DIGITAL CIRCUITS DELAY ANALYSIS

Miljana Sokolović, Milunka Damnjanović, *Faculty of Electronic Engineering, University of Niš,*

Abstract − *The speed of digital circuit is one of the most restricting factors in the deep sub-micron and multi-gigahertz integrated circuits design. It is directly dependent on the circuit delay. Scaling down the technologies and increasing the operational frequency make the delay problems more important. This paper studies the delay in digital circuits, starting from the design process, scaling down the dimensions of base elements, through the delay models and delay calculation techniques. Finally, some comparisons of the delays for the particular technology are given.*

1. INTRODUCTION

The design process of one VLSI chip has a number of steps. First, the design is described at the behavioral level. It is followed by functional partitioning, architectural synthesis, logic synthesis, floorplanning, placement and routing. It ends with a detailed specification of the mask-level circuit layout. In each of these steps, different techniques can be applied to increase performance or reduce power consumption or delay. The design representation can go through the series of transformations that should improve the design quality [2]. Nevertheless, every wrong decision made at a high level of abstraction may bring the design so far from some optimal solution that no subsequent optimization can recover it [2].

Since the delay nowadays has the increasing role in overall circuit performances [1], the accurate estimation of the delay in a digital circuit raised the importance. In order to complete each design stage successfully, we need to have a detailed view to the delay characteristics of every part of the circuit after every design stage. This also needs proper delay modeling and proper techniques for delay calculation.

A short overview of most popular techniques for modeling and calculating the delay in digital circuits is presented onwards. They consider different effects and influences (technology, user requirements, design technique, scaling, etc.) and have different accuracy. Based on their application, a conclusion is made about the level of those influences to the overall circuit delay. Those conclusions can be later used for characterizing the accuracy of delay calculation and modeling techniques at different parts of the circuit and different stages of the circuit design. This analysis is necessary if we want to improve the delay modeling and calculation, and though the design itself.

In the following section, a list of delay types is presented. After, some delay modeling techniques and most important delay calculation methods are listed. When a specific design tool is used, all delays are kept in a file in the **standard delay format** (.sdf) [3]. This file should be generated after each design phase. One such file is discussed, also. After that, the attention is focused on the influence of technology scaling down on circuit parameters and delay. Some comparisons of the different delay calculation methods are given after that, aiming to give a clue about the importance of considering different factors while estimating the delay during the design. This is followed by some practical technology and design improvements that can help to overcome the potential timing problems.

2. DIGITAL CIRCUIT DELAY TYPES AND MODELS

The delay is defined as a time interval between the event occurrence at some input and the change of the output signal caused by that input event [4]. The overall delay of each digital circuit has two components:

- active gate delay
- interconnect delay

Along with the improvement of the integrated circuit technology, the emphasis is moving from the gate delay modeling to the interconnect delay modeling. There are a number of delay models. They differ in accuracy and purpose: different calculations need different mechanisms incorporated into the delay description.

Considering logic simulation of the digital circuit, the simplest delay model for a single gate is 'no delay'. Gates described with zero delay are only used to verify the logic function by the simulation. The next, also very simple delay model is the 'unit delay'. It assumes that both rising and falling edges at the output are late exactly one time-unit related to the input signal change. The delay model that define different delays for the rising and falling edges of the output signal is more complex but more realistic. Due to the tolerances in circuit manufacturing process and the environmental variations, this model needs to be extended to deal with a delay range of a particular signal-edge, instead of exact delay. It means that the delay of the rising or falling edge of the output signal can be found between its minimal and maximal value. Further delay model development introduced the load of the particular gate. If a single gate drives one gate-input, it is faster than when it drives two or more gate-inputs. This approaches to the fan-out dependent delay model. To use this delay model the netlist of the complete digital circuit must be known. The delay of an edge of the output signal depends also on the cause of this output transition. Two and more events at the inputs of the gate can cause some event at the output of that gate. It further means that the same edge of the output signal can have different delays depending on the cause of that change. Figure 1 gives gate delay modeled as a look-up table which is a fan-out dependent delay function for a 3 input NAND gate in MTC45000 technology. Three different rise-edge delays at the gate output Z originate from the events at different inputs. Since the delay function is not linearly dependent on the number of fan-outs, it can also be concluded that loads for each gate are not purely capacitive. The loads probably have a resistive part due to the interconnections [5].

