

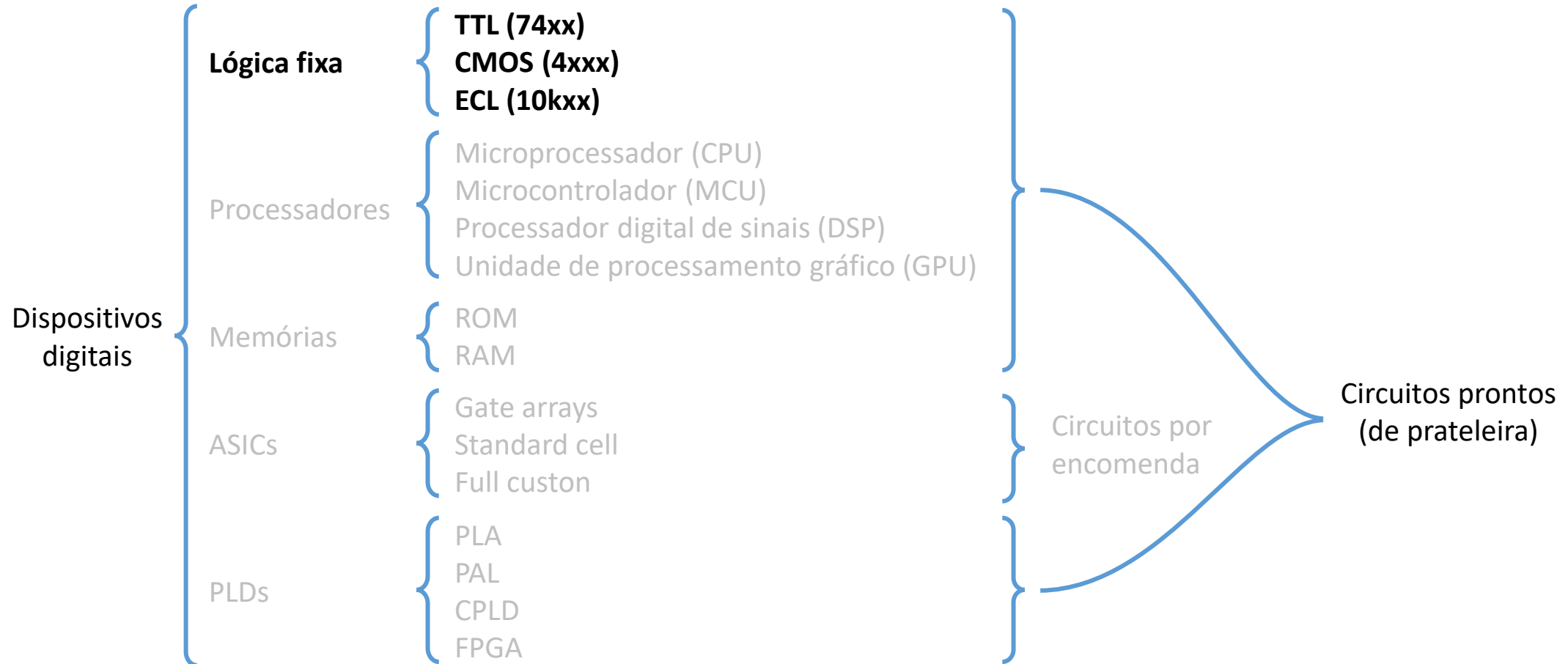
Sistemas Reconfiguráveis

Eng. de Computação

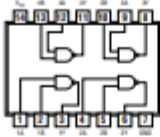
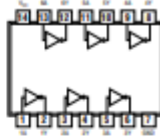
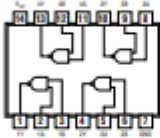
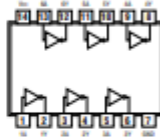
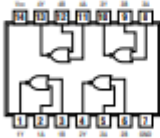
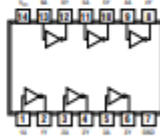
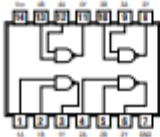
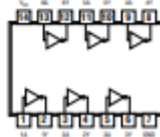
Profs. Francisco Garcia e Antônio Hamilton Magalhães

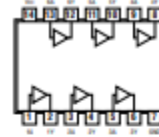
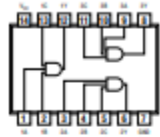
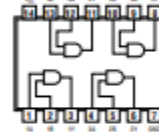
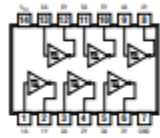
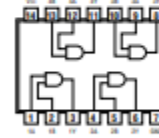


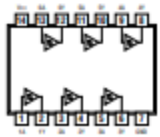
Aula 1 - Circuitos digitais

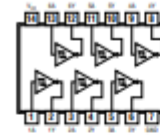
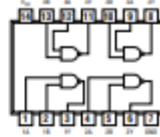
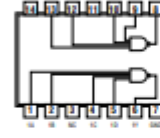
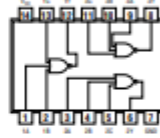
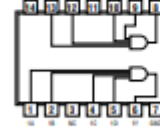
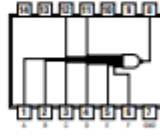
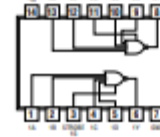

Circuitos integrados digitais



CIs digitais de lógica fixa

00 QUADRUPLE 3-INPUT POSITIVE-NAND GATES positive logic: $Y = ABC$  <p>See page 221</p>	04 HEX INVERTERS positive logic: $Y = A$  <p>See page 228</p>
01 QUADRUPLE 3-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = ABC$  <p>See page 222</p>	04H HEX INVERTERS positive logic: $Y = A$  <p>See page 228</p>
02 QUADRUPLE 3-INPUT POSITIVE-NOR GATES positive logic: $Y = A + B + C$  <p>See page 223</p>	05 HEX INVERTERS WITH OPEN-DRAIN OUTPUTS positive logic: $Y = A$  <p>See page 228</p>
03 QUADRUPLE 3-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = ABC$  <p>See page 224</p>	06 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-DRAIN OUTPUTS positive logic: $Y = A$  <p>See page 227</p>

07 HEX BUFFERS/DRIVERS WITH OPEN-DRAIN OUTPUTS positive logic: $Y = A$  <p>See page 227</p>	11 TRIPLE 3-INPUT POSITIVE-AND GATES positive logic: $Y = ABC + C$  <p>See page 221</p>
08 QUADRUPLE 3-INPUT POSITIVE-AND GATES positive logic: $Y = ABC$  <p>See page 228</p>	14 HEX SCHMITT-TRIGGER INVERTERS positive logic: $Y = A$  <p>See page 242</p>
09 QUADRUPLE 3-INPUT POSITIVE-AND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = ABC$  <p>See page 228</p>	16 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = A$  <p>See page 243</p>
10 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-DRAIN OUTPUTS positive logic: $Y = A$  <p>See page 242</p>	17 HEX SCHMITT-TRIGGER BUFFER positive logic: $Y = A$  <p>See page 243</p>

19 HEX SCHMITT-TRIGGER INVERTERS positive logic: $Y = A$  <p>See page 244</p>	26 QUADRUPLE 3-INPUT HIGH-VOLTAGE INTERFACE POSITIVE-NAND GATES positive logic: $Y = ABC$  <p>See page 247</p>
20 DUAL 4-INPUT POSITIVE-NAND GATES positive logic: $Y = ABCD$  <p>See page 248</p>	27 TRIPLE 3-INPUT POSITIVE-NOR GATES positive logic: $Y = A + B + C$  <p>See page 247</p>
21 DUAL 4-INPUT POSITIVE-AND GATES positive logic: $Y = ABCD$  <p>See page 248</p>	30 8-INPUT POSITIVE-NAND GATES positive logic: $Y = ABCDEFGH$  <p>See page 248</p>
25 DUAL 4-INPUT POSITIVE-NOR GATES WITH STROBE positive logic: $Y = G(A + B + C + D)$  <p>See page 248</p>	31 DELAY ELEMENTS  <p>See page 248</p>

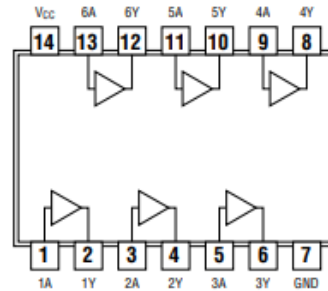
Cis digitais de lógica fixa

07

HEX BUFFERS/DRIVERS WITH OPEN-DRAIN OUTPUTS

positive logic:

$$Y = \bar{A}$$



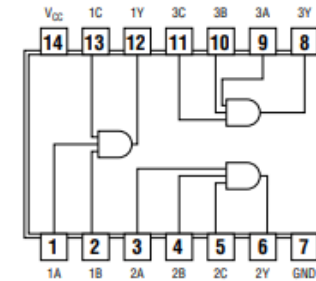
See page 237

11

**TRIPLE 3-INPUT
POSITIVE-AND GATES**

positive logic:

$$Y = A \cdot B \cdot C$$



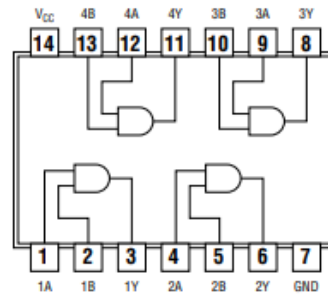
See page 241

08

QUADRUPLE 2-INPUT POSITIVE-AND GATES

positive logic:

$$Y = A \cdot B$$



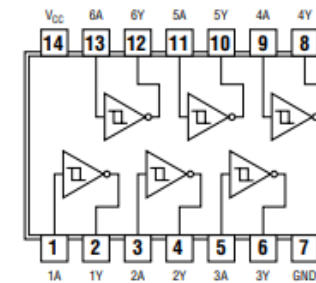
See page 238

14

**HEX SCHMITT-TRIGGER
INVERTERS**

positive logic:

$$Y = \bar{A}$$



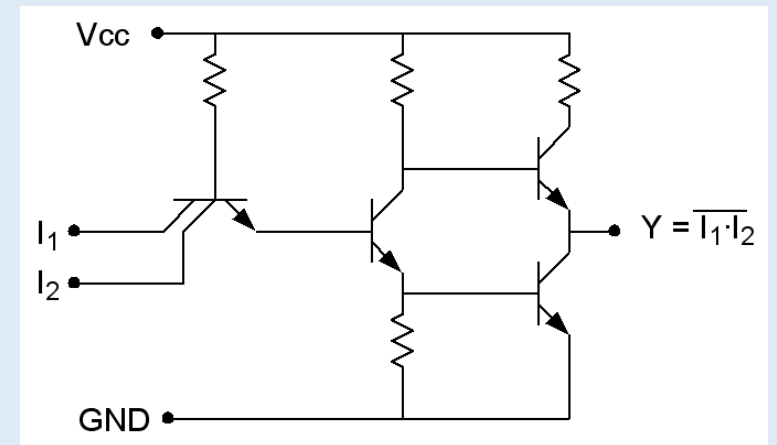
See page 242

Cis digitais de lógica fixa

- Família TTL (Transistor-Transistor Logic)

- Introduzida em 1963
- Tensão de trabalho 5V
- Consumo típico: 10 mW por porta
- Sub-tipos (evolução):
 - LS (Low power Schottky)
 - S (Schottky)
 - H (High speed)
 - F (Fast)
 - AS (Advanced Schottky)
 - ALS (Advance low-power Schottky)

