The Evolution Of Logic Synthesis

*Dots and Dashes, … Zeros and Ones*

Dr. Antun Domic
EVP & GM, Design Group
Synopsys, Inc.

Max Ernst, *Compendium of the History of the Universe*, 1953 (Detail); Galleria Nazionale d'Arte Moderna e Contemporanea, Rome, Italy
Charles R. Darwin

“The Great Tree of Life”
Smartphones & Logic Synthesis

Have More in Common than You May Think!

Smartphones

Logic Synthesis

Claude E. Shannon
“A Symbolic Analysis of Relay and Switching Circuits”

Logic Compiler, ca. 1986
Optimal Solutions, Inc. a.k.a. Synopsys, Inc.

Power Compiler, 1997
Convergence of Logic & (Dynamic) Power Synthesis

Physical Compiler, 1999
Convergence of Synthesis & Implementation

Design Compiler, 2001-2015
The Evolution of Synthesis!

<table>
<thead>
<tr>
<th>Year</th>
<th>Area</th>
<th>Timing</th>
<th>Power Dynamic</th>
<th>Power Static</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2005</td>
<td>78</td>
<td>91</td>
<td>82</td>
<td>78</td>
<td>20</td>
</tr>
<tr>
<td>2010</td>
<td>65</td>
<td>74</td>
<td>66</td>
<td>49</td>
<td>1.6(1)</td>
</tr>
<tr>
<td>2015</td>
<td>53</td>
<td>63</td>
<td>66</td>
<td>38</td>
<td>0.6(1)</td>
</tr>
</tbody>
</table>
Once Upon A Time…
Samuel F.B. Morse’s Telegraph and George Boole’s Algebra

Dots and Dashes…

…Zeros and Ones

“An Investigation of the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities”, 1854

\[ f(A,B,C,D) = \]
\[ = (\overline{ABC}D) + (\overline{ABCD}) + (\overline{ABCD}) + (\overline{A\overline{B}C\overline{D}}) + \]
\[ + (\overline{A\overline{B}C\overline{D}}) + (\overline{AB\overline{C}D}) + (\overline{ABC\overline{D}}) + (\overline{ABC\overline{D}}) \]
Once Upon A Time…
A. Graham Bell’s Telephone – the “Speaking Telegraph”
Once Upon A Time…

*Bell Labs Radio-Telephone – the “Mobile Phone”*

Source: Bell Labs, Radio-Telephone, 1924
Once Upon A Time...

Almon B. Strowger’s Rotary Dial Telephone, and…

Electromechanical Telephone Exchange!
The second problem is that of synthesis. Given certain characteristics, it is required to find a circuit incorporating these characteristics. The solution of this type of problem is not unique and it is therefore additionally desirable that the circuit requiring the least number of switch blades and relay...
The Sampling Theorem
a.k.a. Nyquist-Shannon-Kotelnikov Theorem
The Foundation of Digital Communications

If a function $x(t)$ contains no frequencies higher than $B$ Hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart.

$$f\left( \frac{n}{2W} \right) = \frac{1}{2\pi} \int_{-2\pi W}^{2\pi W} F(\omega) e^{i\omega \frac{n}{2W}} \, d\omega.$$
Maurice Karnaugh
“The Map Method for Synthesis of Combinational Logic Circuits”, Bell Labs, 1953
Edward J. McCluskey

“Algebraic Minimization and the Design of Two-Terminal Contact Networks”

Ph.D. Thesis, MIT, 1956

In addition it appears that the method to be presented here can be programmed on a digital computer without requiring excessively large storage capacity or computing time.
…Multi-Level Minimization,…

IBM, UCB, and University of Colorado at Boulder

Motorola DynaTAC


Source: Motorola DynaTAC
...And, Eventually : Logic Synthesis!

*General Electric, and AT&T*

**Nokia 100**


Kurt Keutzer, e.g. “DAGON: Technology Binding and Local Optimization by DAG Matching”, AT&T, 1988
Logic Compiler, ca. 1986
The Beginning of a New Era! Just the Beginning...
Logic Compiler, ca. 1986
Optimal Solutions, Inc. a.k.a. Synopsys, Inc.

What can LOGIC COMPILER do?

