INTERNATIONAL STANDARD

IEC 61523-2

First edition
2002-05

Delay and power calculation standards –

Part 2:
Pre-layout delay calculation specification for CMOS ASIC libraries

Reference number
IEC 61523-2:2002(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/catlg-e.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/JP.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 61523-2

First edition
2002-05

Delay and power calculation standards –

Part 2:
pre-layout delay calculation specification for CMOS ASIC libraries

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE W

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

DELAY AND POWER CALCULATION STANDARDS –

Part 2: Pre-layout delay calculation specification for CMOS ASIC libraries

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61523-2 has been prepared by IEC technical committee 93: Design automation.

The ASIC Library Representation Working Group of EIAL EDA Technical Committee also participated in the preparation of this standard.

This standard is a revision of the EIAJ[1] document: *ASIC Library Representation (ALR):1994.*

The text of this standard is based on the following documents:

FDIS	Report on voting
93/151/FDIS	93/153/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This standard does not follow the rules for the structure of international standards given in Part 2 of the ISO/IEC Directives.

NOTE This standard has been reproduced without significant modification of its original content or drafting.

[1] Electronic Industries Association of Japan.

IEC 61523 consists of the following parts, under the general title: *Delay and calculation standards:*

IEC 61523-1:2001, Part 1: *Integrated circuit delay and power calculation systems*

IEC 61523-2, Part 2: *Pre-layout delay calculation specification for CMOS ASIC libraries*

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

DELAY AND CALCULATION STANDARDS –

Part 2: Pre-layout delay calculation specification for CMOS ASIC libraries

1. **Scope and object**

This standard specifies the pre-layout delay calculation method for CMOS1[1)]ASIC[2)] Libraries which contains cell based primitives and memories to be used during the pre-layout design phase of Logic simulation, Timing verification, and Logic synthesis.
The delay calculation method addressed in this standard consists of
1) Estimation of wire capacitance and
2) Delay calculation method based on tablelook-up.
With use of DCL and SDF, this delay calculation method helps the user have a unified timing model for various EDA tools in the pre-layout design phase.

This standard is consistent with existing standards and accepts existing standard formats, like SPEF, DCL, and SDF.
Scope of this standard covers the CMOS ASIC front end timing design for using logic synthesizer, simulators, timing verifiers.
The delay calculation method specified is based on the input slew rate calculation step and the port to port calculation step.
During these calculation steps, the table lookup method is used.
The table method of this standard specifies two interpolation methods for delay calculation. One is bi-linear interpolation which is widely used through the industry. Another is a linear interpolation using neighboring 3 points.
The nature of the delay value has monotonously increasing function of convex surface. This linear interpolation has a few percent of differences between linear interpolation and SPICE result.

2. **Normative references**

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

IEEE Standard 1481:1999, *Integrated Circuit (IC) Delay and Power Calculation System.*

---

1) Complementary metal oxide semiconductor (CMOS).
2) Application – Specific Integrated Circuits (ASIC).

3. **Relations with other companion standards activities**

The input to the delay calculator are net list and library.

The net list is described on either Verilog or VHDL.

The library consists of a functional part and a delay parameter part.

The functional part of the library is covered by Verilog or VHDL.

NOTE The delay parameter part of the library has not been standardized , because it depends strongly on the delay calculation method. EIAJ/ALR version1.1 described the delay calculation method , and the delay calculation method of EIAJ/ALR version1.1 is represented by DCL and DCL-PI standard(IEEE 1481).

This part of IEC 61523 specifies in detail a table look up calculation formula for CMOS ASIC library[1)].

The output of the delay calculator is a Standard Delay Format (SDF).

4. **Terms and Definitions**

capacitance of the net: Net means equipotential signal pins which will be connected by routing. The capacitance of the net is the capacitance of all the signal pins tha tare connected by routing.

CMOS: Complementary Metal Oxide Semiconductor.

DCL: Delay Calculation Language.

front end design: Logical design phase of ASIC design. Back end design means layout design.

gate: module containing only one output which is a simple boolean function of its inputs. Some basic simple boolean functions are and/or/not.

input slew rate: Slope of the input signal of the gate.In CMOS,the gate output delay is the function of its input slope of the signal.

load capacitance: Capacitance driven by gate. Usually it is separated into two items: i.e. wiring load capacitance and the sum of input load capacitance.

logic synthesizer: CAD package function performing the translation from RTL-level descriptions to Gate-level descriptions.

port to port delay: One meaning is pin to pin delay inside of gate. The other is pin to pin delay between gates.

pre-layout: Design phase before layout, i.e.logical design phase.

propagation delay: Traveling time of a given edge of a signal. Usually it is separated into two items: i.e.propagation delay inside a gate and propagation

1) This is not defined in EIAS/ARL version 1.1.

delay from the output of a gate to the input of another gate which is driven by it.

SDF: Standard Delay Format.

simulator: CAD package function of the circuit simulator based on behavior, network, and stimulus. There are Digital and Analog Simulators.

SPEF: Standard Parasitic Exchangeable Format.

SPICE: Simulation Program similar to the program with the same name developed at UC Berkeley. The simulation results are in terms of continuous waveforms representing current or voltage. It emphasizes Integrated Circuit timing and waveforms.

timing verifier: CAD package function that checks register to register timing violations

of setup and hold time. Network description and clock timing are necessary.

transient timing group: Group of signal values. The delay will be definedwhen the signal changes from one value to the other.

5. **Pre-layout delay calculation method for CMOS ASIC libraries**

Timing design is the critical issue in sub-micron CMOS ASIC .This clause specifies the detail pre-layout delay calculation.

5.1 **Delay model**

When considering a sub-micron pre-layout (capacitance-based) timing model, two items should be considered, (1)port to port delay timing and (2) input slew rate effect.

In this model, it is necessary to first calculate the capacitance of wires, and then, calculate delays. As a first step of delay calculation, input slew rate is calculated, nd then, port to port delay can be calculated by using the input slew rate.
Therefore, a two step approach is necessary as shown in Figure 1.

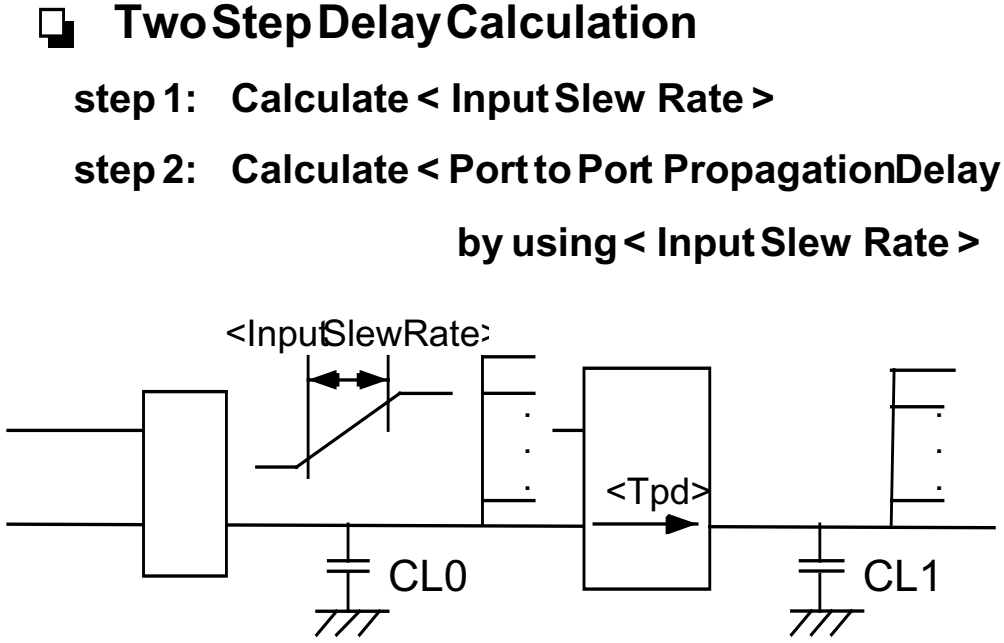


Figure 1: CMOS Delay Model & Calculation Steps

5.2. **Table Look-Up delay calculation method**

The table look-up model of delay calculation specification uses 3 types of table models. First is the 'Net capacitance table' (named Cn table). This table is used for 'load capacitance' estimation. Second is 'Input slew rate table' (named Ts table), and 3rd is 'Port to port propagation delay time table' (named Tpd table). As a first step of delay calculation, input slew rate is calculated by using net capacitance, and Ts table. And then, port to port propagation delay can be calculated by using net capacitance, input slew rate, and Tpd table.

The propagation delay is calculated using a Tpd table by applying one of the methods for interpolation approximation. One is bilinear interpolation approximation by 4 points. This method will be de facto standard from major EDA vendors. The other is linear interpolation approximation by 3 points. This approximation is more accurate than bilinear interpolation, and both linear and bilinear methods can use the same Tpd table.

5.2.1 **Load capacitance estimation**

First step is to estimate the load capacitance of each net. Load capacitance is estimated by the following rule.

Load Capacitance = (Input port capacitance) + estimated net capacitanc

where (Input port capacitance) is the summation of input port capacitance

in the net

The estimated capacitance is a function of fanout and estimated size which is calculated by summing up the cell size of all cells in the top hierarchy to which the net belongs. So, to estimate capacitance, a two dimensional table is used. Indices of the table are fanout and sum of cell size. Different tables should be prepared according to chip size(standard cell) or type of base array(gate array). As shown in figure 2, each net capacitance is calculated by step interpolation using Cn table.

