Engineering Simulation, 1999, Vol. 16, pp. 541–545
Reprints available directly from the Publisher
Photocopying permitted by license only

Processing Method for Bi–directional Lines in Logic Simulation of Discrete Devices

A. I. ANDRYUKHIN

Institute of Applied Mathematics and Mechanics of the National Academy of Sciences of Ukraine, Donetsk.

(Received March 12, 1997)

The method of bi-directional line processing is considered in logic simulation of discrete devices which provides their compact description and convenient form of simulation results.

Key words: logic simulation, bi-directional lines, bus element.

A wide development of microprocessor technique resulted in the necessity of automation of bi-directional line simulation. Its peculiarity is the use of bus structures. The activity of elements on bus may be controlled by special microcircuits, the so called bus arbiters [1]. But in practice it is controlled using elements of small and average integration.

A widely used algorithm of event-driven simulation [2,3] operates in its general formulation with the notions of element inputs and outputs and consequently, assumes the one-directional signal propagation. The construction of a device model taking into account the possibility of signal transmission in both directions, demands the understanding of the device operation. It involves the doubling of these lines and inclusion of an additional fictitious valves [3], i.e. an additional work and non-coincidence of structural model description in electric schematic diagram of device.

In [4] the data structures are described which permit us to realize the event-driven simulation taking into account bi-directedness of some outputs of integral microcircuits and some other peculiarities. But here the necessity arises in correcting software realization of simulation algorithm which is made for one-directed signal transmission. The progress in technology (in particular, functional electronics) can result in the elements appearing with properties which are not represented in rigid tables of interpreting realization systems.

In an hierarchical compilation simulation system (HCSS) [5] the algorithm of event-driven simulation is realized which makes it possible to simplify the scheme description and to automatize the simulation process. It is made together with syntactical and semantic analyzer of device description and reference information of the system. Thus one of the requirements to simulation systems is satisfied — the

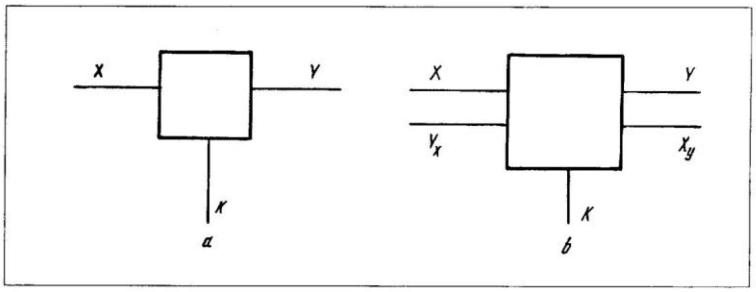

Figure 1. Transform of logic function realization.

realization methods of different logic functions of join should be implemented, namely, wire AND, wire OR, the join which is peculiar for systems with a common bus when only one output is active and others should be in a state with high output resistance [4, 6].

One of the main defects of the compilation approach to realizing the systems of logic simulation, namely, high time costs by changing the device design, became less appreciable due to development of compilators of Turbo family. As it is possible to quickly obtain the necessary data structures for processing elements with new properties by a fixed simulation algorithm, the changes in the system can be much more easily inserted. Here the main work is in creating the program of element operation. This property is important taking into account rapid progress in integral technique and functional electronics. This approach was used in automation of elements with bi-directional outputs inclusion into event-driven simulation algorithm without changing the latter. The algorithm involves two main elements: a) construction of element models with bi-directional outputs; b) construction of one-digital bus program.

The operation of one-directional element is described by its program which determines new values of internal variables and output terminals of the element. The necessary parameters of the program are the fields of input signal values and internal variables of the element describing the state of the latter. These fields are formed by the algorithm.