Convert equations into working circuits:

```
.contents pla
1=0 100
0=1  100

.contents function
.inputs 4
.outputs 3
.inputnames a b c d
.outputnames f0 f1 f2
f0 = b d' + a'c + a'b';
f1 = c + a'd + a'b ';
f2 = b c d';
```

LOGIC COMPILER

CONRAINTS
“MIS: A Multiple-Level Logic Optimization System”, 1987


Source: 1st A. Richard Newton Technical Impact Award in Electronic Design Automation, DAC 2009
...Binary Decision Diagrams (BDD)...

Randal E. Bryant ⇒ Olivier Coudert, and Jean-Christophe Madre

Logic Synthesis (and Formal Verification)

The computational cost of this new procedure is independent of the number of minterms of the function f and of its number of prime implicants, which allows us to treat functions for which these numbers are so large that it has never been possible to perform a 2-level minimization.
A Revolutionary... Evolution
Convergence!
"Our goal was to solve the problem of developing highly testable designs with minimum penalties in chip speed and size," said Aart de Geus, co-founder and senior vice-president of marketing at Synopsys.
Convergence Of Logic And Power Synthesis

Abstract
With the proliferation of portable devices and increasing levels of chip integration, reducing power consumption is becoming of paramount importance. We describe a technique to automatically synthesize gated clocks for finite-state machines (FSM) to reduce power in the final implementation. This technique extends the techniques of [10] to FSMs that either have a single transition network or have the functions described by the authors to provide gated clocks. The clock transition function is then used to determine information to schedule the clock in the FSM for additional power savings. We applied these techniques to standard ISCAS89 benchmarks and found an average reduction in power dissipation of 30% at the cost of a 9% increase in area.

1 Introduction
As portable devices proliferate and desire sizes continue to shrink, allowing more devices to fit on a chip, power consumption has taken an increased importance. Most recent work has focused on accurate estimation of power consumption and on its maximization reduction at all levels of abstraction. From high level synthesis down to physical layout [1, 3, 5].

Most power reduction techniques have emphasized reducing the level of activity in some portion of the circuit. We extend this research by concentrating on reducing the activity level of the clock by selectively stopping the clock. Because many sequential machines are implementations of reactive systems which wait for a certain event to occur before changing state, each power is wasted during this waiting period [9]. Let us see this clock circuit in comparison, reducing unnecessary power and waiting for change until the reactive event occurs. By resetting the clock during this period, we can realize substantial savings in many finite state machines (75%)
Convergence Of Logic And Power Synthesis

*Power Compiler, 1997*

Source: R. Zafalon, STMicroelectronics, SNUG1999
Convergence of Synthesis And Implementation

Physical Compiler, 1999

Physical Compiler
Evaluation Vehicle #1

- Evaluation Objective
  - Stress robustness, capacity and limits of the tool
- Design Characteristics
  - 250 nanometers
  - Transmission application
  - 220K placeable instances, ~800K gates, 18 RAMs
  - 70% utilization, irregular, non-rectilinear floorplan
  - 70+ clocks, fastest @ 65 MHz
  - Traditional flow converged after 4 weeks
- Evaluation Results
  - Compiled in 27 hours
  - Timing in one pass
  - Design routable, no congestion

Source: Motorola Timeport; M. Casale-Rossi, STMicroelectronics, 1999
“Some” Placement in Synthesis: Same Critical Paths, Correlation ±5%
Convergence Continues, 2007
“Some” Global Routing In Synthesis: Improving Planarity in a Graph
Convergence Continues, 2014
Congestion Analysis Pre-Synthesis Technology
Detects RTL Structures Causing Congestion down in the Implementation Flow

Large ROMs  Large Data Switches  Large MUXs  Large Selectors
Congestion Analysis Pre-Synthesis Technology

*E.g. Large MUX Structures*  
*Same-Inputs Sharing & Connectivity-Based Decomposition*

Before

After

Source: Synopsys Research, 2014
## Design Compiler, 2001-2015

*The Evolution of Synthesis!*

<table>
<thead>
<tr>
<th>Year</th>
<th>Area</th>
<th>Timing</th>
<th>Dynamic Power</th>
<th>Static Power</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2005</td>
<td>78</td>
<td>91</td>
<td>82</td>
<td>78</td>
<td>20</td>
</tr>
<tr>
<td>2010</td>
<td>65</td>
<td>74</td>
<td>66</td>
<td>49</td>
<td>1.6(^{(1)})</td>
</tr>
<tr>
<td>2015</td>
<td>52</td>
<td>63</td>
<td>66</td>
<td>38</td>
<td>0.6(^{(1)})</td>
</tr>
</tbody>
</table>

\(^{(1)}\) Design Compiler Multicore

Source: Synopsys Research 2015
Where Do We Stand?