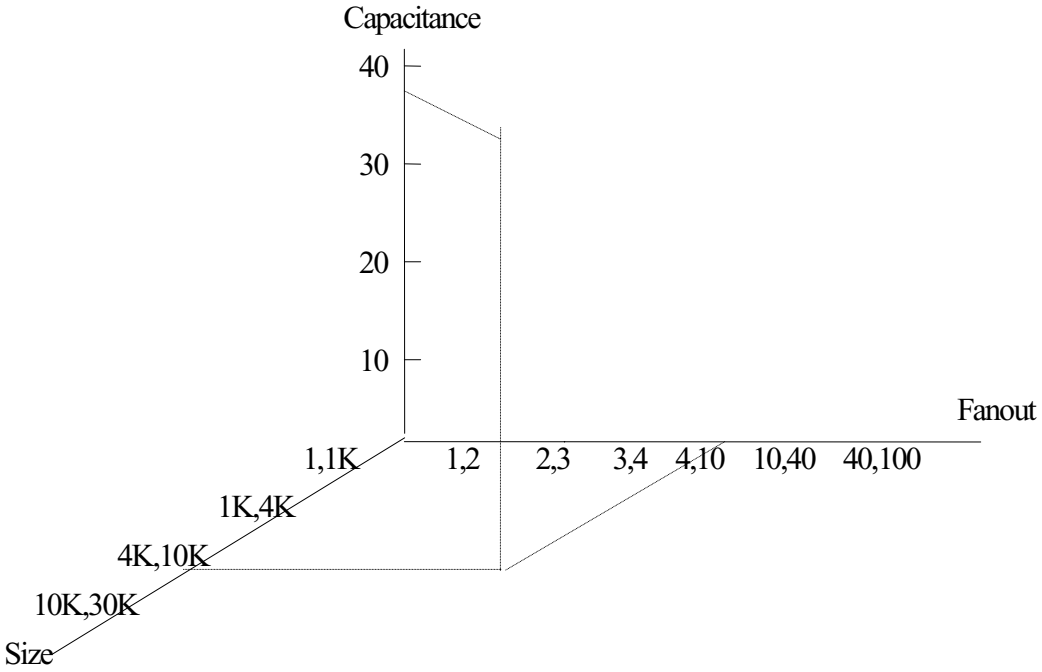


Figure 2. Example of net capacitance estimation

5.2.1.1 **Cn table specification**

Cn table is a 2 dimensional matrix specified for each design methodology, i.e. gate array, standard cell.(See Annex F.1)
The 1st index is size(S[i]): sum of cell size, or sum of number of gates, or base array size for gate array.
The 2nd index(Fo[j]) is the number of fanout in net.
The value(C[i][j]) is pre-defined capacitance value.
where
1 i M (M is effective maximum number of size values used),
1 j N (N is effective maximum number of fanout values used).
C[i][j] has a real value.

Values of S[i] and Fo[j] have 2 integer values each, which are in the following relationship;
0 first value second value,
second value of S[i] = first value of S[i+1],

second value of Fo[j] = first value of Fo[j+1],
first value effective value second value,
Cn[i][j] is real value, unit is pF or fF.

5.2.1.2 **Net capacitance(CnE) estimation rule**

The Net capacitance estimation (CnE) is specified below.
If the size is ranged in S[i] and the fanout is ranged in Fo[j]
then
CnE = Cn[i][j] . .....(1)

If the size is less than the first value of S[1], then set i to 1.
If the size is greater than or equal to the second vale of S[M], then set i to M.
If the fanout is less than the first value of Fo[1], then set j to 1.
If the fanout is greater than or equal to the second value of Fo[N],
then set j to N.
Then apply it to function (1).

5.2.2 **Input slew rate calculation**

Input slew rate is calculated by linear interpolation shown in figure 3.

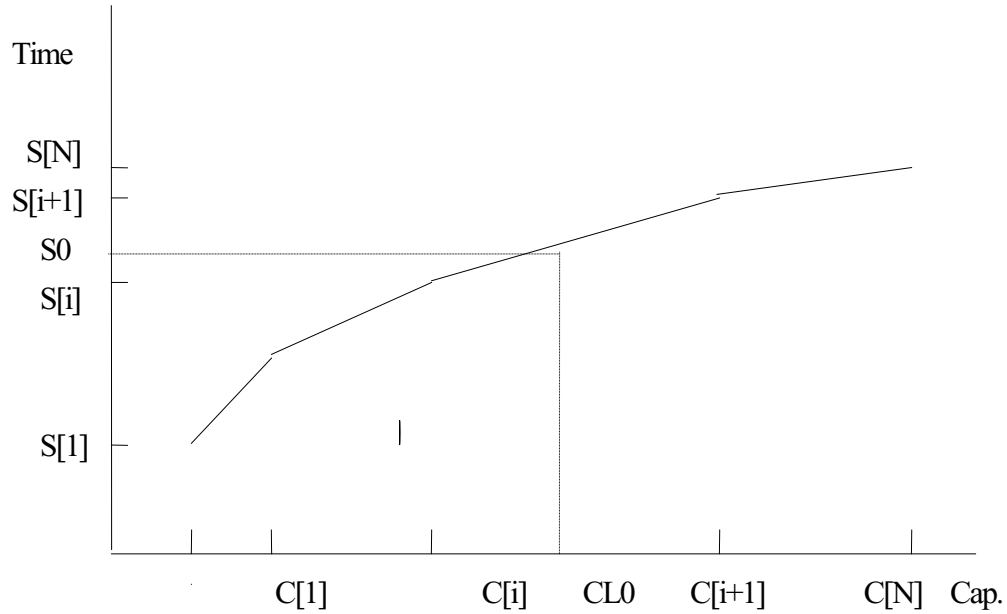


Figure 3. Example of input slew rate calculation

5.2.2.1 **Ts table specification**

A Ts table is a one dimensional matrix for each transient timing group.(See F.2)
The index(C[i]) is the capacitance of the net which includes the input of the target gate,
The value(S[i]) is the characterized input slew rate
where
2 i N (N is effective maximum number of capacitance values used,
C[i] has 1 real value of capacitance, unit is pF or fF,
0 C[i] C[i+1],
S[i] has 1 real value of time, unit is ns.

5.2.2.2 **Input slew rate(Ts0) calculation rule**

To calculate input slew rate by Ts table, the linear interpolation method will be applied between C[i] and C[i+1].

If target input capacitance(C0) is ranged between C[i] and C[i+1]

then

Ts0 = a C0 + b .....(2)

where a = (S[i+1] S[i]) (C[i+1] C[i])

b = (C[i+1] S[i] C[i] S[i+1]) (C[i+1] C[i]).

If target input capacitance is less than C[1], then set i to 1.

If target input capacitance is greater than C[N],

then set i to N-1.

Then apply it to function of (2).

5.2.3 **Port to Port propagation delay time calculation**

Port to port delay is calculated using as Tpd table ,shown in Figure 4, using either a linear or a bilinear interpolation method.

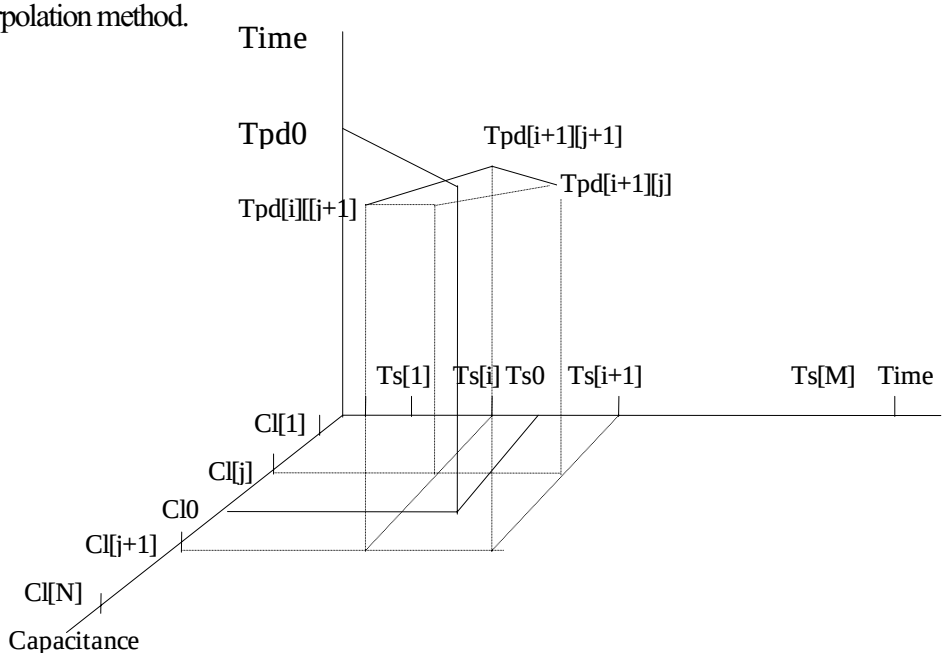


Figure 4. Example of propagation delay time calculation

5.2.3.1 **Tpd table specification**

Tpd table is a two dimensional matrix for each transient timing group.(See F.3)

The 1st index (Ts[i]) is input slew rate ,
The 2nd index(Cl[j]) is load capacitance of output of gate,
The value (Tpd[i][j]) is characterized propagation delay time,
where
2 i M (M is effective maximum number of input slew rates used),
Ts[i] has 1 real value of time, unit is ns,
0 Ts[i] Ts[i+1],
2 j N (N is effective maximum number of load capacitance values of gate output used),
Cl[j] has 1 real value of capacitance, unit is pF or fF,
0 Cl[j] Cl[j+1],
Tpd[i][j] has 1 real value of time, unit is ns,
Tpd[i][j] Tpd[i+1][j], Tpd[i][j+1] Tpd[i+1][j+1].

