

CS623: CAD FOR VLSI DESIGN

Lecture 30

Gain-Base Synthesis

- RTL to Silicon in two-to-three months
- Why is it difficult to take the functional description to physical layout, routing and implementation?

- Deep sub micron (DSM) forces physical design to move into the logical design space - Fix that and you fix this problem.
- Fixing in the logical design space becomes extremely difficult.
- Imparting knowledge of physical design to front-end users may reduce the number of iterations but may not eliminate it.
- every iteration is time consuming.

Requires a new technology for design but that is a challenge.

Old tools are more than 15 years old and can scale upto limited degree

For processes less than 180nm these tools are not scalable

Different steps

- An integrated system that takes RTL to GDSII
- A unified data model that presents a complete picture of the chip to all technologies in the design flow.
- Technologies with enough capacity to support the implementation of multi-million gate chips without forcing artificial partitioning.
- Technologies designed from the ground up to optimize for interconnect and routing resources.

Gain based Synthesis addresses all the four issues

What are the difficulties with existing tools.

- Inaccurate wire-load models to estimate interconnect parasitics. Only cell size is fixed before physical design - Computation intensive and bloated chips due to over-sized logic gates - meaningless if timing closure is not achieved.
- The work in size/delay plane while place and route tools work in load/delay plane - lot of time to find out that a device cannot be implemented at all.
- Lack of capacity to process chips more than a few hundred thousand gates - Artificial partitioning forced - Non optimized clock and power distribution and skew - Long time to partition.

- Operate on databases different than what is used by place and route tools Some tools have common database, using sequential files. This means one tool has to finish before some other tool can operate on it.

Problems with Traditional Synthesis

- Synthesis tools work with a cell library specific to a technology.
- the library can have different variations of the cells - say four versions of (AND, OR, NAND and NOR) and 8 versions of NOT.
- Increasing transistor size results in faster gates with faster signal slews and occupy more silicon real estate and consume more power.

Now to implement a function $w = !(x + y)z$ one may use three to four gates. There may be several choice for these, that is combinatorially large.

Large gates results in increased input capacitance, slowing down the previous stage.

Smallest number of gates need not imply the fastest path.

Synthesis users over-constrain to get timing closure after layout. Results in high power gates with huge drive strengths occupying large area, resulting in routing congestion - "ripples" back to reduce the size.

Placement-savvy synthesis tools use wireload models that are design dependent - but this leaded to minor improvement. These models are derived from initial placement and global routing of the design.

Synthesis and Placement tools should work on (size/load/delay) three-dimensional model - else problems will be recurring

The Approach - Logical Effort

Normalize everything to a base function - which is an inverter driving exactly similar another inverter as a load with no parasitics.

The delay of this inverter is $d * T$, where T is a process-specific constant. $d = (LE \times G) + p$, where LE is called the Logical Effort, G (Electrical effort) is the gain and p is the intrinsic parasitic delay.

LE of a gate describes how much worse the gate is at producing output current when compared to an inverter.

p is a constant for gates and independent of size of the gates - predominantly internal capacitance associated with its transistors. Increased size gives the ability to overcome this increased internal capacitance.

G is the gain - C_{out}/C_{in} . C_{out} is the load - more load decreases performance, more size increases performance (drive larger loads) but increases C_{in} value.

Gain Based Synthesis

The delay depends on Gain of the gate and not on its exact parasitics. Maintain ratio of capacitances constant by choosing gates of appropriate sizes even at the design implementation stage.

How to fine tune synthesis tools towards this end.