Well, Telephones Have Made a Great Deal of Progress!
Smartphones

A Revolutionary… Evolution : Convergence !

50 Things We Won’t Do/Use Anymore (Mostly Because of/Thanks to Smartphones)

© 2015 Synopsys, Inc.
Where Do We Stand?

Even Synthesis Went Far Beyond Logic, Isn’t It?
Logic Synthesis
A Revolutionary… Evolution Too: Convergence!

Logic Compiler

What can LOGIC COMPILER do?

Convert equations into working circuits:

- contents.pla
  - C= 100
  - C= 100

- contents.function
  - inputs 4
  - outputs 1
  - equations: a, b, c, d
    - equation: F1, F2
    - F1 = b' * c' * d + a' * c
    - F2 = b * c' * d

Constraints

Design Compiler

Look-Ahead & Physical Guidance
1 Mx, 2 My, and 2 Mz (Almost Only P&G)
> 80% Routing Utilization, > 10% Smaller Die

DC-T + ICC not Routable
DC-G + ICC Routable

from Equations to Gates, to… “Placed” and Routable Gates
From Telephone To Logic Synthesis…

…And from Logic Synthesis to… Smartphones
The Evolution Of Logic Synthesis

Convergence!

2003
90 Nanometers
“Interoperability”

2005
65 Nanometers
“Correlation”

2007
45/40 Nanometers
“Look Ahead”

2009
32/28 Nanometers
“In-Design”

2011
22/20 Nanometers
“Convergence”
The Evolution Of Logic Synthesis
Convergence… But Not Done Yet : Still Differences!

\[ G_{ij} = \bigcup_{k=0}^{p_i-1} (r_i')c_j^k = 0 \]

\[ (c \# p)_j = \begin{cases} c_j \cap \overline{p}_j & \text{if } i = j \\ c_j & \text{if } i \neq j \end{cases} \]

\[ f(x, y) = \frac{k}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} D(x', y') \frac{\overline{r} - \overline{r'}}{\overline{r}^2 - \overline{r'}^2} dx'dy' \]
Looking Into The Next Decade

There Is a Great Deal Of New Technology Ahead!
Looking Into The Next Decade

1T Transistors per Die by the End of This Decade, “1” nm by the End of the Next
Infinitely Large, Infinitely Small, Infinitely Many

<table>
<thead>
<tr>
<th></th>
<th>2013</th>
<th>2015</th>
<th>2017</th>
<th>2019</th>
<th>2021</th>
<th>2023</th>
<th>2025</th>
<th>202x</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Technology Node” (nm)</td>
<td>“14”</td>
<td>“10”</td>
<td>“7”</td>
<td>“5”</td>
<td>“3.5”</td>
<td>“2.5”</td>
<td>“1.8”</td>
<td>“1.3”</td>
</tr>
<tr>
<td>DRAM ½ Pitch (nm)</td>
<td>28</td>
<td>24</td>
<td>20</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>10</td>
<td>7.7</td>
</tr>
<tr>
<td>MPU/ASIC ½ Pitch (nm)</td>
<td>40</td>
<td>32</td>
<td>25</td>
<td>20</td>
<td>16</td>
<td>13</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>FLASH ½ Pitch (nm)</td>
<td>18</td>
<td>15</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MPU Printed Gate Length (nm)</td>
<td>28</td>
<td>22</td>
<td>18</td>
<td>14</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>MPU Physical Gate Length (nm)</td>
<td>20</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Theoretical Integration Capacity (BT)</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
<td>4096</td>
<td>8192</td>
</tr>
</tbody>
</table>

(Assuming 450mm Wafers in Production in 2018, and EUV in Production after 10nm)

<table>
<thead>
<tr>
<th>Size of Internet (IP Addresses)</th>
<th>2013</th>
<th>2015</th>
<th>2017</th>
<th>2019</th>
<th>2021</th>
<th>2023</th>
<th>2025</th>
<th>202x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25B</td>
<td>~ 50B</td>
<td>100B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Looking Into The Next Decade
Making the Transition to High-Level Design… Again!