5.2.3.2 **Selection rule of 4 points**

To calculate port to port propagation delay time by Tpd table, both linear and bilinear interpolation method are applied among 4 points .

(Tpd[i][j],Tpd[i+1][j],Tpd[i][j+1],Tpd[i+1][j+1]) as shown in Figure 4.
If calculated Ts0 is ranged between Ts[i] and Ts[i+1] and
if target load capacitance(CL1) is ranged between Cl[j] and Cl[j+1],
then select
Tpd[i][j],Tpd[i+1][j],Tpd[i][j+1],Tpd[i+1][j+1].

If Ts0 is less than Ts[1], set i to 1.
If Ts0 is greater than Ts[M], set i to M-1.
If CL1 is less than Cl[1], set j to 1.
If CL1 is greater than Cl[N], set j to N-1.
Then select
Tpd[i][j],Tpd[i+1][j],Tpd[i][j+1],Tpd[i+1][j+1].

5.2.3.3 **Propagation delay time(Tpd0) approximation**

Two methods are specified here.
One is to solve bilinear interpolation (Z = a X + b Y + c X Y + d).
See Annex A in details.
Another method is the linear interpolation based on 3 points which are selected from 4 points shown in figure 5. (See Annex F.4 for detail interpolation example)

5.2.3.3.1 **Selection rule of 3 points**

Tpd is slowly increasing convex function with negative second derivatives.
In this case, 3 point linear approximation is more accurate than bilinear interpolation

,especially more accurate for smaller tables.
See Annex B for details.

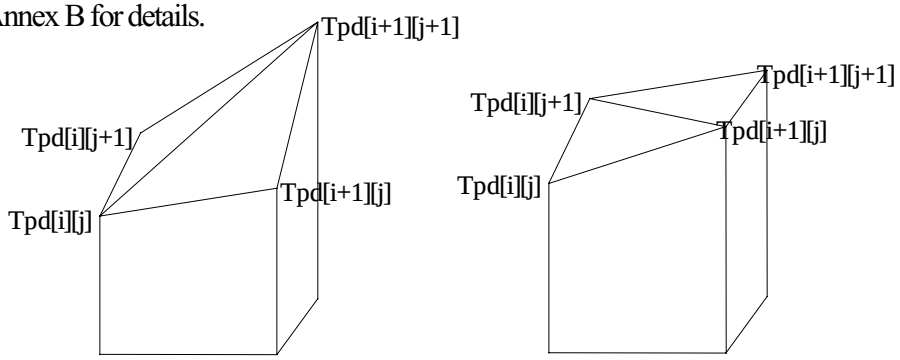


Figure 5. Selection of 3 points

* 3 points are selected using plane function G(slew, load) which consist of Tpd[i][j], Tpd[i+1][j], and Tpd[i][j+1] and apply following rule. (See Annex C.)

If G(Ts[i+1], Cl [j+1]) is greater than or equal to Tpd[i+1][j+1],
Then select 2 planes which consist of the following 3 points.
One is Tpd[i+1][j], Tpd[i][j+1], and Tpd[i][j]
Another is Tpd[i+1][j], Tpd[i][j+1] and Tpd[i+1][j+1]

If G(Ts[i+1], Cl [j+1]) is less than Tpd[i+1][j+1],
then select 2 planes which consist of the following 3 points.
One is Tpd[i][j], Tpd[i+1][j+1], and Tpd[i+1][j].
Another is Tpd[i][j], Tpd[i+1][j+1], and Tpd[i][j+1] .

After that ,select 3 points which include point (Ts0, CL1).
See Annex E for examples.
See Annex D for the evaluation of the accuracy comparison between two method.

5.2.3.3.2 **Tpd calculation by linear interpolation method**

Tpd is calculated using the following equation. (See Annex B)

$$Z = Tpd[i][j+1] + (Tpd[i][j] \quad Tpd[i][j+1]) \quad (Cl[j] \quad Cl[j+1]) \quad (CL1 \quad Cl[j+1])$$

$$Tpd0 = Z + (Tpd[i+1][j]-Tpd[j][j])/(Ts[i+1]-Ts[i])x(Ts0-Ts[i]) \quad .....(3)$$

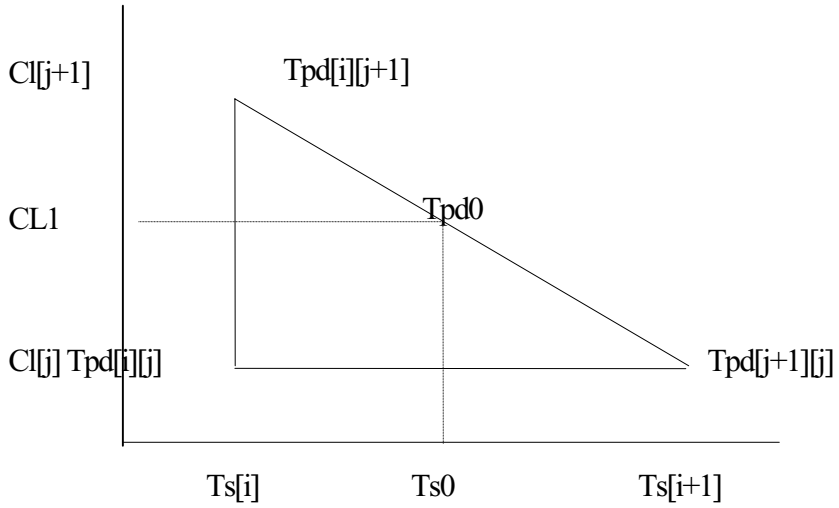


Figure 6. Interpolation for right angle triangle

Annex A. (Informative)
Four points interpolation

Here, we explain bilinear interpolation by four trapezoidal points.

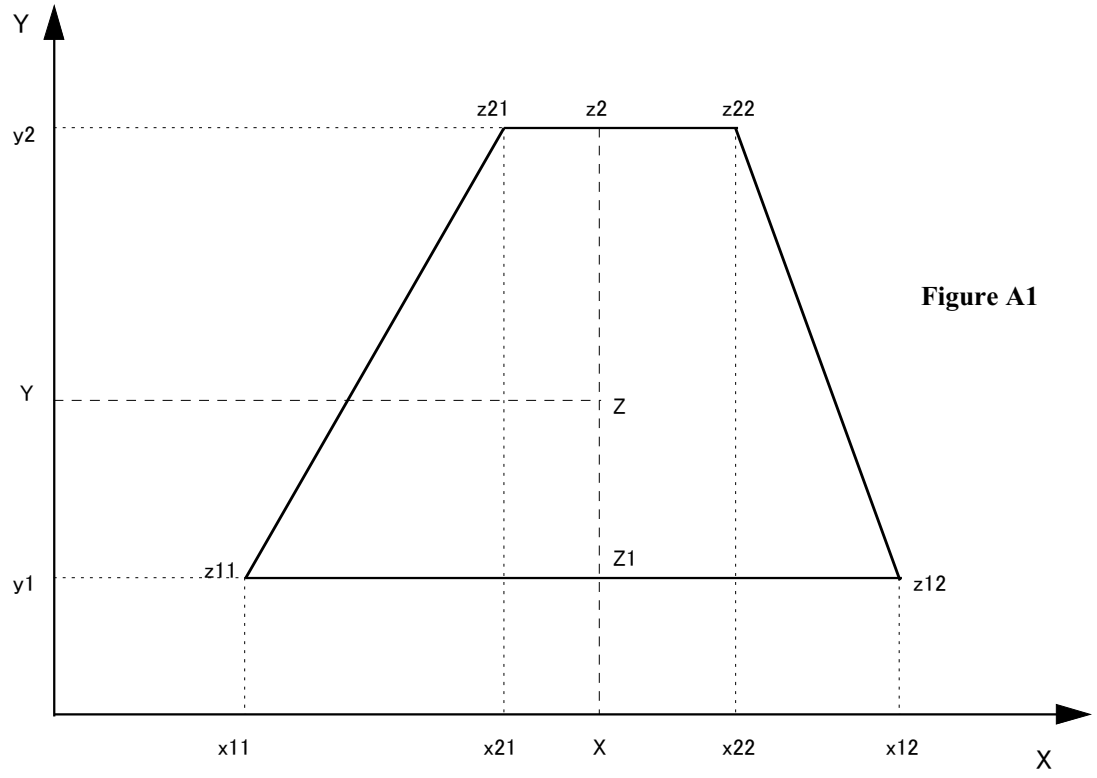


Figure A1

When four trapezoidal points(x11,y1), (x12,y1), (x21,y2), (x22,y2) are given and functional values of each points z11,z12,z21,z22 are known, we estimate functional value z at (x, y) by following bilinear interpolation method.(see Figure A.1)

Bilinear interpolation method uses following bilinear approximation formula.

$$z = Ax+By+Cxy+D$$

A.1.Bilinear formula is "linear" on Y=y1, so we can evaluate Z1 by linear interpolation(x11,z11)and(x12,z12).

$$Z1 = z11 + \frac{z12 - z11}{x12 - x11}(x - x11)$$

A.2. Bilinear formula is "linear" on Y=y2, so we can evaluate Z2 by linear interpolation(x21,z21)and(x22,z22).

$$Z2 = z21 + \frac{z22 - z21}{x22 - x21}(x - x21)$$

A.3. Bilinear formula is "linear" on X=x, so we can evaluate z by linear interpolation(y1 ,Z1) and (y2 ,Z2).

$$z = Z1 + \frac{Z2 - Z1}{y2 - y1}(y - y1)$$

Consequently, we can evaluate z at(x, y) by the following three-steps formula.