Table 1. *Delay model for a MTC45000 library NAND gate*

TIMINGS nS @typical P, 3.30v, 25.0C, 7.89fF SL (output load), 0.500nS SS (input slope)

Timing/load		1.0xSL	5.0xSL	10.0xSL	25.0xSL	maxLoad
A to Z	Fall delay	0.12	0.21	0.30	0.53	0.62
A to Z	Rise delay	0.15	0.24	0.33	0.61	0.72
B to Z	Fall delay	0.12	0.19	0.28	0.52	0.60
B to Z	Rise delay	0.17	0.25	0.34	0.61	0.71
C to Z	Fall delay	0.10	0.18	0.26	0.50	0.58
C to Z	Rise delay	0.19	0.26	0.35	0.61	0.71

As discussed, all described delay models are related to the logic simulations. It should be mentioned, that since logic simulations are used for description or verification at the high level of the design, the interconnect delays are not taken into account here. If we want delay models to have different purpose, those models should consider other factors. The

delay modeling is also very important in the timing analysis. **Static timing analysis** (STA) is a method of computing the expected timing of a digital circuit without requiring simulation [6]. It uses the delay models that incorporates gate delays as well as the circuit netlist. Nevertheless, the behavior of an electronic circuit is often dependent on many factors in its environment like temperature or power supply variations. Then STA must be prepared to work with a delay models that represent range of possible delays for each component.

The further improvement in the analysis leads to more complex delay modeling. Statistical STA (SSTA) is a procedure that is becoming increasingly necessary to handle the complexities of process and environmental variations in integrated circuits [6]. Since the STA represents a technique for determining only timing information of the circuit, without considering its correct logic function, it can be imagined as a simplified simulation. In the SSTA, gate delays should be randomly chosen with a given probability function (for example Gaussian), many times in order to simulate real parameter variations of the gates from mass production. The delay models here should contain random-number generator that will generate different gate delays for each of the number simulations. In [7], [8] a new method is proposed that modifies standard logic simulator to perform delay corner-analysis, STA and SSTA.

3. DELAY CALCULATION

Before the presentation of different delay calculation techniques, we need to consider some important definitions that can help understanding them [6].

- The *path* in a digital circuit is defined as any signal route that connects two circuit nodes, with respect to the unilaterality of the signals in the circuit. There can be a large number of signal paths between two circuit nodes. There can even be more than one path between two circuit nodes, and that is referred to as reconvergence.
- The *critical path* is defined as the path between an input and an output that has maximal delay. When one of the techniques for timing analysis computes the circuit timing, the critical path can easily be found by tracking back.
- The *arrival time* of a signal is the time that signal-change needs to arrive to a particular point. To calculate this time, delays of all components and wires of the path must be known. Arrival times, and indeed almost all times in timing analysis, are normally kept as a range between the earliest and latest possible time at which a signal can change..
- *Required time* is the latest time at which a signal should arrive without need the clock cycle longer than desired. The computation of this time is done in the following manner. At each primary output, the required rise/fall times are set according to the circuit specifications. Next, a backward tracing through the topology is carried out, processing each gate and calculating the required times at all of its fanouts.
- The *slack* associates with each connection, and represents the difference between the required and arrival time. A positive slack at a node means that the arrival time at that node may be increased by the obtained slack value without affecting the overall circuit delay. On the other hand, negative slack implies that a path is too slow, and that it must be speeded up (or the reference signal delayed) if the whole circuit should work with a required speed.

There are many methods for gate delay calculation. The choice depends primarily on required speed and accuracy [6].