C / C++ / M / SystemVerilog
Formal Verification
High-Level Synthesis
RTL
Formal Verification
Synthesis
Gates
Looking Into The Next Decade

Evolving Towards Two, Closely Connected “Sub-Systems”
Explore & Analyze, then Implement
Looking Into The Next Decade

Evolving Towards Two, Closely Connected “Sub-Systems”
Explore & Analyze, then Implement

• Until now:
  – Mostly “preserve” the gates
  – “Change” placement, and/or routing to close timing, power, …
    – Very, very time consuming, and
    – Leads to an infinite number of iterations

• In the future:
  – Once the objective is “within reach” …
  – “Hold” placement and routing
  – Systematically “change” the gates
    – Same footprint, different timing, power, temperature inversion point, etc.
    – The richness of the library is fundamental
Looking Into The Next Decade

**RTL Exploration Is ~ 6X Faster Than Full Synthesis**

Slack Distribution Comparison, Correlation ±8%

**Floorplan Information**

**Slack Histogram**
Looking Into The Next Decade

Today Libraries Contain Thousand of Elements
Many Variants with the Same Footprint but Different “Performance”

\[ T = 125^\circ C \]

\[ T = \text{Temperature Inversion Point} \]
Looking Into The Next Decade
Extending the Use of Multi-Bit Structures Will Save Area and Power, and Will Alleviate Congestion, Simplifying Routing

Multi-bit Flip-Flop (MBFF)
Fewer pins for the router to connect

Non-MBFF vs. MBFF - Area Results
1.75% Area Savings

<table>
<thead>
<tr>
<th>Savings</th>
<th>DC-G</th>
<th>ICC (post-CTS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Area</td>
<td>- 7.3%</td>
<td>- 7.10%</td>
</tr>
<tr>
<td>Overall Std-cell Area</td>
<td>- 2.6%</td>
<td>- 1.75%</td>
</tr>
</tbody>
</table>

Total Area

Source: K. Umino, Hyon Han, Samsung, SNUG Austin 2014
Convergence Of Synthesis And Implementation

RC Variation Among the Many Metal Layers Makes Estimates Extremely Difficult Forcing to Bring Global Routing, and Probably Detailed, into the Synthesis Picture

Interconnects

Interconnect Stacks

- \( \Rightarrow M_z^2 \) RC Delay = 2ps/um
- \( \Rightarrow M_z^1 \) RC Delay = 5ps/um
- \( \Rightarrow My \) RC Delay = 7ps/um
- \( \Rightarrow Mx \) RC Delay = 8ps/um

Multiple interconnect stack offerings to optimize for cost, density, or performance
Looking Into The Next Decade

Today MPU Have 4-8 Cores, 16 Cores Are Just Around the Corner
Logic Synthesis Algorithms Must Be Suitable for Fine Parallelization
Looking Into The Next Decade
Buffers Insertion Optimization

A 2003 Estimate…

A 2010 Data Point…

@ 32 Nanometers Buffers ≈ 30% of Instances

Source: P. Saxena, Intel, IDPS 2003; Synopsys Research 2010
Looking Into The Next Decade

Temperature-Aware Synthesis

Source: P. Gelsinger, Intel, DAC 2004
Looking Into The Next Decade

Andreas Kuehlmann, Robert K. Brayton, Alan Mishchenko, Luca Amarù, Pierre-Emmanuel Gaillardon, Giovanni De Micheli
“It may be said that natural selection is daily and hourly scrutinizing .... every variation, even the slightest; rejecting that which is bad, preserving and adding up all that is good ... silently and insensibly working [...] at the improvement of each [...] . We see nothing of these slow changes in progress, until the hand of time has marked the long lapses of ages, and then [...] we only see that the forms of life are now different from what they formerly were.”
The Evolution Of Logic Synthesis

Dots and Dashes, … Zeroes and Ones

Workshop on Logic Synthesis and Verification
December 11th, 2015
Lausanne, Switzerland