

**Departamento de Engenharia Elétrica e de Computação  
SEL 405 – Lab. de Introdução aos Sistemas Digitais I**

**Profa. Luiza Maria Romeiro Codá**

**“DISPOSITIVOS LÓGICOS PROGRAMÁVEIS”**

**1. Introdução:**

A tecnologia dos circuitos digitais tem se desenvolvido rapidamente nas últimas décadas. Os avanços constantes têm transformado de forma radical todo o processo de projeto de hardware. A microeletrônica apresenta uma história longa de evolução num período muito curto, passando pelo descobrimento do efeito transistor em 1947 na Bell Labs e, pelo desenvolvimento do processo planar para a fabricação de CI's (circuitos integrados) em 1959 na Fairchild, resultando nos primeiros CI's comerciais em 1962.

A revolução da microeletrônica foi possível por causa do significativo progresso no desenvolvimento de muitas áreas como: a tecnologia de processos no silício, componentes semicondutores e ferramentas de projeto. Os componentes dos circuitos digitais evoluíram de transistores individuais para circuitos integrados VLSI (*very large scale integration*). A criação do microprocessador possibilitou uma alavanca para o desenvolvimento de chips de memória de alta densidade e, o avanço na tecnologia de produtos lógicos tais como Dispositivos Lógicos Programáveis (PLDs).

Antigas gerações, de tecnologia de hardware, para projetos digitais apresentavam um grande número de chips contendo portas básicas (AND e OR). Com o avanço tecnológico os projetos mais complexos passaram a usar circuitos de alta densidade, que em geral incluem dispositivos como controladores, contadores, registradores e decodificadores, proporcionando o desenvolvimento de alternativas para obter flexibilidade em uma implementação. Com o uso de módulos programáveis (portas básicas que permitem a implementação de sistemas digitais). Estes módulos têm uma estrutura padrão e são personalizados para uma função particular.

Exemplos de tipos e classes de dispositivos programáveis (configuráveis) com capacidade de implementar funções lógicas são:

- PLA (Programmable Logic Array),
- PAL (Programmable Array Logic),
- CPLD's (Complex Programmable Logic Devices)
- MPGA (Mask Programmable Gate Array)
- FPGA (Field Programmable Gate Array)

A alta capacidade e a versatilidade dos dispositivos programáveis fazem destes uma alternativa considerável no projeto de sistemas digitais (desde circuitos combinacionais simples até circuitos seqüenciais complexos, p.ex.: microprocessadores).

A utilização de ferramentas EDA (Eletronic Design Automation) ou ECAD((Electronic Computer-Aided Design) tem simplificado e acelerado todo o ciclo de projeto. Atualmente, não é mais necessário desenhar portas lógicas individuais e planejar todas suas interconexões. As linguagens de descrição de hardware (HDLs) estão hoje consolidadas no meio acadêmico e industrial como forma padrão na elaboração de projetos. Existem também, ferramentas de síntese lógica automática, disponíveis para mapear circuitos em diversas tecnologias.

Todas essas mudanças na tecnologia exigem uma prototipação cada vez mais rápida, pois o ciclo de vida dos produtos modernos está tomando-se cada vez mais curto em relação ao tempo necessário para o projeto e desenvolvimento dos mesmos. As implementações de circuitos podem ser agrupadas em diversas categorias:

- ◆ **CI's customizados ou ASICs (*application specific integrated circuits*):** São aqueles que necessitam de um processo de fabricação especial, que requer máscaras específicas para cada projeto. Outras características desse tipo de implementação são o tempo de desenvolvimento longo e os custos extremamente altos. Em aplicações que requerem um grande volume de produção, o alto custo do projeto e dos testes é amortizado.
- ◆ **MPGAs (*mask-programmable gate arrays*) :** Nesse tipo de implementação, o processo de fabricação é agilizado pelo uso de máscaras genéricas de módulos pré-projetados, mas ainda necessita de máscaras específicas para a interconexão dos módulos. O projeto é normalmente facilitado por uma biblioteca de células, proporcionando um tempo de desenvolvimento mais curto e custos mais baixos em relação aos CI's customizados.
- ◆ **Standard Cells :** Essa tecnologia se assemelha muito a das MPGAs, o projeto também é facilitado pelo uso de módulos pré-projetados. Os módulos (*standard cells*) são geralmente salvos em bancos de dados. Os projetistas selecionam as células desejadas (nesses bancos de dados) para realizar seus projetos. Em comparação aos CI's customizados, os circuitos implementados em *standard cells* são menos eficientes em tamanho e desempenho, entretanto, seu custo de desenvolvimento é mais baixo.
- ◆ **PLDs (*programmable logic devices*):** Essa tecnologia possui como principal característica a capacidade de programação (configuração) pelo usuário, eliminando o processo de fabricação e facilitando assim as mudanças de projetos. Em comparação com outras tecnologias, os PLDs apresentam um ciclo de projeto muito curto e custos muito baixos.

O mercado de PLDs encontra-se em plena expansão, de forma que atualmente existem diversos fabricantes e modelos de dispositivos desse tipo. Uma das principais tarefas do projetista hoje é pesquisar e selecionar, dentre as opções disponíveis no mercado, qual a que melhor atende às suas necessidades.

## 2. Histórico:

Em um rápido histórico pode-se observar que o conceito de programação de hardware se materializou com a necessidade de se construir uma unidade de memória cujo conteúdo fixo não se perdesse ao se desligar o sistema.

As **ROMs (Memória de Apenas Leitura)**, surgiram no final dos anos 70, as quais eram fornecidas pela fábrica com seu conteúdo já determinado, (através da máscara de metalização). A Figura 2.1 mostra a estrutura de um dos tipos mais antigos de ROM. Ela é composta por uma matriz de células de diodos ou transistores (Figura 2.2). A programação da memória, com "UM" ou "ZERO", é feita pela conexão ou não de dispositivos à grade de memória na fase de metalização (fabricação).

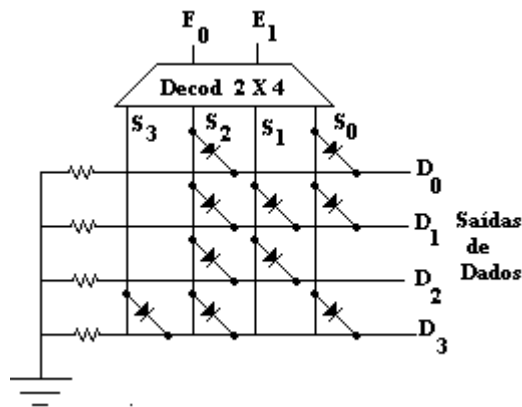


Figura 2.1. Estrutura de uma ROM programada por máscara de metalização.

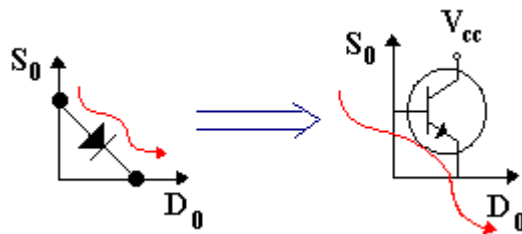
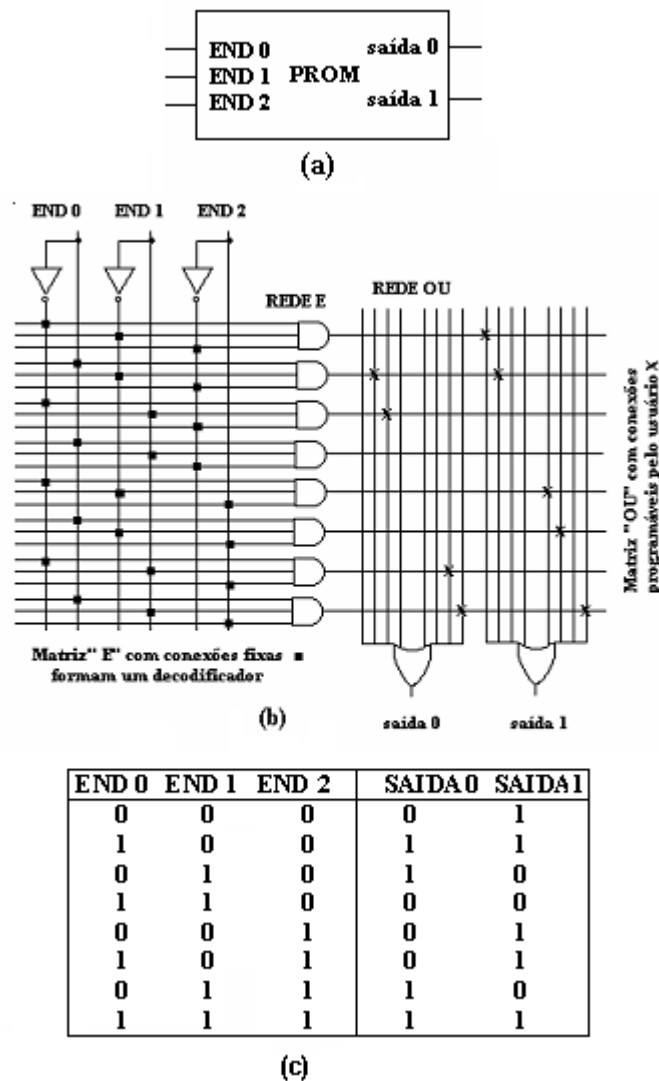


Figura 2.2. Transistores substituindo diodos na matriz de uma ROM.

Logo depois, na mesma década, surgiram as **Memórias de Apenas Leitura Programáveis (PROM)**, esquematizada na Figura 2.3a, elas eram semelhantes às memórias ROM, porém eram programadas pelo usuário. A programação era obtida "queimando-se" fusíveis internos, isto é, desfazendo-se conexões internas, Figura 2.3b.