- The slowest, but the most accurate method is the SPICE simulation.
- For applications like logic synthesis and placement and routing two dimensional tables are commonly used. These tables take into account the input slope and the output load, and then generate a circuit delay and output slope.
- A quite simple method that uses so called K-factor model is often used. This approximates the delay as (constant) + k · (load capacitance or the fan-out).
- One method that is more complex uses Delay Calculation Language DCL [9]. The user-defined program calls the DCL whenever a delay value is required. This allows arbitrarily complex delay models to be represented. However, this can make significant software engineering problems.
- Logical effort provides a simple delay calculation that accounts for gate sizing and is analytically tractable [10].

Similarly, there are many ways to calculate the delay of a wire. The most common methods increasing accuracy and decreasing calculation time are [6]:

- In the lumped C method, the entire wire capacitance is applied to the gate output, and the delay through the wire itself is ignored.
- Elmore delay [11] is a simple approximation, often used where faster calculations are required, and where the delay through the wire cannot be ignored. It takes into account the R and C values of the wire segments into a calculation. The delay of each wire segment is (R of the segment) · (downstream C).
- Padé approximation is a more complex analytical method. There are many variants of this method such as AWE [12], PRIMA [13] and PVL as more recent and sophisticated. These methods are faster than circuit simulation and more accurate than Elmore.
- DCL, as defined above, can be also used for interconnect delay calculation.
- Circuit simulators such as SPICE can also be used. As indicated, it is the most accurate, but slowest method.

It often makes sense to combine the delay calculation for a gate and all the wire connected to its output. This combination is called the *stage delay*. The delay of a wire or a gate can also depend on the behavior of the nearby components. This is one of the main effects that are analyzed during signal integrity checks.

4. STANDARD DELAY FORMAT

The Standard Delay Format (SDF) is a textual file format for representing the delay and timing information of electronic systems in different design stages. Both human and machine can read it, but in its most common usage it will be machine written and machine read as the support for timing analysis and verification tools, and for other tools requiring delay and timing information [3]. The Standard Delay Format (SDF) was designed to serve as a simple textual medium for exchanging the timing information and constraints between different EDA tools.

A timing calculator tool is responsible for SDF file generation. To do this, it examines the specific design for which it has been instructed to calculate timing data. In Fig.1, the arrow from the design description (netlist) illustrates this. The timing calculator must locate each region within the design, for which exists a timing model and calculate values for the parameters of that timing model. Strategies for computation vary from technology to technology. Knowledge of the timing models can be obtained by

accessing them directly (not shown) or can be built into the timing calculator and/or cell characterization data [3].
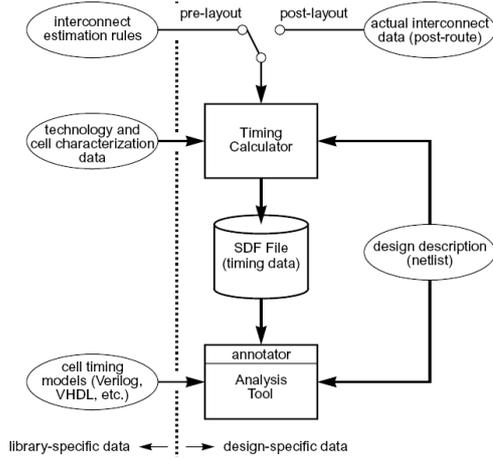
Fig.1. SDF files in timing back-annotation

