Component Estimator (CompEst)
Interface Report
Version 1.1

Champaka Ramachandran
Fadi Kurdahi

Technical Report #93-xx
Feb 3, 1994

Dept. of Electrical and Computer Engineering
University of California, Irvine
Irvine, CA 92717
(714) 856-5106

champaka@ece.uci.edu

Abstract

This report provides instructions for interfacing with CompEst (Version 1.1). CompEst is a component estimator which provides the height, width and propagation delay of a GENUS component. The user can obtain a list of all possible height, width, delay tuples for the component for a large number of aspect ratios.

Contents

List of Figures

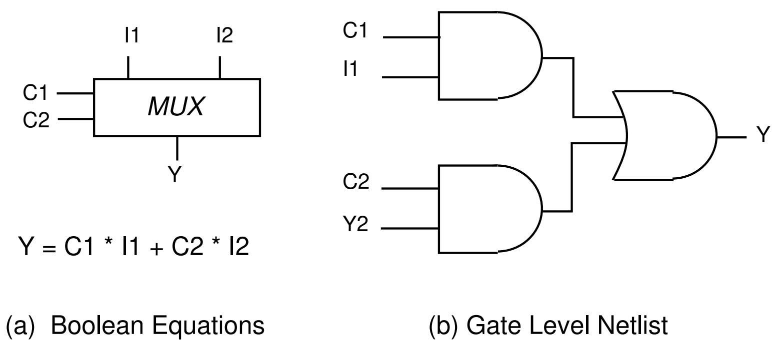

Figure 1: Component Description Formats

1 Overview

CompEst is a component estimator which produces the physical estimates and timing estimates of the component.

CompEst accepts a description of a GENUS component [1] written in VHDL or in EQN format. The detailed description of the GENUS components can be found in [1]. ALU, a register, a multiplexer are some example GENUS components. These components can be described using two formats. They can be described as a netlist of logic gates like and, or, nand, nor, using a VHDL structural description format or they can be described using simple *Sum of products* expression of boolean equations, using the EQN format [2] as shown in Figure 1.

Given the above description, the component can be implemented in layout using standard cell layout style. In this style, the logic gates in the description are layed out into rows and the nets connecting the gates are routed in between the rows. By varying the number of rows, the component can be implemented in different shapes as shown in Figure 2.

CompEst, produces height and width estimates for each of these different row implementations. This estimate is called the shape function of the component. Thus, given the description of the component, CompEst produces the shape function of the component. To produce the shape function, the number of rows is varied such that the height to width ratio of the component varies between 0.1 to 10. Thus the shape function contains points that have aspect ratios between 0.1 and 10.

In addition to the shape function, CompEst also produces timing estimates for the component. The timing estimates includes the worst case delay from an input to an output of the component. This worst case delay is dependent on the delay of the gates and the delay of the nets in the component description. The gate delay is dependent on the type of gate, however the net delay is dependent on the length of the nets. The length of the nets changes when the component is implemented in different shapes. Thus, when the shape of the component changes when the component is implemented with varying number of rows, the worst case delay of the component also changes. CompEst estimates the worst case delay for each of the row configurations.

This report provides, gives a brief description of CompEst System overview, its input files and the kind of outputs it produces. Section 3, illustrates the usage of CompEst through a simple example. The appendix gives the actual interface format that can be used to communicate with CompEst.

