpression[dimsName];
   If[MatchQ[dimsValue, _List?VectorQ] === False,
    Message[DiscreteSystemImplementationProcessing2::"invdims",
     procedureName, dimsValue];
    Return[{{}, {}}];];
   If[Length[First[inputSequence]] ≠ dimsValue[[1]],
    Message[DiscreteSystemImplementationProcessing2::"invinp",
     Length[First[inputSequence]], dimsValue[[1]]];
    Return[{{}, {}}];];
   If[Length[initialConditions] ≠ dimsValue[[2]],
    Message[DiscreteSystemImplementationProcessing2::"invcond",
     Length[initialConditions], dimsValue[[2]]];
    Return[{{}, {}}];];
   If[Length[systemParameters] ≠ dimsValue[[3]],
    Message[DiscreteSystemImplementationProcessing2::"invparam",
     Length[systemParameters], dimsValue[[3]]];
    Return[{{}, {}}];];
   Do[response =
     procedureName[inputSequence[[n]], finalConditions,
      systemParameters];
    AppendTo[outputSequence, First[response]];
    finalConditions = Last[response];, {n, Length[inputSequence]}];
   {outputSequence, finalConditions}];

DiscreteSystemImplementationProcessing2[_, _, _, _] :=
  Module[{},
   Message[DiscreteSystemImplementationProcessing2::"invargs"];
   {{}, {}}];

DiscreteSystemImplementationProcessing2[a___] :=
  Module[{n}, n = Length[{a}];
   Message[DiscreteSystemImplementationProcessing2::"argx", n, 4];
   {{}, {}}];
```