The interconnect effects strongly influence the timing characteristics of ASIC's. Thus the timing calculator must use the estimation rules (pre-layout) or actual interconnect data (post-layout), Fig. 2. Therefore, the SDF is suitable for both pre-layout and post-layout applications [3].

```
(CELL
  (CELLTYPE "DFFSR")
  (INSTANCE a_reg_reg_1)
  (TIMINGCHECK
    (WIDTH (negedge S) (0.264082:0.264082:0.264082))
    (RECOVERY (posedge S) (posedge R) (0.0531:0.0531:0.0531))
    (RECOVERY (posedge S) (posedge CLK) (0:0:0))
    (REMOVAL (posedge S) (posedge CLK) (0.1719:0.1719:0.1719))
    (RECOVERY (posedge R) (posedge S) (0.0523:0.0523:0.0523))
    (WIDTH (negedge R) (0.22713:0.22713:0.22713))
    (RECOVERY (posedge R) (posedge CLK) (0:0:0))
    (REMOVAL (posedge R) (posedge CLK) (0.3601:0.3601:0.3601))
    (SETUP (posedge D) (posedge CLK) (0.1655:0.1655:0.1655))
    (SETUP (negedge D) (posedge CLK) (0.0935:0.0935:0.0935))
    (HOLD (posedge D) (posedge CLK) (0:0:0))
    (HOLD (negedge D) (posedge CLK) (0.0939:0.0939:0.0939))
    (WIDTH (posedge CLK) (0.488409:0.488409:0.488409))
    (WIDTH (negedge CLK) (0.332997:0.332997:0.332997))
  )
  (DELAY
    (ABSOLUTE
      (IOPATH (posedge CLK) Q (0.6766:0.6766:0.6766)
(0.7745:0.7745:0.7745))
      (IOPATH R Q (0.413:0.413:0.413) (0.601:0.601:0.601))
      (IOPATH S Q (0.5714:0.5714:0.5714) ()))
    `
```

a)

```
(DELAY
  (ABSOLUTE
    (INTERCONNECT clk clk__L1_I0/A (0.0014:0.0014:0.0014)
(0.0014:0.0014:0.0014))
    (INTERCONNECT rst i_145/A (0.0009:0.0009:0.0009)
(0.0009:0.0009:0.0009))
    (INTERCONNECT add i_33/A (0.0029:0.0029:0.0029)
(0.0029:0.0029:0.0029))
    (INTERCONNECT add i_2/A (0.0042:0.0042:0.0042)
(0.0042:0.0042:0.0042))
    (INTERCONNECT add i_41/A (0.0043:0.0043:0.0043)
(0.0043:0.0043:0.0043))
    (INTERCONNECT add i_39/A (0.004:0.004:0.004)
(0.004:0.004:0.004))
  .        .        .
```

b)

Fig.2. Descriptions of a) D flip-flop, and b) interconnects in SDF format

An example of the gate description for the SDF file is shown in figure 2a, while the interconnect SDF description is shown in figure 2b.

5. SCALING DOWN AND DELAYS

Timing performances of the sub-micron integrated circuit in the CMOS technology are known after the place&route design step. Here, loading delays become grater than the intrinsic gate delays. Wire RC delays become very important, as well as the good clock distribution when the frequencies are above 100MHz [14]. Figure 3 illustrates the dominant

delays for different CMOS technologies [15]. As we can see from the figure, scaling down decreases gate delays and increases wire delays in the overall circuit delay [15].
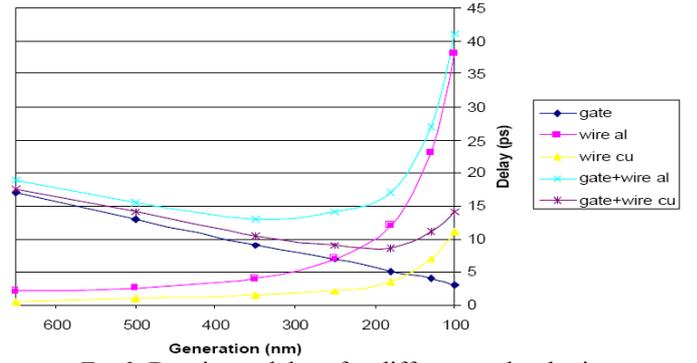
Fig.3. Dominant delays for different technologies

We can observe scaling down from two different points of view. First, we consider the influence that scaling down has to gate delays. Scaling the technology decreases the gate delays even with a lower VDD due to the smaller gate length and gate oxide thickness. Gate capacitance becomes smaller, while diffusion capacitance becomes relatively larger [15].

