

Circuitos Programáveis

PROM / PAL / PLA

Circuitos Programáveis

PLDs (Programmable Logic Devices) são circuitos integrados cujas conexões internas podem ser programadas pelo usuário de modo a realizar diferentes projetos de circuitos lógicos.

Vantagens

- Facilidade para o desenvolvimento de protótipos
- Simulação do projeto intrínseca (no próprio hardware)
- Baixo risco financeiro de desenvolvimento

Tipos de de Programmable Logic Devices (PLDs)

Diferem...

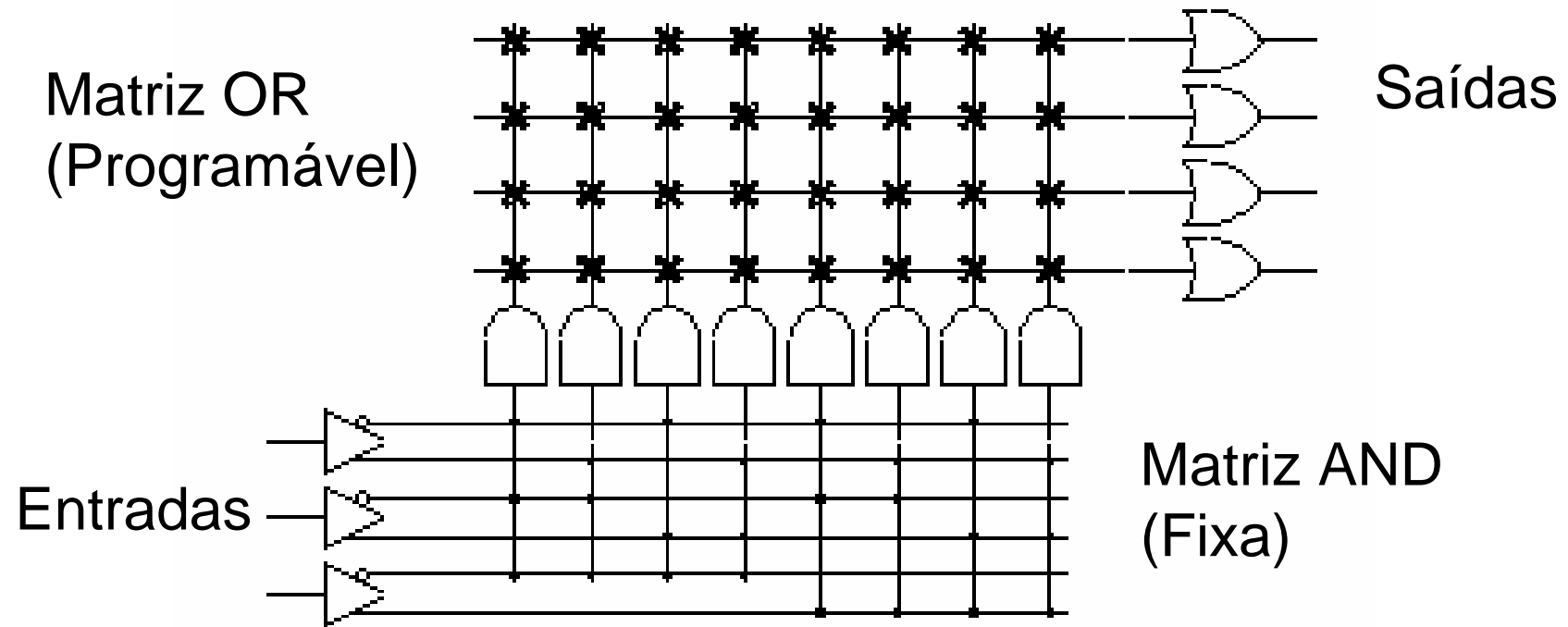
- Na organização dos arrays AND e OR, memória ou blocos lógicos.
- Na programabilidade dos arrays (colocação dos fusíveis ou transistores)
- Programáveis pelo fabricante ou pelo usuário

PROM

Programmable Read Only Memory

- O primeiro chip programável pelo usuário.
 - linhas de endereço servem como inputs
 - linhas de dados como output
- PROM pode implementar qualquer circuito combinacional
- Geralmente utilizadas quando as equações precisam um grande número de termos produtos.
 - Não volátil
- Ex: Microprogramação