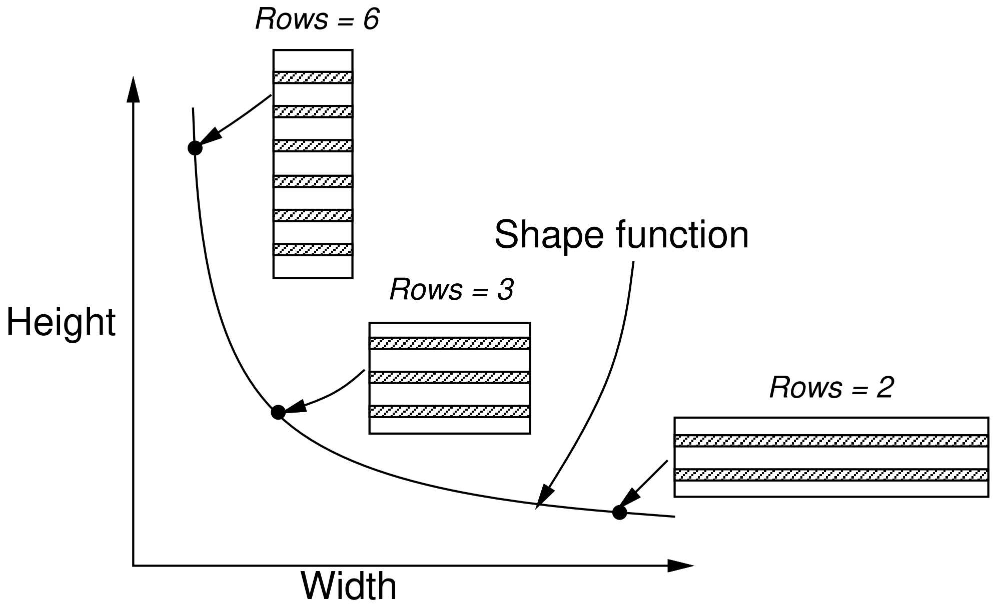

Figure 2: Component Shape Function

2 Component Estimation System

As was seen in section 1, CompEst reads the input description of the component produced by GENUS, and produces the shape function of the component and the worst case delay for all the points on the shape function.

The Component Estimator's interfaces with GENUS as shown in Figure 3. CompEst classifies all its input information into two categories, namely, *mandatory inputs* and *default inputs*.

Mandatory inputs are inputs without which CompEst cannot run. An example of the mandatory input is the component description file name. The *mandatory inputs* are specified in the *Required Inputs File*(RIF). The complete list of mandatory inputs can be obtained in Appendix A.

Default inputs are inputs which may or may not be specified. The specified *default inputs* are present in the *Defaults Modifier File* (DMF). If the *default inputs* are not specified, then CompEst uses its own the defaults values for these inputs. An example of the *default inputs* includes the file containing the characteristic information about the gates present in the component description. This is also called the gate library information. The complete list of these default inputs can be obtained in Appendix A.

The Component Estimator reads these two files and produces the shape function and worst case delays and stores the results in the *Results Output File*(ROF). The results consists of width, height, worst case delay tuples for each of the row configurations present in the shape function. The actual format of the ROF can be obtained in Appendix A.

The figure 3 shows the interfacing technique between GENUS and CompEst, which includes the RIF, ROF and DMF.