As the chip area increases (due to the increase of the functionality), the number of wires also increases, as well as the average wire length [15]. Since the interconnect delay is determined by the product of the line resistance and line capacitance (Fig.4), the following equations stand:
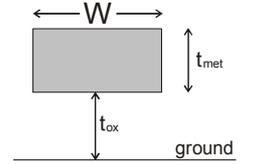
Fig.4. A wire

$$R = \rho_{met} \cdot L / W \cdot t_{met} , \quad \text{and} \quad C = \kappa_{ox} \cdot L \cdot W / t_{ox} . \quad (1)$$

$$RC = \rho_{met} \cdot \kappa_{ox} \cdot L^2 / t_{met} \cdot t_{ox} \quad (2)$$

Although these equations do not consider the capacitance and the resistance as the distributed parameters, the following technology improvement should be considered. In order to achieve lower interconnect delays, the metal resistance must be lower down. It means that aluminum lines should be replaced with the copper ones. Lower dielectric constant interlayer should be used, instead of standard SiO_2 dielectric, with appropriate polymers. This layer should also be made thicker, as well as the metal layer. In order to reduce the substrate losses, the high resistive substrate should be used [16]. Beside the technological issues, the impedance matching can decrease the delay, as well as the elimination of the cross-talk and unrelated circuit elements. Wires need to be pipelined (repeaters with states) to maintain synchronization eliminating delay variations [17].

6. COMPARISON

Table 2 shows that large error can happen in digital circuit simulation when fan-outs are not taken into account for the gate delay description. The estimated worst-case delays and the errors for not considering fan-out [5] for ISCAS benchmark circuit are presented.

From SDF file of one complex adder-subtracter circuit implemented in CMOS035 technology, we can evaluate the percentage of the delay introduced by the wires, comparing it to gates delay and the overall delay of the circuit. This is shown in Table 3. The file is extracted after place&route design phase.

The IC delays depend on circuit area. Figure 5 shows how the area (and though the delay) depends on the

complexity of the circuit. The abscissa in this figure denotes the size of the logic block expressed in the number of the equivalent NAND gates, while the ordinates gives the difference of the block size (expressed in percents) before and after routing lines. Given examples are implemented in AMI CMOS035 technology.

Table 2. Estimated worst-case delays when considering fan-out information and errors

Circuit name	D_{fmn} [ns]	ε_{fmx} %	D_{fmx} [ns]	ε_{fmn} %	D_{rmn} [ns]	ε_{rmx} %	D_{rmx} [ns]	ε_{rmn} %
c17	1.9	0	6	500	1.9	0	6	200
c432	4.6	142	53	214	2.8	47	53	212
c499	0.9	0	29.3	159	1	0	29.4	158
c880	1.8	0	50.1	109	2	0	50.3	108
c1355	2.8	0	53.9	120	2.9	0	54.1	126
c2670	0	0	80.5	149	0	0	80.6	149
c3540	1.8	0	93.8	97	1.9	0	94.8	99
c5315	1.8	100	89.9	82.3	1	0	88.2	81.5
c6288	5.5	511	262	261	6.8	580	262.7	112
c7552	0	0	101.1	136	0	0	102	137

Table 3. Physical of the adder-subtracter circuit in CMOS035 technology

	Rising edge delay		Falling edge delay	
delay type	[ns]	%	[ns]	%
wires	0.2114014	1.32	0.2114014	1.61
gates	15.830108	98.68	12.920864	98.39
average total delay	16.0415094 ns		13.1322654 ns	
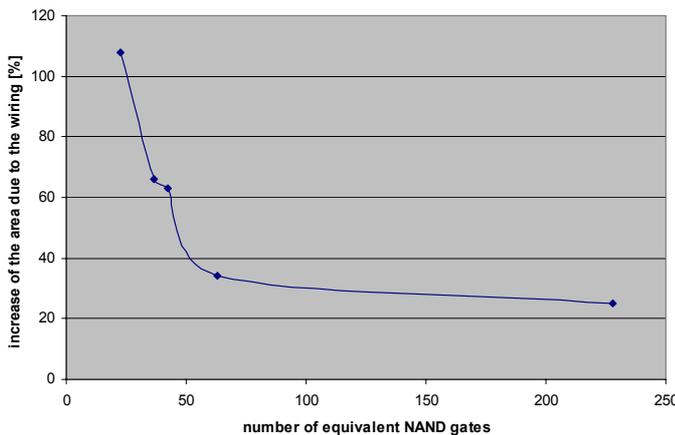
Fig.5. Area increase of different IC's after routing wires