The program is an instruction which describes the functions of transitions and outputs of automatic element model. The construction of an element model with bi-directional outputs is based on the possibility of representing the operation of these elements as the appropriate one-directional models for using them in event-driven one-directional algorithm. Consider the element E which has X ($\underline{Y}$), the set of purely input (output) terminals. $\overline{W}$ is the set of bi-directional terminals, $\underline{S}$ the set of internal state variables. The operation of E element may be described by the expressions

$\overline{Y} = Y(\overline{X}, \overline{W}, \overline{S})$, $\overline{W} = W(\overline{X}, \overline{W}, \overline{S})$, $\overline{S}^{+} = S(\overline{X}, \overline{W}, \overline{S})$, where $\overline{S}^{+}$ ($\overline{W}^{+}$) are the new values of the state variables and bi-directional terminals, and $\overline{Y}$ are the values of E outputs. The set of outputs and inputs are designated as $\overline{Y}' = (\overline{Y}, \overline{W}^{+})$ and $\overline{X} = (\overline{X}, \overline{W})$, and output and transition functions as $Y' = (Y, W)$ and $S' = S(\overline{X}, \overline{W}, \overline{S})$, then the one-directional element E' is defined. But the number of inputs and outputs is increased by $|W|$ compared to E. Therefore it should be shown in reference information about an element with bi-directional terminals. This is achieved by double description of the bi-directional element contacts: input and output ones.

The information is also necessary about controlling contacts of elements of various types with bi-directional terminals. When the program element module with bi-directional terminals is being written, it should be taken into account that the doubling of bi-directional contacts in reference information results in their appearance in the field of input signals and in the necessity of definition of their new values in the field of output terminals. So, for the key in Fig, 1,a, which operation is described by the expressions $Y = T(K, X, Y)$ and $X = T(K, X, Y)$, where $T(K, X, Y)$ is X with Y and is Y with $K=0$, we have the form shown in Fig. 1*b*. It is obtained after its contacts doubling in reference information. Here $X, Y_x (Y, X_y)$ are the inputs (outputs) of the element. The operation of this element can be described by the expressions $Y = T(K, X, Y_x)$ and $X_y = T(K, X, Y_x)$. Taking into account that the value Y_x is the old value Y, following the algorithm and comparing old and new values of the key outputs, we include in a queue of active elements the successors of contacts X or $K = 1$ if they change their values. We should add that the lists of contact predecessors $X(Y)$ coincide with those of contact successors $X_y (Y_x)$, which are bus lines or element outputs with the third state for bi-directional terminals.

On the basis of an approach used in a model of bi-directional key, the models of elements are built where the controlling element contacts play the role of generated key. They determine the activity of element and transfer direction. The main classes of digital elements have been analyzed which are directly connected with buses. These are: a) the element with outputs which have the third state; b) the element with bi-directional informational contacts and controlling contacts OE (the control of connection with bus) and T (the control of transfer direction); c) the element with bi-directional informational contacts and controlling contacts CS (control of connection with bus) and R(W) (control of transfer direction, i.e. reading or writing). The examples are microcircuits К555АП3, К589АП26 or К580ВА86, К580ВИ53, respectively.

The construction of the bus element model and implementation of its operational logic necessitates the inclusion of additional data and changing the pre-processor. The basis of models of logical schemes are the information of the device elements and interconnections. In *HCSS* they are tables ELEMENTS and INTERCONNECTIONS used for program generation in C language which

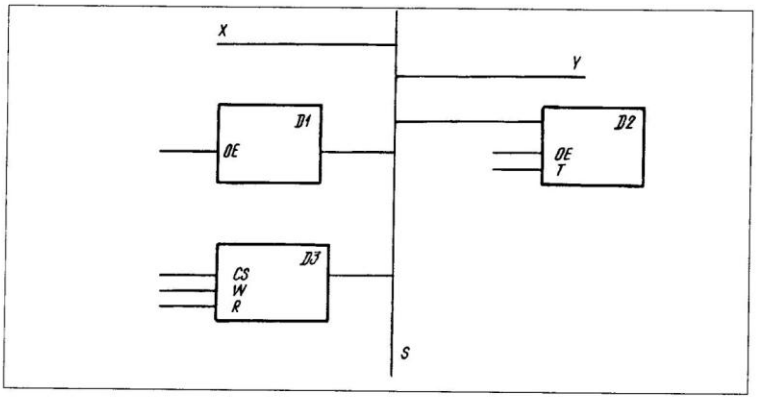

Figure 2. One-digital scheme with switched elements.