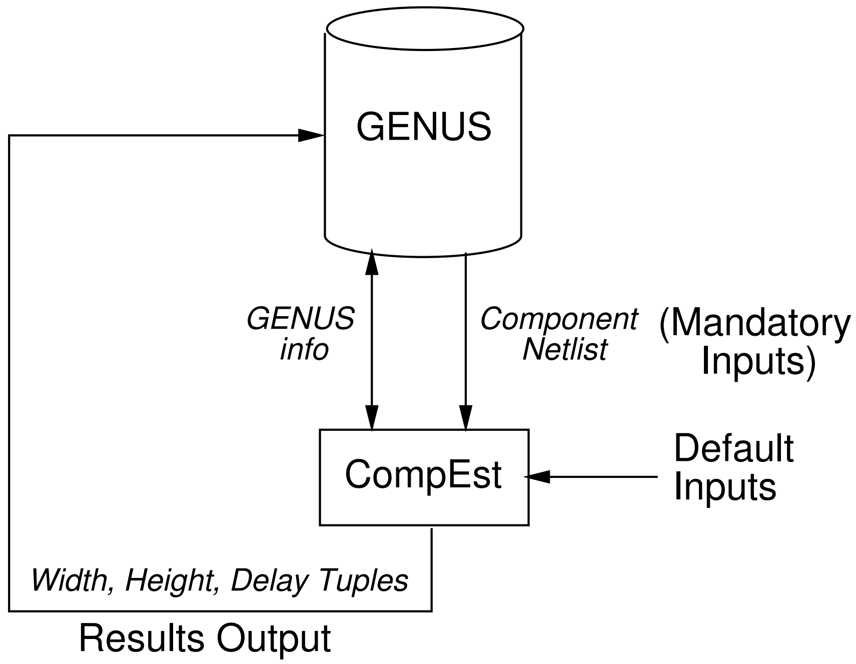

Figure 3: GENUS – Component Estimator Interface

3 Interface Files

This section will examine the information stored in each of the files shown in Figure 3. It will examine the contents of RIF, ROF and DMF as shown in Figure 4. The exact file formats can be obtained in Appendix A.

3.1 Required Inputs File

The RIF consists of three types of information, namely, component information, netlist information, and technology information.

The netlist information section consists of the component description file name and the description format which could be either a netlist of logic gates in VHDL format or a boolean equation description in EQN format.

The technology information section consists of information regarding the layout implementation style of the component. It indicates whether the component is implemented in layout as rows of standard cells, or FPGA etc. In case it is implemented as FPGA, the FPGA library name is specified.

The component information is for the sole purpose of ease of handling information with GENUS. This set of information is specified by GENUS so that CompEst can append this information along with the results. This set of information is not processed by CompEst. The set of information belonging here includes the compiler name of the component, component bit width, functionality of the component and the number of functions that the component implements.

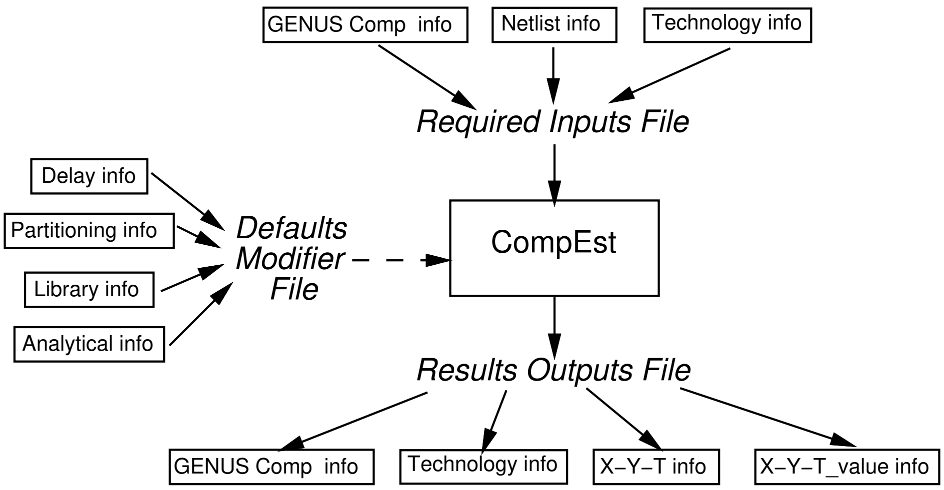

Figure 4: Details of Component Estimator Interface

3.2 Defaults Modifier File

The DMF has four types of information, which include delay information, partitioning information, analytical information and library information

The delay information includes the number of worst case paths that need to be computed for every shape function point and the type of delay analysis that need to be performed. One can choose between a fast, static analysis algorithm, or a slow and more accurate false path analysis algorithm.

The partitioning information includes information about the accuracy of the estimator. The higher the slicing level number, the more accurate the height and width values. It also includes user specified bounds of the number of rows the component can be implemented in, as a upper aspect ratio bound and a lower aspect ratio bound.

The library information describes the pathnames of the various libraries that CompEst uses. These include the library that contains the gate height and width characteristics, the technology library and the gate delay characteristics. The technology library contains some design rule information like the width of metal lines, the spacing between metal lines, the capacitance and the resistance of the metal lines and some standard cell technology specific information.

Finally, the analytical information includes some empirical constants which the user can change. This include a number that specifies the type of circuit as highly parallel, highly serial or as an intermediate one. The Rent's parameter specifies this nature of the circuit. One can also choose the formula that is used to calculate another empirical number called average wire length.