[Trapezoidal points interpolation formula]

$$Z1 = z11 + \frac{z12 - z11}{x12 - x11}(x - x11)$$

$$Z2 = z21 + \frac{z22 - z21}{x22 - x21}(x - x21)$$

$$z = Z1 + \frac{Z2 - Z1}{y2 - y1}(y - y1)$$

Annex B. (Informative)
Three points interpolation

Here, we explain linear interpolation by three triangular points.

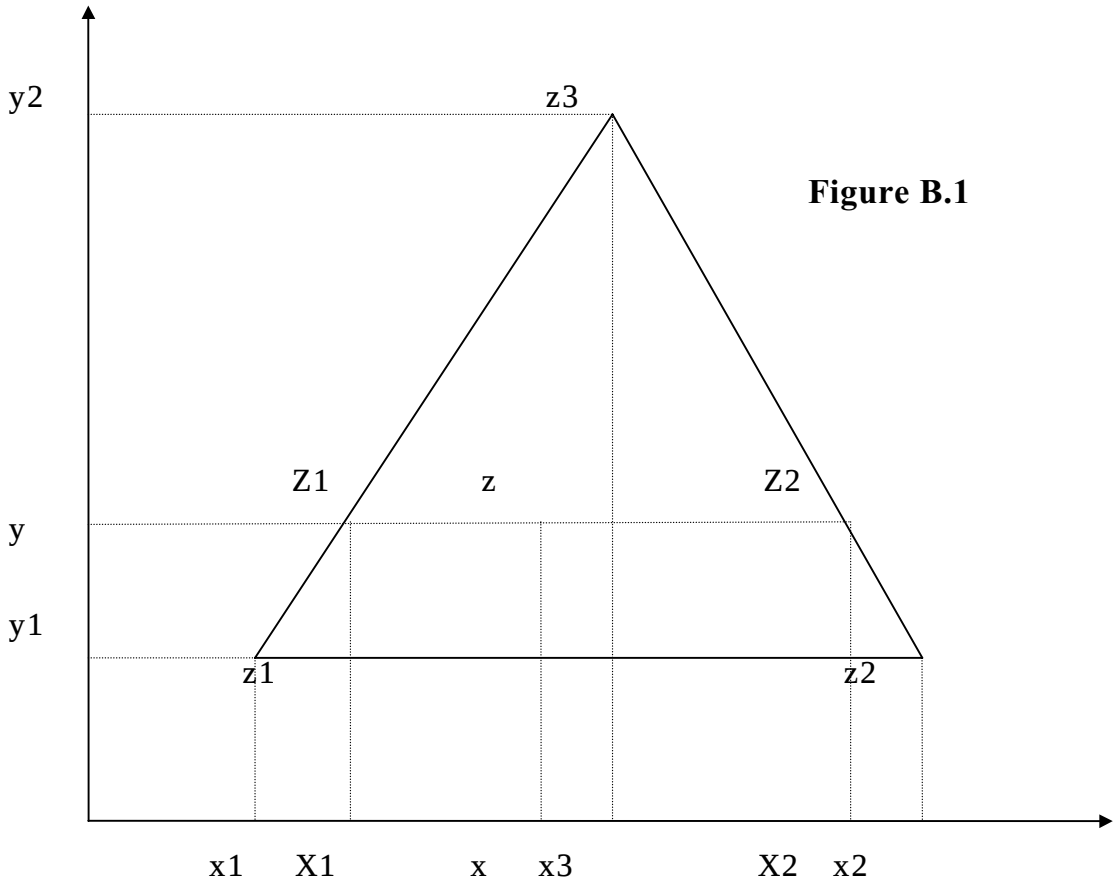


Figure B.1

When three points (x1,y1),(x2,y1),(x3,y2) are given and functional values of each points z1 ,z2 ,z3 are known, we estimate functional value z at (x, y) by following linear interpolation method.(see Figure B.1)

B.1. Evaluate X1 by linear interpolation (y2 ,x3) and (y1 ,x1).

$$X1 = x3 + \frac{x1 - x3}{y1 - y2} (y - y2)$$

B.2. Evaluate Z1 by linear interpolation (y2 ,z3) and (y1 ,z1).

$$Z1 = z3 + \frac{z1 - z3}{y1 - y2} (y - y2)$$

B.3. Evaluate X2 by linear interpolation (y2 ,x3) and (y1 ,x2).

$$X2 = x3 + \frac{x2 - x3}{y1 - y2} (y - y2)$$

B.4. Evaluate Z2 by linear interpolation (y2 ,z3) and (y1 ,z2).

$$Z2 = z3 + \frac{z2 - z3}{y1 - y2} (y - y2)$$

B.5. Evaluate z by linear interpolation (X1 ,Z1) and (X2 ,Z2).

$$z = Z1 + \frac{Z2 - Z1}{X2 - X1} (x - X1)$$

Where,

$$Z2 - Z1 = z3 + \frac{z2 - z3}{y1 - y2}(y - y2) - z3 - \frac{z1 - z3}{y1 - y2}(y - y2)$$

$$= \frac{z2 - z1}{y1 - y2} (y - y2)$$

$$\text{Similarly,}\quad X2 - X1 = \frac{x2 - x1}{y1 - y2}(y - y2)$$

$$\text{So,}\quad \frac{Z2 - Z1}{X2 - X1} = \frac{z2 - z1}{x2 - x1}$$

$$\text{Therefore,}\quad z = Z1 + \frac{z2 - z1}{x2 - x1}(x - X1)$$

Consequently, we can evaluate z at(x, y) by the following three-steps formula.

[Three points interpolation formula]

$$X = x3 + \frac{x1 - x3}{y1 - y2}(y - y2)$$

$$Z = z3 + \frac{z1 - z3}{y1 - y2}(y - y2)$$

$$z = Z + \frac{z2 - z1}{x2 - x1}(x - X)$$

If three points form right-angled triangle(x1=x3), the formula becomes easier.

[Three points interpolation formula (right-angled)]

$$Z = z3 + \frac{z1 - z3}{y1 - y2}(y - y2)$$

$$z = Z + \frac{z2 - z1}{x2 - x1}(x - x3)$$

Annex C (Informative)
Selection method of interpolation plane

C.1. the contents of this annex

We explain selection method of interpolation plane in calculation of the gate delay value using table model. In the delay calculation that uses table model, the interpolation method between points constituting a table is important. We analyzed the accuracy of the delay value after interpolation using 3 points selected from 4 points surrounding the point to be calculated.

C.1.1 A selection of interpolation plane (No. 1).

By choosing 3 points among 4 points surrounding the point to be calculated, we interpolate a delay value. In general, there are two candidate planes of interpolation for 4 points unless these 4 points are on one plane. In this case, we need to select which interpolation plane to use as a plane to interpolate a point to be calculated.

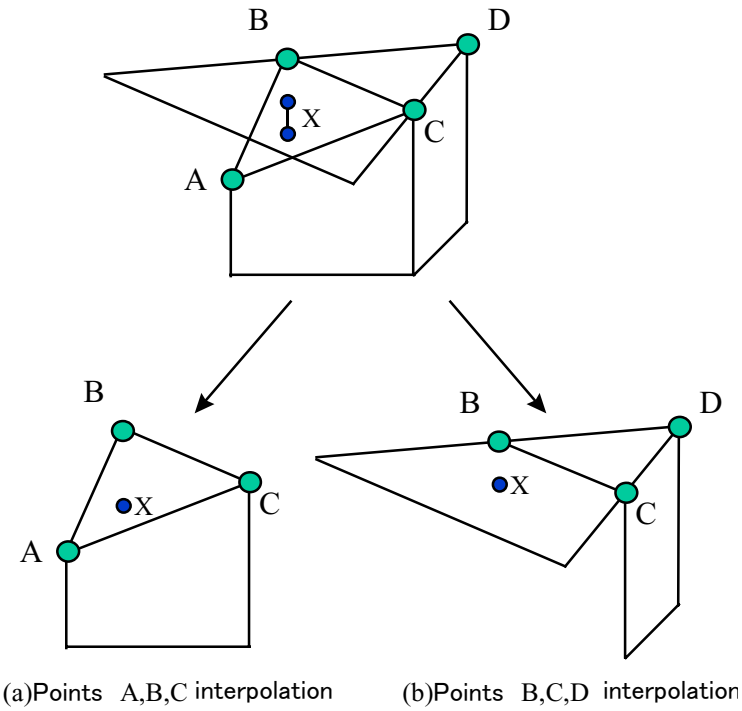


Figure C.1 selection of an interpolation plane (No.1)

. C1.2. A selection of an interpolation plane (No. 2)

. A combined plane formed by two planes by selecting 3 points among 4 points become convex or concave according to the combination of the planes as shown in Fig. C.1. Hence, the selection of the combined plane shown in Fig. C.2 becomes the second problem when calculating the delay from 4 points.

