



Evento	XXI FEIRA DE INICIAÇÃO À INOVAÇÃO E AO DESENVOLVIMENTO TECNOLÓGICO – FINOVA/2012
Ano	2012
Local	Porto Alegre - RS
Título	Sínteses lógica e física para comparação de desempenho entre standard cells de 350nm comerciais e da ferramenta Astran (UFRGS)
Autores	JULIA CASARIN PUGET Daniel Silva Guimarães Júnior
Orientador	RICARDO AUGUSTO DA LUZ REIS

Sínteses lógica e física para comparação de desempenho entre *standard cells* de 350nm comerciais e da ferramenta Astran (UFRGS)

Julia Casarin Puget, Daniel Guimarães Jr., Ricardo Augusto da Luz Reis

Este trabalho consiste em realizar o fluxo de concepção de um circuito integrado a fim de comparar o desempenho em consumo de potência, área utilizada e efeitos de *timing* para duas bibliotecas de portas lógicas, uma disponibilizada comercialmente, e outra produzida a partir de portas lógicas geradas automaticamente pela ferramenta Astran [1]. A síntese lógica e a síntese física são processos que fazem parte do fluxo de concepção de um circuito integrado em microeletrônica, iniciado a partir da sua descrição em linguagem de *hardware*. No caso deste trabalho, a linguagem de *hardware* utilizada foi Verilog.

No processo de síntese lógica, uma representação em Verilog, com suas conexões (*wires*) e variáveis, é mapeada para uma representação de mesma natureza, mas constituída por portas lógicas, seguindo as ligações descritas no circuito original. A saída deste processo é chamada de Verilog estrutural, ou *netlist*. A ferramenta utilizada para realizar este processo é o RTL Compiler© (Cadence). As portas lógicas que serão utilizadas para mapear o circuito são selecionadas pela ferramenta, respeitando restrições determinadas pelo usuário, e a partir de uma biblioteca de portas lógicas em formato *liberty*, que contém informações de potência, temporização e área de cada porta lógica. A síntese lógica tem, como entradas, a biblioteca em formato *liberty*, um documento de restrições determinadas pelo usuário da ferramenta (formato SDC – *Synopsis Design Constraints*), e o circuito em questão, descrito em uma linguagem de *hardware*. Esta síntese permite uma análise da viabilidade do projeto para as restrições antes que seja feita a sua síntese física.

No processo de síntese física, é feito o leiaute do circuito. Uma das entradas para este processo é a *netlist* mencionada anteriormente. Aqui, o *design* é desenvolvido em várias etapas, desde a sua planta baixa (*floorplanning*), planejamento da rede de alimentação (*powerplanning*), posicionamento das instâncias de portas lógicas presentes na descrição ao longo da área do chip (*placement*), inserção de relógio (CTS – *ClockTreeSynthesis*) e roteamento (*routing*). A ferramenta utilizada para construir a estrutura física do circuito foi SoC Encounter (Cadence)©, versão 8.1. Neste processo, é possível modificar as etapas anteriores, para um melhor desempenho, otimizando o circuito para temporização, consumo de potência e ocupação de espaço físico.

Este trabalho faz parte de uma dissertação de mestrado em curso do mestrando Daniel Guimarães Jr. Referente às portas lógicas (também chamadas células, por serem a constituição básica do circuito), foram utilizadas duas bibliotecas diferentes para elaborar os experimentos. Uma biblioteca é comercial para tecnologia de comprimento de canal 350nm, disponibilizada pela empresa AMS (*Austria Microsystems*). A outra biblioteca, também de tecnologia 350nm, foi caracterizada a partir de células geradas automaticamente por uma ferramenta desenvolvida na Universidade Federal do Rio Grande do Sul, chamada Astran. Todas as células são denominadas *standard cell*, logo, tendo todas a mesma altura, para facilitar seu posicionamento e alimentação.

Foram selecionados circuitos de *benchmark* em Verilog para serem sintetizados, passando pelos dois processos aludidos. Os experimentos consistem em sintetizar cada *benchmark*, passando pelos processos aludidos, mas efetuando a síntese física duas vezes. A síntese lógica é feita a partir da biblioteca comercial. Daí, segue-se a primeira síntese física da sequência, com a *netlist* mapeada apenas para células comerciais. Na segunda vez, o processo de síntese física é repetido de forma idêntica, inclusive preservando-se as mesmas medidas escolhidas para largura de anéis de potência e de porcentagem de ocupação de área por posicionamento de células que na primeira síntese, seguindo-se exatamente os mesmos passos. A diferença é que, neste caso, a *netlist* adquirida a partir da primeira síntese lógica é modificada por uma ferramenta que se baseia no uso de redes neurais artificiais. Estas redes absorvem a *netlist*, e, a partir do seu aprendizado, modificam-na, substituindo as células comerciais presentes na descrição por células geradas automaticamente pela ferramenta Astran, mas apenas nos casos em que é estimado pelas redes neurais que as células Astran teriam um rendimento melhor em relação a *timing*, potência ou área. Nos casos em que isso não aconteceria, a célula original seria mantida. Assim, apenas poucas células da *netlist* são substituídas, mas as análises de *timing*,

potência e área obtidas a partir de dez *benchmarks* sintetizados evidenciam, em algumas amostras, uma maior economia de potência ou redução de área da concepção do circuito desde uma *netlist* mista em relação a uma *netlist* puramente comercial. Isso mostra que é possível utilizar células geradas automaticamente por uma ferramenta como o Astran para concepção de circuitos, mas ainda é necessário aprimorar a ferramenta e realizar mais experimentos. O uso de uma ferramenta de geração automática como esta representaria uma redução considerável de tempo dispendido na etapa de leiaute de células, que não são o produto final do circuito integrado, e sim, sua constituição básica. Com automatização, não é necessário realizar o *design* de cada porta lógica à mão, um processo dispendioso e sujeito a erros humanos, e também reduz a dependência de uma biblioteca disponibilizada, quando fosse demandada uma porta lógica que não constasse na biblioteca, seria possível utilizar a ferramenta automática para gerá-la.

Referências bibliográficas:

[1] ZIESEMER, Adriel Mota. **Geração Automática de Partes Operativas de Circuitos VLSI**, dissertação de mestrado, Universidade Federal do Rio Grande do Sul, 2007.