3.3 Results Output File

The ROF consists of four types of information. The component information and the technology information section are exact duplicates of the RIF and are present here so that GENUS can process this information easily. The other two sets of information include the X-Y-T information and X-Y-T value information.

The X-Y-T section stores the number of width, height, delay tuples and the units for measuring width, height and delay. The X-Y-T value section stores the actual width, height, worst case delay tuples.

4 Tutorial

This section illustrates the RIF and ROF for obtaining the shape function and worst case delay of a 4-bit ALU. The input description for a 4-bit ALU in EQN format is given below :

```
INORDER = CADD[0] I0[0] I0[1] I0[2] I0[3] I1[0] I1[1] I1[2] I1[3] ICIN[0];

OUTORDER = O0[0] O0[1] O0[2] O0[3] OCOUT[0] O1[0] OZERO[0]
OVF[0];

C0 = (CADD[0]*ICIN[0]);

X[0] = (CADD[0]*I0[0]);
X[1] = (CADD[0]*I0[1]);
X[2] = (CADD[0]*I0[2]);
X[3] = (CADD[0]*I0[3]);

Y[0] = (CADD[0]*I1[0]);
Y[1] = (CADD[0]*I1[1]);
Y[2] = (CADD[0]*I1[2]);
Y[3] = (CADD[0]*I1[3]);

C1 = (X[0] * Y[0]) + (C0 * (X[0] + Y[0]));
C2 = (X[1] * Y[1]) + (C1 * (X[1] + Y[1]));
C3 = (X[2] * Y[2]) + (C2 * (X[2] + Y[2]));
C4 = (X[3] * Y[3]) + (C3 * (X[3] + Y[3]));

Z[0] = C0 X̂[0] Ŷ[0];
Z[1] = C1 X̂[1] Ŷ[1];
Z[2] = C2 X̂[2] Ŷ[2];
Z[3] = C3 X̂[3] Ŷ[3];

O0[0] = Z[0];
O0[1] = Z[1];
O0[2] = Z[2];
O0[3] = Z[3];

OCOUT[0] = C4;
O1[0] = (0);
OZERO[0] = !(Z[0] + Z[1] + Z[2] + Z[3]);
OVF[0] = C4 Ĉ3;
```

The contents of RIF for obtaining the shape function of a 4-bit ALU is shown below :

```
%COMPONENT INFORMATION
GC_COMPILER_NAME : ALU
GC_INPUT_WIDTH : 4
GC_NUM_FUNCTIONS : 2
GC_FUNCTION_LIST : GC_ADD,GC_LNOT
GC_STYLE : GC_CARRY_LOOKAHEAD

%TECHNOLOGY INFORMATION
CE_TECH_NAME : FPGA
CE_LAYOUT_STYLE : BIT_SLICE
CE_LIB_NAME : Xilinx4000

%NETLIST INFORMATION
CE_NETLIST_FILENAME : alu-4bit.vhdl
CE_NETLIST_FORMAT : VHDL
```

The contents of ROF produced by CompEst for a 4-bit ALU is shown below:

```
%X_Y_T INFORMATION
CE_NUM_IMPLEMENTATION : 2
CE_LENGTH_UNIT : microns
CE_TIME_UNIT : ns

%X_Y_T
X-Y-T_VALUE : 10,20,30
X-Y-T_VALUE : 100.25,200.25,300
```

5 Appendix A

COMPEST Users Manual – 7 December 1993 ("U. C. Irvine")

CompEst – Perform Component Area and Timing Estimation

SYNOPSIS

CompEst [**-D** *Defaults* Modifier File Name] **-I** *Required* Inputs File name **-O** *Results* Output File Name

DESCRIPTION

CompEst performs a Area and Timing Estimation for the component whose netlist (or connectivity description in any other format) has been given.

CompEst uses two types of information, one is the mandatory information and the other optional modified defaults. CompEst produces the output in a Results Outputs File (ROF).

The mandatory inputs are present in the The Required Inputs File (RIF). The RIF consists of three sections, namely, component information, netlist information, and technology information.