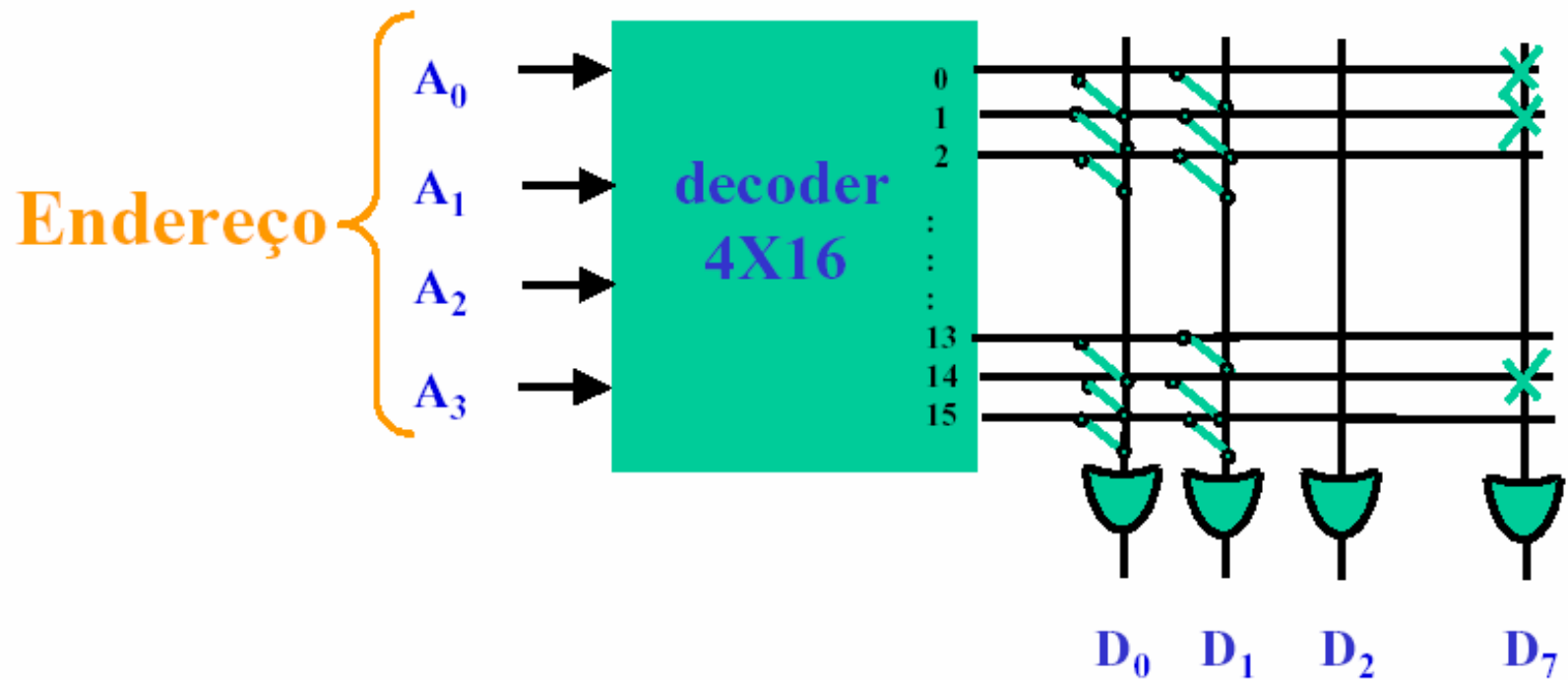
Estrutura de uma PROM



Memória PROM usada como PLD

PROM - Programmable Read-Only-Memory

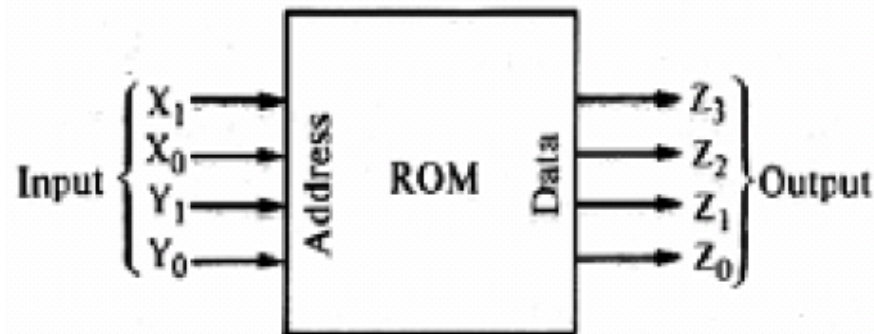
Representação simbólica



- **Notar que se trata de uma memória, onde cada célula tem originalmente todos os bits iguais a 1.**
- **Programação corresponde à queima dos “fusíveis”**
- **Encarando a PROM como um PLD**
 - **Bits de endereço A_i → variáveis de entrada da função**
 - **Decodificador: efetua a decodificação de todos os mintermos**
cada palavra corresponde então a um mintermo
 - **Bits de dados de saída → saída de múltiplas funções**
 - **Por exemplo: $F = \sum m(0,1,14) = m_0 + m_1 + m_{14}$**
Implementada por D_7 na figura
Precisa memória com 16 mintermos → 4 variáveis de entrada
4 bits de endereço
- **Comparando com a forma geral de um PLD**
 - **PROM - decodifica todos os mintermos**
 - **ARRAY AND é fixo (decodificador) - saída dele são mintermos**
 - **ARRAY OR é programável - soma dos mintermos**
 - **Número de funções = número de bits das palavras**
 - **Número de variáveis de entrada = número de bits de endereço**

Exemplo de PROM implementando circuito

- Geralmente são consideradas dispositivos de memória onde cada endereço aplicado na entrada retorna uma palavra programada no dispositivo.



Input				Outputs			
X_1	X_0	Y_1	Y_0	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

PLA e PAL- Como funciona?

Estrutura básica:

- um arranjo de portas **AND** e um arranjo de portas **OR**, cujas entradas podem ser interconectadas para produzir funções nas saídas das variáveis de entrada.
- Conexões programáveis.

Introdução

Especificação de uma função como SDP (Soma de Produtos)

