LEITURA E TRATAMENTO EFICIENTE DE AIGS COM MAIS DE UM MILHÃO DE NODOS

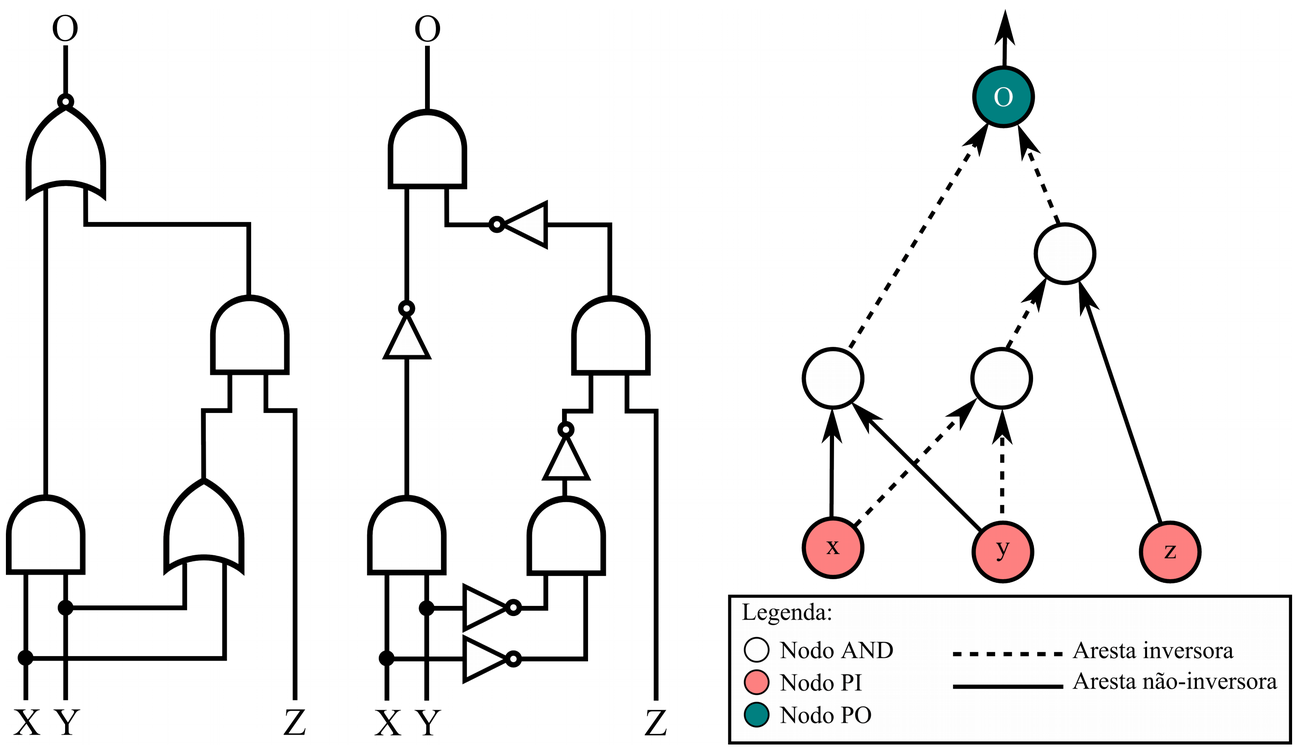
Marcos Henrique Backes, André Inácio Reis
Instituto de Informática – UFRGS, Porto Alegre
{mhbackes, andreis}@inf.ufrgs.br

MOTIVAÇÃO

- ✓ Circuitos digitais estão presentes em diversos dispositivos utilizados no dia a dia: telefones celulares, computadores, carros, eletrodomésticos.
- ✓ Como o projeto desses circuitos trata-se de uma tarefa complexa, os projetistas de circuitos integrados desenvolvem programas de computador que os auxiliam na realização dessa tarefa.
- ✓ Devido ao aumento da complexidade dos circuitos nos últimos anos, no intuito de torná-los mais rápidos, o número de componentes necessários para sintetizá-los também aumentou.
- ✓ Atualmente, existem projetos de circuitos com milhões e até bilhões de transistores. Dessa forma, para que a concepção de circuitos integrados seja feita de maneira rápida, é necessário que o circuito seja representado de forma eficiente no computador, através de uma estrutura de dados.