The outputs are available in the Results Outputs File(ROF). ROF consists of either a status code if CompEst did not run to completion or the results of the CompEst run, otherwise. a successful completion. The RO file for a successful run consists of four sections, namely, component information, technology information, X-Y-T information and X-Y-T values. The component information and the technology information are exact replica of the information given in the RIF.

The defaults that need to be modified are present in the Defaults Modifier File (DMF). The DMF consists of four sections, namely, Partitioning information, Analytical Estimator information, Timing Information and the Library information.

OPTIONS

-D Defaults Modifier File Name This is an optional input. The options provides the path name of the file that contains information that would need to redefine any default entries.

-I Required Inputs File Name This is a essential input option. This option provides the path name of the file that contains information that are mandatory for running CompEst.

-O Results Output File Name This provides the path name of the file containing the results.

FILE FORMAT

Each section consists of a header line followed by a list of field name-value pairs, one per line. The header line begins with a "

Each field line consists of a field name, a separator ("COLON"), and a field value. Field names in component information section in RIF are GENUS names with "GC_" prefix. Field names in the other section are CompEst names with "CE_" prefix. Field values could be a single word or a list of words separated by comma delimiter.

FILES

Required Inputs File

The RIF consists of three sections, namely, component information, netlist information, and technology information.

The netlist information section has two fields (in this order):

CE_NETLIST_FILENAME : unix path name /* file name */
CE_NETLIST_FORMAT : VHDL / EQN /* netlist format */

The technology information section consists of three fields (in this order):

CE_TECH_NAME : Character string /* technology library name */
CE_LAYOUT_NAME : Character string /* layout style */
CE_LIB_NAME : Character string /* gate library name */

The valid field names for the component information section are :

GC_COMPILER_NAME : Character string /* component name */
GC_INPUT_WIDTH : Character string /* input width */
GC_NUM_FUNCTIONS : Character string /* number of functions */
GC_FUNCTION_LIST : Character string /* function list */
GC_STYLE : Character string /* function style */
GC_FILE_PATHNAME : Character string /* file name */

Results Output File
The ROF consists of four sections. The component information section and the technology information section are exact duplicates of the RIF. The X-Y-T information and X-Y-T value section stores the shape and delay of the component.

The valid field names for the X-Y-T information section are :

CE_NUM_IMPLEMENTATION : Int /* number of implementation */
CE_LENGTH_UNIT : Character string /* wire length unit */
CE_TIME_UNIT : Character string /* time unit */

The X-Y-T value section has a list of three valued tuples, with one tuple per line. An example tuple is : 10.5,20.3,300

The valid field names for the X-Y-T value section are :

X-Y-T_VALUE : Float,Float,Float /* X, Y, T */

Defaults Modifier File The DMF has four sections, which include delay information, partitioning information, analytical information and library information

The valid field names for the Delay information section are :

CE_ANALYSIS_TYPE : STATIC_PATH/FALSE_PATH /*analysis type */
CE_NUM_WORSTCASE_DELAYS : Int /*number of worst case */
CE_INCLUDE_LINEDELAYS : Yes/No /* include line delay or not */

The valid field names for the partitioning information section are :

CE_SLICING_LEVEL : Int /* number of slicing level */
CE_UPPER_ASP : Float /* upper aspect ratio */
CE_LOWER_ASP : Float /* lower aspect ratio */

The valid field names for the analytical information section are :

CE_RENTS_PARAMETER : Float /* rent slots */
CE_WIRELENGTH_FORMULA : FEUER/GURA /*wirelength formula */
CE_MAX_STD_ROWS : Int /* maxium rows */

The valid field names for the library information section are :

CE_TECH_LIB : file path name /* technology library name */
CE_GATE_TIMING_LIB : file path name /* timing library name */
CE_GATE_SHAPE_LIB : file path name /* gate shape library name */

6 References

[1] N. Dutt and P. Jha, "Genus: A generic component library for high level synthesis," tech. rep., ICS Department, UC Irvine, 1992.

[2] R. Brayton *et al.*, "MIS: a multiple level logic optimization system," *IEEE Trans. CAD*, vol. CAD-6, pp. 1062–1081, Nov. 1987.