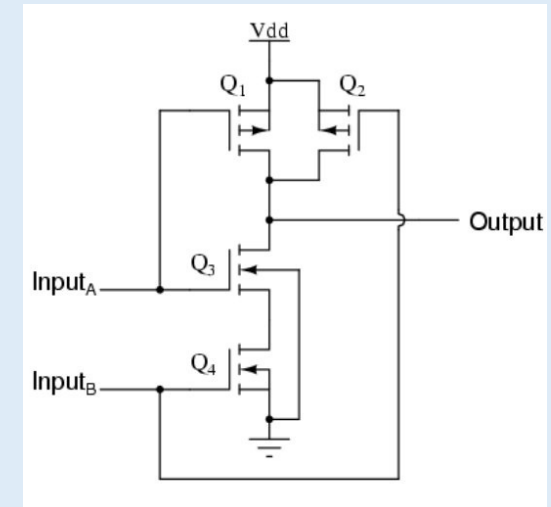
Porta NAND de duas entradas



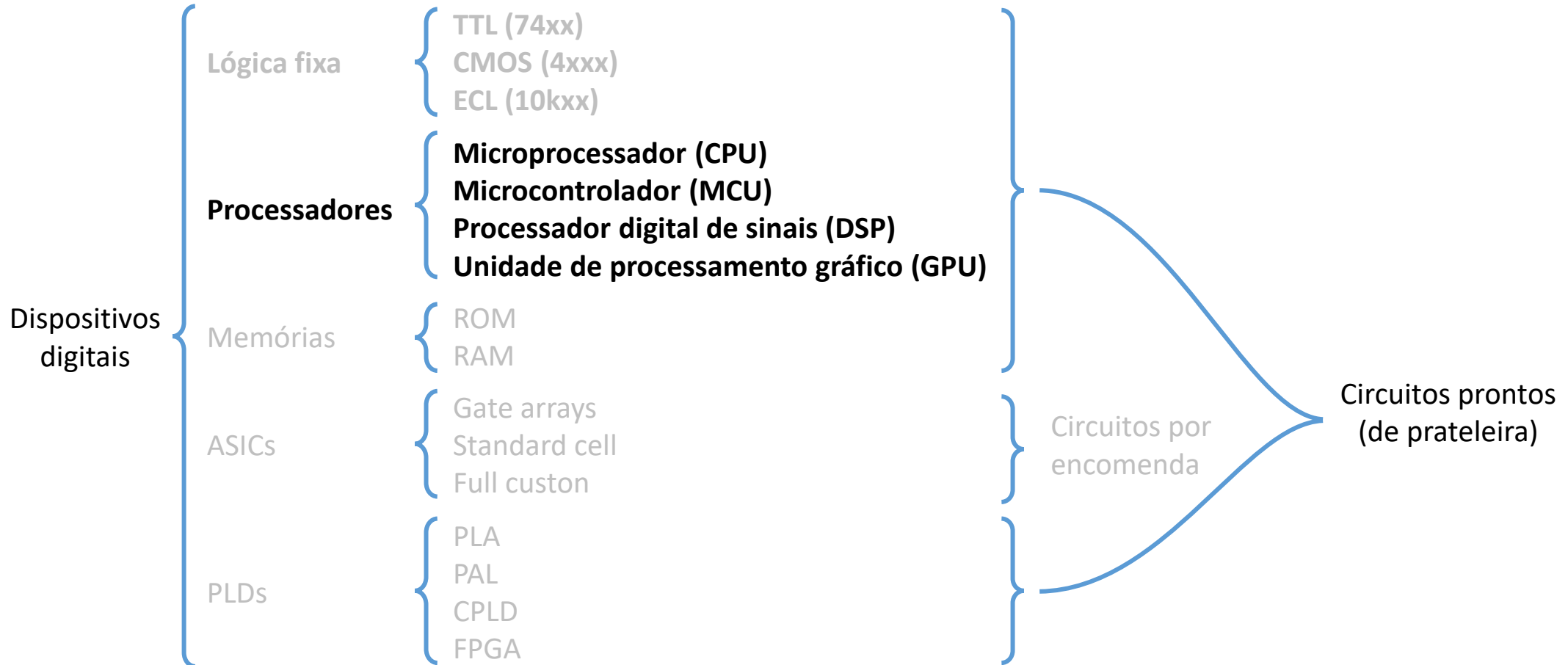
Cis digitais de lógica fixa

- Família CMOS (Complementary metal–oxide–semiconductor)
 - Introduzida em 1968 pela RCA e, posteriormente, fabricada por quase todas as companhias de semicondutores ativas na época
 - Tensão de trabalho de 5 a 15V
 - Consumo estático muito baixo (menos de 1uW por porta)
 - Sub-tipos (evolução):
 - HCMOS (High speed)

Porta NAND de duas entradas



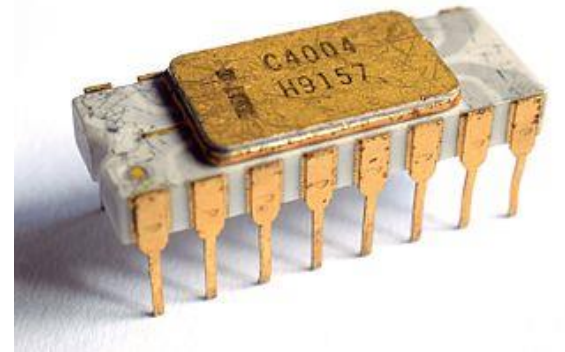
Circuitos integrados digitais



Processadores

Um pouco de história

- Em 1968, Robert Noyce e Gordon Moore fundaram a NM Electronics, que mais tarde se tornaria Intel Corporation.
- Em 1971, a Intel lançou o primeiro microprocessador na forma de um circuito integrado, o 4004, por solicitação da empresa Busicom Corp. do Japão, para ser usado em uma calculadora. Esse CI trabalhava com registradores de 4 bits, 46 instruções, clock de 740kHz e possuía cerca de 2300 transistores.
- Em 1973, a Intel lançou o 8008, o primeiro microprocessador de 8 bits e que era quase duas vezes mais complexo que o Intel 4004.



Intel 4004

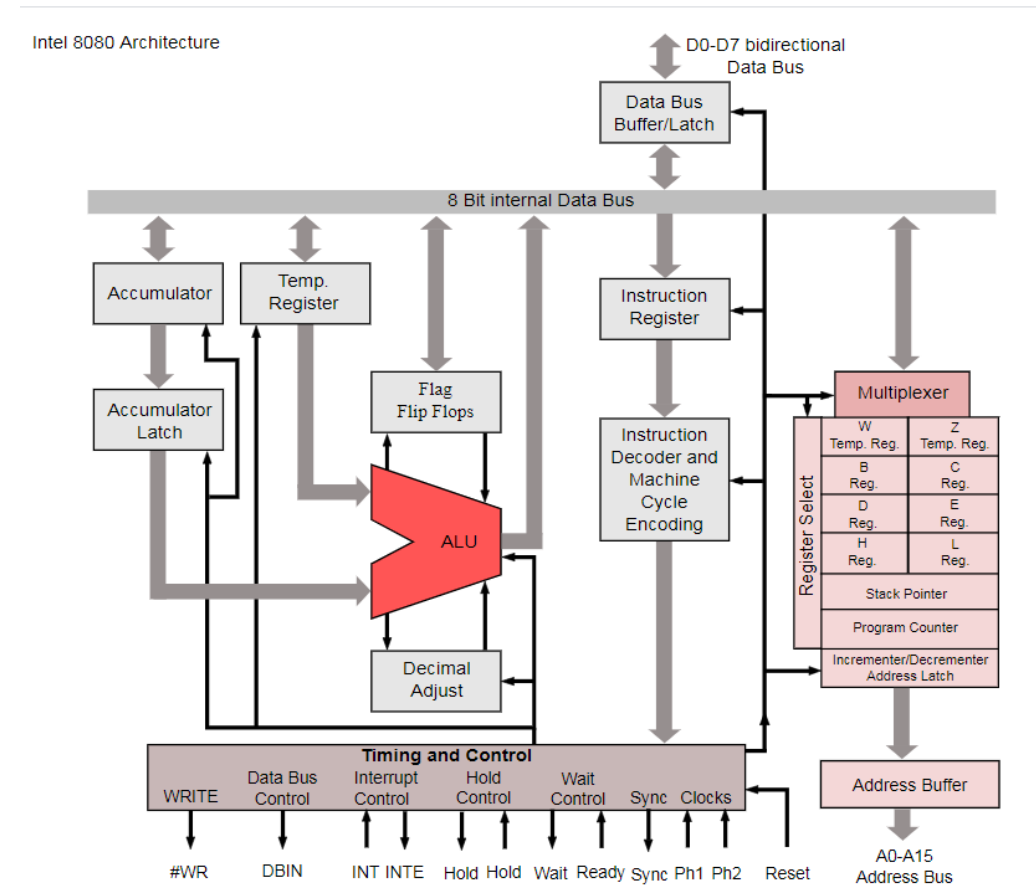


Intel 8008

Processadores

Um pouco de história

- Em **1974**, a **Intel** lançou o **8080**, uma evolução do 8008, que podia trabalhar com frequência de *clock* de até **2MHz**. A alimentação desse processador era feita por três fontes, **+5V**, **+12V** e **-12V**. Usava cerca de **6000 transistores P-MOS**.
- Em **1976** Intel lançou o **8085**, compatível com o 8080, mas com duas instruções adicionais. Tinha como vantagem o fato de necessitar de apenas uma fonte de alimentação de **+5V**, por causa do uso de transistores **N-MOS**.



Arquitetura do Intel 8080

Processadores

Um pouco de história

- O **Z80**, baseado no 8080, mas bem melhor do que ele, foi lançado em **1976 pela Zilog**, uma empresa criada na California em 1974 por **Federico Faggin e Ralph Ungermann**, ex funcionários da Intel.
- Seguindo o Intel 8080, a partir de 1974 várias empresas lançaram processadores de **8 bits**, tais como a **Motorola** com o **6800** e a **MOS Technology com o 6502**. Esse último era bem mais barato do que os seus concorrentes da época e forçou uma queda nos preços de toda a indústria.
- Na década de **1970**, vários **computadores pessoais**, (TRS-80, Commodore, Sinclair, Apple II) e **console para jogos** (Atari, Nintendo) foram desenvolvidos utilizando processadores de **8 bits**.



TRS-90 Model 4p

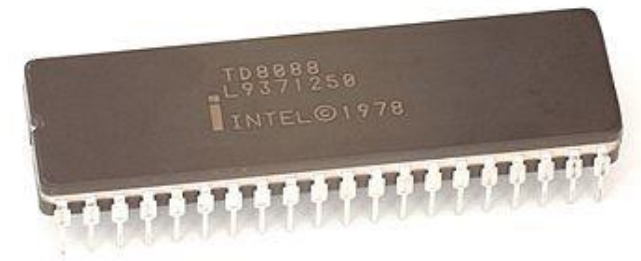


Atari 2600

Processadores

Um pouco de história

- A **Advanced Micro Devices (AMD)**, fundada em **1969** por **Jerry Sanders** e sete amigos, desenvolveu, em 1975, uma versão de **engenharia reversa** do processador padrão **8080**.
- Em **1976** a **Intel** e a **AMD** assinaram um **acordo de troca de licenças**, em que concordam em licenciar para a outra parte todas as patentes que cada uma das empresas detinham.
- A Intel lançou, em **1978**, o **8086** e, em **1979**, o **8088**, processadores de **16 bits**, sendo que o 8088 tem o barramento de dados externo de 8 bits.
- Nessa mesma época outros processadores de **16/32 bits** estavam sendo desenvolvidos, como o **68000 da Motorola**, o **Z8000 da Zilog** e o **32016 da National Semiconductors**.



Processadores

Um pouco de história

- Em 1982, a IBM escolheu o microprocessador Intel 8088 para seu PC, mas somente com a condição de que houvesse um fornecedor alternativo confiável para atender às suas necessidades. O resultado disso é que a AMD renovou seu acordo abrangente de troca de licenças com a Intel e se tornou fabricante alternativo dos microprocessadores 8086 e 8088 para a IBM.
- A partir do 8086/8088, a Intel desenvolveu outros microprocessadores, como o 80286, em 1982 (que introduziu na linha Intel o modo protegido), o 80386 (1985) e o 80486, de 32 bits (1989).
- Para evitar a concorrência da AMD, em 1987 a Intel notificou a AMD sobre o fim do acordo de licenças. Isso iniciou uma batalha judicial entre as empresas.