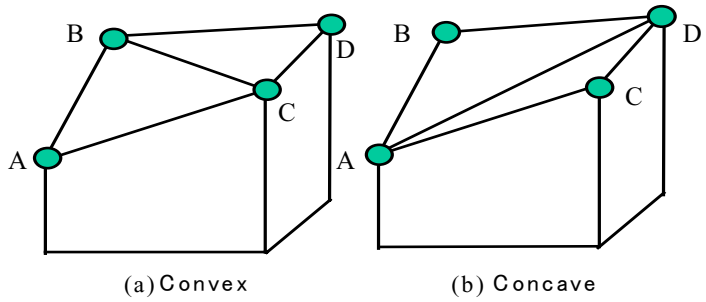
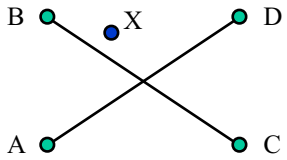


Figure C.2 A selection of interpolation plane (No. 2)

C.2 Precision evaluation

C 2.1. A selection of interpolation plane (No. 1)

We compare the result of the interpolation for each combination of the planes for 4 points A,B,C and D, and the result of the SPICE simulation for the point to be calculated based on the same 4 points whose relation of coordinates and delay values are given in Fig. C.4. In this comparison, we used the delay value of an actual cell.



A:(load1,slew1,delay1)
B:(load1,slew2,delay3)
C:(load2,slew1,delay3)
D:(load2,slew2,delay4)
load1<load2,slew1<slew2
delay1<delay2,delay3<delay4
X:(load3,slew3,delay5)
load3>(load1+load2)/2
slew3<(slew1+slew2)/2

Figure C.3 Relation of each point

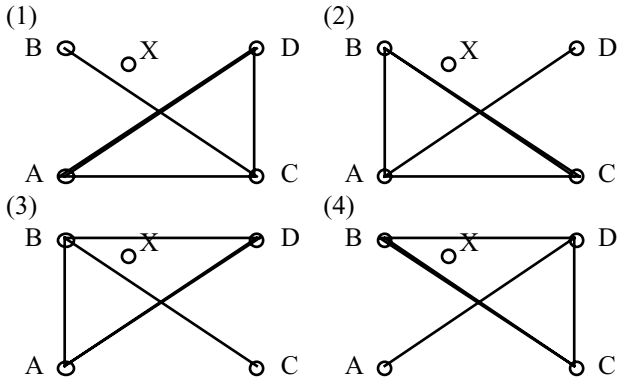


Figure C.4 A plane for interpolation of a point X

A discrepancy of a delay value at a point X calculated using each plane shown in Fig. C.4

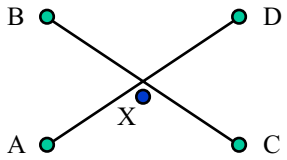
(1), (2), (3) and (4) and an delay value of SPICE simulation is as follows.

(1) the plane which is formed by point A,C,D	24.3[%]
(2) the plane which is formed by point A,B,C	-22.5[%]
(3) the plane which is formed by point A,B,D	-7.6[%]
(4) the plane which is formed by point B,C,D	-12.9[%]

From the above-mentioned result, better result is obtained for the interpolation using the planes (3) and (4) which contain point X than the combination of the planes (1) and (2) which don't contain point X.

C.2.2. A selection of interpolation plane (No. 2)

We compare a result of two kinds of interpolation method described before and the result of the SPICE simulation for the 4 points with the following relation. In this comparison, we also used the delay value of an actual cell.



A:(load1,slew1,delay1)
B:(load1,slew2,delay3)
C:(load2,slew1,delay3)
D:(load2,slew2,delay4)
load1<load2,slew1<slew2
delay1<delay2,delay3<delay4
X:(load3,slew3,delay5)
load3<(load1+load2)/2
slew3<(slew1+slew2)/2

Figure C.5 Position relation of each point (a plane)

We assume the position relation of the points A, B, C and D where a point D is above the plane consisting of points A,B,C (cf. Figure C.3). And we also assume that the point X at which delay value is calculated exists near the crossing point of a line A-D and a line B-C.

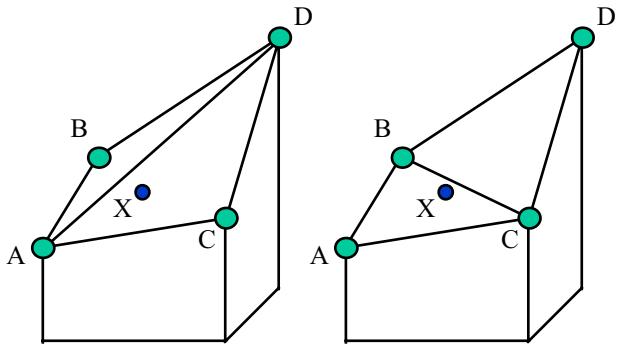


Figure C.6 Position relation of each point (a cube)

We show an error of result obtained by the interpolation using two kinds of planes shown in the Fig. C.3 against the delay value in the point X obtained by SPICE simulation under the above assumption.

interpolation plane (convex)	-3.3[%]
interpolation plane (concave)	-29.7[%]

A good result of the interpolation is obtained when the combination of the planes are so selected that the surface of delay curve become convex in the area surrounded by points A,B,C and D .

C.3. Consideration

A characteristic of a delay value table used for the above precision evaluation satisfies the following conditions when expressed a delay value in F(load,slew) as a function of the load capacitance and a input signal slew.

(1) $F^{(1)}(load,slew) > 0$

(2) $F^{(2)}(load,slew) < 0$

In other words, a curved surface of delay value expressed in F(load,slew) is monotonously increasing in load capacitance and an input signal slew and convex in all areas. Under such condition, interpolation plane which uses two planes divided by a line A-D gives a good result when a point D is above the plane formed by points A,B and C. On the contrary, interpolation plane which uses two planes divided by a line B-C gives a good result when a point D is under the plane formed by points A,B and C. From these consideration, the curved surface of delay value that is convex is well interpolated by applying the above method in the opposite manner.

C.4. A summary

From precision evaluation and a result of consideration, we can summarize the division method as follows. Let a delay curved surface in one domain be f(load,slew), a plane formed by points A,B and C be g(load,slew), a load capacitance, an input slew and a delay value in point D be load_d and slew_d,delay_d respectively. In this case, we assume that $f^{(1)}(load,slew)$ is 0 or positive.

(1) $F^{(1)}(load,slew)$ 0 && $f^{(2)}(load,slew)$ 0

(a) g(load_d,slew_d) > delay_d Division by a line B-D

(b) g(load_d,slew_d) < delay_d Division by a line A-D

(c) g(load_d,slew_d) = delay_d A plane formed by points A,B,C and D

(2) $F^{(1)}(load,slew)$ 0 && $f^{(2)}(load,slew)$ 0

(d) g(load_d,slew_d) > delay_d Division by a line A-D

(e) g(load_d,slew_d) < delay_d Division by a line B-C

(f) g(load_d,slew_d) = delay_d A plane formed by points A,B,C and D

And, in the domain where $f^{(1)}(load,slew) > 0$ is not satisfied, we need to determine the selection method of the plane according to the position relation of points A,B,C and D. For this purpose, implementing the information of derivatives in the library will be helpful.

Annex D. (Informative)

Theoretical Accuracy Comparison Between Two Interpolation Methods

As an example of theoretical accuracy comparison between local-linear interpolation and bilinear interpolation, we estimate these precision in case of the quadric surface $z = Ax^2 + Bxy + Cy^2 + Dx + Ey$ on the rectangle [0,S]X[0,T] (S,T>0).

We show that the local-linear interpolation is better than bilinear interpolation if $|B|$ is sufficiently small.

D.1. Test surface

We compare accuracy between two interpolations by applying them to the following quadratic surface $z = Ax^2 + Bxy + Cy^2 + Dx + Ey$ on [0,S]X[0,T] (S,T>0). By scaling x and y -direction, it is enough to consider in case of S=T=1. So we analize the following test surface on [0,1]X[0,1].

$z = Ax^2 + Bxy + Cy^2 + Dx + Ey$

This test surface is constrained by following properties of delay surface.

a) Delay surface is convex. $\Rightarrow A < 0, C < 0$

b) Delay surface is monotone increasing.

$$\Rightarrow \begin{cases} 2Ax + By + D > 0 \\ 2Cy + Bx + E > 0 \end{cases} \qquad (0 \le x, y \le 1)$$

Especially, $D > 0, E > 0$.

D.2. Error evaluation

D.2.1 Bilinear interpolation

Applying bilinear interpolation to the test surface, we give following approximation formula.

$$z = (A + D)x + (C + E)y + Bxy$$

And the approximation error is

$$E_4(x, y) = Ax^2 + Cy^2 - Ax - Cy \ .$$

We estimate the maximum error of bilinear interpolation.

