Skew-Free Clock Distribution for Standard-Cell VLSI Designs

Gerard M Blair

Department of Electrical Engineering
University of Edinburgh
The King's Buildings
Edinburgh, EH9 3JL
Scotland, UK

ABSTRACT

The control of skew in globally distributed signals is essential to the safe design of VLSI. This paper describes a novel technique using Cadence EDGE software for the creation of a skew-free distribution network in standard-cell designs which is achieved after initial placement by modification of the netlist and insertion of buffers whose drive strength is finally determined after the component is completely routed.

1. INTRODUCTION

The increase in complexity of VLSI components has led to an increase in the problems of clock signal distribution where the larger number of clocked subcells on a single component has resulted in a high relative fanouts for the clock circuitry. If no corrective design is undertaken, this leads to long switching delays, which reduces the overall circuit speed, and poor clock edges, which cause some types of latches to fail.

The first step is to buffer the clock signal in a chain of buffers of increasing size to attain the optimum speed for driving the given capacitance. Hedenstierna jeppson 1987 However, if this final signal is distributed through out the component on a single track, it can lead to high currents requiring wide tracks to prevent breakages due to electromigration; and also to clock skew introduced in the track due to transmission line effects over the long resulting lengths (of the order of several centimetres).

A solution is to distribute the buffering at each level into several smaller buffers which drive independent nodes. Friedman powell 1986 Thus the current flowing from the final stage of buffering is distributed over the distinct nodes and at a level with can be supported by the technology. Similarly, the final track capacitance is distributed between the final buffers. However, the independence of these buffers may lead to different delays along the distinct distribution paths from the common, original clock signal. This may cause clock skew with the danger falsely of clocking data transferred between latches which are adjacent but driven by different, final clock-buffers.

There are two main methods to overcome this problem. The first method is to increase the system's tolerance to possible skew by introducing multi-phase clocking schemes. weste 1985 These remove the problem by ensuring that adjacent latches are not changing at the same clock phase, and so there is no danger of falsely latching a changing value. However, a single phased clocking scheme offers the advantages of a simpler design style and reduced routing overheads in comparison with multiphase approaches. afghahi 1990

The second method is to use Computer Aided Design (CAD) tools to minimize the clock skew by careful placement and routing (P&R) of the buffer hierarchy, boon 1989 kuh 1990 and to distribute buffers on a row by row basis of the standard cell placement with buffer sizes matched according to the number of clock inputs in each row. saigo watanabe 1990

The emphasis so far has been on the use of either the design structure or on careful control of the P&R algorithms to minimize skew. The approach of this paper, however, is to focus instead upon the layout itself as produced by existing Placement and Routing (P&R) software and to render the skew negligible by augmenting the netlist with variable sized buffers.

2. DESIGN SEQUENCE

The automatic buffering scheme is aimed at a standard-cell design environment and provides each input for a given net with a skew-free signal by building a fully balanced distribution network. This is achieved by modification of the cell- and net-lists after placement and modification of buffer drive strengths after routing.

From the designer's point of view, this scheme allows the completion of the design using global nets without any considerations for the buffering or distribution of these signals, and the verification of the design by simulation (without interconnect capacitances) so that the clock distribution simply becomes part of the P&R operation. The user performs the initial placement and calls the buffering software which prompts for the name of each of the global nets. Distribution networks are then created and the user proceeds with the normal operations to the completion of routing. Finally, the second phase of the buffering scheme is called which modifies the previously created buffers according to the actual interconnect capacitances without altering the existing placement or routing.

2.1. CAD prerequisites

The procedure is implemented in the Cadence EDGE software which was made available under the EUROCHIP initiative. The procedure relies upon several features of that suite:

1) The use of the high-level data-base language: SKILL, Skill 1990 DAC the Cadence compromise between LISP and C.

2) The ability of SKILL to interrogate and modify both the cell-list and netlist of the design during P&R.