Processadores

Um pouco de história

- Na década de **1990** a **Intel** lançou o **Pentium** e a **AMD** desenvolveu o **K5** sem, no entanto, conseguir tirar a supremacia da Intel.
- Na década de **2000**, a AMD foi pioneira no lançamento de um microprocessador de **64 bits** para computadores de mesa, o **Athlon 64**.
- Com a dificuldade de se projetar dispositivos com desempenho cada vez maior, a solução adotada pelas indústrias foi desenvolver microprocessadores de **vários núcleos**, tais como a série Core **i3**, **i5**, **i7** e **i9** da Intel ou a série **Ryzen**, da AMD.

Processadores



Os processadores são dispositivos complexos (desde milhares até bilhões de transistores), que executam uma **lista de instruções** (*software*) para implementar a funcionalidade desejada. Seu projeto é muito **caro** (milhões de dólares são investidos no projeto dos processadores mais avançados) e **demorado** (anos). Os processadores se dividem em:

- **Microprocessador (μ P)**
- **Microcontrolador (μ C)**
- **Processador digital de sinais (DSP)**
- **Unidade de processamento gráfico (GPU)**

Microprocessadores (µP)



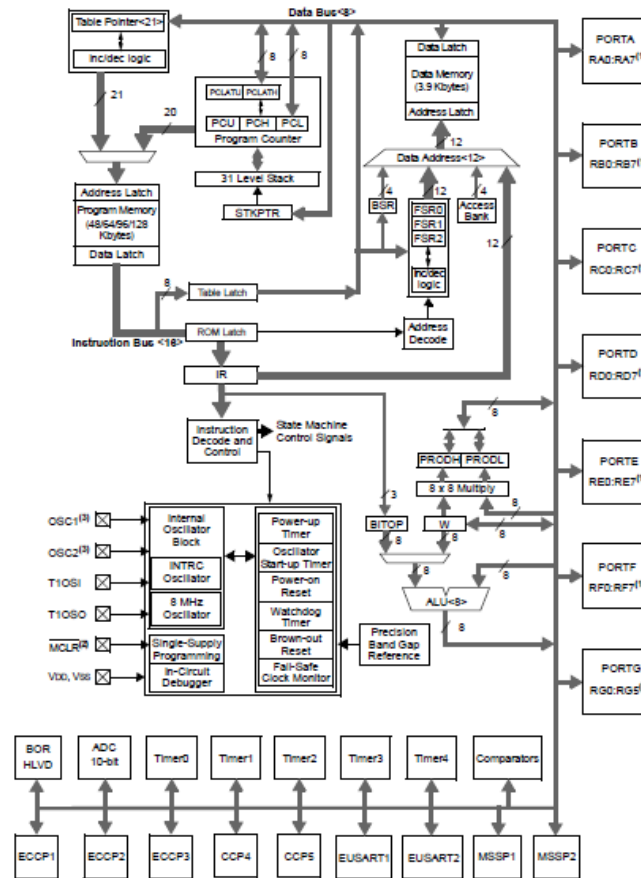
- O microprocessador é um circuito integrado que é programável pelo **uso de um programa**.
- Um microprocessador é um processador de **propósito geral** que é projetado para empreender uma **grande variedade de tarefas**. Sua arquitetura pode **não** ser, necessariamente, **otimizada** para uma **tarefa específica**.
- **Circuitos externos** tais como memória e interface de entrada e saída devem ser adicionados para se obter a funcionalidade do µP.
- Quanto ao conjunto de instruções, existem dois tipos: **CISC** (*complex instruction set computer*) e **RISC** (*reduced instruction set computer*).
- Um µP muitas vezes é chamado de **CPU** (*central processing unit*).

Microcontroladores (uC)



- É um tipo de microprocessador que reúne, em um **mesmo circuito** integrado, **CPU, memória, interface de entrada e saída** e controlador de comunicação. Alguns modelos possuem também conversores **A/D** (analógico para digital) e **D/A** (digital para analógico).
- O objetivo, nesse caso, **não é ser tão flexível** ou tão poderoso quanto um microprocessador de propósito geral, mas ser um computador em uma só pastilha (***computer-on-a-chip***).
- É usado em **aplicações embutidas**, onde o **custo**, o **tamanho** e o consumo são considerações importantes.
- É também chamado de **MCU** (*microcontroller unit*).

Microcontroladores (uC)



Processadores digitais de sinais - DSP



- É um tipo **especializado de microprocessador**, que visa aplicações de **processamento digital de sinais em tempo real**, tais como **filtragem** digital ou transformada rápida de Fourier (**FFT**). Essas aplicações requerem que um grande número de multiplicações e adições seja realizado de forma rápida.
- Um processador de propósito geral, executando essas operações em *software*, pode não conseguir a velocidade necessária. A arquitetura de um **DSP é otimizada de forma a obter a máxima performance para esse tipo de operação**, empregando **células especializadas** em realizar, no *hardware*, multiplicações e adições.

Processadores digitais de sinais - DSP



Algumas aplicações de DSP:

- **Filtragem digital**

- Resposta a impulsos finitos (FIR)
- Resposta a impulsos infinitos (IIR)
- Filtros casados (correlatores)
- Transformadas de Hilbert
- Filtros adaptativos

- **Processamento de sinais**

- Compressão e reconhecimento de voz
- Expansão
- Média
- Cálculos de energia

- **Processamento de dados:**

- Criptografia e embaralhamento (*scrambling*)
- Codificação (codificação Trellis)
- Decodificação (decodificação Viterbi)

- **Processamento numérico**

- Escalar, vetorial, matriz aritmética e computação com funções transcendentais
- Funções não lineares
- Geração de números pseudo-aleatórios

- **Análise espectral**

- Transformada rápida de Fourier (FFT)
- Transformada discreta de Fourier (DFT)
- Transformadas de seno/cosseno

Processadores digitais de sinais - DSP



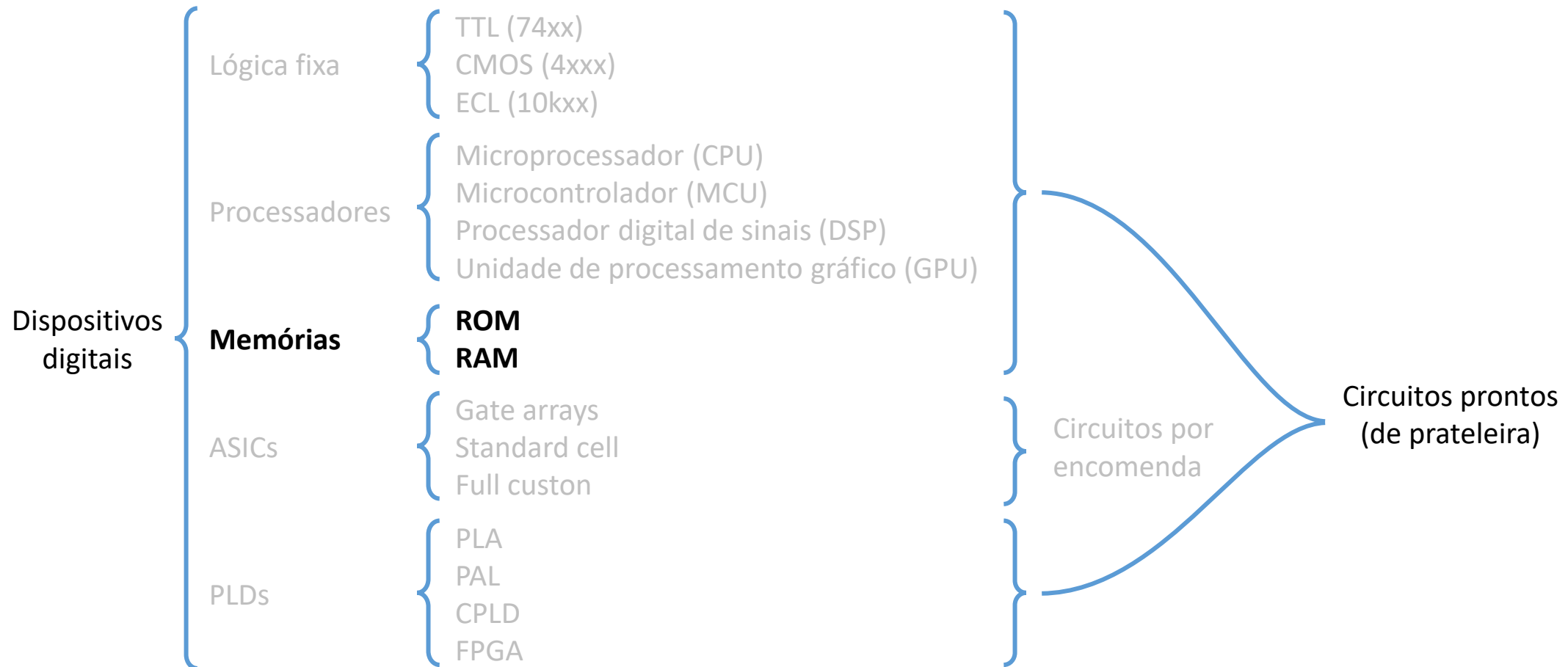
Algumas áreas de uso para DSP:

- Telecomunicações
 - Comunicação de dados:
 - Radiocomunicações:
 - Computadores:
 - Processamento de imagem
 - Instrumentação
- Processamento de som
 - Controle
 - Medicina
 - Vídeo digital
 - Radar e Sonar

Unidade de processamento gráfico - GPU

- Tipo de microprocessador **especializado em processar gráficos** em computadores pessoais, estações de trabalho ou videogames
- As GPUs trabalham com um **grande número de unidades de cálculos** em uma arquitetura **massivamente paralela**.
- Recentemente **placas de vídeo** incorporaram tecnologias para programação de propósito geral, como **OpenCL e CUDA**, servindo como um co-processador massivamente paralelo para **cálculos intensivos**, sendo essa técnica conhecida como **GPGPU** (*general purpose graphic processing unit*).

Circuitos integrados digitais



Memórias



Memórias são dispositivos usados para **armazenar dados ou códigos de programas em um circuito eletrônico** ou sistema baseado em processador. Os dois tipos básicos são:

- ROM (*read-only memory*)
 - Mask ROM
 - PROM
 - UV-EPROM
 - EEPROM ou E²PROM
- RAM (*random access memory*).
 - Estática
 - Dinâmica

Memórias ROM

ROM (*read-only memory*)

As ROMs são usadas para armazenar **programas** ou dados de forma não volátil (mantém a informação mesmo na ausência de alimentação).

- **Mask ROM**
 - Os dados são colocados na memória no **processo de fabricação, através das máscaras**.
- **PROM** (programmable ROM)
 - Os dados são colocados na memória **pelo usuário, uma única vez**.
- **EPROM** (erasable PROM)
 - Os dados são colocados na memória **pelo usuário**, podendo a memória ser **apagada e regravada**
 - **UV-EPROM** (ultraviolet erasable PROM): A memória é apagada através de luz **ultra-violeta**
 - **EEPROM** ou E²PROM (electrically erasable PROM): A memória é apagada **eletricamente**



UV-EPROM

Memórias RAM

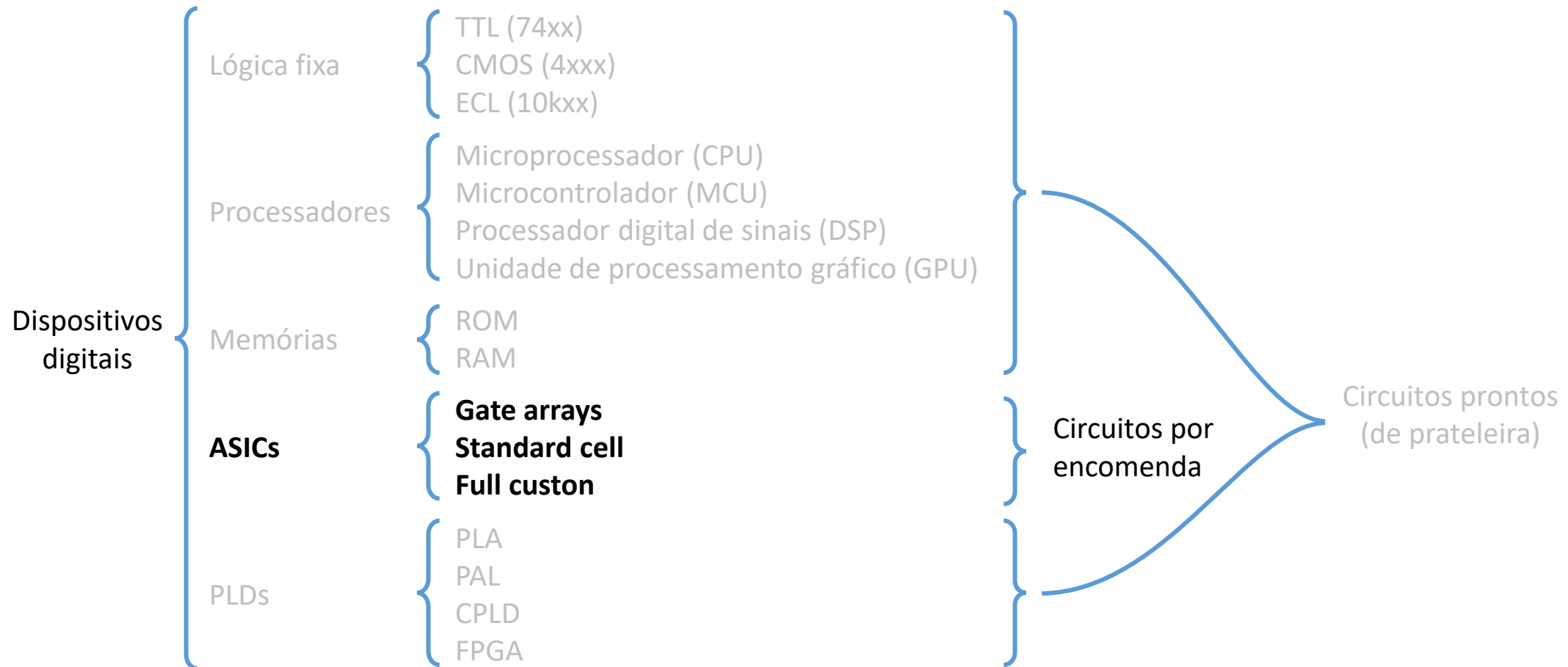


RAM (*random access memory*).

RAMs são usadas para manter dados que requerem acesso rápido e habilidade de modificar o seu conteúdo durante o operação normal.

- Estática: **mantém os dados mesmo quando não há atividade na memória**. As células de memória utilizam vários transistores.
 - Menor densidade
 - Muito rápidas
 - Baixo consumo
- Dinâmica: Os dados são armazenados na **forma de uma carga elétrica**. Perdem os dados se não estiverem sendo frequentemente usadas (**refresh**).
 - Células muito simples => maior densidade
 - Tempo de acesso para leitura e escrita maior do que na RAM estática
 - Consumo elevado
 - Podem ter um funcionamento síncrono, chamadas então de SDRAM (synchronous DRAM)

Circuitos integrados digitais



ASICs



Um circuito integrado de aplicação específica, em inglês *application-specific integrated circuit* (ASIC), é um dispositivo feito **sob encomenda** por um fabricante de semicondutores para um determinado cliente, visando um uso particular.

Foram introduzidos em **1967** pelas empresas **Ferranti e Interdesign**, utilizando tecnologia de transistores **bipolares**, enquanto que o primeiro circuito desse tipo feito com tecnologia **CMOS** foi desenvolvido por Robert Lipp, em **1974** para a **International Microcircuits, Inc. (IMI)**.

Os ASICs se dividem em:

- *Gate array*
- *Standard cell*
- *Full custom*

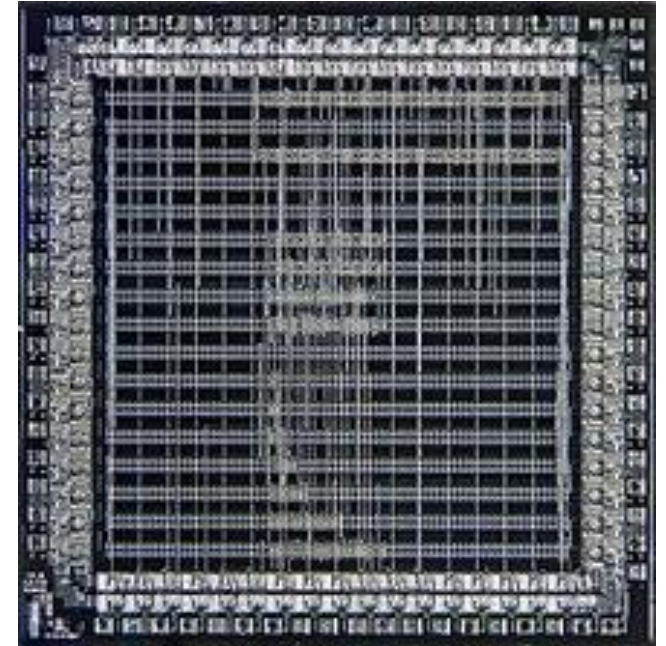
ASIC – *Gate array*

Uma **etapa inicial** de fabricação cria um **número grande** (de milhares a centenas de milhares) de **portas básicas não conectadas**.

Wafers com circuitos inacabados são usualmente **pré-fabricados e estocados em grandes quantidades**, independentemente dos pedidos dos clientes.

Em uma **etapa final**, essas **portas são conectadas** de acordo com o projeto do cliente, através da **camada de interconexão metálica**.

Essa abordagem **reduz os custos não recorrentes** de engenharia, pois **poucas máscaras personalizadas** são produzidas.



Fotografia de microscópio de um ASIC *gate array* mostrando as células lógicas predefinedas e interconexões personalizadas. Este projeto específico usa menos de 20% das portas lógicas disponíveis.

ASIC – *Standard cell*



No tipo *standard cell* não há o conceito de células básicas e nenhum componente é pré-fabricado na pastilha de silício. O fabricante **cria máscaras por encomenda a partir de bibliotecas de funções lógicas.**

Uma biblioteca de células padrão é uma coleção de **funções lógicas eletrônicas de baixo nível, como AND, OR, inversores, flip-flops, latches e buffers.** Essas células são realizadas como células totalmente personalizadas de **altura fixa e largura variável.** O principal aspecto dessas bibliotecas é que elas têm uma altura fixa, o que permite que sejam **colocadas em fileiras,** facilitando o **processo de layout** digital automatizado.

ASIC – *Standard cell*



A biblioteca geralmente contém **várias implementações da mesma função lógica, diferindo em área e velocidade**. Essa variedade aumenta a **eficiência das ferramentas automatizadas de síntese**, localização e roteamento (SPR). Indiretamente, também dá ao projetista maior liberdade para lidar com os compromissos de implementação (**área vs. velocidade vs. consumo de energia**).

Para um projeto específico, nesse tipo de **ASIC todas as máscaras de fabricação são personalizadas**, o que faz com que os custos iniciais de produção sejam maiores do que o tipo *gate array*.

ASIC – *Full custom*



O **projeto totalmente personalizado** é uma metodologia para projetar circuitos integrados, especificando o **layout de cada transistor** individual e as interconexões entre eles.

Maximiza potencialmente o desempenho do chip e minimiza sua área, mas é **extremamente trabalhoso** para implementar. O projeto totalmente personalizado é limitado a CIs que devem ser fabricados em **volumes extremamente altos**.

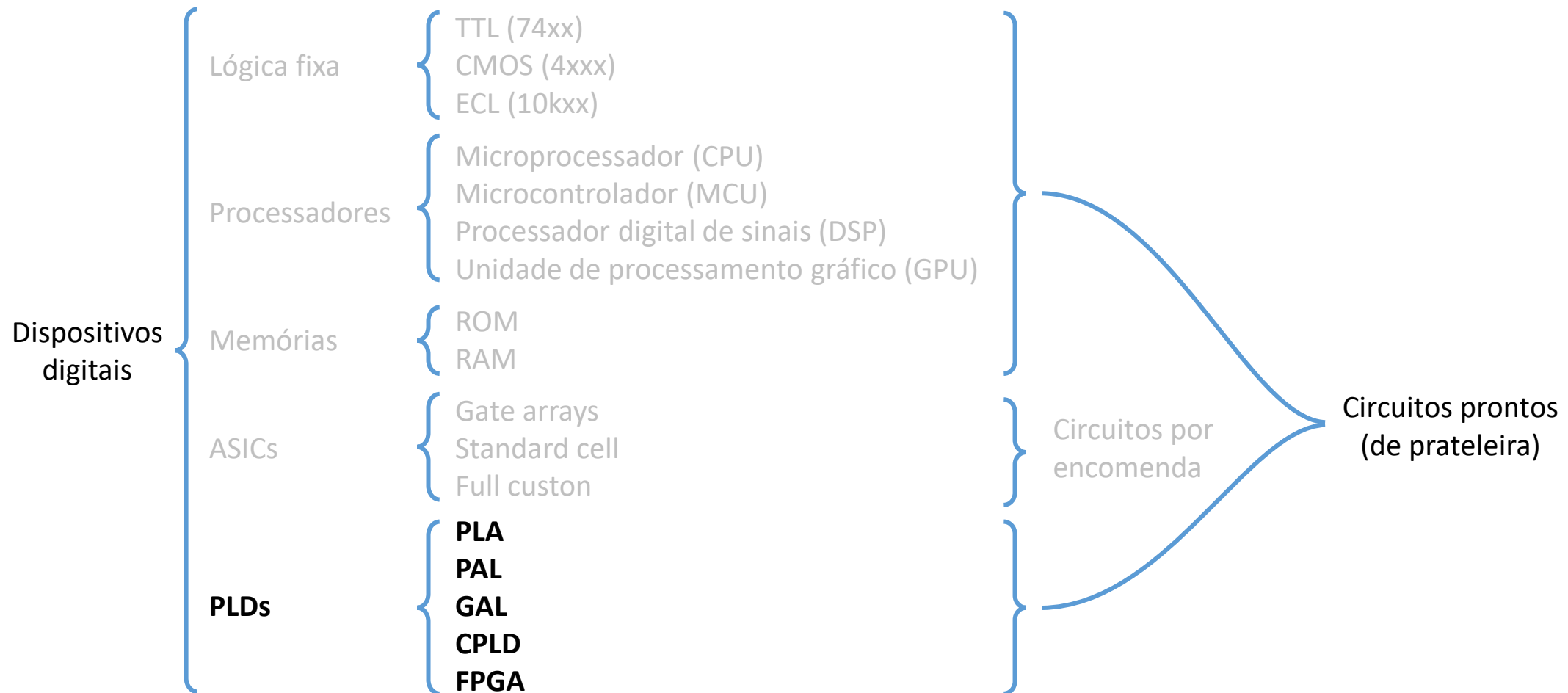
ASICs



O principal fator que afeta o projeto e a produção de ASICs é o **alto custo dos conjuntos de máscaras e das ferramentas de projeto** (*electronic design automation* – EDA). Os conjuntos de máscaras são necessários para transferir os projetos do ASIC para o *wafer*.

ASICs tem um **custo inicial de projeto alto e um custo de produção baixo**. São recomendadas para produtos com um **ciclo de vida longo** e **grandes volumes de produção**.

Circuitos integrados digitais



PLDs



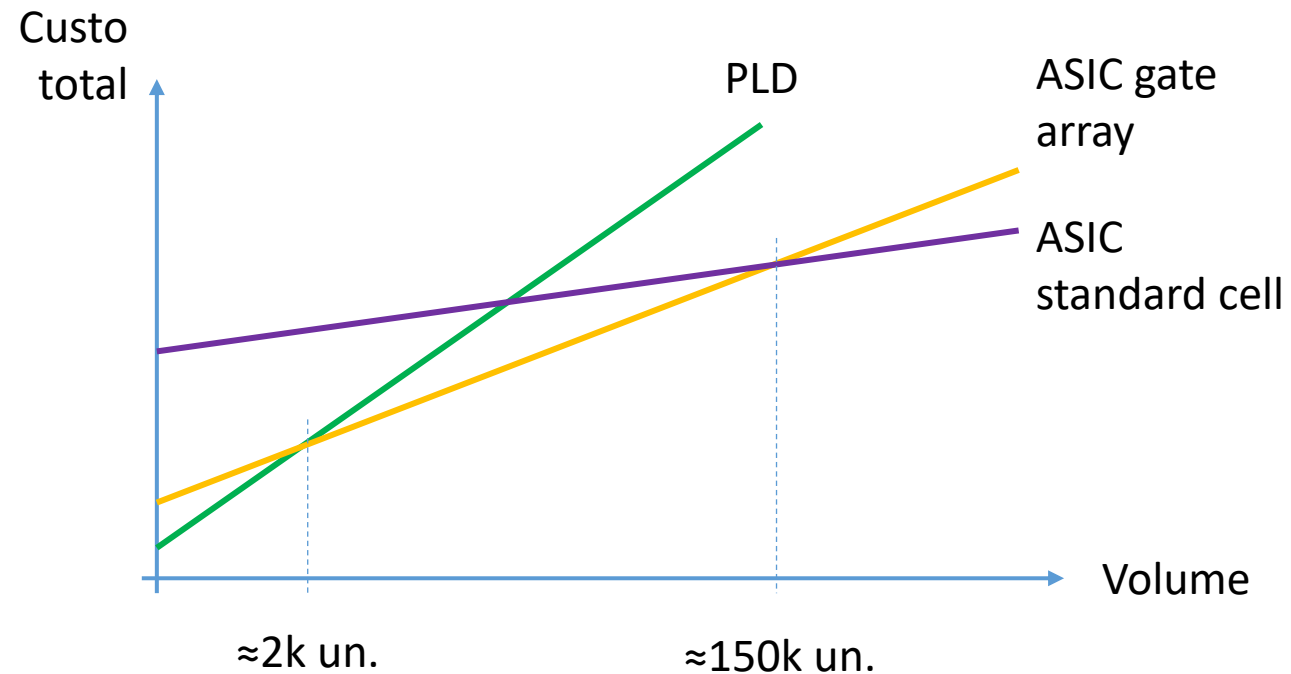
Um dispositivo lógico programável, (***programmable logic device* - PLD**) é um circuito integrado que pode ser **programado pelo usuário** para implementar **qualquer função lógica digital**.

Tem um **custo** de componente **mais elevado** do que um ASIC, porém com um custo de desenvolvimento menor. Assim, os PLDs são indicados para produtos com **volume de produção pequenos ou médios**.

Quase todos os tipos de PLDs atuais podem ser **reprogramados**, facilitando, dessa forma, a revisão dos projetos, mesmo depois do produto montado.

PLDs

Comparação de custos



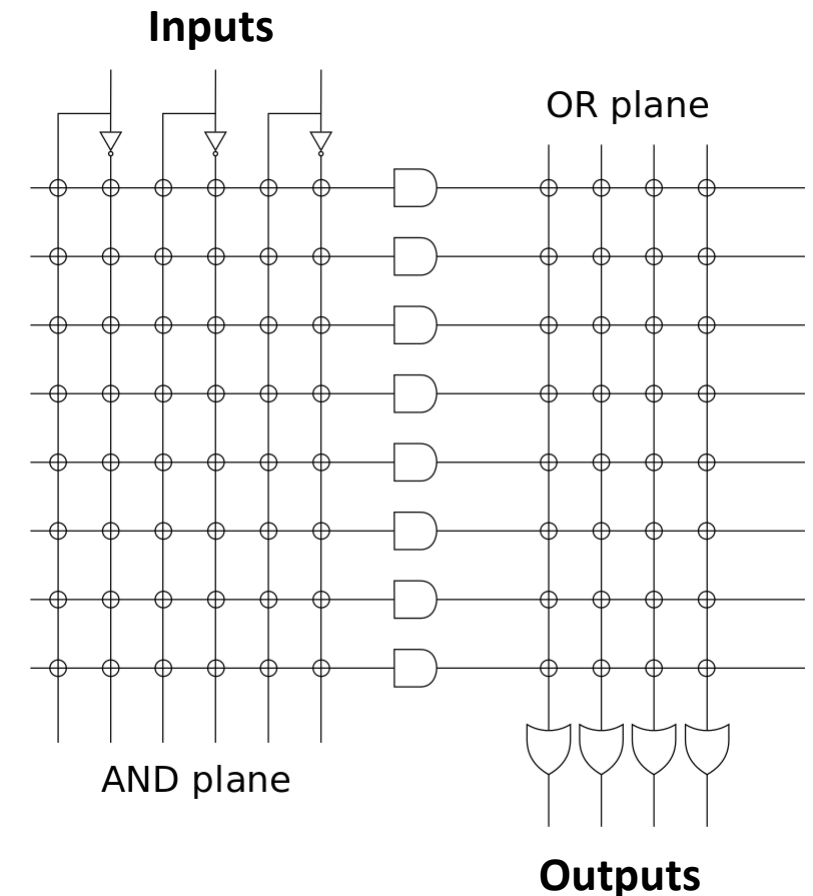
PLD - PLA

Programmable logic array é um tipo de PLD usado para implementar lógica **combinacional**.

O PLA tem um conjunto de planos de portas **AND programáveis**, que se conectam a um conjunto de planos de portas **OR programáveis**.

Para **N entradas**, tem **2^N portas AND** e para **M saídas**, deve haver **M portas OR**, cada uma com entradas programáveis de todas as portas AND.

Este layout permite que muitas funções lógicas sejam sintetizadas na forma canônica de **soma de produtos**.



PLD - PLA

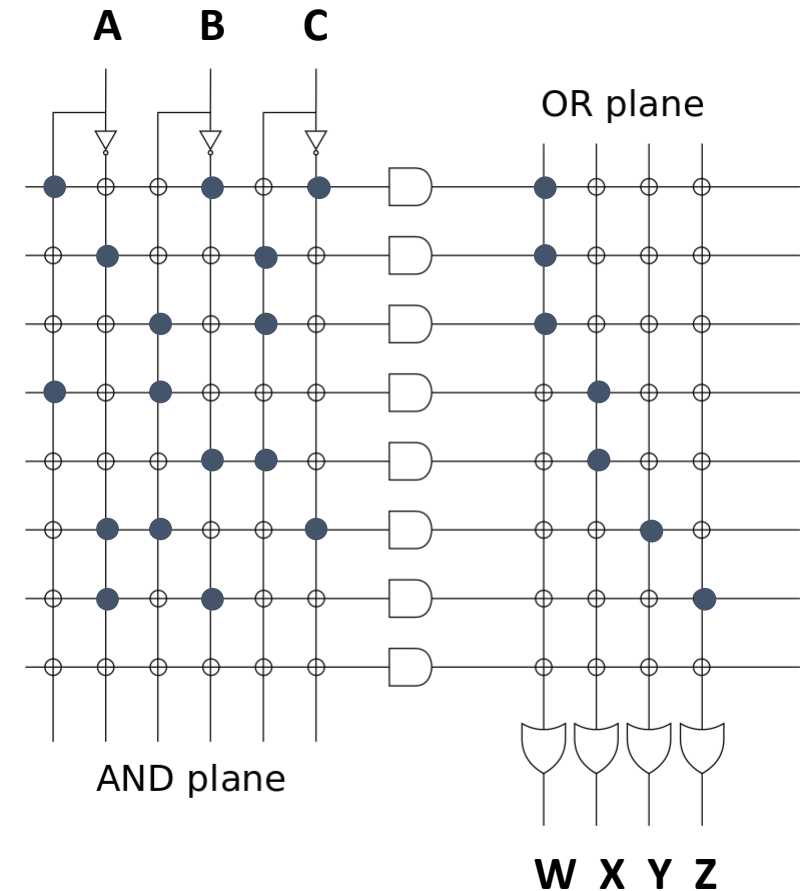
Exemplo:

$$W = A\bar{B}\bar{C} + \bar{A}C + BC$$

$$X = AB + \bar{B}C$$

$$Y = \bar{A}B\bar{C}$$

$$Z = \bar{A}\bar{B}$$

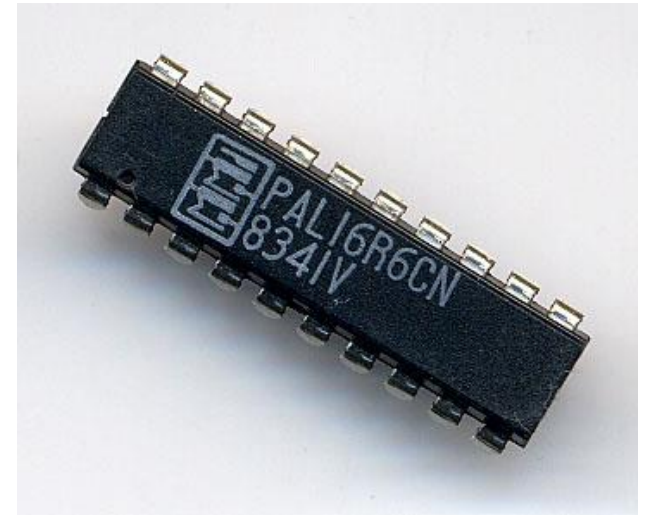


PLD - PAL

Programmable Array Logic (PAL) é uma família de PLDs usada para implementar funções lógicas em circuitos digitais, introduzida pela **Monolithic Memories, Inc. (MMI)** em março de **1978**.

Através de um equipamento apropriado, pode ser programável em campo.

- Programável apenas uma vez (one time programming - **OTP**)
- Reprogramável
 - Apagável através de luz ultravioleta (como uma **UV-EPROM**)
 - Apagável eletricamente (como uma **E²PROM**)

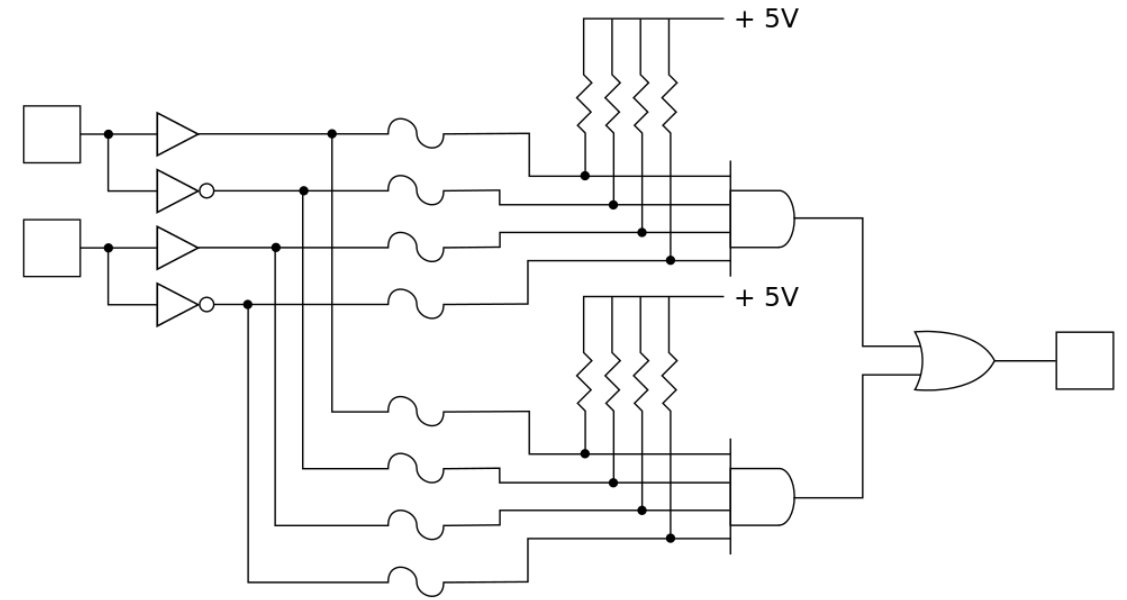


CI PAL 16R6, fabricado pela MMI

PLD - PAL

Diferentemente do PLA, nesse caso **somente o plano de portas AND é programável**.

A lógica das portas **OR é fixa**, cada porta ligada a um número definido de portas AND



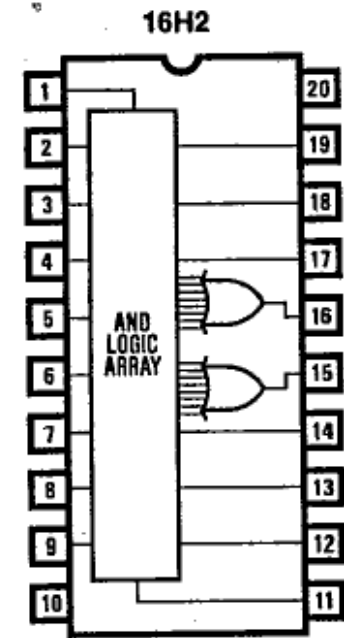
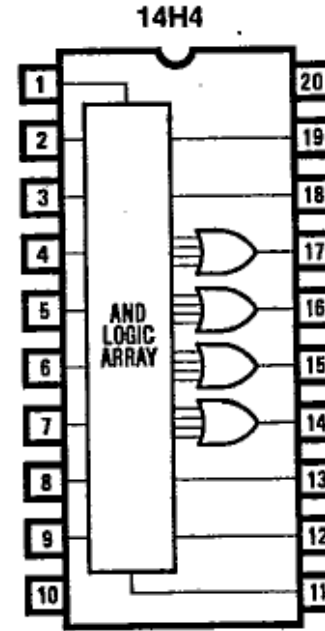
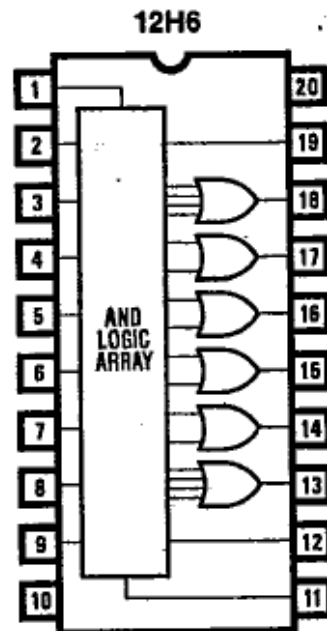
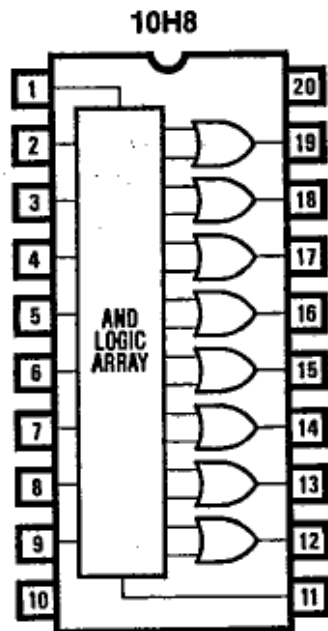
Simplified programmable logic device

PLD - PAL

PAL comercial: 10H8, 12H6, 14H4 e 16H2

10, 12, 14 ou 16 entradas / 8, 6, 4 ou 2 saídas

Lógica combinacional apenas – Saídas ativas em alto (não invertidas)

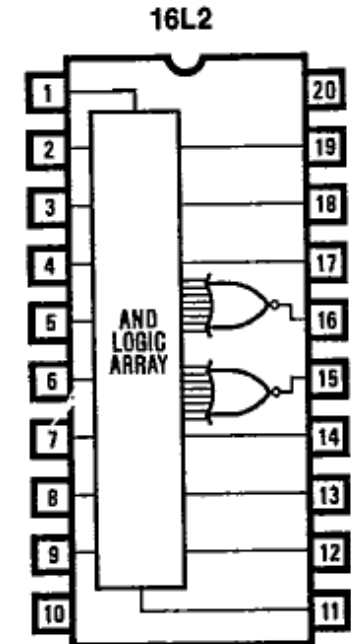
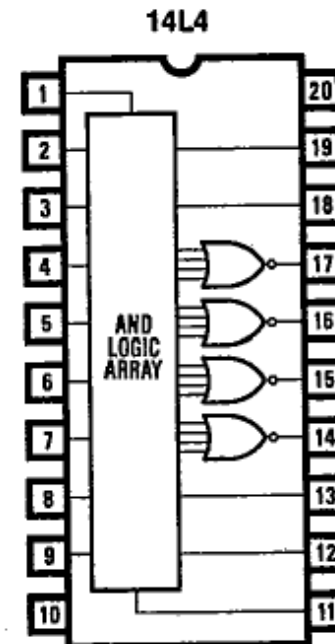
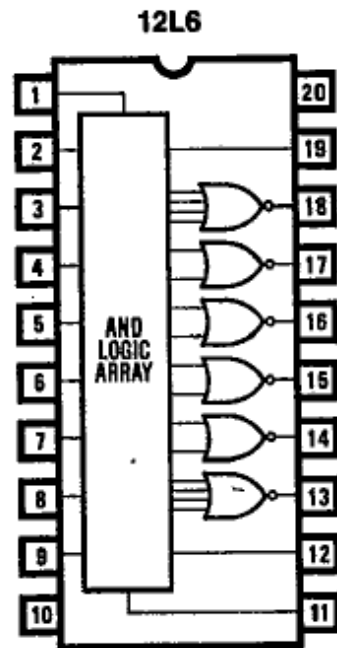
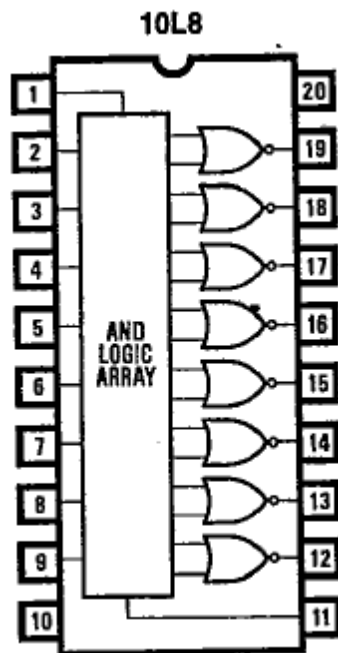


PLD - PAL

PAL comercial: 10L8, 12L6, 14L4 e 16L2

10, 12, 14 ou 16 entradas / 8, 6, 4 ou 2 saídas

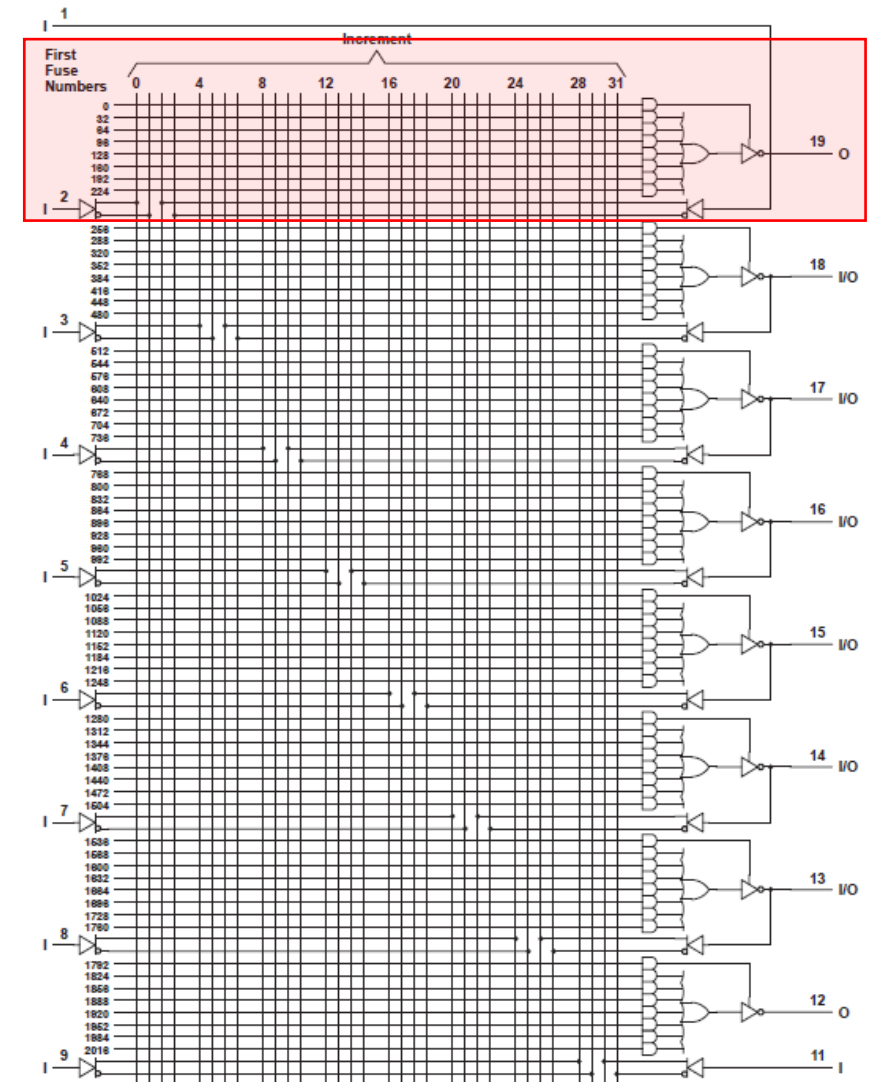
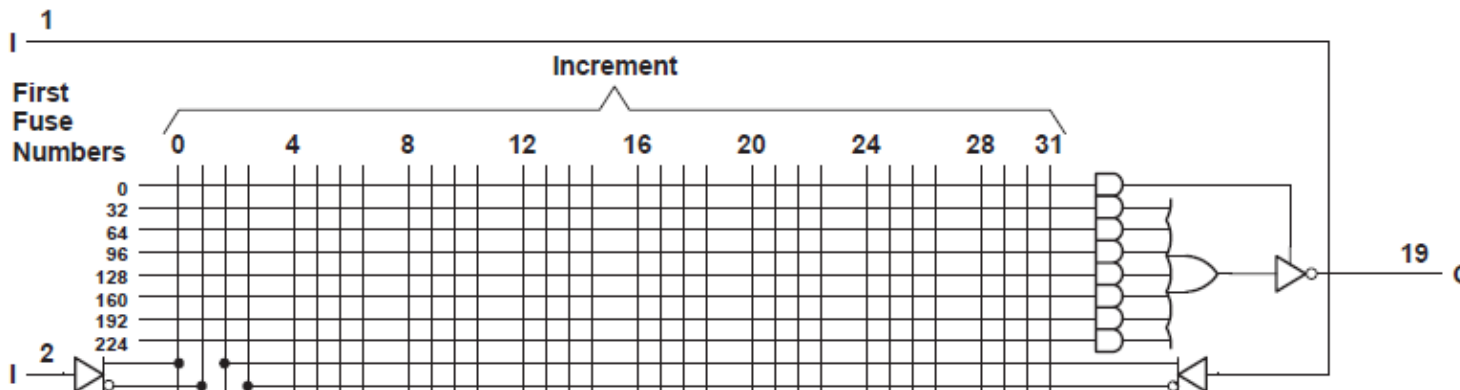
Lógica combinacional apenas – Saídas ativas em baixo (invertidas)



PLD - PAL

PAL comercial: 16L8 - 16 entradas e 8 saídas (lógica combinacional).

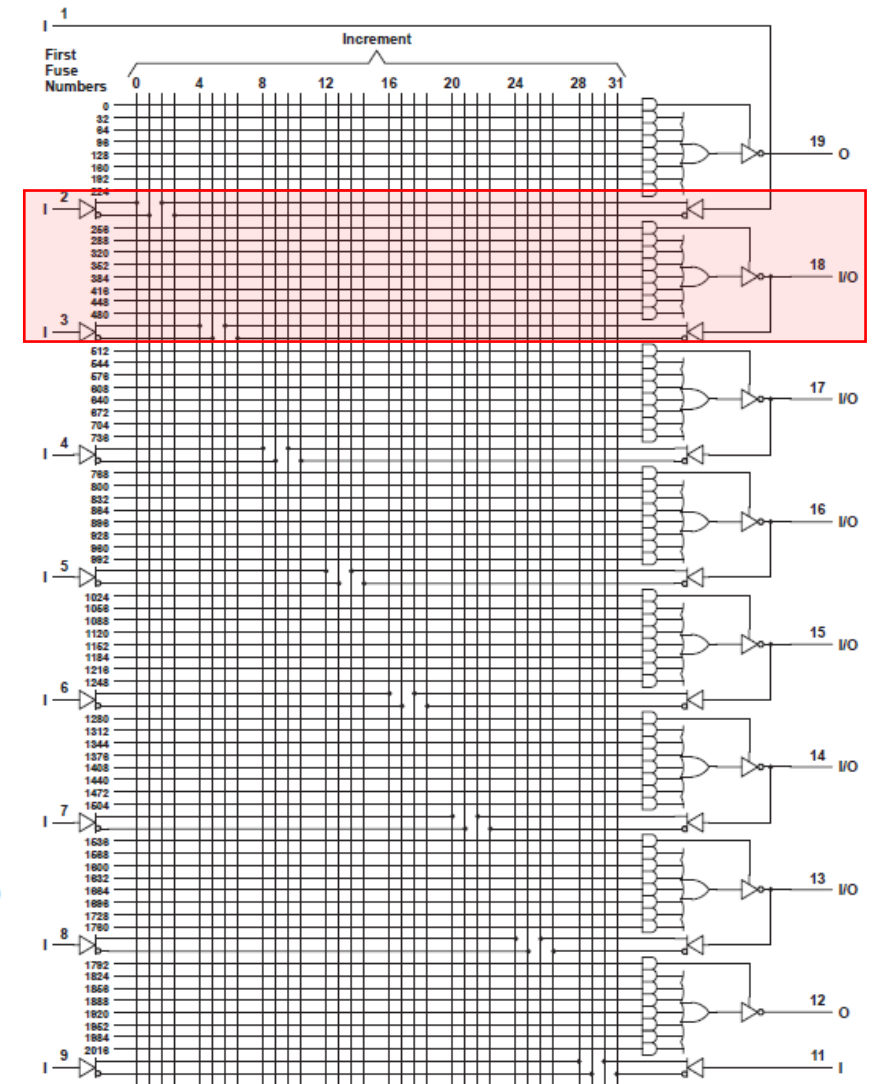
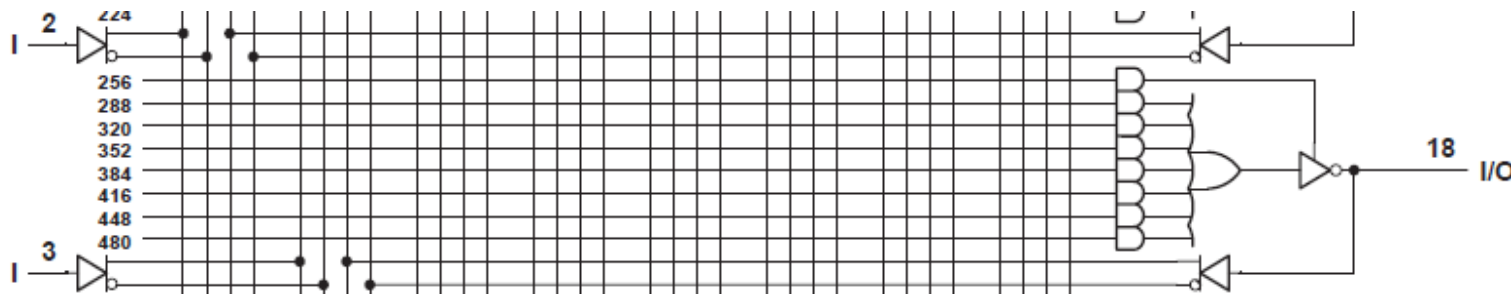
- 10 entradas
- 6 entradas/saídas
- 2 saídas



PLD - PAL

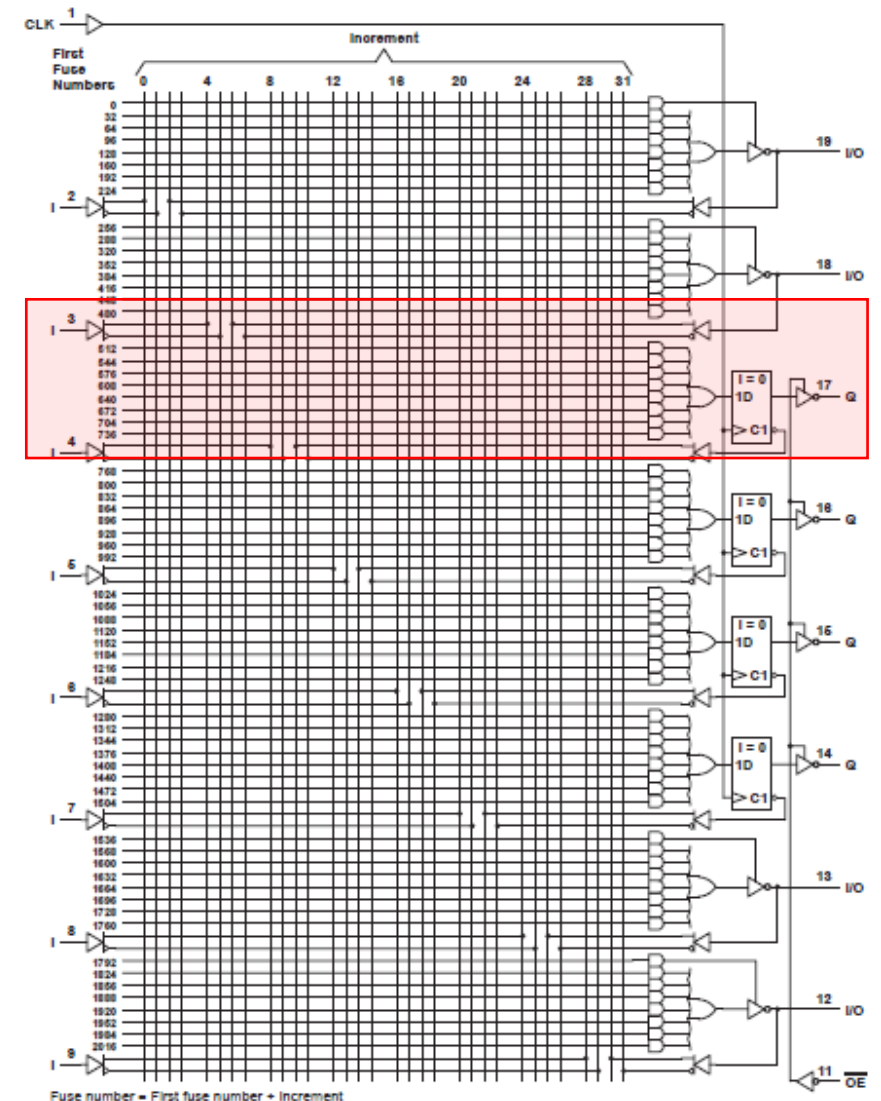
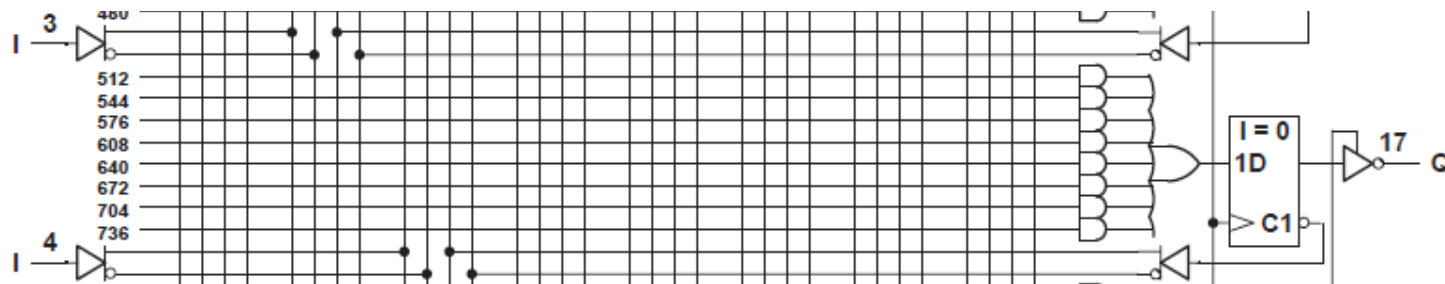
PAL comercial: 16L8 - 16 entradas e 8 saídas (lógica combinacional).

- 10 entradas
- 6 entradas/saídas
- 2 saídas



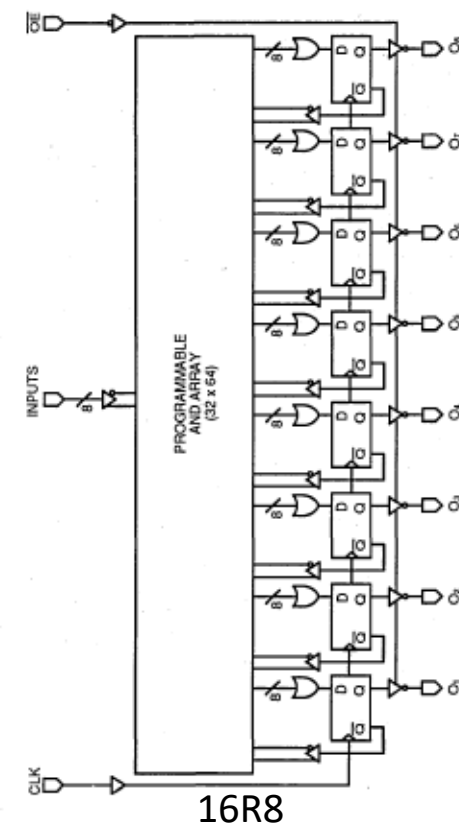
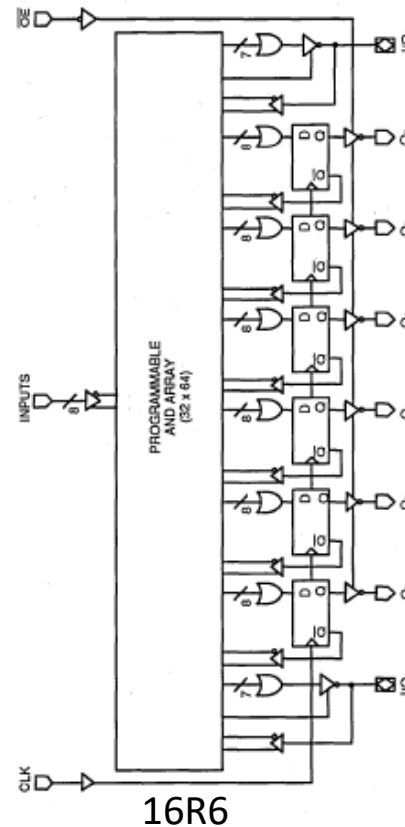
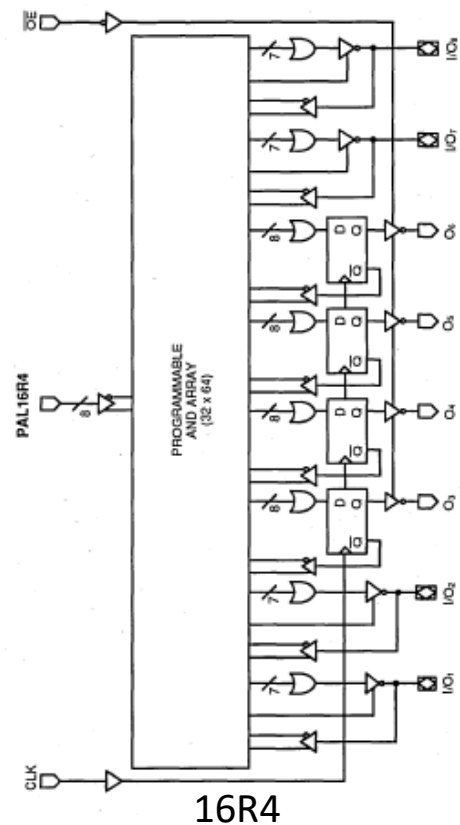
PLD - PAL

PAL comercial: 16R4 - 16 entradas e 8 saídas (4 flip-flops).