First, on the boundary of square,

$$E_4(x,0) = E_4(x,1) = Ax^2 - Ax$$

$$E_4(0,y) = E_4(1,y) = Cy^2 - Cy$$

Hence,

$$\text{Boundary maximum error} = \max\left\{\frac{|A|}{4}, \frac{|C|}{4}\right\}$$

Next, the maximum error in interior region is evaluated by stationary value of $E_4(x, y)$.

Hence,

$$\text{Interior maximum error} = \left|E_4\left(\frac{1}{2}, \frac{1}{2}\right)\right| = \frac{|A + C|}{4}$$

Therefore, the maximum error of bilinear interpolation is

$$ME_4 = \max\left\{\frac{|A|}{4}, \frac{|C|}{4}, \frac{|A + C|}{4}\right\} = \frac{|A + C|}{4}$$

because, $A < 0, C < 0$.

D.2.2 **RF local linear interpolation**

We evaluate approximation formula and approximation error in case of rightward falling (RF) local-linear interpolation. (Figure D.1)

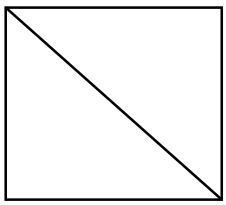


Figure D.1

Applying RF local linear interpolation to the test surface, we give following approximation formula.

$$z = \begin{cases} (A + D)x + (C + E)y & (x + y \le 1) \\ (A + B + D)x + (B + C + E)y - B & (x + y > 1) \end{cases}$$

And the approximation error

$$E_{3RF}(x, y) = \begin{cases} Ax^2 + Cy^2 - Ax - Cy + Bxy & (x + y \le 1) \\ Ax^2 + Cy^2 - Ax - Cy + B(xy - x - y + 1) & (x + y > 1) \end{cases}$$

Especially, on diagonal line ($x + y = 1$)

$$E_{3RF}(x, 1 - x) = (A + C - B)(x^2 - x)$$

And on the boundary of square,

$$E_{3RF}(x, 0) = E_{3RF}(x, 1) = Ax^2 - Ax$$
$$E_{3RF}(0, y) = E_{3RF}(1, y) = Cy^2 - Cy$$

D.2.3 **RR local linear interpolation**

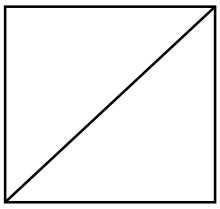


Figure D.2

We evaluate approximation formula and error in case of rightward rising (RR) local linear interpolation. (Figure D.2)

Applying RR local linear interpolation to the test surface, we give following approximation formula.

$$z = \begin{cases} (A+B+D)x + (C+E)y & (x \le y) \\ (A+D)x + (B+C+E)y & (x > y) \end{cases}$$

And the approximation error

$$E_{3RR}(x,y) = \begin{cases} Ax^2 + Cy^2 - Ax - Cy + B(xy - x) & (x \le y) \\ Ax^2 + Cy^2 - Ax - Cy + B(xy - y) & (x > y) \end{cases}$$

Especially, on diagonal line ($x = y$)

$$E_{3RR}(x,x) = (A+C+B)(x^2 - x)$$

And on the boundary of square

$$E_{3RR}(x,0) = E_{3RR}(x,1) = Ax^2 - Ax$$
$$E_{3RR}(0,y) = E_{3RR}(1,y) = Cy^2 - Cy$$

Hence,

$$E_{3RR}(x,y) = E_{3RF}(x,y) = E_4(x,y)$$ ***on the boundary of the square***.

D.3. **Accuracy Comparison when $|B|$ is sufficiently small**

If $B = 0$, $E_4(x,y) = E_{3RF}(x,y) = E_{3RR}(x,y) = Ax^2 + Cy^2 - Ax - Cy$.

Here, we compare accuracy between the local linear interpolation and the bilinear interpolation when $|B|$ is sufficiently small, by using perturbation method. In the following, put $B = b\varepsilon$ where b is real number and ε is very small positive number (order parameter).

D.3.1 **Bilinear interpolation**

The maximum approximation error of bilinear interpolation is

$$ME_4 = \frac{|A+C|}{4}.$$

D.3.2 RF local linear interpolation

The maximum error on the boundary of square is $\max\left\{\frac{|A|}{4}, \frac{|C|}{4}\right\}$ and that on the diagonal line ($x+y=1$) is $\frac{|A+C-b\varepsilon|}{4}$. The maximum error in interior region is evaluated by stationary value of $E_{3RF}(x,y)$.

The stationary condition $\frac{\partial E_{3RF}}{\partial x} = \frac{\partial E_{3RF}}{\partial y} = 0$ is

$$\begin{cases} 2A(x-\frac{1}{2}) + b\varepsilon(y-\frac{1}{2}) = -\frac{b\varepsilon}{2} \\ b\varepsilon(x-\frac{1}{2}) + 2C(y-\frac{1}{2}) = -\frac{b\varepsilon}{2} \end{cases} \qquad (x+y<1)$$

$$\begin{cases} 2A(x-\frac{1}{2}) + b\varepsilon(y-\frac{1}{2}) = \frac{b\varepsilon}{2} \\ b\varepsilon(x-\frac{1}{2}) + 2C(y-\frac{1}{2}) = \frac{b\varepsilon}{2} \end{cases} \qquad (x+y>1)$$

By using perturbation approximation, the solution (x_0, y_0) of above equation is

$$(x_0, y_0) = \begin{cases} (\frac{1}{2} - \frac{b}{4A}\varepsilon, \frac{1}{2} - \frac{b}{4C}\varepsilon) & (x+y<1) \\ (\frac{1}{2} + \frac{b}{4A}\varepsilon, \frac{1}{2} + \frac{b}{4C}\varepsilon) & (x+y>1) \end{cases}$$

From $A<0, C<0$, the solution (x_0, y_0) is consistent with range condition if $b<0$.

Then the stationary value is

$$E_{3RF}(x_0, y_0) = -\frac{A+C-b\varepsilon}{4} - \frac{b^2}{16}(\frac{1}{A} + \frac{1}{C})\varepsilon^2 + O(\varepsilon^3)$$

Therefore the maximum approximation error of RF local linear interpolation is

$$ME_{3RF} = \frac{|A+C-b\varepsilon|}{4} + O(\varepsilon^2)$$

D.3.3 RR local linear interpolation

The maximum error on the boundary of square is $\max\left\{\frac{|A|}{4},\frac{|C|}{4}\right\}$ and that on the diagonal line ($x=y$) is $\frac{|A+C+b\varepsilon|}{4}$. The maximum error in interior region is evaluated by stationary value of $E_{3RR}(x,y)$.

The stationary condition $\frac{\partial E_{3RR}}{\partial x}=\frac{\partial E_{3RR}}{\partial y}=0$ is

$$\begin{cases} 2A(x-\frac{1}{2})+b\varepsilon(y-\frac{1}{2})=\frac{b\varepsilon}{2} \\ b\varepsilon(x-\frac{1}{2})+2C(y-\frac{1}{2})=-\frac{b\varepsilon}{2} \end{cases} \qquad (x<y)$$

$$\begin{cases} 2A(x-\frac{1}{2})+b\varepsilon(y-\frac{1}{2})=-\frac{b\varepsilon}{2} \\ b\varepsilon(x-\frac{1}{2})+2C(y-\frac{1}{2})=\frac{b\varepsilon}{2} \end{cases} \qquad (x>y)$$

By using perturbation approximation, the solution (x_0,y_0) of above equation is

$$(x_{0,},y_0)=\begin{cases} (\frac{1}{2}+\frac{b}{4A}\varepsilon,\frac{1}{2}-\frac{b}{4C}\varepsilon) & (x<y) \\ (\frac{1}{2}-\frac{b}{4A}\varepsilon,\frac{1}{2}+\frac{b}{4C}\varepsilon) & (x>y) \end{cases}$$

From $A<0,C<0$, the solution (x_0,y_0) is consistent with range condition if $b>0$.

Then the stationary value is

$$E_{3RR}(x_0,y_0)=-\frac{A+C+b\varepsilon}{4}-\frac{b^2}{16}(\frac{1}{A}+\frac{1}{C})\varepsilon^2+O(\varepsilon^3)$$

Therefore the maximum approximation error of RR local linear interpolation is

$$ME_{3RR}=\frac{|A+C+b\varepsilon|}{4}+O(\varepsilon^2)$$

D.3.4 Accuracy Comparison

By above evaluation, maximum approximation errors of interpolations are as follows.

$$ME_4 = \frac{|A+C|}{4} \qquad \text{(bilinear)}$$

$$ME_{3RF} = \frac{|A+C-b\varepsilon|}{4} + O(\varepsilon^2) \qquad \text{(RF local linear)}$$

$$ME_{3RR} = \frac{|A+C+b\varepsilon|}{4} + O(\varepsilon^2) \qquad \text{(RR local linear)}$$

From $A<0$ and $C<0$, we give following accuracy comparison between interpolations.

i) If $b>0$, then RR local linear interpolation is better than bilinear interpolation.

If $b<0$, then RF local linear interpolation is better than bilinear interpolation.