3) The parameterization of the layout of cells allowing the design of a family of buffers whose drive strength may be fine-tuned by a parameter set *after* final placement and routing.

2.2. Outline of procedure

The following describes the procedure in terms of the distribution of a single (clock) signal although it should be repeated in a design for all widely distributed clock and control signals.

The user enters the design using schematic capture with the used clock signal designated as a single, global net. Care must be taken during the design verification by simulation to ensure that the high capacitance loading on the global net does not alter the simulation results. Typically, this could be done by driving the net explicitly as an external input until the final design step when the (unique) driver is connected to the net.

The design then enters P&R and the placement proceeds as normal except that the priority of the global net is reduced so that it does not perturb the placement algorithm. After placement the first part of the buffering scheme is run:

1) The number of cells driven by the global net is counted and an appropriate buffering hierarchy is determined.

2) The cells driven by the net are clustered according to their position in the layout into equally sized groups.

3) To each group is assigned a unique buffer and the netlist is updated so that these cells are driven by the output of that buffer.

4) Each new buffer is added to the placement database so that it is physically close to the cluster it is driving.

5) Steps 2-4 are then repeated for each level of created buffers until a complete distribution tree is built.

6) Finally the netlist is modified so that the last level of buffers is driven by the original global signal.

The user now continues with the standard Routing program, either routing the design as normal or giving preference to the nets on the distribution network. After the design is fully routed, the second part of the buffering scheme is run:

7) The interconnect capacitances for the generated sub-nets are extracted and a common ratio of output capacitance to input capacitance (that is, drive strength) is derived for each independent level of the distribution tree.

8) A parameter is then added to the instance of each buffer network buffers which modifies the transistor sizes in accordance with the actual capacitance it drives *without altering the placement or routing* so that the delays are equalized across each level of the network.

3. IMPLEMENTATION

The general objective is to create a buffer tree as shown in figure 1 where the overall size of the buffers increases at each layer (to minimize total delay) and the sizes of individual buffers are adjusted according to the final output capacitance (to equalize the delay across each level). This section will describe the details of the procedure which rely most heavily upon the EDGE software.

3.1. Clustering

The standard cells are initially placed by the standard EDGE routines and the automatic buffer scheme is then called.

Firstly, a minimum spanning tree is generated for the nodes driven by the global net. This is used as a the basic framework for clustering these nodes into groups to be driven by independent buffers. The generation of the minimum spanning tree using the metric of rectilinear distance corresponding to the Manhattan routing style. The algorithm of the prototype system is currently quite crude since even large bit-serial DSP chips have no more than about 3k cells driven by the same clock blair 1991 esscirc and the procedure should not need to be repeated many times in any design.

The number of inputs which the designated signal must drive is used to derive the size of the groups into which the nodes are clustered and thence the expected size of the buffer at this level. The clusters are formed by coalescing the nodes in the reverse order to which they were added to the spanning tree. When a node is as large as, or larger than, the cluster size, that number of nodes are selected to form a single cluster and the rest are added back to the remaining tree.

For each cluster, a new net is created and connected to the node of each cell in the cluster. A new instance of a buffer is created and its output is connected to the new net, while its input is connected to the net which originally drove the cluster. Finally, the instance is added to the standard-cell array in the locality of the cluster it is driving. This whole procedure is then repeated for each level of the distribution tree until the final level alone is driven by the original signal.

The partitioning of the network by minimum spanning tree contrasts with other authors who divide the layout array into simple blocks. The difference is that the previous systems rely upon symmetric paths to each clusters to minimize the variation in interconnect capacitance; the procedure reported below compensates for any variation by modification of buffer size, and so facilitates a more general (and hierarchical) solution.

3.2. Buffer Design

The P&R routines in EDGE work upon the *abstract* view of the standard cells. This is derived from the layout and is effectively the bounding box and the positions of the input and output I/O pins. After a component is completely placed and routed, the abstracts are replaced by their corresponding layout.

The buffers are actually taken from a whole family which must be designed specifically for each standard-cell library. There are two degrees of variability: the size of the abstract for P&R, and the actual size of the transistors after the layout replaces the abstract. Clearly the size of the abstract limits the size of the final transistors.

An example of a buffer abstract is shown in figure 2 showing the bounding box and pin position information (I/O and pwr/gnd). The height of the abstract must be the same as the common height of all the standard-cell library; the family of abstracts correspond to the various widths which provide a range of maximum-drive options.

Once the abstract size has been selected in the procedure and it has passed through P&R, neither the size nor the positions of the I/O pins may be altered, since this would break the connections established in the component layout. However, the other features within the abstract may be freely altered. Specifically, the transistor widths of the buffer may be altered to provide any desired drive strength, within the limits imposed by the abstract, by the setting of a parameter associated with the abstract instance in the P&R environment.

Figure 3 shows the simplified layout for three different drive strengths corresponding to the same abstract and parameterized layout. The Cadence software allows both the stretching of features (as in the diffusion and polysilicon) and the conditional repetition (as with the contacts). In this example, the p-type transistors are simply twice the width of the n-type but other relations can be easily encoded. The use of horizontal polysilicon gates allows the transistor widths to be varied with as small a granularity as the layout grid.

The abstract size must be selected when it is instanced before being inserted into the array and routed. The output capacitance for a buffer is estimated from the number of nodes it must drive (the cluster size), the likely value of the interconnect capacitance, and an additional margin for error; this is then used with the chosen ratio of output to input capacitance to estimate the buffer transistor sizes, and the next smallest abstract to accommodate these sizes is selected. Thus the aim is to provide the necessary drive strength while minimizing area. With these buffers inserted into the array and netlist, the routing proceeds to completion.

For each buffer, the *actual* output capacitances is extracted from the routed design. For each level in the signal distribution hierarchy there is an ideal C_{out}/C_{in} ratio (R) calculated in terms of a speed/area trade-off.

At each level, the feasibility of achieving the ideal ratio is checked by the relationship:

$$\frac{C_{out(max)}}{C_{in(max)}} \leq R \leq \frac{C_{out(min)}}{C_{in(min)}}$$

The output capacitances are now fixed and the maximum and minimum are taken over all buffers on the given level; the input capacitances are variable within the bounds of the fixed abstract. If this relationship holds then R is used, otherwise the value of the condition it violates is used as the common ration.

The final stage is to adjust the actual transistor size in each buffer by selecting each abstract instance in turn and setting the *size* parameter to alter the transistor sizes of the actual layer so that $C_{out(actual)}/C_{in}$ equals the common ratio for each corresponding level.

4. Concluding Remarks

The elegance of this approach stems from the fact that the buffer sizes may be altered until skew is removed, without affecting the remaining layout. Thus, if simulation reveals skew on the first pass, the sizes may be altered accordingly until the skew is negligible. The only limit on this is the "rounding error" associated with the layout's grid-size which might typically be in the region of 1-2% of the size and hence the delay at each stage.

This buffering scheme provides a skew-free distribution network for general standard-cell VLSI designs and has been embedded within an industry standard CAD tools suite. The use of post-routing modification of the layout allows for the creation of a truly balanced signal distribution network which takes full account of the variations in interconnect capacitance on the independent signal paths. With this system, simple single-phase design strategies may be adopted without the inherent problems of clock distribution.

Unfortately the figures for this paper have been lost in the mists of time (well since 1992) - the original paper may be found in:

G.M. Blair; "Skew-Free Clock Distribution for Standard-Cell VLSI Designs", Proc. IEE Pt. G, 139, No 2, pp265-268, April 1992.

G.M. Blair, E.G. Friedman(Ed); "Skew-Free Clock Distribution for Standard-Cell VLSI Designs", Clock Distribution Networks in VLSI Circuits and Systems, IEEE Press, 1995, (reprint of paper) ISBN 0-7803-1058-6.