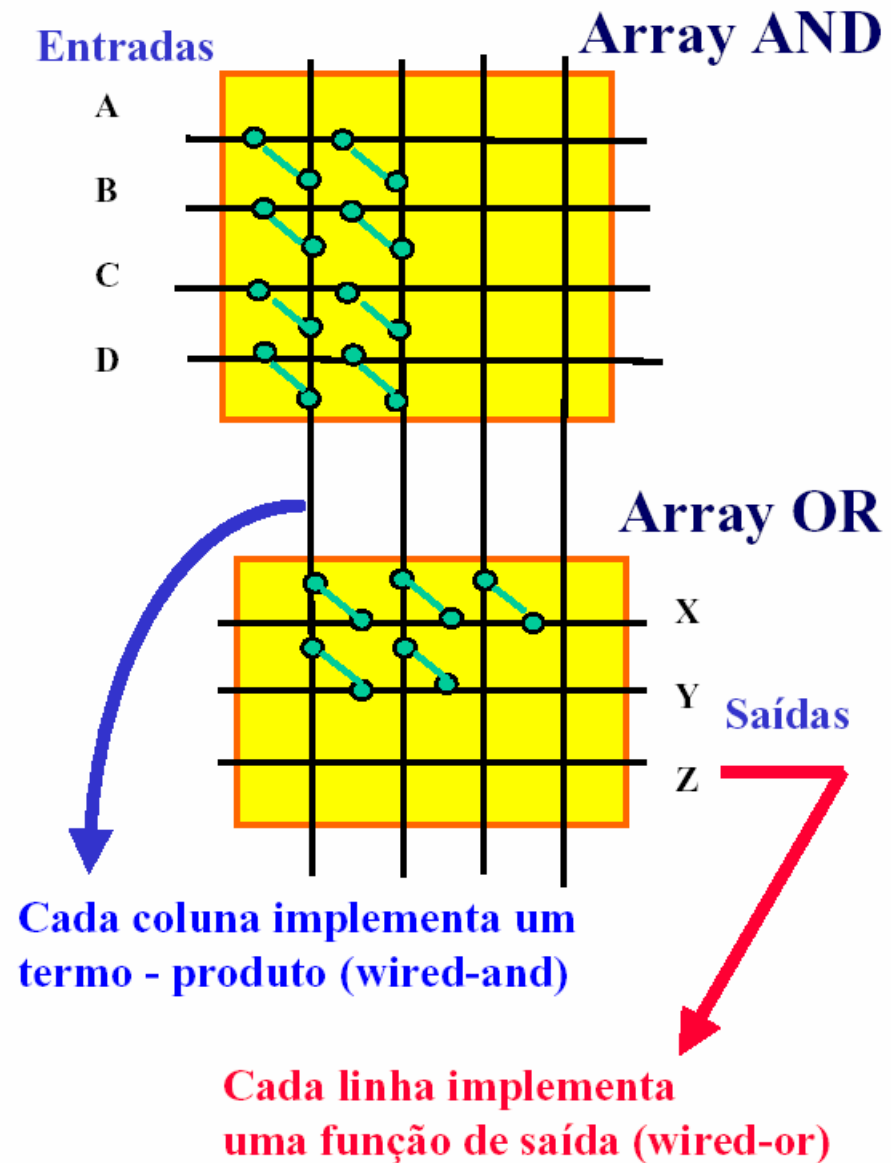


Implementação trivial por um circuito AND-OR

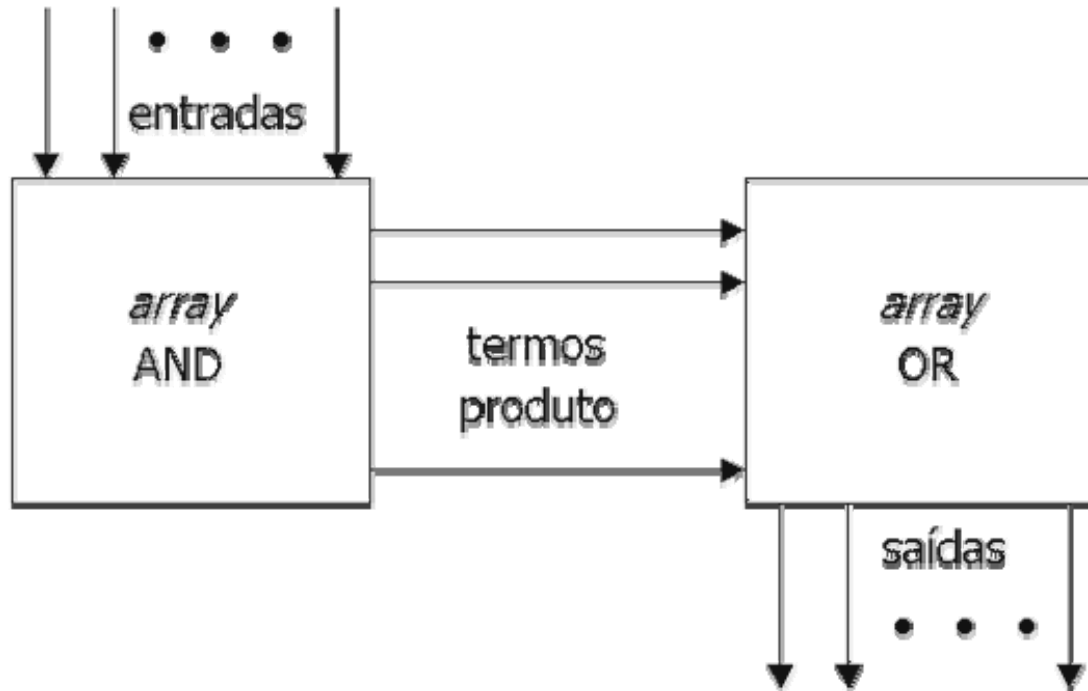
PLD's



- dispositivos pré-difundidos
- arrays AND-OR
- “fusíveis” inicialmente intactos
(fusíveis são na verdade implementados por transistores/diodos)
- programação corresponde à queima dos “fusíveis”



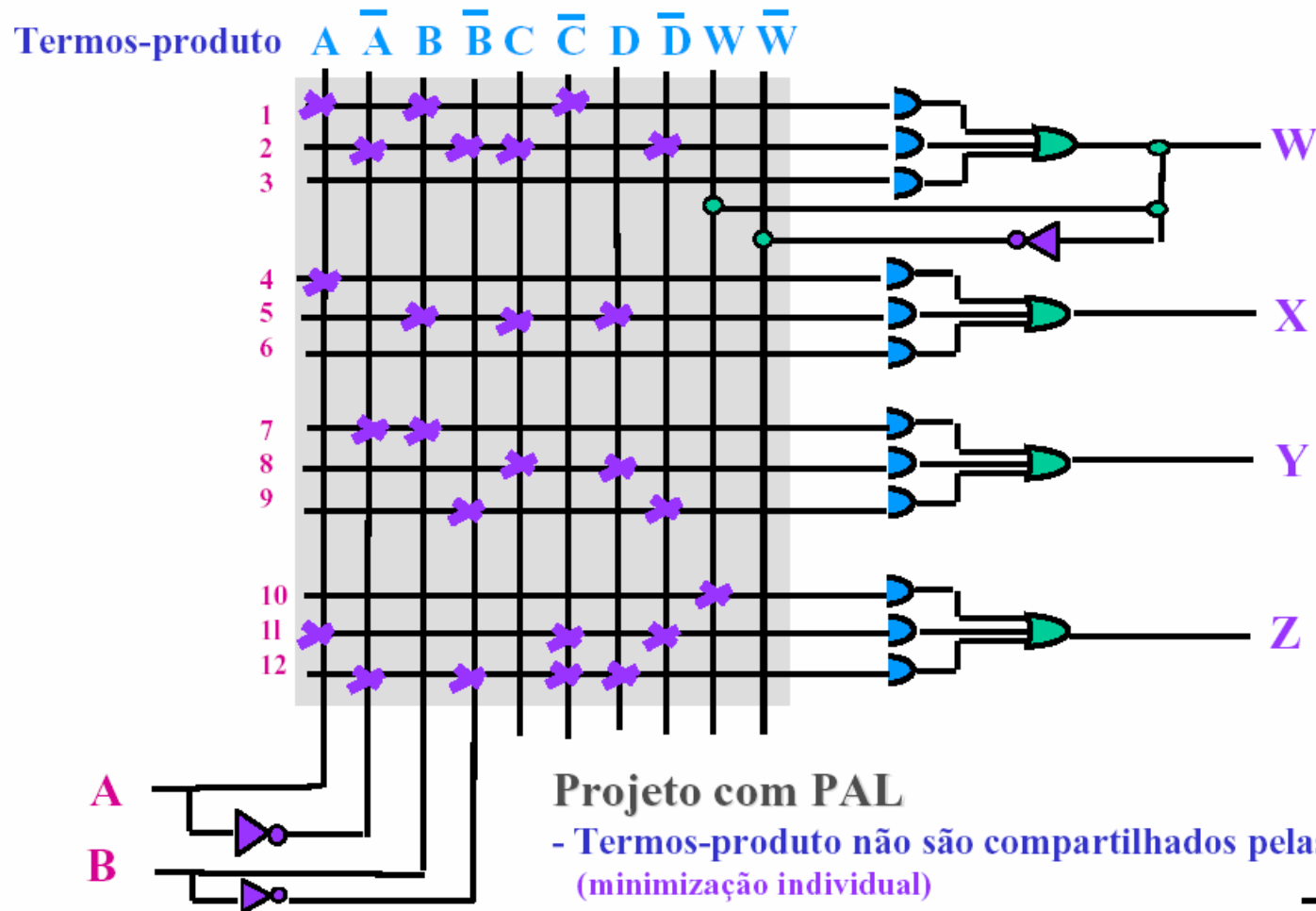
PLA x PAL



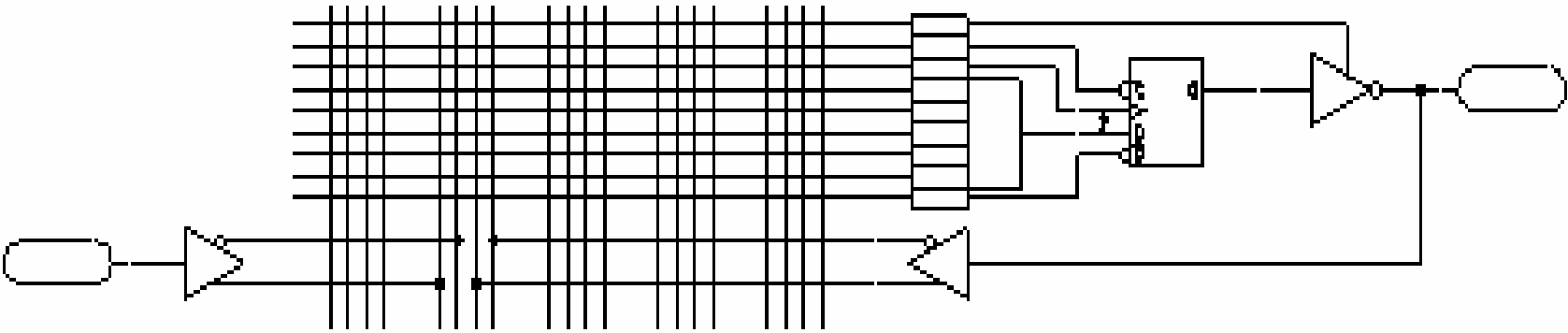
AND PROGRAMAVEL	OR FIXO	-----	PAL
AND PROGRAMAVEL	OR PROGRAMAVEL	-----	PLA
AND FIXO (MEM)	OR PROGRAMAVEL	-----	PROM

PAL - Programmable Array Logic

- Características:
- Array AND é programável
 - Array OR é fixo (cada OR é limitado a n termos-produto)
 - Feedbacks das saídas para o array and permitem funções com mais de n termos



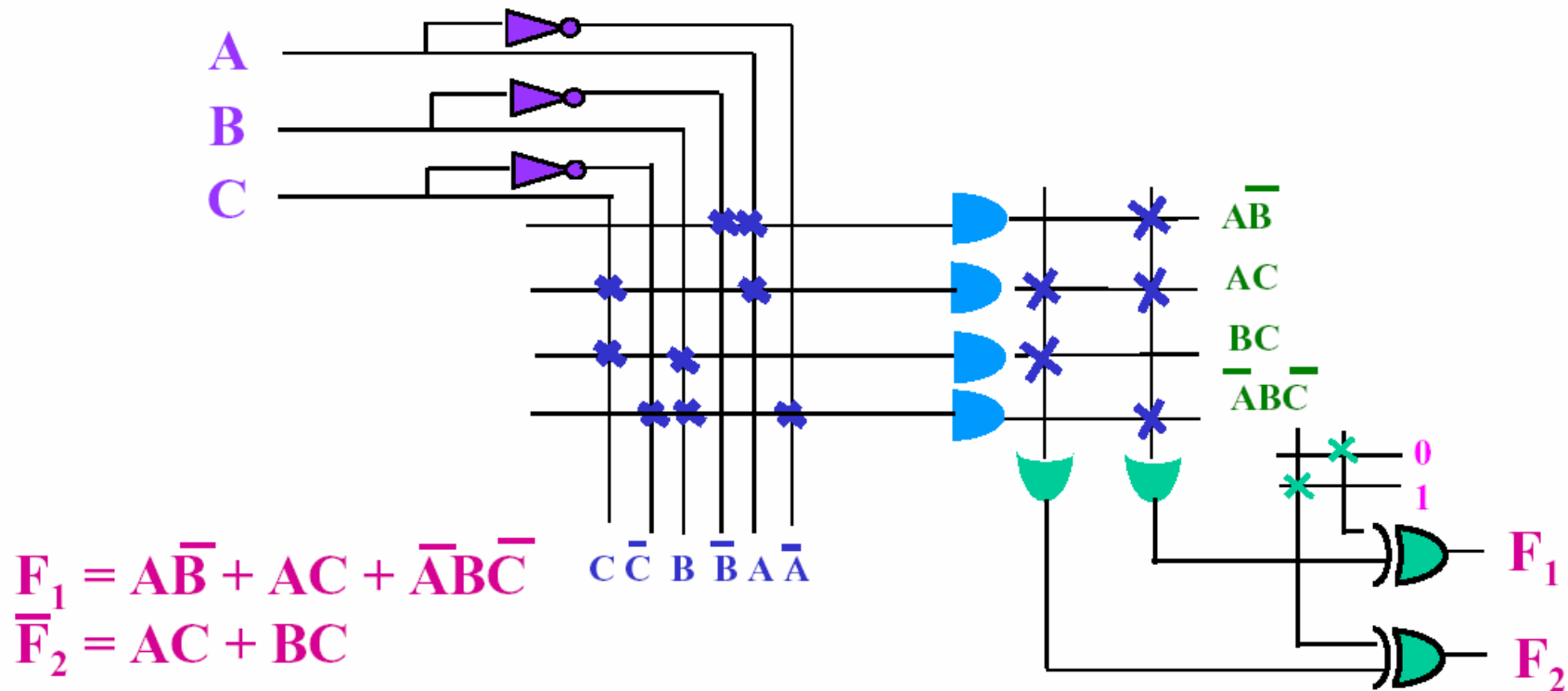
PAL Sequential (c/ Flip-Flop)



PLA - Programmable Logic Array

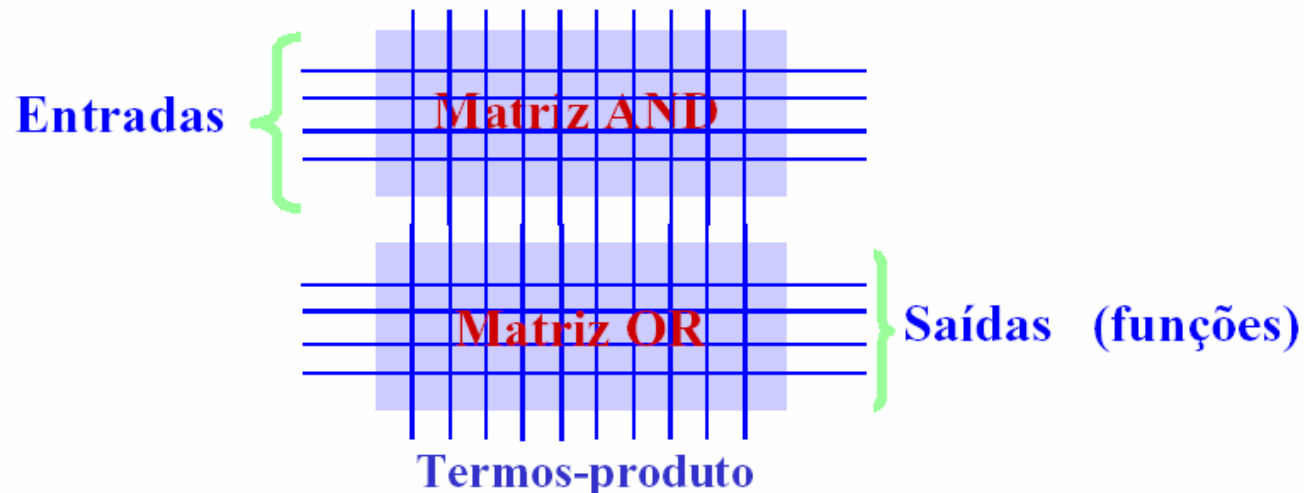
Características

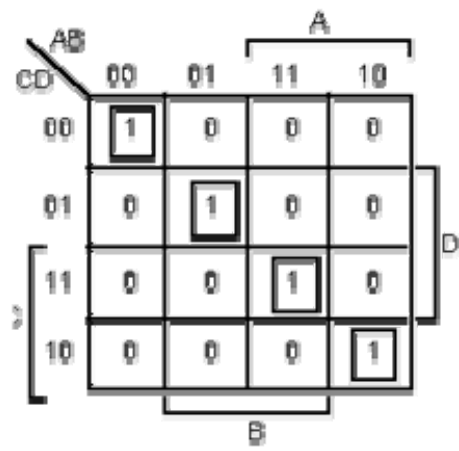
- Arrays AND e OR são programáveis
- Para n variáveis de entrada, o array AND tem $2n$ linhas (variável e variável complementada)
- São decodificados apenas os termos-produto necessários



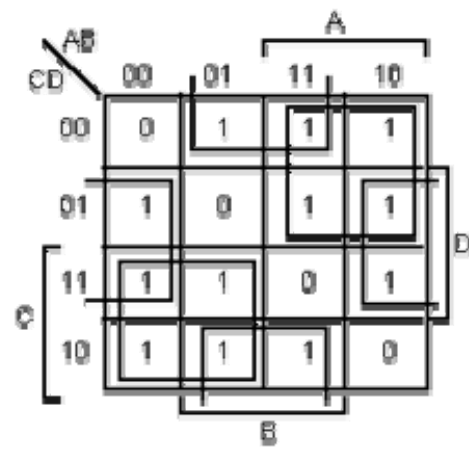
Projeto com PLA

- Deve-se minimizar todas as funções de saída simultaneamente, procurando-se o menor número de termos-produto comuns a todas as funções
- Deve-se tentar solução ótima tanto com F como com \overline{F} , procurando:
 - menor número de termos
 - termos comuns a outras funçõesNo exemplo: AC é comum a $F1$ e $\overline{F2}$
- Tamanho do PLA
 - Nro. entradas = número de variáveis da função
 - Nro. linhas do array AND = nro. máximo de termos-produto no conjunto das funções
 - Nro. saídas = número de funções simultâneas

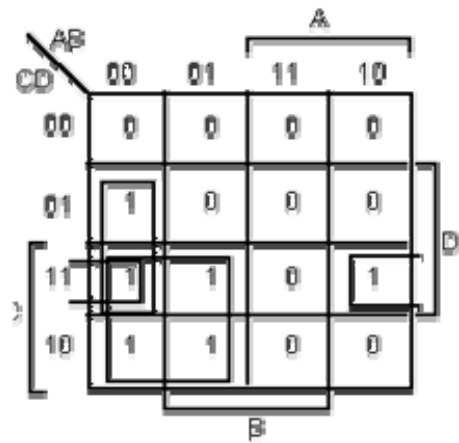




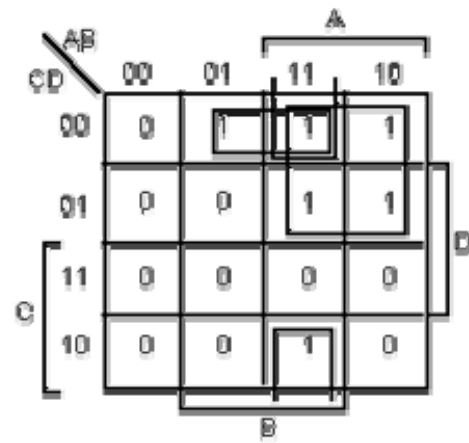
K-map for EQ



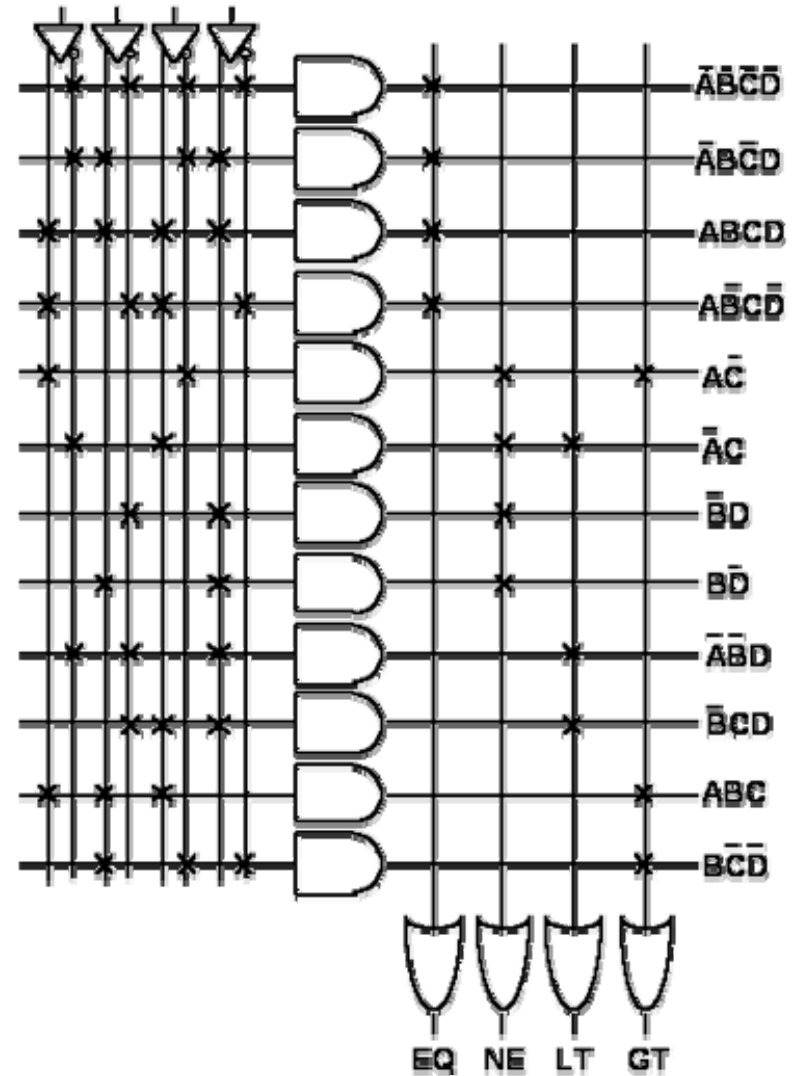
K-map for NE

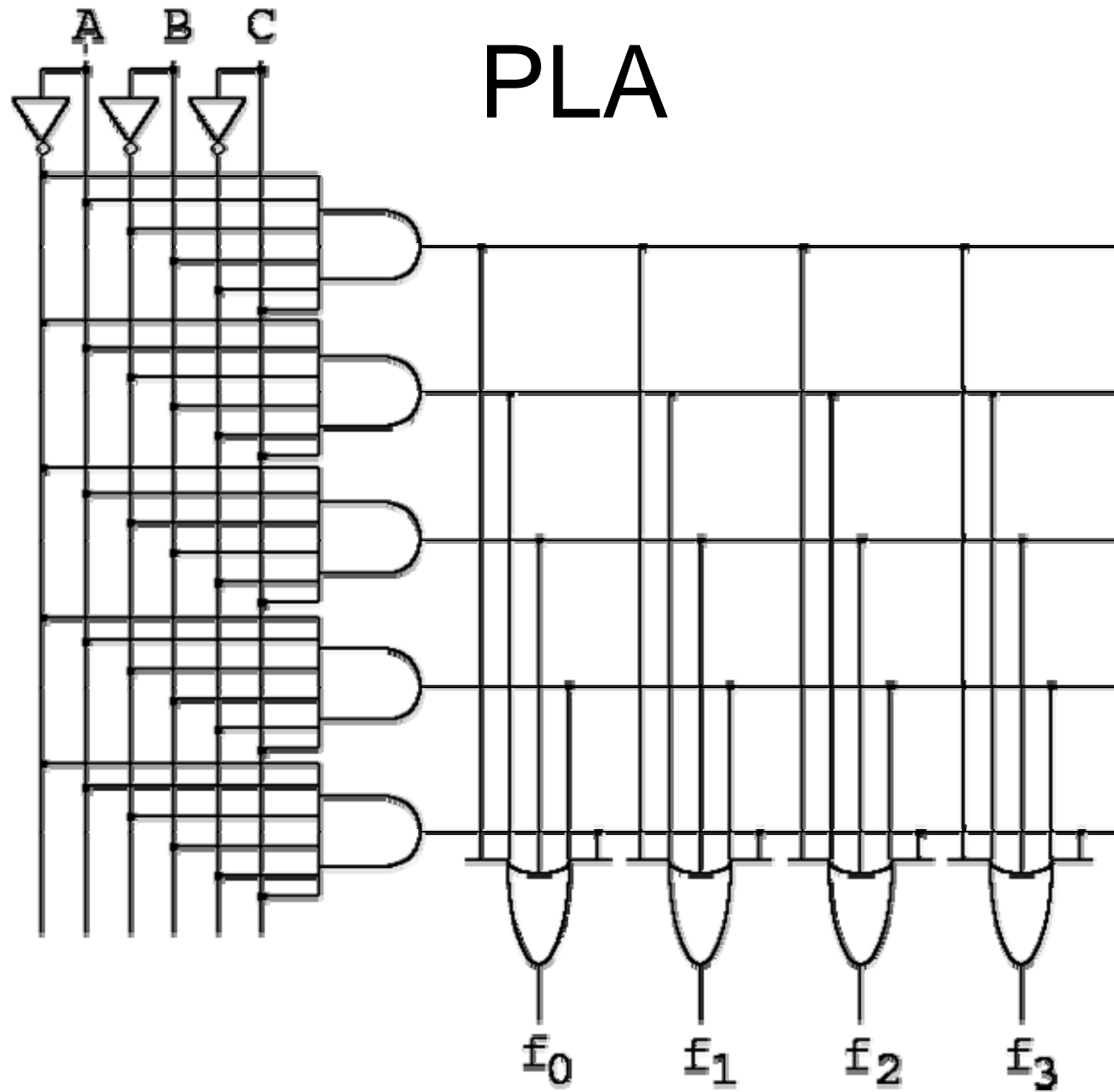


K-map for LT

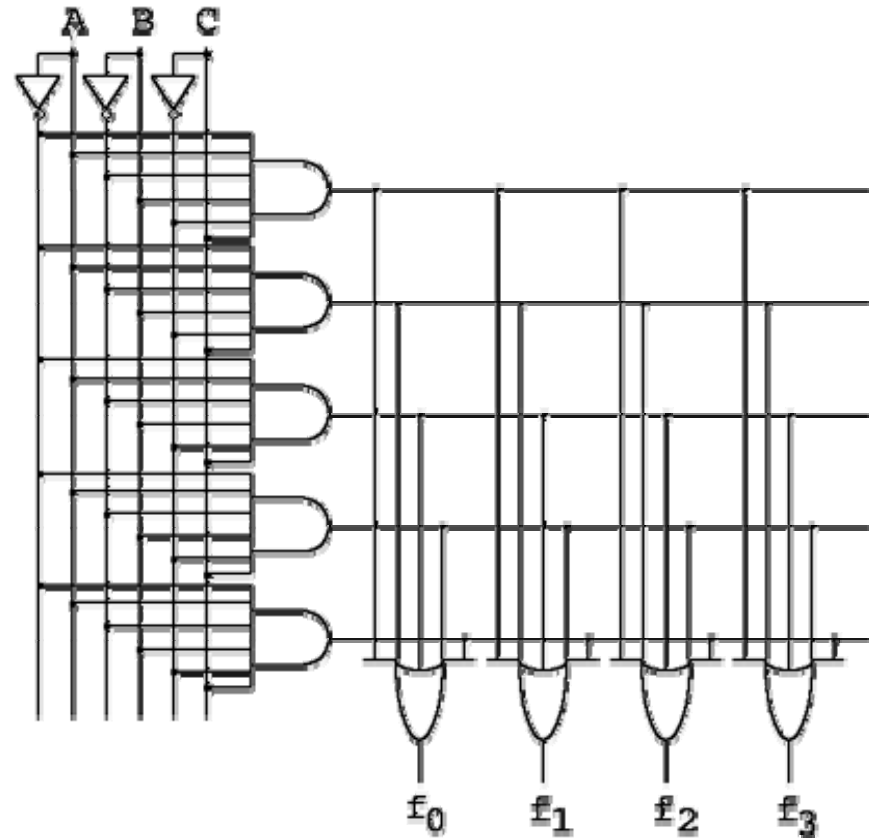


K-map for GT





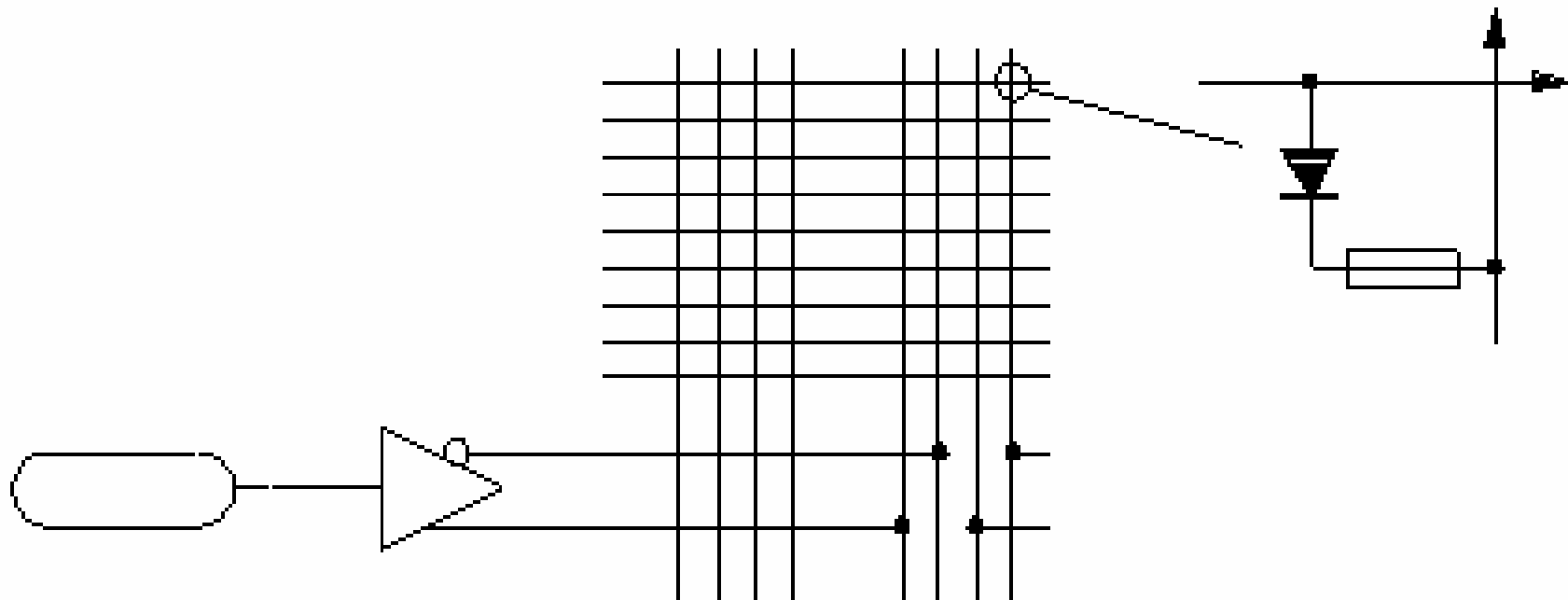
- $F_0 = A + B' C'$
- $F_1 = A C' + A B$
- $F_2 = B' C' + A B$
- $F_3 = B' C + A$



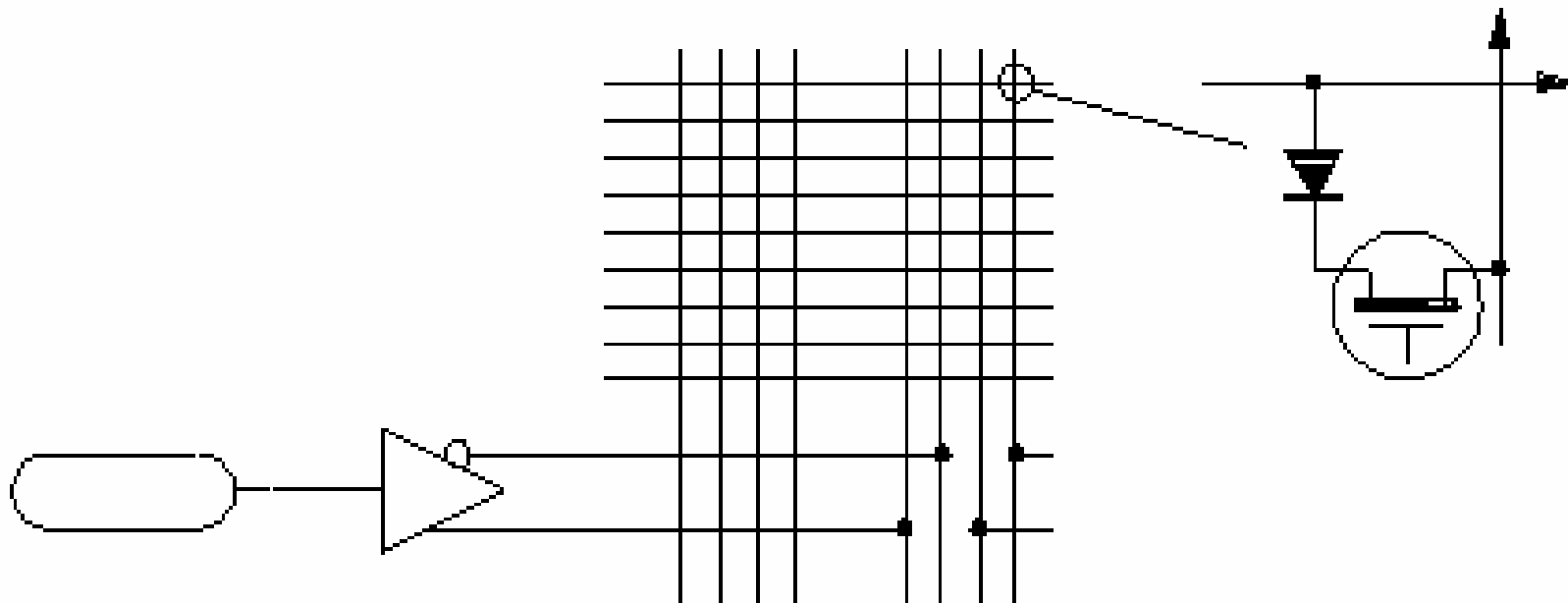
Elementos de Programação

- **Anti-fusível:** não volátil, entre metais
- **EPROM e EEPROM:** transistor que liga ou desliga conforme tensão aplicada no gate. Pode ser apagado por ultra-violeta ou eletricamente.
- **SRAM:** célula de memória conectada a um transistor de passagem.

Elemento Programável com Fusível (PAL)



Elemento Programável com MOSFET de Porta Flutuante (EPLD)



Elemento Programável com RAM (FPGA)