D.4. Conclusion

We give the following accuracy comparison between local linear interpolation and bilinear interpolation in case of quadratic surface $z = Ax^2 + Bxy + Cy^2 + Dx + Ey$ on the rectangle [0,T]X[0,T] (S,T>0).

i) If $B>0$ and $|B|$ is sufficiently small, then RR local linear interpolation is better than bilinear interpolation.

ii) If $B<0$ and $|B|$ is sufficiently small, then RF local linear interpolation is better than bilinear interpolation.

iii) If $B=0$, then the accuracy of local linear interpolation and that of bilinear interpolation are same.

Annex E (Informative)
Application example

E.1. The contents of this Annex

We demonstrate the interpolation selection method by showing the example of delay calculation of gates.

E.1.1 Inverter

We show the result of applying above described method to the actual library as an example. The library cell used here as an example has the following characteristics.

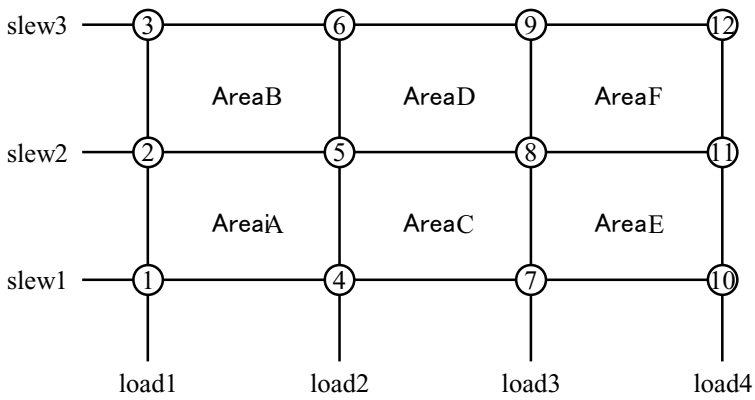
Cell type is inverter cell,

A table of 4 3 for a load capacitance input signal slew ,

The load capacitance and input signal slew are monotonously increasing,

Convex in all area.

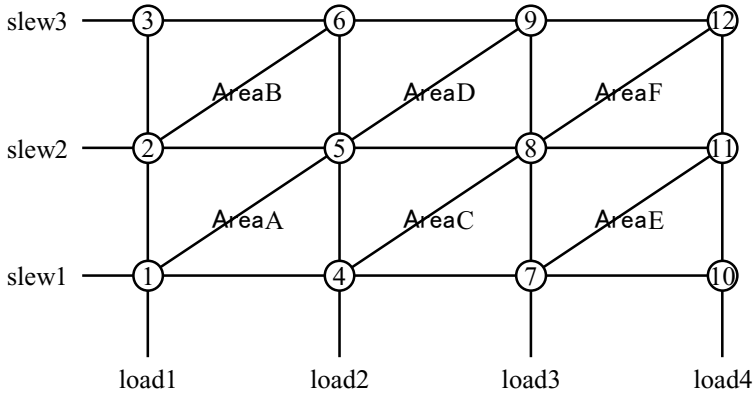
[Table]



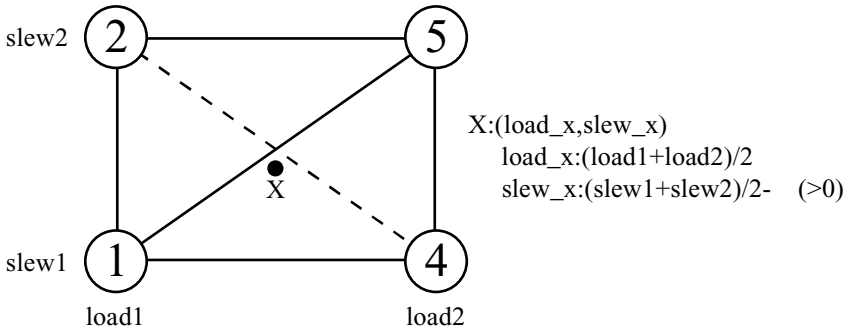
load1 < load2 < load3 < load4 , slew1 < slew2 < slew3

We examine the position relation of four points forming each area A - F. For example, we divide the area A so that the condition of convex is satisfied according to the position relation of the point 5 and the plane formed by the points 1, 2 and 4. We will show the result of the area division by examining the relation of the points for each area using the actual delay data.

[Table after area division]



In this table, a result of the dividing method is to do "the division of upward slant to the right" or rightward rising in order to meet the condition from position relation of points in all domains. In order to verify the correctness of the division of the areas, SPICE simulation is performed for points in the center of each area and compared with the result of the interpolation.



An interpolation of the delay value for the point X is performed using two kinds of planes formed by points 1,2 and 4 which are rightward rising and points 1,4 and 5 which are rightward falling. Error of interpolation is examined against the result of SPICE simulation. In the above figure, a simulation condition on domain A is shown. The error analysis is also performed on the other areas.

	A	B	C	D	E	F
	-29.340	-5.153	-0.746	-17.789	-5.151	-3.388
	-2.921	-0.743	0.293	-3.527	-0.198	-0.419

From the above-mentioned result, we conclude that the interpolation is well performed by dividing all areas by the rightward rising lines. The reason why a difference of precision is small in areas B,C,E and F is that four points to constitute each domain are near to one plane.

E.1.2 Buffer

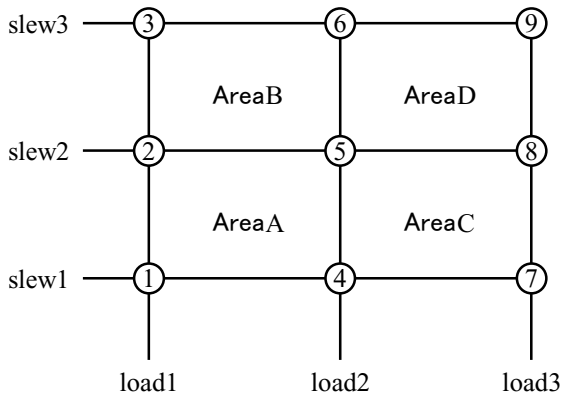
[Characteristic]

Cell type is buffer cell,

A table of 3 2 for a load capacitance input signal slew ,

Convex in the all domain.

[Table]

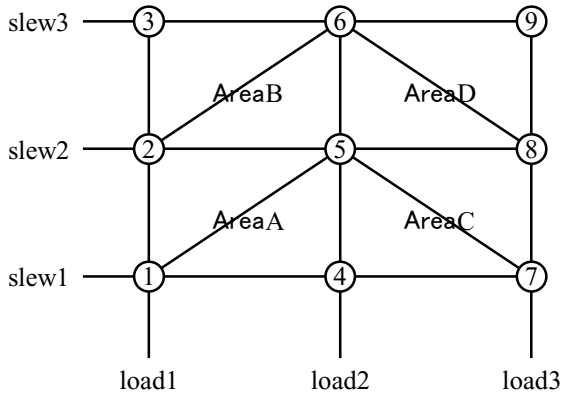


load1 < load2 < load3 < load4 , slew1 < slew2 < slew3

For the above table, we show the result of the division of area using the actual delay

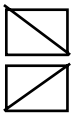
value as performed for inverter.

[Area division table]



In areas C and D, the result is different from the case of the inverter. This is because that the point 8 is under a plane formed by the points 4, 5 and 7. We show a result compared with SPICE simulation result using an equal point in case of inverter next.

	A	B	C	D
	-4.8656	-1.8065	-2.1576	-0.2706
	-4.8025	-0.9651	-2.1654	-0.2735



As the result, the difference of rightward rising division and the rightward falling division is as small as none. Even in this case, the error is smaller in the case of the rightward rising line under our early assumption. This result gives a good example that the method of the area division is satisfactory if each division of area is upward slant to the right.

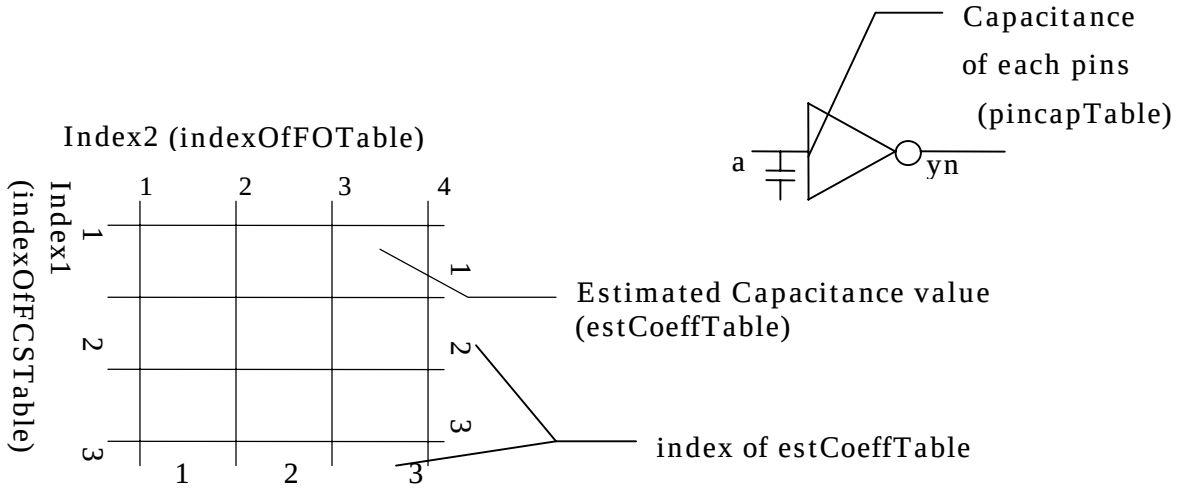
Annex F (Informative)

Example of Cn, Ts, Tpd tables by Delay Calculation Language

This standard is focused in interpolation method. And this standard is also independent of table syntax. However it is easy to understand this standard if we will show the example of tables. Following is the example of tables which was written by DCL (IEEE par1481)(Delay Calculation Language).