7. CONCLUSION

Delays for the sub-micron, multi-gigahertz technologies are difficult to model accurately in the early design stages, since the wire length, wire capacitance, fan-out, hierarchy data, net type, circuit activity factors and many other hardly predictable influences define them. The improvement of the dynamic behavior can be achieved by improving the fabricating process, technology and design. Every wrong decision made at the beginning of the design process, increases its influence later in the design. The same is with gate delay analysis in the early design stages of sub-micron circuit, where major part of the total delay goes to the interconnections. Further research would be achieving the better accuracy of the present tools for delay calculation in the early design phases.

REFERENCES

[1] D. Sylvester, "Measurement Techniques and Interconnect Estimation", *Proceedings of the 2000 international workshop on System-level interconnect prediction* SLIP '00, April 2000.

[2] A. Narayan "Logic Level Delay Modeling", EE241 Interim Project Report.

[3] Standard for Standard Delay Format (SDF) for the Electronic Design Process, *IEEE Standards Board.*

[4] V. Litovski, "Electronic circuit design" (in Serbian), DGIP "Nova Jugoslavija" - Vranje, Niš, 2000.

[5] M. Sokolović, V. Litovski, M. Zwoliński, "Fan-out based delaya estimation in digital circuits", VI Symposium on industrial electronics, INDEL'06 November 2006, Banja Luka.

[6] L. Scheffer, L. Lavagno, G. Martin, *Electronic Design Automation for Integrated Circuits Handbook*, CRC Press/Taylor and Francis, March 2006.

[7] M. Sokolović, V. Litovski, "Using VHDL simulator to estimate logic path delays in combinational and embedded sequential circuit", *Proc. of the International Conference on Computer as a Tool, EUROCON* 2005, Belgrade, November 2005, pp. 1683-1686.

[8] M. Sokolović, V. Litovski, "Efficient computation of the statistical worst case delay in complex digital circuit", *Proc. of the XLX Conf. of ETRAN,* Belgrade, 2006, vol. 1. pp 23 – 26.

[9] IEEE standard including DCL

[10] I Sutherland, B Sproull, D Harris, *Logical effort: designing fast CMOS circuits*, San Francisco, CA, Morgan Kaufmann Publishers Inc. USA,1999.

[11] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *Journal of Applied Physics*, January 1948, Volume 19, Issue 1, pp. 55-63.

[12] L.T. Pillage, R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis", *IEEE Transactions on the Computer-Aided Design of Integrated Circuits and Systems*, Volume 9, Issue 4, April 1990, pp. 352 - 366.

[13] A. Odabasioglu, M. Celik, L.T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume 17, Issue 8, Aug. 1998, pp. 645 – 654.

[14] www.public.asu.edu/~ashriva6/teaching/LPCA/Presentations/p218-heo.ppt

[15] humanresources.web.cern.ch/Humanresources/external/training/tech/special/ELEC2002/ELEC-2002_25Apr02_2_PDF.pdf

[16] H. Ruecker, Summer School of Microelectronics, Frankfurt (Oder), June 2005.

[17] http://vlsicad.ucsd.edu/courses/ece260b-w07

Sadržaj – Brzina digitalnih kola predstavlja jedan od ograničavajućih faktora kod deep sub-mikronskih tehnologija za integrisana kola koja rade na frekvencijama od nekoliko gigaherca. Ona direktno zavisi od kašnjenja u kolu. Skaliranje tehnologije i povećanje radne frekvencije kola čini problem kašnjenja još važnijim. Ovaj rad predstavlja jedan pokušaj analize kašnjenja digitalnih kola od procesa projektovanja, kroz opis različitih modela kašnjenja kao i različitih metoda za izračunavanje kašnjenja. Ukazano je i na uticaj koji skaliranje ima na kašnjenja u kolu.

ANALIZA KAŠNJENJA U DIGITALNIM KOLIMA
Miljana Sokolović, Milunka Damnjanović