**Figura 2.3.(a) Esquema em blocos de uma PROM de 8x2 bits**  
**(b) Esquema de ligação interno de uma memória PROM de 8 x 2 bits**  
**(c) Tabela verdade do decodificador (PROM 8X2)**

A memória **PROM** é semelhante a um grande decodificador de endereços, representado pelas portas AND's de múltiplas entradas e, através de conexões apropriadas da saída do decodificador às entradas dos blocos OR, determina-se se a saída será "UM" ou ZERO". A tabela verdade é mostrada na Figura 2.3c. A PROM é fabricada com todas as conexões, isto é, como se estivesse com "UM" armazenado em todos os bits e em todos os endereços. Para programar, aplica-se, de forma apropriada, tensões elevadas, entre 10 a 30V, através de um pino especial da pastilha, queimando-se as conexões nos bits cujo conteúdo desejado fosse "ZERO", mostrado na Figura 2.4.

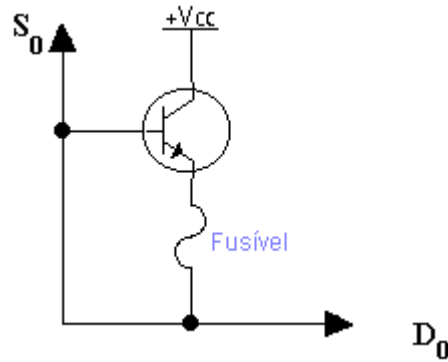


Figura 2.4 Célula de uma PROM

Inconvenientemente, uma vez queimado o fusível, em uma PROM, não seria mais possível o retorno dele para "UM". A evolução da tecnologia permitiu contornar esse problema com o desenvolvimento das memórias **EPROMs** que são PROMs apagáveis por radiação ultravioleta e das **EEPROM** apagáveis eletricamente.

A memória PROM foi o primeiro tipo de *chip* programável pelo usuário, que podia implementar circuitos lógicos. As linhas de endereço eram utilizadas como entradas do circuito lógico, e as linhas de dados como saídas desses circuitos. As funções lógicas, entretanto, raramente requerem mais que alguns termos produto, e uma PROM contém um decodificador completo para seus endereços de entradas. As PROMs, portanto, se mostram como uma arquitetura ineficiente para o projeto e realização de circuitos lógicos, e são raramente utilizadas para esse fim .

Na **EPROM**, ao invés de fusíveis como na PROM, a programação é feita por armazenamento de carga, onde cada bit da memória possui um transistor MOS com dois Gates, um deles flutuante, não conectado a grade de memória, e isolado por material de altíssima impedância, Figura 2.5.

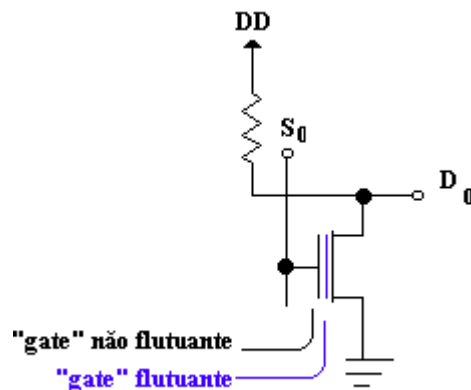


Figura 2.5 Estrutura do transistor MOS com gate flutuante de uma EPROM.

Em estado "apagado", como ao sair da fábrica, esses transistores não conduzem quando selecionados, e o conteúdo da posição de memória é levado à "UM" por resistores chamados de "pull-up". Para gravar um valor "ZERO" aplica-se uma alta tensão no gate-flutuante, o que causa uma ruptura (*break-down*) no material isolante, permitindo assim um acúmulo de cargas no gate flutuante, as quais ali permanecem mesmo após o término do pulso de tensão, devido a alta impedância do material isolante. A presença dessas cargas no gate do transistor provoca a condução quando a posição daquele bit for selecionada.

A alta impedância do material isolante permite que uma EPROM mantenha sua programação por mais ou menos 10 anos. Para apagá-las utiliza-se luz UV. Os fótons de alta energia colidem com os elétrons armazenados na gate flutuante e provocam seu espalhamento, desfazendo o acúmulo de cargas e portanto, a programação. Existe um número máximo de vezes para apagar e regravar. Após a programação cobre-se a janela da pastilha com uma etiqueta opaca, pois a exposição ao sol apaga a EPROM em uma semana, e a exposição à luz fluorescente do ambiente de trabalho apaga a EPROM em mais ou menos 3 anos.

A **EEPROM (ou E<sup>2</sup>PROM)**, a qual é uma PROM eletricamente apagável, pode ser apagada e gravada no próprio circuito, sem equipamentos adicionais (apagadores e programadores). Pode-se programar apenas 1 bit ou 1 byte sem apagar a memória inteira. A EEPROM tem transistores com gate flutuante com material isolante de espessura muito menor. Desta forma, cada bit pode ser apagado pela aplicação de uma tensão no gate não flutuante, oposta a que gerou as cargas na gravação. A EEPROM não é volátil e é útil para armazenamento de dados preservados quando o equipamento for desligado.

A EPROM ou EEPROM, embora seja uma ROM, permite leitura e escrita, mas não substitui uma memória do tipo que lê e escreve (RAM) pois a EPROM tem tempo de escrita muito superiores, custo maior e aceita um número limitado (10 mil) de ciclos de apagamento/gravação.

Com a PROM passou-se a dispor além de uma memória de apenas leitura, também de um circuito combinatório genérico de n entradas e m saídas, cuja função de transferência poderia ser facilmente programável. Portanto, O dispositivo programável mais simples e que foi o primeiro chip a ser programado pelo usuário é o PROM. Uma das principais aplicações desse primeiro tipo de PLD foi para implementar circuitos lógicos.

Introduzido no mercado pela Philips na década de 70, o PLA (*Programmable Logic Array*) foi o primeiro dispositivo desenvolvido para implementar funções lógicas definidas. Um PLA é constituído por arranjos AND e OR, onde ambos são programáveis. Por esse motivo o PLA é adequado para a implementação de funções na forma de soma de produtos, sendo muito versátil, pois tanto os arranjos AND como os OR podem ter muitas entradas. Porém, ela não foi bem aceita pelos projetistas pelo seu alto custo.

Em 1978 foi introduzido pela Monolithic um inovador dispositivo de lógica programável chamado PAL (*Programmable Array Logic*). Este é constituído também por arranjos AND e OR, sendo que o arranjo AND é programável e o arranjo OR é fixo. Esta disposição barateou em muito a fabricação dos dispositivos de lógica programável, contudo para compensar a impossibilidade de programar os arranjos OR, alguns PAL's são fabricados com diferentes números de entradas e saídas, bem como vários tamanhos de portas OR.

Hoje, tanto os PLA's e PAL's, bem como outros dispositivos que surgiram a partir dessas estruturas e que possuem nomenclaturas diferentes, são classificados como SPLD's (*Simple PLD's*), cujas características principais são o baixo custo e o alto desempenho.

Atualmente existentes diversos PLD's sendo comercializados com diferentes estruturas, com mais ou com menos blocos SPLD's, onde são chamados de HCPLD's

(High Capacity *PLD's*). Os primeiros *CPLD's* foram introduzidos pela Altera Corporation.

Os *HCPLD's* disponíveis atualmente de maior capacidade lógica são os tradicionais *Gate Arrays*, algumas vezes chamados de *MPGA (Mask - Programmable Gate Array)*. Assim como as *MPGA's*, as *FPGA's (Field Programmable Gate Arrays)* são formadas por blocos lógicos e interconexões entre estes blocos, e a diferença está na configuração realizada através de programação pelo usuário final. No ano de 1985, o primeiro *FPGA* foi disponibilizado para ser comercializado pela empresa Xilinx Inc.

Atualmente os *FPGA's* oferecem capacidades lógicas de trabalho elevadas e são os responsáveis pela maior parte das mudanças ocorridas na maneira como os circuitos digitais são desenvolvidos.

No início da década de 80, simples *PLDs* eram utilizados na implementação de múltiplos dispositivos lógicos discretos. Hoje em dia, *PLDs* integram em um único dispositivo grande quantidade de blocos capazes de implementar muitas funções lógicas, e são também geralmente preferidos do que os circuitos integrados de aplicações específicas (*ASICs*). Um dispositivo *ASIC* é desenhado para atender às características solicitadas pelo projetista de circuitos e terá seu uso exclusivo em tal finalidade, ou seja, é um dispositivo dedicado.

Hoje o custo por unidade dos dispositivos *PLDs* reduziu devido ao alto volume de fabricação e também por causa da utilização de novas tecnologias, sendo possível encontrar no mercado *PLDs* com integração, densidade, performance, e custo equivalentes ao de um dispositivo não programável. Tais fatores, combinados com tempo de produção e flexibilidade na programação ajudam a manter o crescimento do uso de *PLDs* de alta densidade no desenvolvimento projetos de eletrônica digital.

Os *PLDs* são confeccionados em lógica *CMOS*, o que garante características de confiabilidade. Com a densidade dos *PLDs* acima de a 250.000 portas (para os mais modernos), é possível projetar subsistemas inteiramente digitais com um único dispositivo *PLD*.

### **3. Dispositivos Lógicos Programáveis (PLD'S):**

Com o número enorme de *CI's* -padrão que estão disponíveis no mercado, o projetista de sistemas lógicos poderia implementar qualquer função necessária em projetos de computadores. Porém, ele pode se deparar com o uso de uma quantidade imensa de *CI's* - padrão, exigindo um espaço considerável na placa de circuito impresso. A redução no número de *CI's* em um projeto acarreta em inúmeras vantagens além de menor espaço ocupado na placa de circuito impresso, também um menor consumo de energia, levando a utilização de fontes menores e mais baratas e maior confiabilidade no projeto, uma vez que possui um número menor de interligações, e velocidades mais altas.

Para diminuir o número de *CI's* de um projeto é necessário colocar mais funções dentro de um chip, o qual foi conseguido com o desenvolvimento de técnicas de

integração LSI, VLSI e ULSI. O desenvolvimento dessas tecnologias e a de programação de hardware, as quais permitem realizar ou desfazer conexões, era o que faltava para desenvolver componentes cuja função fosse programável pelo usuário, para realizar funções diversas e específicas. O desenvolvimento de **Dispositivos Lógicos Programáveis (PLD-Programmable Logic Devices)** tem proporcionado a substituição de diversos chips por um único CI, além de poder ser programado pelo usuário, eliminando o processo de fabricação e facilitando eventuais mudanças no projeto, diminuindo o tempo e custo do projeto.

Portanto, PLD é todo circuito de lógica digital configurado pelo usuário final, incluindo simples, baixa densidade, e alta capacidade, utilizados para implementar funções lógicas. A programação é feita por meio de campos elétricos induzidos no dispositivo, essa programação é feita via softwares especiais fornecidos pelos próprios fabricantes desses produtos. É um CI que pode conter grande número de portas lógicas, flip-flop e latches interconectados dentro do chip, além de e registradores. nas entradas e saídas. A maioria de tais conexões é feita através de fusíveis que podem ser queimados ou mantidos intactos.

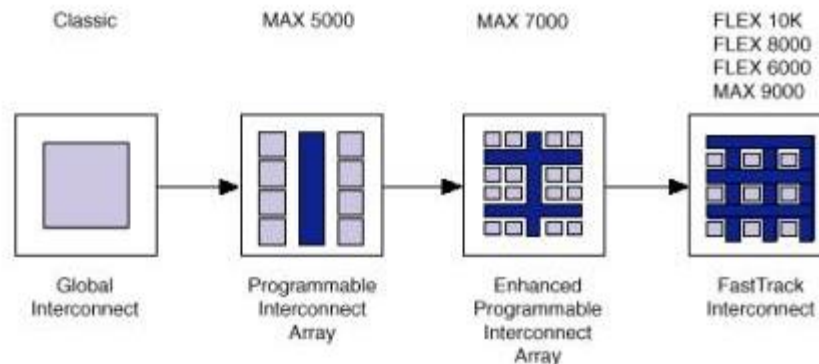
PLD's são oferecidos em diferentes arquiteturas, e uma variedade de elementos de memória são disponíveis para a configuração do dispositivo.

A estrutura interna de um PLD é baseado em um array que conecta os barramentos aos blocos lógicos, aos pinos de entrada e saída, e aos blocos de memória.

Os PLD's, em princípio, possuíam apenas um único bloco lógico (que realizava globalmente as operações), conectado aos pinos de entrada e saída do dispositivo. Com o aperfeiçoamento da tecnologia de fabricação, esses dispositivos receberam uma estrutura de interconexões programáveis, que viriam flexibilizar mais ainda a programação.

Na verdade, o grande desafio de Eletrônica Digital é o desenvolvimento do componente "universal" que alie versatilidade com elevada capacidade, alta velocidade, baixo custo e fácil manuseio.

A evolução trouxe uma otimização do aproveitamento de espaço, aumentando ainda mais as interconexões do array de blocos lógicos. A Figura 3.1 mostra a evolução dos PLD's.



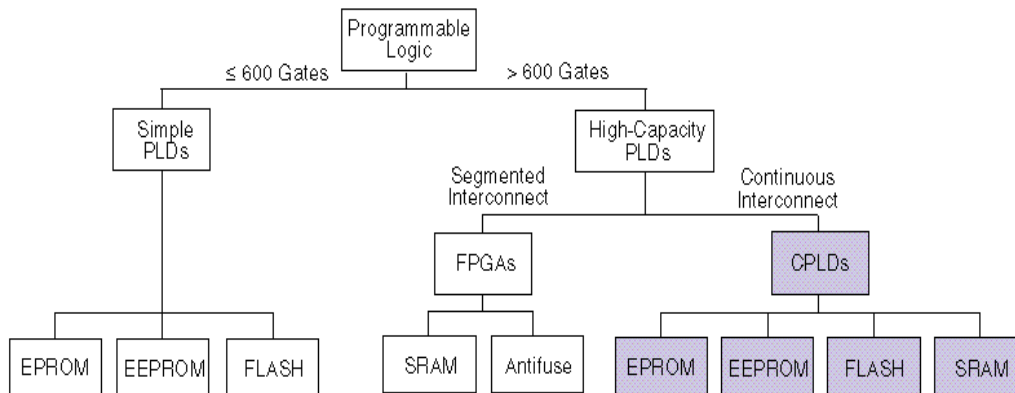
**Figura 3.1. Evolução da arquitetura dos PLD's da ALTERA.**



## 4. Tipos de dispositivos e Lógica Programável:

Os dispositivos de lógica programável podem ser classificados em dois tipos diferentes (mostrados na Figura 4.1):

- Dispositivo de Lógica Programável Simples (“**SPLD- Simple Programmable Logic Device**”)
- Dispositivo de Lógica Programável de Alta Capacidade (“**HCPLD- High Capacity Programmable Logic Device**”).



**Figura4.1 Tipos de Dispositivo de Lógica programável.**

Os PLD's simples usam tecnologia CMOS e oferecem elementos de memória do tipo EPROM, EEPROM, e memória FLASH, possuem uma quantidade de portas abaixo de 600 e englobam os dispositivos PALs, GALs, e outros componente. Os HCPLDs (dispositivos lógicos programáveis de alta capacidade), e oferecem mais de 600 portas disponíveis. Também utilizam tecnologia CMOS com memória EPROM, EEPROM, FLASH, SRAM, e opções anti-fusível. Por HCPLD's entendem-se os dispositivos FPGA e CPLD.

Os dispositivos cuja programação é baseada em anti-fusível não podem ser reconfigurados, pois são programáveis uma única vez. Isso é uma desvantagem, porque os projetistas frequentemente precisam testar o hardware ou incorporar mudanças inesperadas ao projeto. Utilizando um dispositivo que é construído com elementos anti-fusível tais modificações no projeto não poderão ser feitas.

Para realizar a programação desses dispositivos existem vários softwares que são desenvolvidos pelos próprios fabricantes de PLDs e fornecidos aos projetistas, de acordo com o elemento a ser programado.

## 4.1 SPLDs (Simple Programmable Logic Devices):

São circuitos que possuem uma estrutura interna baseada em um conjunto de portas AND/OR denominados arranjos lógicos, podendo ou não possuir flip-flops na saída, dependendo da configuração. É a categoria de todos os pequenos PLDs, onde as características mais importantes são baixo custo e alto desempenho.

### 4.1.1 Matriz Programáveis Logicamente (PLA): (ou Matriz programável logicamente em campo (FPLA))

Observando a PROM existem duas matrizes de interconexões AND e OR. Na PROM a matriz AND é fixa, decodificador completo e a matriz OR é programável. A arquitetura da PROM é adequada àquelas aplicações onde cada possível combinação de entrada é necessária à geração da função de saída. Exemplo: conversores de código e as tabelas de dados utilizadas na geração de funções. Uma PLA, Figura 4.1, tem a estrutura semelhante à da PROM com duas diferenças:

- Dispõe de menor quantidade de portas AND (não é decodificador completo)
- Tem duas matrizes de conexão das portas (AND e OR) programáveis (para compensar o número menor de portas AND).

A PLA, embora seja o mais versátil de todos os Dispositivos Lógicos Programáveis por possuir dois conjuntos de fusíveis, é mais difícil de fabricar, projetar que a PROM, conseqüentemente tem um custo mais alto.

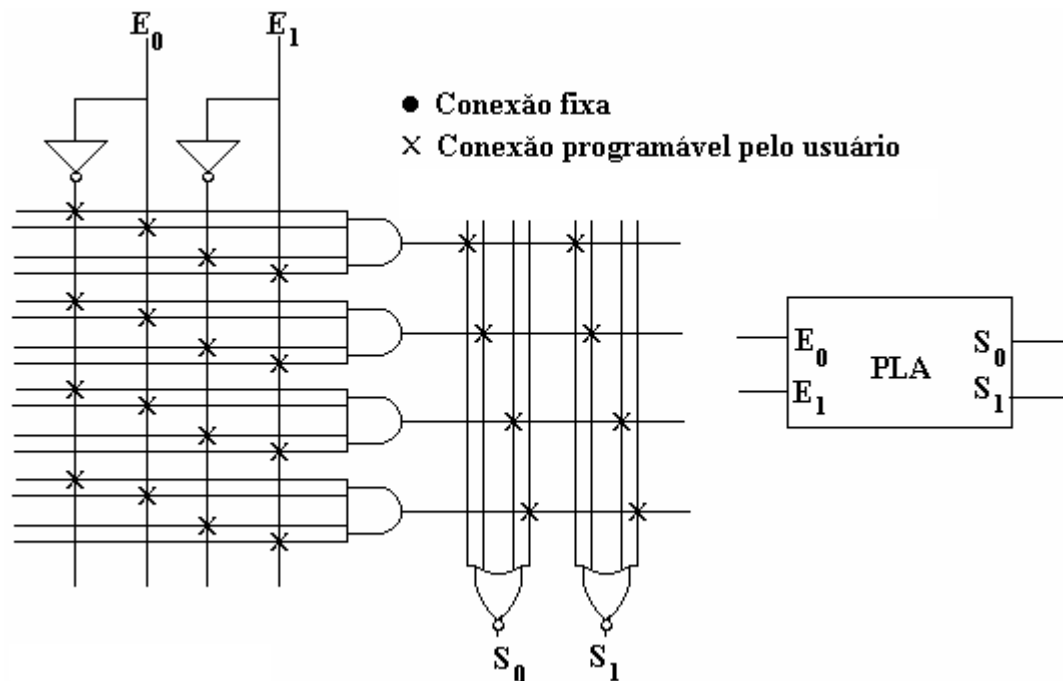
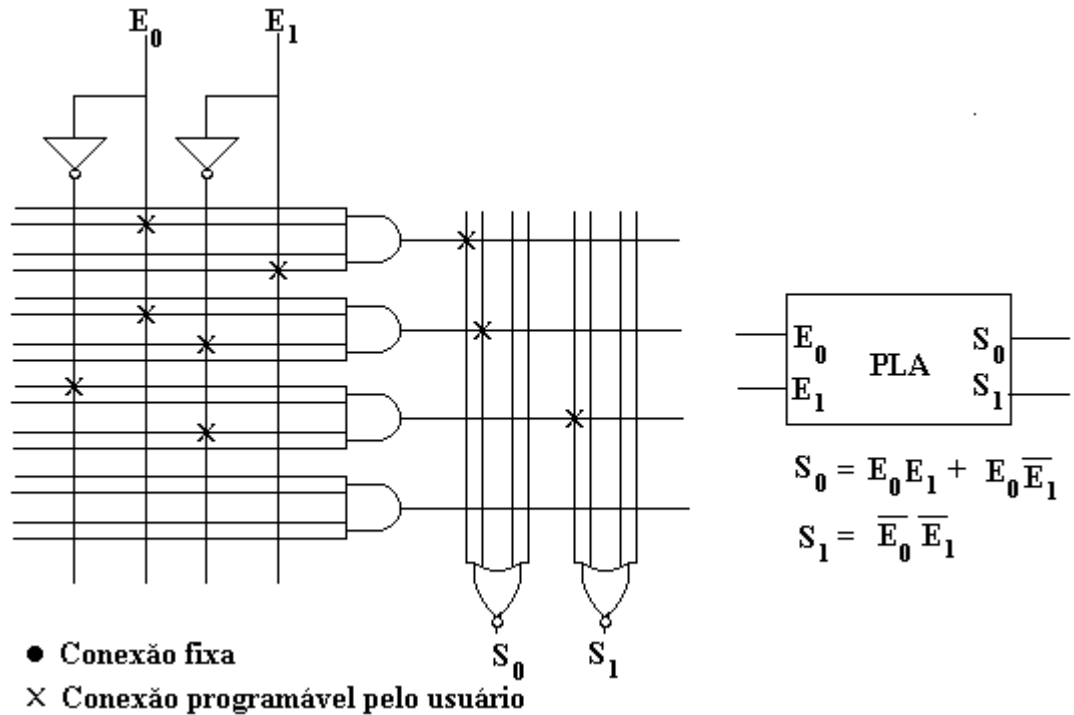


Figura 4.1 Uma PLA ainda não programada, com todos os fusíveis intactos

A Figura 4.2 mostra uma PLA programada para a função:

$$S = E_0E_1 + E_0\bar{E}_1 + \bar{E}_0\bar{E}_1$$

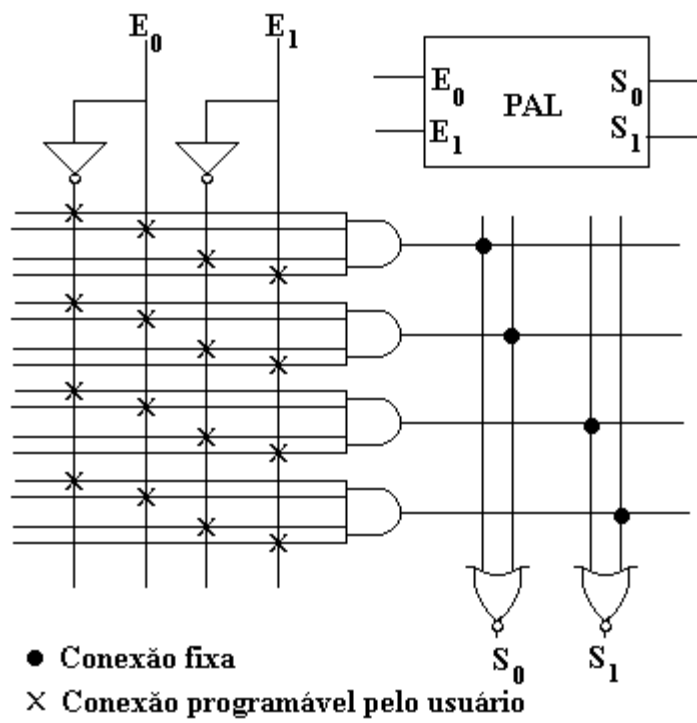
A conexão programável consiste em um fusível o qual intacto apresenta ligação feita, queimado ligação desfeita, rompida.



**Figura 4.2** PLA de 4 entradas programada

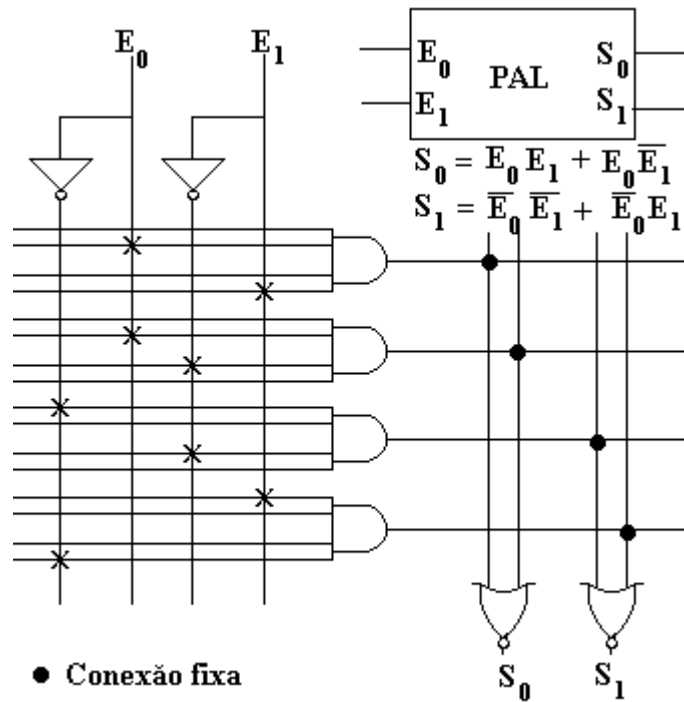
### 4.1.2 Matriz Lógica Programável (PAL):

Muitas aplicações não necessitam de que todas as combinações de entrada sejam programáveis, por essa razão foi criada a PAL, a qual é uma simplificação da PLA, sendo apenas a matriz de conexão das portas AND programável, a matriz OR é fixa, como mostra a Figura 4.3. Cada porta AND pode ser programada para gerar qualquer dos produtos AND das duas variáveis de entrada. E de seus complementos. Cada porta OR está ligada somente a duas das saídas ANDs. Isso limita a função a ser implementada a ter no máximo dois produtos AND. Se a função precisar de mais de dois produtos não pode ser implementada nesta PAL devendo ser escolhido outro modelo que tenha mais entradas para as portas OR. Caso sejam necessárias menos de dois termos para a implementação da função, as saídas não utilizadas devem ser colocadas no nível lógico "0".



**Figura 4.3 Uma PAL ainda não programada, todos os fusíveis intactos**

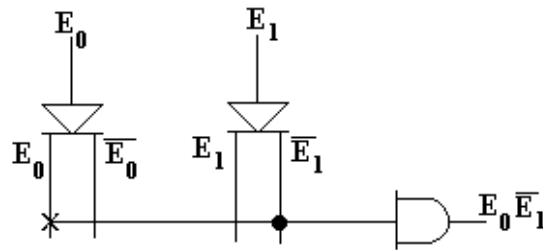
A Figura 4.4 mostra uma PAL programada para a função:  
 $S = E_0E_1 + E_0\bar{E}_1 + \bar{E}_0\bar{E}_1 + \bar{E}_0E_1$



**Figura 4.4 PAL de 4 entradas programada**

As PALs foram inicialmente fabricadas com tecnologia bipolar, portanto possuem alta potência de consumo e configuração da célula de saída que só pode ser usada para lógica combinacional.

A Figura 4.5 mostra um exemplo da simbologia simplificada para uma AND de quatro entradas. Em primeiro lugar observe que os buffers de entrada são representados sem a distinção entre inversores e não inversores. Observe também que uma única linha aparece como entrada da porta AND, representando as quatro entradas existentes. As conexões das variáveis de entrada à porta AND são indicadas ora por um ponto, ora por um X. O X representa um fusível intacto, e o ponto representa uma conexão fixa. A ausência do X e do ponto indica ausência de conexão. Nesse exemplo as entradas A e B estão ligadas à porta AND de maneira a gerar o produto  $A\bar{B}$ .

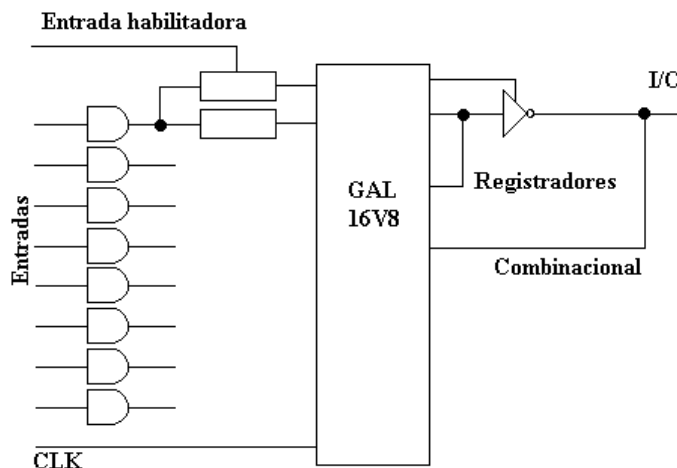


- Conexão fixa
- × Conexão programável pelo usuário

**Figura 4.5 Simbologia simplificada para os PLDs**

### 4.1.3 Arranjo Lógico Genérico (GAL):

AS GAL's ("Generic Array Logic") são estruturas como a PAL, mas possuem macrocélulas de saídas as quais podem ser configuradas como lógica combinacional ou sequencial (registradores), como mostrada na Figura 4.6.



**Figura 4.6 Circuito típico de uma GAL**

Uma única GAL pode substituir 24 tipos diferentes de PAL. A GAL é construída com a tecnologia CMOS, isso reduz o consumo de potência comparado à dispositivos bipolares, mas possui alta taxa de ruído. A Gal possui a mesma quantidade de pinos do que uma PAL, em torno de 20 e 24 pinos.

#### 4.1.4 Dispositivos Lógicos Programáveis Apagáveis (EPLD):

Os EPLD's ("Ereaseable Programmable Logic Device") são dispositivos que possuem quantidades de pinos entre 20 a 66, possuem um consumo mais baixo de potência do que as GALs, em torno de metade da potência, mas não são rápidos, porém possuem um custo baixo e são fáceis de usar em desenvolvimento de sistemas. Possuem os registradores programáveis como tipo D, T, JK ou Flip-flop RS com quatro entradas para clocks independentes. O problema dos EPLDs é que possuem arranjo de portas OR fixas e AND programáveis, as quais apresentam baixa utilização de portas. São utilizados em telecomunicações e aplicação industrial. A Figura 4.7 mostra uma macrocélula como decodificador de endereços. A macrocélula tem oito termos de produtos, mas apenas um deles é usado. Entretanto, quando uma aplicação necessita de mais de oito termos de produtos deve ser feito um "loop" de realimentação o qual torna o projeto muito mais lento.

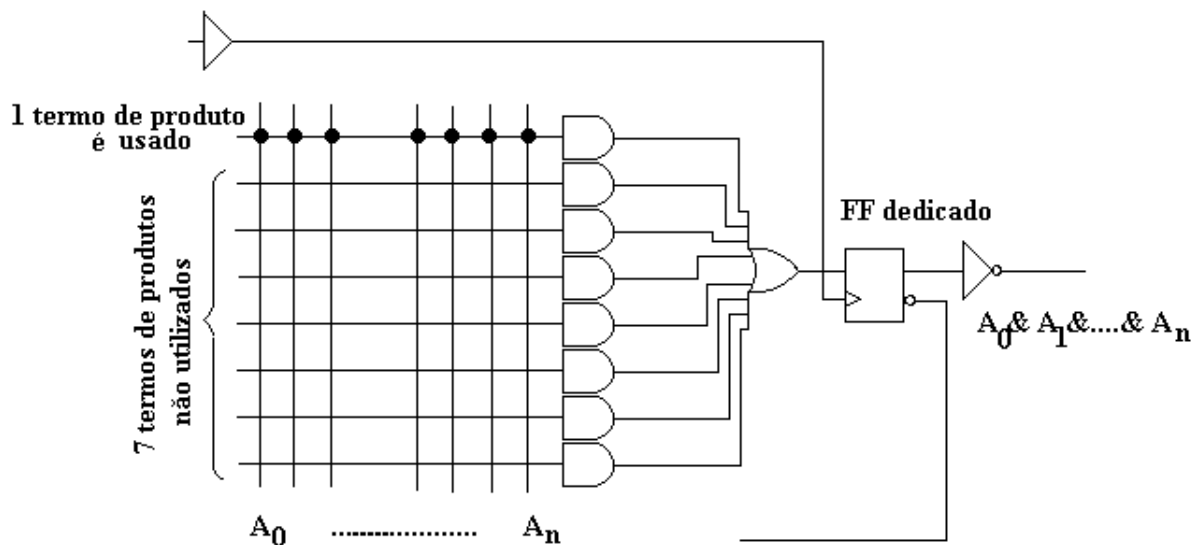


Figura 4.7 Macro célula como decodificador de endereços

## 4.2 Dispositivo de lógica Programável de Alta Capacidade (HCPLD):

Com a necessidade cada vez maior de implementar circuitos cada vez mais complexos, conexões entre vários SPLD's tornou-se comum. Para contornar o velho problema de interconectar vários *chips*, no caso vários SPLD's, os fabricantes de dispositivos lógicos programáveis passaram a integrar em um único *chip* vários SPLD's.

HCPLD São dispositivos lógico-programáveis que possuem mais de 600 portas. Quanto maior o número de portas do *PLD*, maior será sua complexidade, principalmente no que se refere à estrutura de conexão. Os *HCPLD* são divididos, basicamente em:

- CPLD (*Complex Programmable Logic Devices*)
- FPGA (*Field Programmable Gate Array*).

A diferença básica entre CPLD e FPGA reside no método de interligação das células.

### 4.2.1 Estrutura de um HCPLD:

Um HCPLD é constituído por várias estruturas repetidas conhecidas como **células**. Cada célula é constituída de elementos que implementam lógica combinacional acompanhados de *Latches* (através de Memórias e Flip-Flops) destinados a aplicações de lógica sequencial. Interligando estas duas estruturas, tem-se multiplexadores programáveis que realizam a interligação destes blocos entre si e destes com o exterior da célula (para outra célula ou para barramentos).

Um **barramento** é uma estrutura que se destina a promover a interligação entre as células. São conexões físicas (trilhas metalizadas) disponíveis no interior do *chip*. Dependendo do tipo de PLD envolvido (FPGA ou CPLD), de sua complexidade e de seu fabricante, tem-se diferentes recursos e modos de conexão. Basicamente esses blocos de conexão são estruturas de **conexão segmentadas** (FPGA) e **estruturas contínuas**(CPLD).

A interface com o mundo exterior é feita através de unidades conhecidas como **blocos de entrada e saída**. Sua função é conectar as células aos pinos do integrado que normalmente pode ser realizada diretamente (Célula → Bloco I/O) ou indiretamente (Célula → Barramento →Bloco I/O). A seguir serão descritas mais detalhadamente cada uma dessas estruturas

### A. Célula Lógica:

A **célula lógica** é todo conjunto ou bloco lógico que possui conexão com os barramentos do integrado. As variações entre as células identificam os tipos diferentes de HCPLD. As células são divididas em duas partes:

- **Elementos Geradoras de Lógica Combinacional**
- **Estruturas de unidades das células internas.**

#### A.1 Elementos Geradores de Lógica Combinacional:

São utilizados para a implementação de lógica combinacional e podem ser de dois tipos: *matriz AND e OR* e *Look-Up Table (LUT)*.

A **Matriz Programável de ANDs e ORs** é uma estrutura utilizada em *PLDs* mais simples, desde os *PALs*. Esta matriz implementa uma função lógica através da soma de produtos (representação em Mintermos).

O **Look-Up Table (LUT)** é basicamente uma memória pre-programada que fornece uma saída dado um conjunto de variáveis de entrada. Assim, enquanto a matriz de AND e OR produz para cada entrada um valor de saída, o LUT não realiza operação lógica nenhuma, ele apenas consulta a tabela verdade da função que nele foi programada.

## A.2 Estruturas de unidades das células internas:

São estruturas que irão especificar a função seqüencial do circuito após a programação, são diferentes em cada tipo de HCPLD e portanto, identificam o tipo de HCPLD: CPLD ou FPGA.

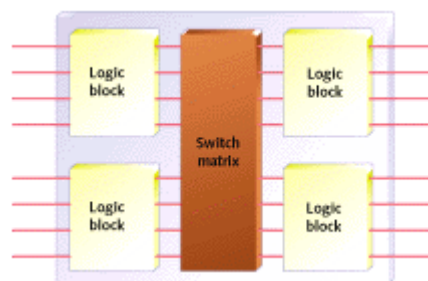
## B. Estruturas de Interconexão:

Para conexão entre as células e entre as células e os blocos de saída, os HCPLDs possuem estruturas conhecidas como barramentos. Para os CPLDs mais complexos e para os FPGA, verificam-se barramentos horizontais e verticais adjacentes a cada célula. Um CPLD é constituído por barramentos contínuos enquanto que um FPGA, por barramentos segmentados.

### 4.2.2 Tipos de HCPLD:

#### 4.2.2.1 Dispositivos Lógicos Programáveis Complexos (CPLDs- Complex Programmable Logic Devices):

CPLD (*Complex Programmable Logic Devices*), Figura 4.8, é um conjunto de múltiplos PLDs em um único *chip* onde cada bloco lógico comunica com o outro através de interconexões programáveis. Esta arquitetura de dispositivo permite tornar mais aproveitável a área de integração (área em silício) permitindo um desempenho melhor e redução de custo. Um único CPLD pode substituir centenas de componentes 74XX (portas lógicas).



**Figura 4.8 Estrutura interna de um CPLD.**



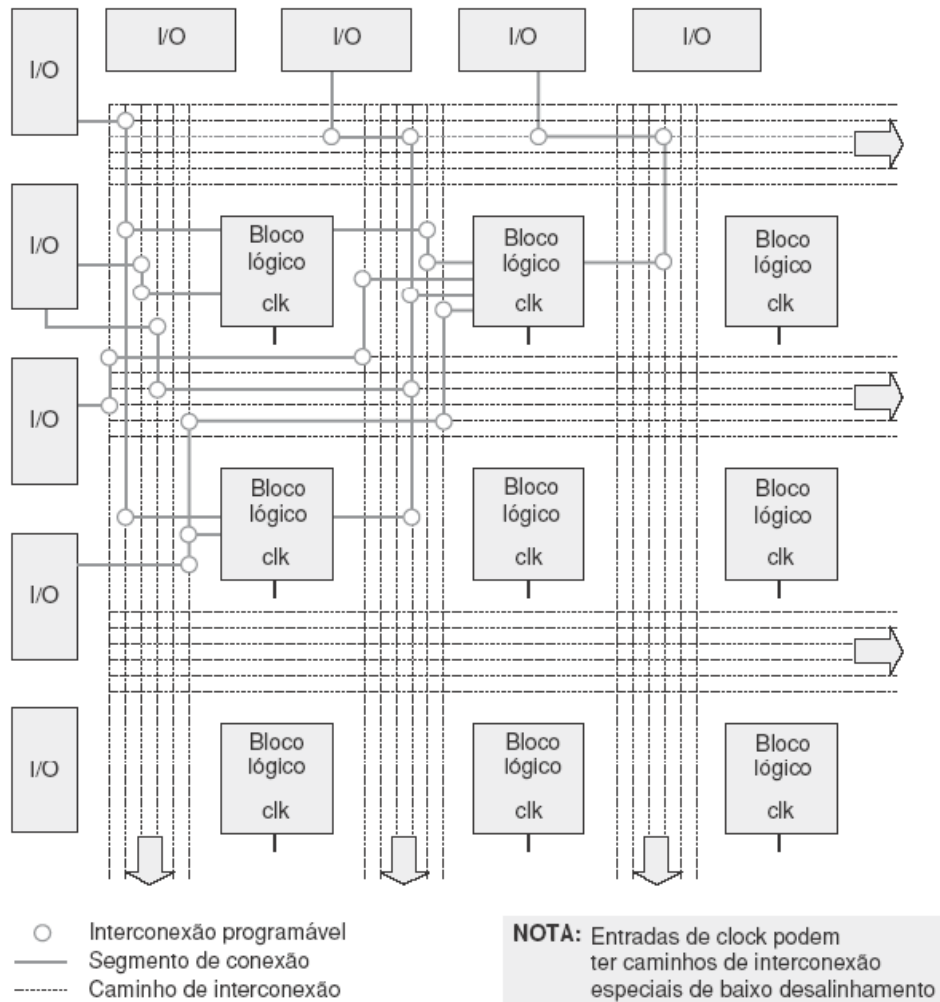
As interconexões programáveis formam uma espécie de barramento que roteia(interliga) sinais das entradas ou saídas para as entradas de um bloco lógico, ou das saídas de um bloco lógico para as entradas do mesmo ou outro bloco. E, cada bloco lógico é equivalente à um SPLD, contendo suas macrocélulas com suas interconexões. Diferente das interconexões configuráveis da SPLD, as interconexões entre os blocos lógicos da CPLD podem não ser totalmente conectáveis, ou seja, algumas conexões entre um bloco e outro, teóricamente pode ser possível, mas na prática não podem ser executadas. Com isso, torna-se difícil utilizar 100% das macrocélulas. A Tabela 1 mostra alguns fabricantes de CPLD.

**Tabela 1.**

Fabricante	Produtos CPLD	URL
Altera	MAX 5000, 7000 & 9000	<a href="http://www.altera.com">www.altera.com</a>
Atmel	ATF & ATV	<a href="http://www.atmel.com">www.atmel.com</a>
Cypress	FLASH370, Ultra37000	<a href="http://www.cypress.com">www.cypress.com</a>
Lattice	ispLSI 1000 to 8000	<a href="http://www.latticesemi.com">www.latticesemi.com</a>
Philips	XPLA	<a href="http://www.philips.com">www.philips.com</a>
Vantis	MACH 1 to 5	<a href="http://www.vantis.com">www.vantis.com</a>
Xilinx	XC9500	<a href="http://www.xilinx.com">www.xilinx.com</a>

#### **4.2.2.2 Arranjos Lógicos Programáveis em Campo (FPGA - Field-Programmable Gate Array):**

FPGA, Figura 4.9 são circuitos programáveis que não possuem planos AND e OR, mas possuem um grande número (da ordem de milhares) de unidades idênticas(blocos lógicos) configuráveis. As unidades lógicas podem ser vistas como componentes padrões que podem ser configurados independentemente e interconectados a partir de uma matriz de trilhas condutoras e chaves programáveis.

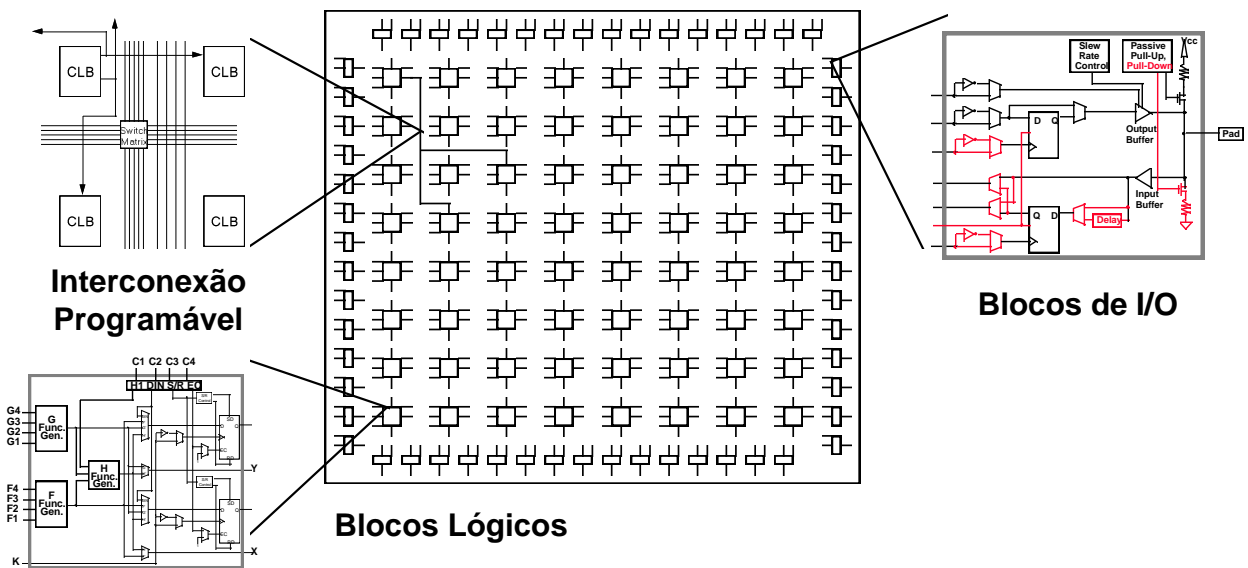


(Figura retirada do site: [http://pessoal.utfpr.edu.br/valfredo/arquivos/01\\_PLDs.pdf](http://pessoal.utfpr.edu.br/valfredo/arquivos/01_PLDs.pdf))

**Figura 4.9 Estrutura de um FPGA**

O FPGA é dividido em duas unidades lógicas configuráveis. Uma delas, a menor, pode ser uma matriz and/or configurável ou pode ser configurada para operar como LUT. A outra unidade, a maior, é um arranjo bidimensional com recursos mais complexos como multiplexadores, blocos de memória, contadores, etc. Essa unidade nada mais é do que um arranjo de elementos de circuitos não conectados, chamados blocos lógicos (mostrados na Figura 4.10), e recursos de interconexão, cuja configuração é obtida através de programação pelo usuário.

Os blocos lógicos, mostrados na Figura 4.9 ou 4.10, são interconectados através de segmentos de trilha e comutadores programáveis. Chama-se de roteamento, a maneira pela qual esses comutadores programáveis e segmentos de trilha são posicionados para permitir a interconexão das células lógicas.



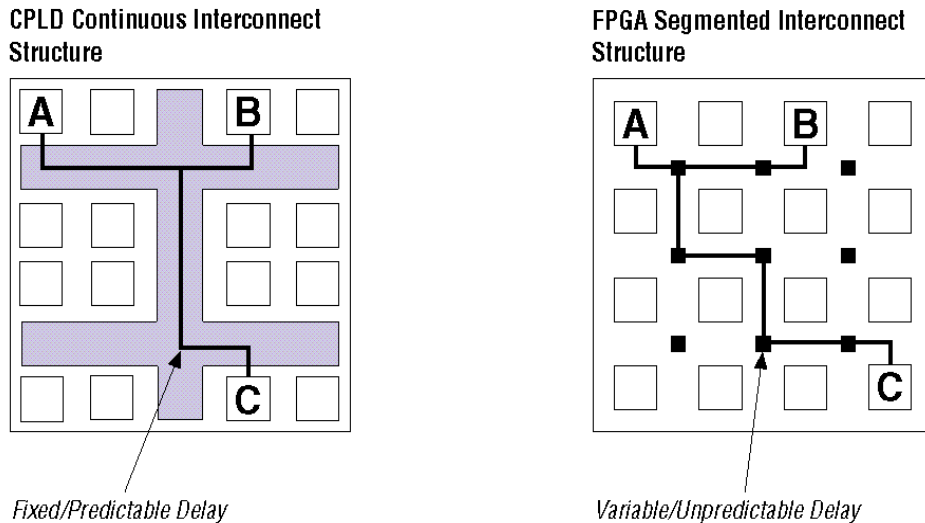
**Figura 4.10 Arquitetura de uma FPGA**

### 4.2.3 Diferenças básicas entre CPLD e FPGA:

Ambos são dispositivos lógicos programáveis e ambos, muitas vezes, são produzidos pelas mesmas companhias. Existem, porém, muitas diferenças entre as tecnologias associadas:

- FPGAs contém muitos pequenos blocos lógicos com flip-flops (até da ordem de 1000000). CPLDs são compostos por um pequeno número (algumas centenas) de grandes blocos lógicos.
- FPGAs, tipicamente, são baseados em RAM, isto significa que perdem sua programação após desligar a alimentação. Portanto, precisam ser reconfigurados (reprogramados) após cada corte de energia. CPLDs são baseados em EEPROM mantendo sua programação após desligar a alimentação;
- CPLDs têm um tempo de resposta melhor, por ser composto de alguns poucos grande blocos lógicos. Em contrapartida esta característica lhe dá menos flexibilidade.
- FPGAs têm recursos de roteamento especiais para implementar de maneira eficiente funções aritméticas e RAM. CPLDs não têm.
- FPGAs podem ser usados em projetos grandes e complexos, enquanto CPLDs estão restritos a projetos bem menores.

A menor granularidade do CPLD, juntamente com sua estrutura de conexão contínua (figura 4.12), facilita sua programação, possibilita melhor desempenho do sistema (*Propagation Delay fixo*) e permite melhor utilização das células lógicas para certas aplicações. A estrutura contínua permite ainda que pequenas modificações na lógica sejam feitas sem degradação no desempenho. Como um FPGA depende fortemente do roteamento, estas mesmas modificações poderiam acarretar numa queda no desempenho devido a necessidade de um novo roteamento. Por outro lado, a granularidade do FPGA dá uma grande flexibilidade e pode ser desejável em certas aplicações. Portanto, a escolha de um ou outro é função direta das necessidades de um projeto. Cabe ao projetista escolher a tecnologia e o dispositivo mais adequado a



implementação em questão.

**Figura 4.12 Comparação entre FPGA e CPLD.**

**Resumo das características dos CPLDs e FPGAs:**

	CPLD	FPGA
Elemento interno de configuração	EPROM /FLASH	SRAM
Interface de configuração	JTAG	JTAG
flexibilidade nas interconexões	BAIXA	MAIS ALTA
Tempos de propagação	UNIFORME	NÃO UNIFORME

## 5. Capacidades Lógicas dos PLDs:

A Figura 5.1 resume as categorias de PLDs com a listagem das capacidades lógicas disponíveis para cada uma das duas categorias disponíveis SPLDs e HCPLDs (CPLDs e FPGAs). Na figura, "Gates equivalentes" se refere ao equivalente a "número de porta NAND de duas entradas". O gráfico serve como um guia para selecionar um dispositivo específico para dada aplicação, de acordo com a capacidade lógica desejada. Entretanto, cada tipo de PLD é inerentemente melhor usado para algumas aplicações do que para outras.

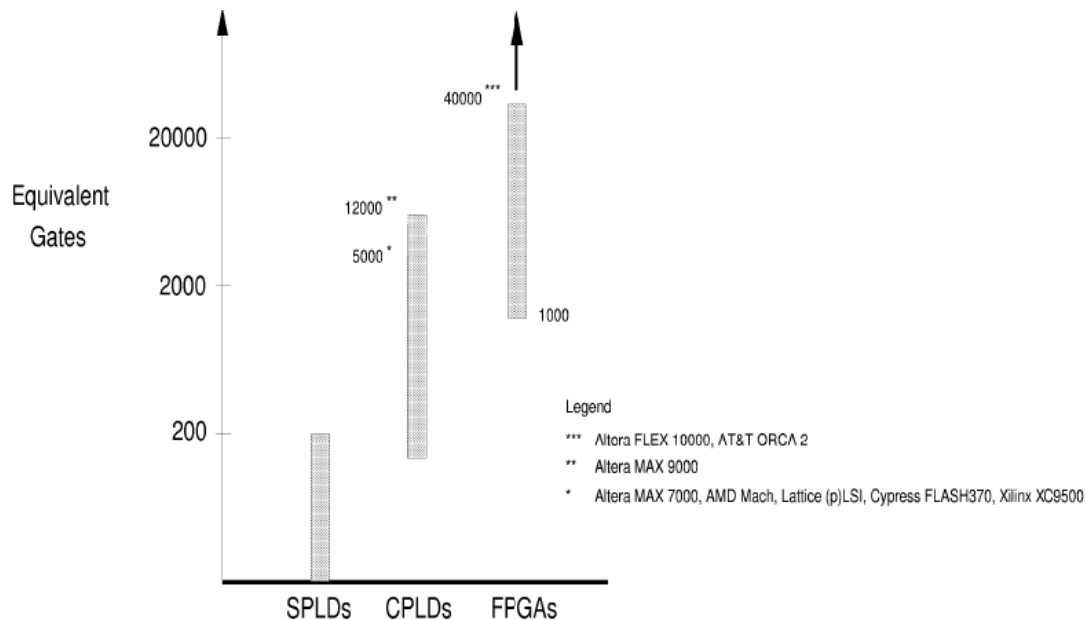


Figura retirada do site: <http://www.demic.fee.unicamp.br/~elnatan/ee610/24a%20Aula.pdf>

**Figura 5.1 Capacidade lógica por tipo de PLD.**

## 6. Tipos de Encapsulamentos:

### 6.1 Encapsulamento dos SPLDs:

- Dual in Line Package (DIP)
- Pin Grid Array (PGA) ou Plastic-Leaded Chip Carrier (PLCC)

### 6.2 Encapsulamento dos HCPLDs:

#### CPLD:

- PLCC :até 100 pinos;
- QFP (Quad Flat Pack): até 200 pinos

#### FPGA:

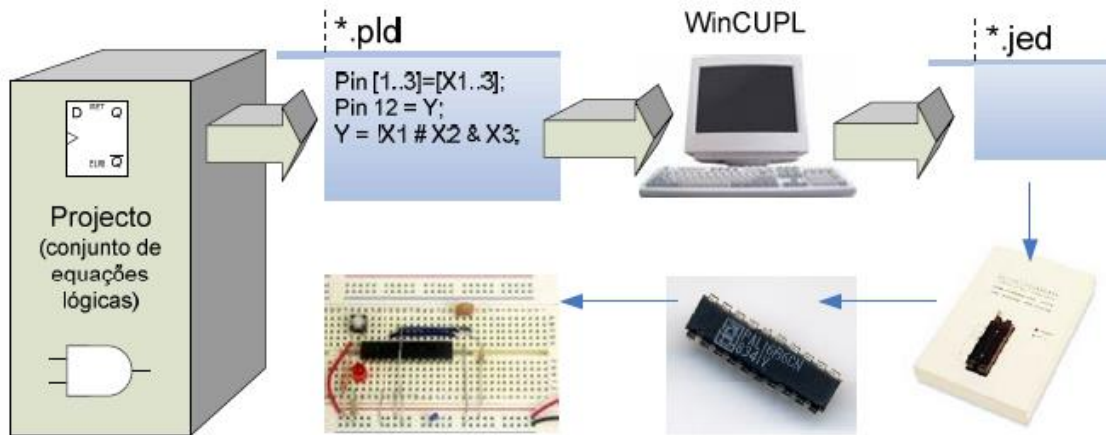
- PGA
- BGA

## 7. Projeto auxiliado por EDA para Programação dos PLDs:

Quando se projeta circuitos para implementar nos PLDs, é essencial empregar programas EDA, os quais são ferramentas (softwares) de projeto para produzir sistemas eletrônicos. Uma visão geral do processo de desenvolvimento de hardware para dispositivos lógicos programáveis (SPDS e HCPLDs) é discutido nos itens a seguir

### 7.1 Projeto e Programação dos SPLDs:

Um sistema de ECAD para SPLDs iria incluir software para as seguintes tarefas: Entrada inicial do projeto, otimização lógica, "fitting" do dispositivo, simulação e configuração. O fluxo de projeto de um SPLD é mostrado na Figura 7.1



(Figura do site: [http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd\\_PLD.pdf](http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd_PLD.pdf))

**Figura 7.1 Fluxo do projeto de um SPLD.**

Para programar um SPLD é necessário inicialmente especificar quais as ligações que devem ser ativadas (para interligar os blocos no seu interior). Essa especificação é realizada num ficheiro de texto com um formato específico que obedece ao padrão designado por JEDEC. Esse ficheiro (\*.jed) é lido pelo programador de circuitos, mostrado na Figura 7.2, que por sua vez gera os sinais eléctricos necessários à programação das ligações entre blocos. Embora seja possível escrever à mão o ficheiro JEDEC é mais fácil e mais confiável utilizar um programa designado por compilador lógico que gera esse ficheiro a partir das equações lógicas que se deseja implementar. Esse programa chama-se WinCUPPL e está instalado nos computadores do laboratório de sistemas Digitais do SEL.



(Figura do site: [http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd\\_PLD.pdf](http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd_PLD.pdf))

**Figura 7.2 Programador de EPROM e SPLD.**

Os passos mostrados na Figura 7.1 estão descritos nos passos de 1 a 4 seguir:

1º Passo: Realizar o projecto lógico (tabelas de verdade, mapas de Karnaugh, diagrama de estados etc.) obtendo as equações lógicas (combinatórias e/ou sequenciais) necessárias. No caso dos circuitos sequenciais deve-se utilizar flip-flops do tipo que o dispositivo disponível possui internamente, por exemplo, no caso da GAL 16v8, o flip-flop é do tipo-D.

2º passo: Utilizando o programa WinCUPL. Escrever um ficheiro de texto (\*.pld) indicando os pinos de entrada e saída e a descrição textual das equações lógicas a implementar. A sintaxe da descrição está explicada no menu de ajuda da aplicação. Uma vez concluída pode-se utilizar o simulador para garantir que a descrição textual está de acordo com a funcionalidade pretendida. Compilar o ficheiro \*.pld de modo a gerar a versão no formato JEDEC, isto é, \*.jed.

3º passo: obtido o arquivo \*.jed, utiliza-se o software (Waccess) do programador (Figura 6.2) para programar o circuito dentro do chip.

4º passo: Testa o circuito na base de montagem.

## 7.2 Projeto e Programação dos HCPLDs:

O fluxo de projeto de circuitos de uma ferramenta EDA para HCPLDs, mostrada na Figura 7.3, pode ser simplificada e dividida em quatro fases:

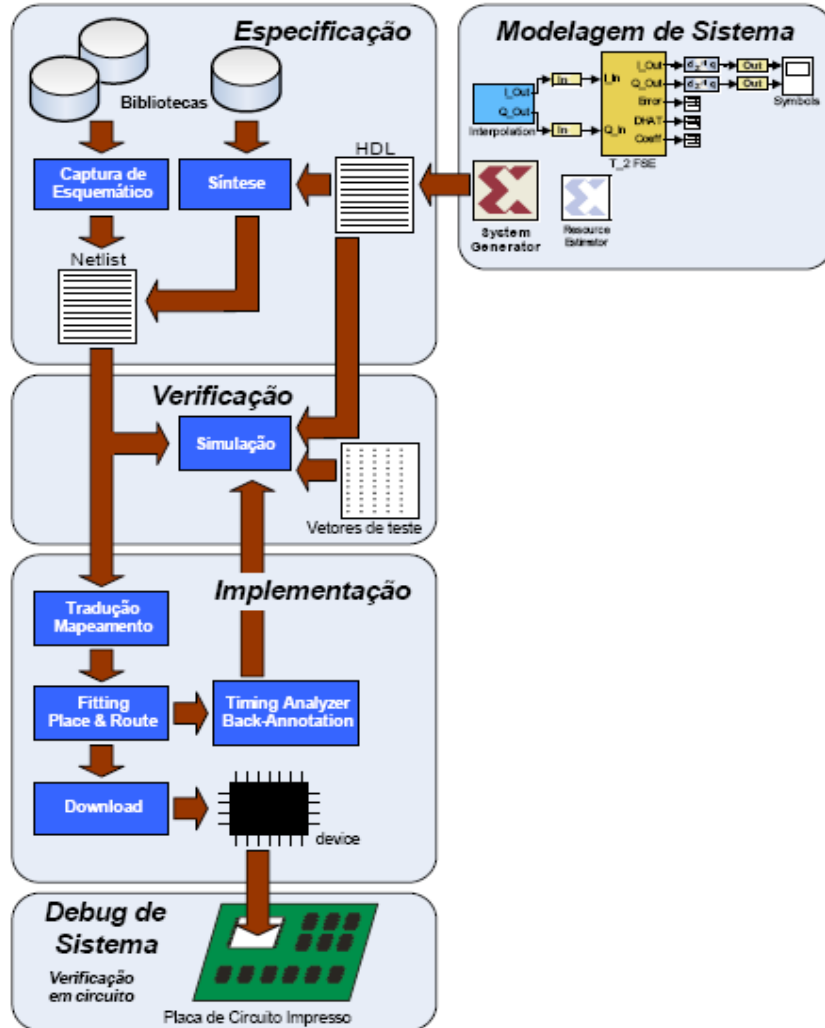
- Especificação
- Verificação
- Implementação
- Debug do sistema

Uma etapa importante do projeto é a geração do netlist, o qual consiste de uma listagem dos componentes do circuito e de como estes componentes estão interconectados, incluindo o nome dos pinos de I/O, e serve para a etapa de verificação e implementação do circuito e é dependente da tecnologia e do fabricante escolhido. A geração do netlist pode ser feita de duas maneiras:

- Captura do esquemático
- Síntese de HDL

A etapa de verificação possibilita acertos na lógica.

A etapa de implementação utiliza o netlist para mapear os blocos lógicos e rotear, interligar os diversos blocos do HCPLD. É obtido então um arquivo em binário que pode ser baixado diretamente no dispositivo, técnica ISP(In System Programming), através de uma interface JTAG(Join Test Action Group: padronizada pelo IEEE), como mostra a Figura 7.4



(Figura do site: <http://www.decom.fee.unicamp.br/~cardoso/ie344b/Aula5%20-FPGA%20e%20Fluxo%20de%20Projeto.pdf>)  
**Figura 7.3 Fluxo de projeto de um HCPLD.**

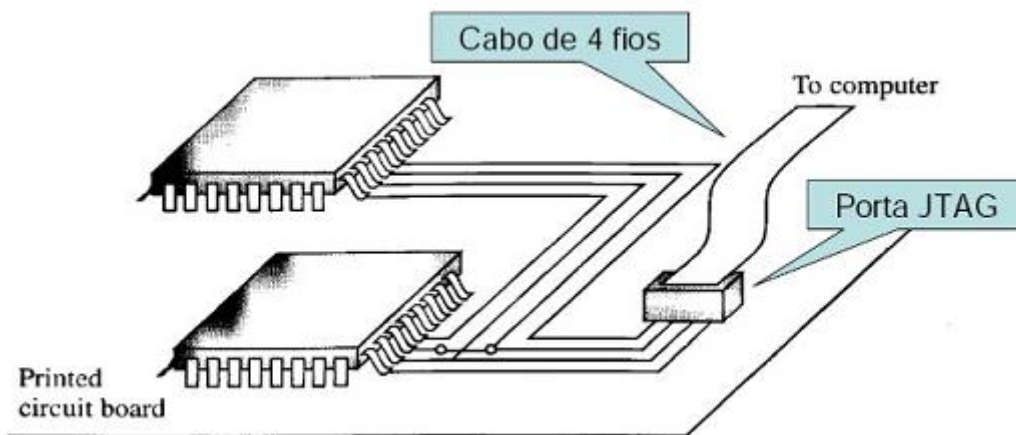


Figura do site: <http://www.dcc.ufrj.br/~gabriel/circlog/DispLogPro.pdf>  
**Figura 7.4 Programação através de JTAG.**



## 8. Escolha do PLD:

Cada tipo de PLD apresenta vantagens que os tomam mais adequados para algumas aplicações do que outros. Um projetista hoje deparar com a difícil tarefa de pesquisar os diferentes tipos de *chips*, entender qual sua melhor utilização, escolher um fabricante específico, aprender a utilizar as ferramentas EDA, para só então começar a projetar o hardware. Em seguida são listados alguns parâmetros que são utilizados na escolha do dispositivo adequado para um projeto específico:

**Número de Portas** : como os fabricantes usam medidas diferentes para a contagem de número de portas (*gates*), é praticamente inútil usar esse parâmetro. Uma melhor comparação a este respeito pode ser feita em relação ao número de *flip-flops* e pinos de I/O.

**Número de Pinos de I/O:** Há suficiente quantidade de pinos de I/O para o seu projeto? Este é frequentemente uma questão feita, e afeta bastante o custo do chip. Conseqüentemente, muitos fabricantes oferecem o mesmo componente com diferentes quantidade de pinos de I/O.

**Custo por Chip:** Obviamente o custo é um fator que precisa ser incluído em um produto final baseado em PLDs . Seria mais barato a longo prazo desenvolver um projeto fixo de ASIC e produzir uma grande quantidade? Ou se escolher o PLD será desejável utilizar o menor componente que possua os recursos mínimos para o projeto em questão.

**Ferramentas Disponíveis:** A ferramenta escolhida precisa ter informações sobre o chip escolhido.

**Desempenho:** Dispositivos mais rápidos (ex: CPLDS) muitas vezes possuem uma flexibilidade menor do que outros com velocidades mais baixas(ex:FPGAs).Desta forma, se a escolha é pela velocidade extra, normalmente perde-se em densidade. Portanto, em grandes projetos é provável que seja necessário utilizar um componente mais lento.

**Consumo de Energia:** O consumo de energia pode ser um fator importante em alguns sistemas. Dispositivos baseados em EEPROM e FLASH tipicamente requerem mais potência que os baseados em PROM, EPROM e SRAM.

**Encapsulamento:** Dispositivos programáveis estão disponíveis em todos os formatos de encapsulamento. A escolha precisa ser direcionada a um deles levando em consideração se é necessário redução de consumo de energia, redução de dissipação, tamanho e/ou custo.

## 9 . Bibliografia:

- Fregni, E. & Saraiva, <sup>a</sup> M., “ Engenharia do Projeto Lógico Digital”, Ed. Edgard Blücher Ltda.
- Tocci, J. R. , “Sistemas Digitais- Princípios e Aplicações”, Ed. Prentice Hall do Brasil
- [www.altera.com](http://www.altera.com)
- [www.xilinx.com](http://www.xilinx.com)
- [http://www.gta.ufrj.br/grad/01\\_1/pld/hcpld.htm](http://www.gta.ufrj.br/grad/01_1/pld/hcpld.htm)
- <http://www.tesequipamentos.com.br/arg/UTP128-84.pdf>
- [http://www.teleco.com.br/tutoriais/tutorialfiltrodig/pagina\\_2.asp](http://www.teleco.com.br/tutoriais/tutorialfiltrodig/pagina_2.asp)
- <http://www.demic.fee.unicamp.br/~elnatan/ee610/24a%20Aula.pdf>
- [http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd\\_PLD.pdf](http://ac.dcti.iscte.pt/AC-2008-09/Material/Tutoriais/Introd_PLD.pdf)
- <http://www.dcc.ufrj.br/~gabriel/circlog/DispLogPro.pdf>
- <http://www2.eletronica.org/artigos/eletronica-digital/dispositivos-programaveis-como-funcionam>