F.1 Example of Cn table

In order to estimate the net capacitance, both Cn and pin capacitance table must be defined.

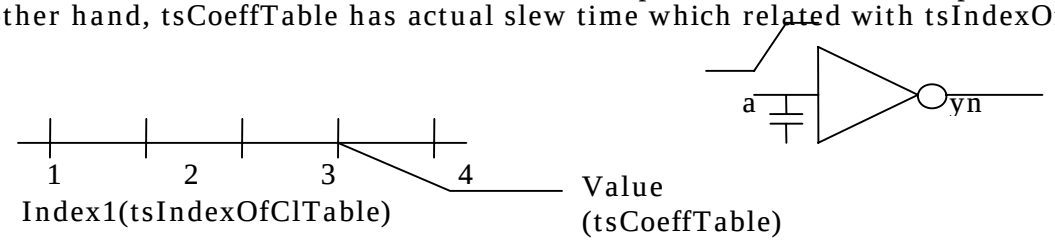



```
/* define port capacitance of each cells */
   table(pincapTable):
/* cell_name pin: capacitance value */
   inv.a: 0.010;
end;

/* define size index */
   table(indexOfCSTable):
/* dieName : ga4x4, ga5x5, ga6x6, ga7x7, bd1, bd2, bd3, bd4 */
   GA_0.35: 'STEP' 'STEP' 'STEP' 'STEP'
   16.00   25.00   36.00   49.00;
end;

/* define fanout index */
   table(indexOFFOTable):
/* dieName : foix1, foix2, foix3, foix4, bd1, bd2, bd3, bd4 */
   GA_0.35: 'STEP'   'STEP' 'STEP' 'STEP'
   1   3   10   50;
end;

/* define estimated wire capacitance value */
   table(estCoeffTable):
/* dieName indexOfEstCoeffTableForCS indexOfEstCoeffTableForFO: wireLoad */
   GA_0.35."1"."1": 0.32;
   GA_0.35."1"."2": 0.42;
   GA_0.35."1"."3": 0.52;
          .....
   GA_0.35."3"."3": 1.02;
```

end;

F.2 Example of Ts table

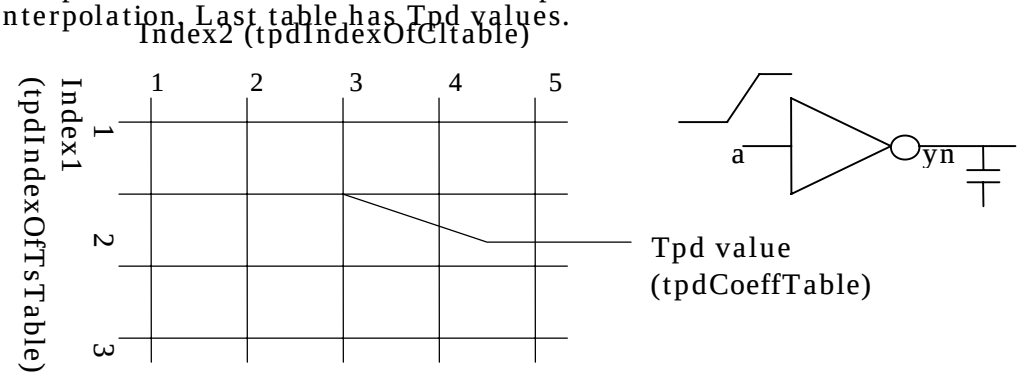
Following is one of the case of Ts table written by DCL. Mainly Ts tables are the index table named tsIndexOfClTable and slew time table named tsCoeffTable. tsIndexOfClTable has actual value of the capacitance for Ts interpolation. On the other hand, tsCoeffTable has actual slew time which related with tsIndexOfClTable.




```
/* define index Cl */
   table(tsIndexOfClTable):
/* cell_name outputPin transition: bd1 bd2 bd3 bd4 bd5 */
   inv.yn.R: 0.0075   0.0300   0.1200   0.3600   0.6000;
   inv.yn.F: 0.0075   0.0300   0.1200   0.3600   0.6000;
end;

/* define Ts values */
   table(tsCoeffTable):
/* cell_name outputPin transition: ts1 ts2 ts3 ts4 ts5 */
   inv.yn.R: 0.1616   0.3655 1.1960   1.4720   3.4100   5.9000;
   inv.yn.F: 0.1616   0.3655 1.1960   1.4720   3.4100   5.9000;
end;
```

F.3 Example of Tpd table

Following is the Tpd table example written by DCL. In this case, Tpd table is consisted 3 type of tables. 1st table has Ts values which is the index of Tpd interpolation. 2nd table has load capacitance values which is the index of Tpd interpolation. Last table has Tpd values.




```
/* define Ts values for index of Tpd interpolation */
   table(tpdIndexOfTsTable):
/* cell_name inputPin outputPin transition: bd1 bd2 bd3 bd4 bd5 */
   inv.a.yn.R: 0.1   0.5   1.8   5.0;
   inv.a.yn.F: 0.1   0.5   1.8   5.0;
end;
```

```
/* define Cl values for index of Tpd interpolation */
   table(tpdIndexOfClTable):
/* cell_name outputPin transition: bd1 bd2 bd3 bd4 bd5 */
   inv.yn.R: 0.0075   0.0300   0.1500   0.3600   0.600;
   inv.yn.F: 0.0075   0.0300   0.1500   0.3600   0.600;
end;

/* define Tpd values */
   able(tpdCoeffTable):
/* cell_name fromPin toPin coeffType indexOfCl indexOfTs: value */
/* RP case */
   inv.a.yn.RP."1"."1":   0.1033;
   inv.a.yn.RP."1"."2":   0.1812;
            ............
   inv.a.yn.RP."1"."4":   0.4981;
   inv.a.yn.RP."2"."1":   0.1990;
            ............
   inv.a.yn.RP."5"."4":   3.8040;
/* RT case */
   inv.a.yn.RT."1"."1":   0.1033;
            ............
   inv.a.yn.RT."5"."4":   3.8040;
/* FP case */
   inv.a.yn.FP."1"."1":   0.1044;
            ............
   inv.a.yn.FP."5"."4":   3.9040;
/* FT case */
   inv.a.yn.FT."1"."1":   0.1044;
            ............
   inv.a.yn.FT."5"."4":   3.8040;
end;
```

F.4 Example of Tpd interpolation

Following is one of the case of Tpd interpolation written by DCL. This example include both plane selection function named selectPlane and Tpd interpolation function named interp.

```
/*
+--------------------------------------+
|     Plane Selection Routine          |
|     RR -> 1,    RF-> -1              |
+--------------------------------------+
*/
calc(selectPlane):
   passed(
   string   : coeffType &
   integer : row, column & )

   Z00 = tpdCoeffTable( coeffType, row, column ).value,
   Z01 = tpdCoeffTable( coeffType, row, column+1 ).value,
   Z10 = tpdCoeffTable( coeffType, row+1, column ).value,
   Z11 = tpdCoeffTable( coeffType, row+1, column+1 ).value,
   when ( Z10+Z01-Z00 < Z11)
      result(integer:   1 ),
   otherwise
      result(integer: -1 );
```

```
/*
+-----------------------------------------+
|        Interpolation   Routine      |
+-----------------------------------------+
*/
calc(interp):
  passed(
       string   : coeffType &
       integer : row, column &
       number   : X1, X2, Y1, Y2, TS, CL
  )
  Z00 = tpdCoeffTable( coeffType, row, column ).value,
  Z01 = tpdCoeffTable( coeffType, row, column+1 ).value,
  Z10 = tpdCoeffTable( coeffType, row+1, column ).value,
  Z11 = tpdCoeffTable( coeffType, row+1, column+1 ).value,
  SRR = (Y2-Y1)*(TS-X1)-(X2-X1)*(CL-Y1),
  SRF = (Y2-Y1)*(TS-X1)+(X2-X1)*(CL-Y2),
  when( selectPlane( coeffType, row, columun ) == 1 && SRR <= 0 )
    result(number:
         Z=(Z11-Z01)*(TS-X1)/(X2-X1)+(Z01-Z00)*(CL-Y2)/(Y2-Y1)+Z01),
  when( selectPlane( coeffType, row, columun ) == 1 && SRR > 0   )
    result(number:
         Z=(Z10-Z00)*(TS-X2)/(X2-X1)+(Z11-Z10)*(CL-Y1)/(Y2-Y1)+Z10),
  when( selectPlane( coeffType, row, columun ) == -1 && SRF <= 0 )
    result(number:
         Z=(Z10-Z00)*(TS-X1)/(X2-X1)+(Z01-Z00)*(CL-Y1)/(Y2-Y1)+Z00),
  otherwise
    result(number:
         Z=(Z11-Z01)*(TS-X2)/(X2-X1)+(Z11-Z10)*(CL-Y2)/(Y2-Y1)+Z11);
```

ISBN 2-8318-6341-4

ICS 35.240.50

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND