

**Introdução:**

A evolução da microeletrônica depende da diminuição dos transistores e melhorias nas ferramentas de CAD. A estrutura de dados adotada por cada CAD, tem impacto direto no resultado final do chip (como área, potência e atraso)

Existem diversas estruturas de dados bem conhecidas pela indústria microeletrônica e em pesquisas acadêmicas como:

- Soma de Produtos(SOP)
- And-inverter Graph (AIG)
- Boolean Network

**Definição de AIG e MIG:**

AIG - Fig1(a):

- É um grafo homogêneo
- Cada nodo representa uma função lógica AND de 2 entradas
- Arestas representam interconexões que podem ser diretas ou negadas

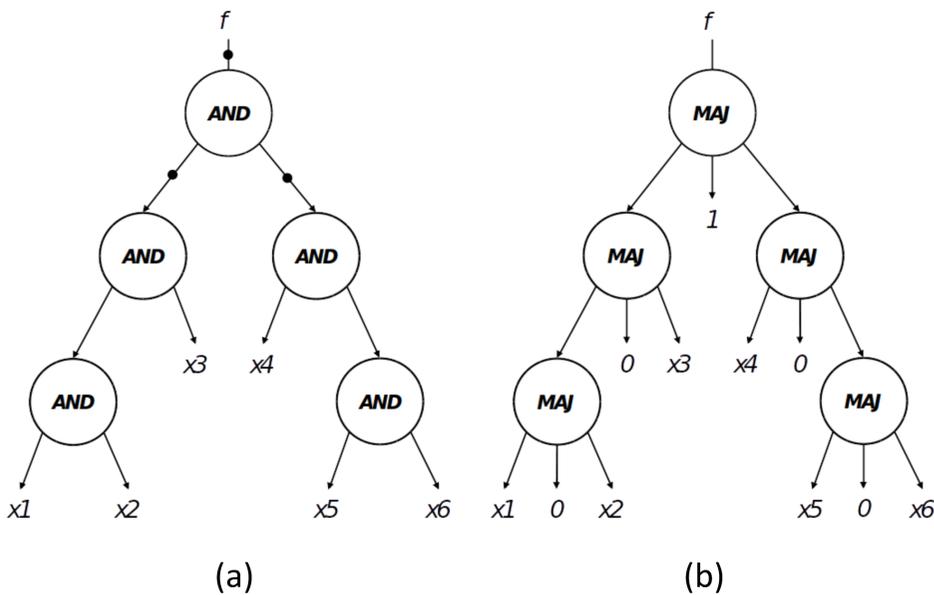


Fig. 1. In (a) AIG and in (b) MIG representation of the function  $f = (x1 \cdot x2 \cdot x3) + (x4 \cdot x5 \cdot x6)$ .

MIG – Fig1(b)

- Estrutura igual a do AIG, entretanto cada nodo representa uma função do lógica MAJ de 3 entradas

No artigo [1], Amarú propôs essa estrutura MIG junto com um conjuntos de operações Booleanas, alcançando avanços em relação a pontos críticos como área, potencia e tamanho.

O primeiro passo para explorar as vantagens do MIG é realizar uma conversão a partir de um AIG.

Existe uma maneira trivial de fazer essa conversão:

$$AND(x,y) = MAJ(x,y,0)$$

$$OR(x,y) = MAJ(x,y,1)$$

Assim temos uma conversão 1-1 para essa nova estrutura, que terá o mesmo numero de nodos da AIG.

**Método proposto:**

Contudo é possível conseguir uma redução dos nodos se explorarmos as 3 entradas, e encontrar majoritárias dentro do AIG e transformá-los em só uma MAJ.

Para encontrar as majoritárias, o método realiza uma busca em profundidade identificando dois padrões de Majoritária: ISOP(1) e a fatoração da ISOP (2):

$$MAJ(x,y,z) = !((!(x \cdot y) \cdot !(x \cdot z)) \cdot !(y \cdot z)) \quad (1)$$

$$MAJ(x,y,z) = !(x \cdot !(y \cdot !z)) \cdot !(y \cdot z) \quad (2)$$

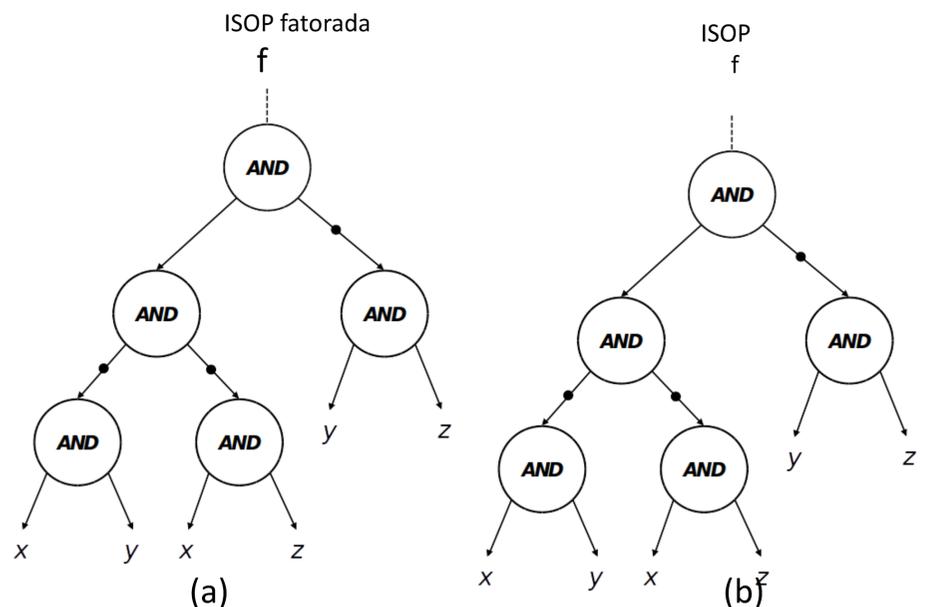


Fig. 2. Duas representações de padroes para MAJ de 3-entradas: em (a) AIG obtido a partir da Eq. (1) e (b) AIG obtido a partir da Eq. (2)

**Resultado:**

Os resultados obtidos estão apresentados na Tabela 1. Foi obtida uma redução em relação a conversão 1 para 1. Espera-se que a redução no numero de nodos do MIG inicial gere uma otimização nos resultados gerados pela ferramenta MIGhty proposta no trabalho de Amarú[1]

Tabela 1. Com alguns resultados da ferramenta proposta sobre circuitos de benchmark em [2]

Benchmark	AIG	MAJ	MAJ <sub>fac</sub>	MAJ <sub>ISOP</sub>	MAJ <sub>cov</sub>	MIG	% Node Reduction
vMIG.des_perf	77454	2565	0	2565	12825	67194	13.25
vMIG.diffeq1	22857	1300	17	1283	6483	17657	22.75
vMIG.div16	9486	1278	0	1278	6390	4374	53.89
vMIG.pci_spoci_ctrl	1660	182	0	182	910	932	43.86
vMIG.sqrt32	4616	615	0	615	3075	2156	53.29
Average	16852.39	859.52	7.71	851.81	4289.87	13414.32	25.77
Stand. Dev.	18907.41	740.07	35.70	728.42	3688.10	16405.05	11.06

Referências:

- [1] L. Amarú, P. E. Gaillardon and G. De Micheli. "Majority-Inverter Graph: A New Paradigm for Logic Optimization," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 5, pp. 806-819, May 2016.

[2] <http://lsi.epfl.ch/MI>

Agradecimentos:

Pesquisa parcialmente financiada por CNPq, FAPERGS