simulates the appropriate device. Table ELEMENTS of the scheme for each i-th element out of N available contains the line of data of the $\langle T_i, Q_i, D_i \rangle$ kind, where T_i is the sign of the element; G_i is the name of element by standard (type); D_i is the name of element in device. Table INTERCONNECTION is the set of lines of $\langle DQ_j, KQ_j, DI_i, KI_i \rangle$ kind, where $DQ_j\,(DI_i)$ are the numbers of the device elements interconnected by the contacts KQ_j and KI_i. On the basis of the above tables and reference information about the device elements the program-pre-processor builds the program in C language, where it generates the data structure necessary for each element (for example, the lists of predecessors and successors of element contacts).

Fig. 2 presents the one-digital bus S with switched elements $D1, D2, D3$. The activity of these elements on bus is determined by the values of controlling contacts designated by the symbols taken in practice: $-OE, T, CS, W, R$. For defining the value on the bus it is necessary to have the values of element contacts $D1, D2, D3$, connected with the bus and the values of their controlling contacts. To do this, it is necessary to generate for each bus element the list of elements-successors which are directly interconnected. This is performed in the scheme description in addition to tables ELEMENTS and INTERCONNECTIONS. The list of elements-successors of bus element represents the data structure $\langle i, k, I\,(i, T_i) \rangle$, where T_i is the sign of the element i; k is the number of element contact i directly connected to bus and $I\,(i, T_i)$ are the data describing the controlling contacts of the element i. Just the information $I\,(i, T_i)$ made it possible to uniformly process the bus elements without changing the simulation algorithm.

In the program realization of the bus element module the following parameters are given: a) the address of the value array for the device element terminals; b) the queue of the active scheme elements and its characteristics (the quantity, the number of the first free element in the queue etc); c) the address of the array of bus elements-successors.

The algorithm of bus operation involves the calculation of the activity property $S_i = S(k, I(i, T_i))$ for each bus element-successor and the value of its output on bus $V_i = V(k, I(i, T_i))$. For verification of logical design, the information about the state on bus is given, if the number of active elements is more than one. After definition of active element, when the value of its active output on bus is equal to the old bus value, the program of bus element finishes its work. Otherwise it includes the non-cut-off elements- successors in the list of active elements and finishes the work giving the control to properly event-driven monitor. All the above operations provide the form of simulation object description convenient for users and visualization of its results.

REFERENCES

1. Shevkoplyas B. V. *Mikroprotsessornye struktury. Inzhenernye reshenia:Spravochnik.* (Microprocessor structures. Engineering solutions. Reference book). — M. : Radio i svyaz, 1990. — 512 p. (In Russian).
2. Savelyev P. V., Konyakhin V. V. *Funktsionalno-logicheskoe proektirovanie BIS* (Functional-logical design of LSI)/Ed. G. G. Kazenkov. —M.:Vysh. shk.,1990. — 156 p. (In Russian).
3. Ioffe M. I. *Diagnostirovanie logicheskikh skhem. Algoritmy odelirovaniya i avtomaticheskogo sinteza testov* (Diagnosis of logical schemes. Algorithms for simulation and automatic test synthesis). — M.: Nauka, 1989. — 158 p. (In Russian).
4. Ivannikov A. D. *Modelirovanie mikroprotsessornykh ustroistv* (Simulation of microprocessor devices). — M.: Energoatomizdat, 1990. — 144 p. (In Russian).
5. Andryukhin A. N., Speranskiy D. V. *Ierarkhicheskaya kompilyativnaya sistema modelirovaniya i generatsiya testov* (Hierarchical compilation system of simulation and test generation) // Tekhn. diagnostika i nerazrushayushiy kontrol. — 1994. — No 2. — p. 71—78. (In Russian).
6. Armstrong J. R. *Modelirovanie tsifrovykh sistem na yazyke VHDL* (Chip-level modelling with VHDL). M.: Mir, 1992. — 175 p. (Russian translation).