GRAFO DE ANDs E INVERSORES

- ✓ A estrutura de dados mais popular para a representação de um circuito, na etapa de síntese lógica, é o Grafo de ANDs e Inversores (do inglês And Inverter Graph – AIG), que é um grafo dirigido e acíclico composto e nodos AND de duas entradas (AND2), nodos de entrada (PI) e de saída (PO), conectados por arestas dirigidas e negadas. Um exemplo de AIG é mostrado na Figura 1.



(a) (b) (c)

Figura 1: Um circuito combinacional (a), sua representação com portas AND2 e inversores (b), e uma possível representação de AIG (c).

ESTRUTURA DE DADOS

- ✓ A estrutura de dados criada para representar o AIG tem como principal objetivo prover uma interface simples para o programador e, ao mesmo tempo, utilizar o mínimo de memória, uma vez que a quantidade de memória utilizada é proporcional ao tempo de processamento.
- ✓ Para isso, como os AIGs utilizados apresentam um número muito grande de nodos e arestas, tentou-se representá-los com o mínimo de memória necessário.

RESULTADOS EXPERIMENTAIS

- ✓ Foi realizada uma comparação entre a ferramenta ABC [1] e a estrutura de dados criada, medindo o tempo necessário para carregar a estrutura do AIG a partir do arquivo que o descreve. Foram utilizados AIGs pertencentes ao conjunto de circuitos *benchmarks* da EPFL [2].
- ✓ Dentre os resultados obtidos, percebe-se que, em geral, o tempo de execução do nosso algoritmo foi menor ou igual ao do ABC. A memória utilizada também foi menor, exceto para os circuitos maiores: *sixteen, twenty e twentythree*, que possuem entre 16 e 23 milhões de portas AND2.

Tempo de Execução		
AIG	ABC	NOSSO
adder	0.00 s	0.00 s
arbiter	0.00 s	0.00 s
bar	0.00 s	0.00 s
cavlc	0.00 s	0.00 s
ctrl	0.00 s	0.00 s
dec	0.00 s	0.00 s
div	0.02 s	0.02 s
hyp	0.15 s	0.08 s
i2c	0.00 s	0.00 s
int2float	0.00 s	0.00 s
log2	0.01 s	0.01 s
max	0.00 s	0.00 s
mem_ctrl	0.02 s	0.03 s
multiplier	0.01 s	0.01 s
priority	0.00 s	0.00 s
router	0.00 s	0.00 s
sin	0.00 s	0.00 s
sixteen	20.99 s	8.29 s
sqrt	0.01 s	0.01 s
square	0.01 s	0.01 s
twenty	25.00 s	10.89 s
twentythree	27.63 s	12.58 s
voter	0.00 s	0.00 s

Memória		
AIG	ABC	NOSSO
adder	39.2 MB	13.5 MB
arbiter	40.5 MB	17.0 MB
bar	39.8 MB	14.2 MB
cavlc	39.4 MB	13.2 MB
ctrl	39.3 MB	13.1 MB
dec	39.2 MB	13.2 MB
div	45.7 MB	30.7 MB
hyp	63.8 MB	78.8 MB
i2c	39.5 MB	13.6 MB
int2float	39.3 MB	13.1 MB
log2	43.0 MB	23.0 MB
max	39.8 MB	14.2 MB
mem_ctrl	44.8 MB	28.7 MB
multiplier	42.6 MB	21.5 MB
priority	39.2 MB	13.3 MB
router	39.3 MB	13.2 MB
sin	39.9 MB	14.8 MB
sixteen	1.9 GB	4.9 GB
sqrt	42.1 MB	20.8 MB
square	41.2 MB	18.8 MB
twenty	2.3 GB	6.3 GB
twentythree	2.6 GB	7.1 GB
voter	40.8 MB	17.7 MB

Referências

[1] Berkeley Logic Synthesis and Verification Group, Berkeley, Calif. ABC: A System for Sequential Synthesis and Verification. http://www.eecs.berkeley.edu/~alanmi/abc/.

[2] The EPFL Combinational Benchmark Suite, “Multi-output PLA benchmarks”. http://lsi.epfl.ch/benchmarks .