PLD - PAL

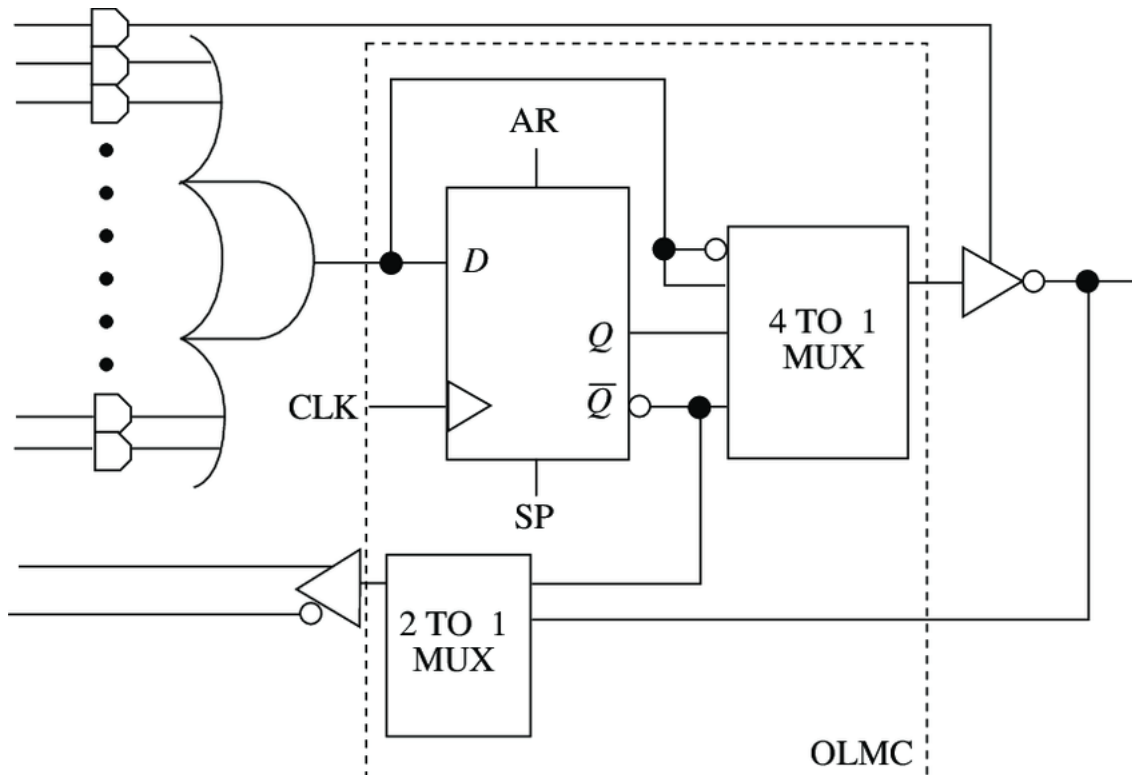
PAL comercial: 16R4, 16R6 e 16R8
16 entradas / 8 saídas (4, 6 ou 8 flip-flops).



PLD - GAL

Generic Array Logic (GAL) → Evolução do PAL

As saídas têm uma **macrocélula** => muito mais flexibilidade de uso:



O pino serve como entrada ou saída
Saídas:

- Combinacional: não invertida ou invertida
- Com flip-flop: não invertida ou invertida

Entrada na lógica:

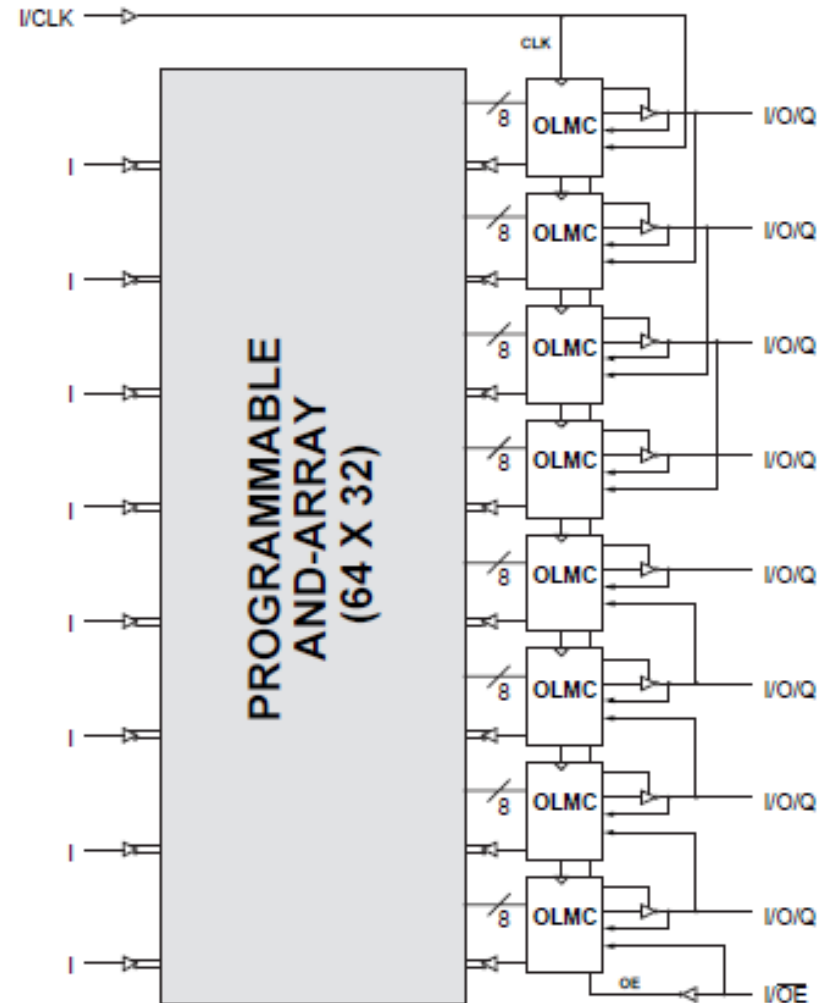
- Do pino
- Do flip-flop

PLD - GAL

GAL comercial: 16V8

8 macrocélulas

Substitui qualquer um dos PALs apresentados anteriormente

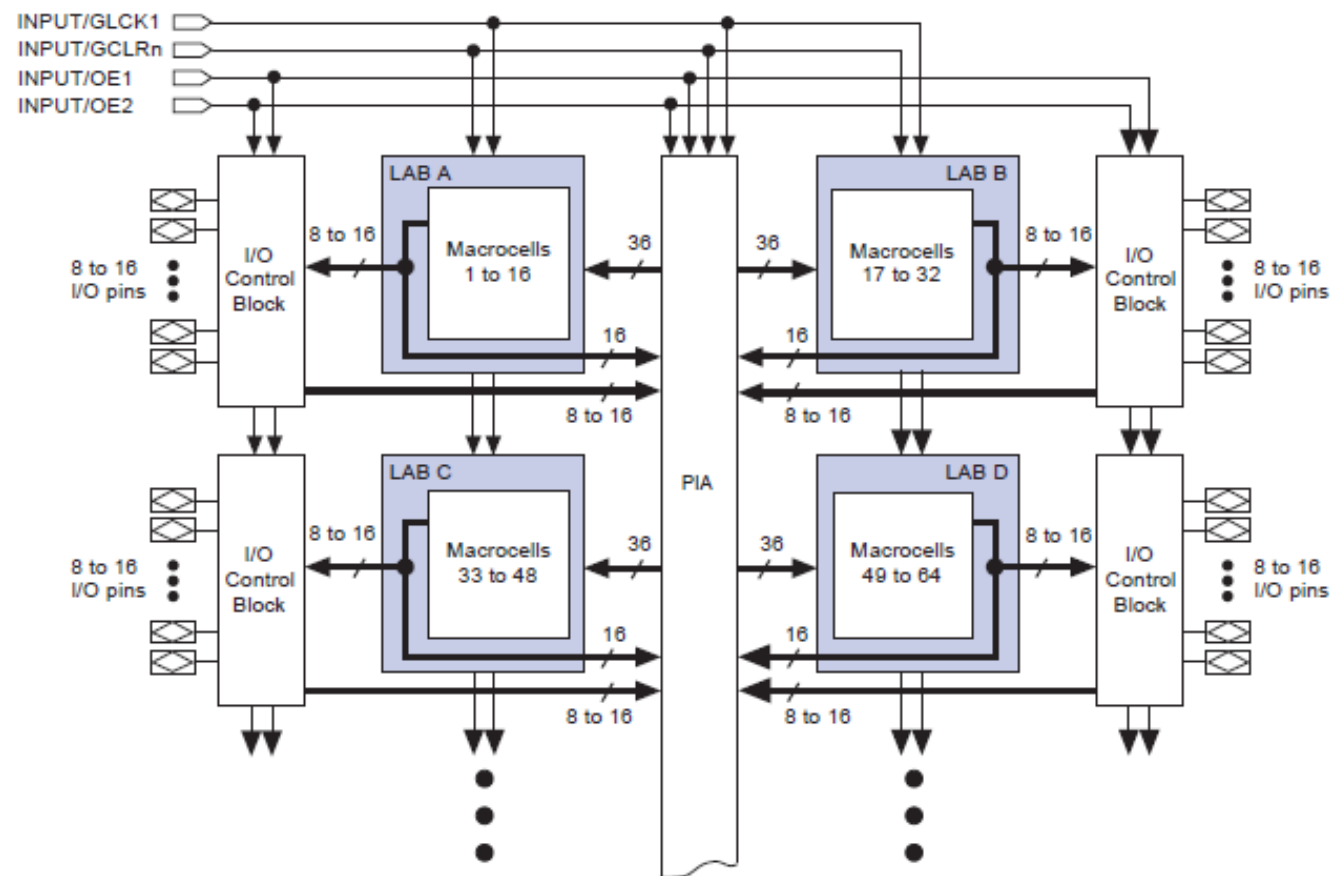


PLD - CPLD

Complex PLD (CPLD)

Reúne, em um mesmo circuito integrado, vários blocos lógicos semelhantes a um PAL com **macrocélulas**, e uma matriz de **interconexão programável**.

Obs. Os dispositivos PLA, PAL e GAL são chamados de simple PLD (SPLD)



LAB: Logic array block

PIA: Programmable interconnect array

PLD - CPLD

Complex PLD (CPLD)

A programação de um CPLD fica armazenada em uma **memória não volátil**, do tipo **UV-EEPROM** (nos modelos mais antigos) ou **E²PROM**.

Uma característica dos CPLD é que as interconexões são feitas através de **barramentos contínuos de comprimento uniforme**, o que faz com que o tempo de **propagação (*delay*)** entre quaisquer duas células lógicas do dispositivo seja **constante**.

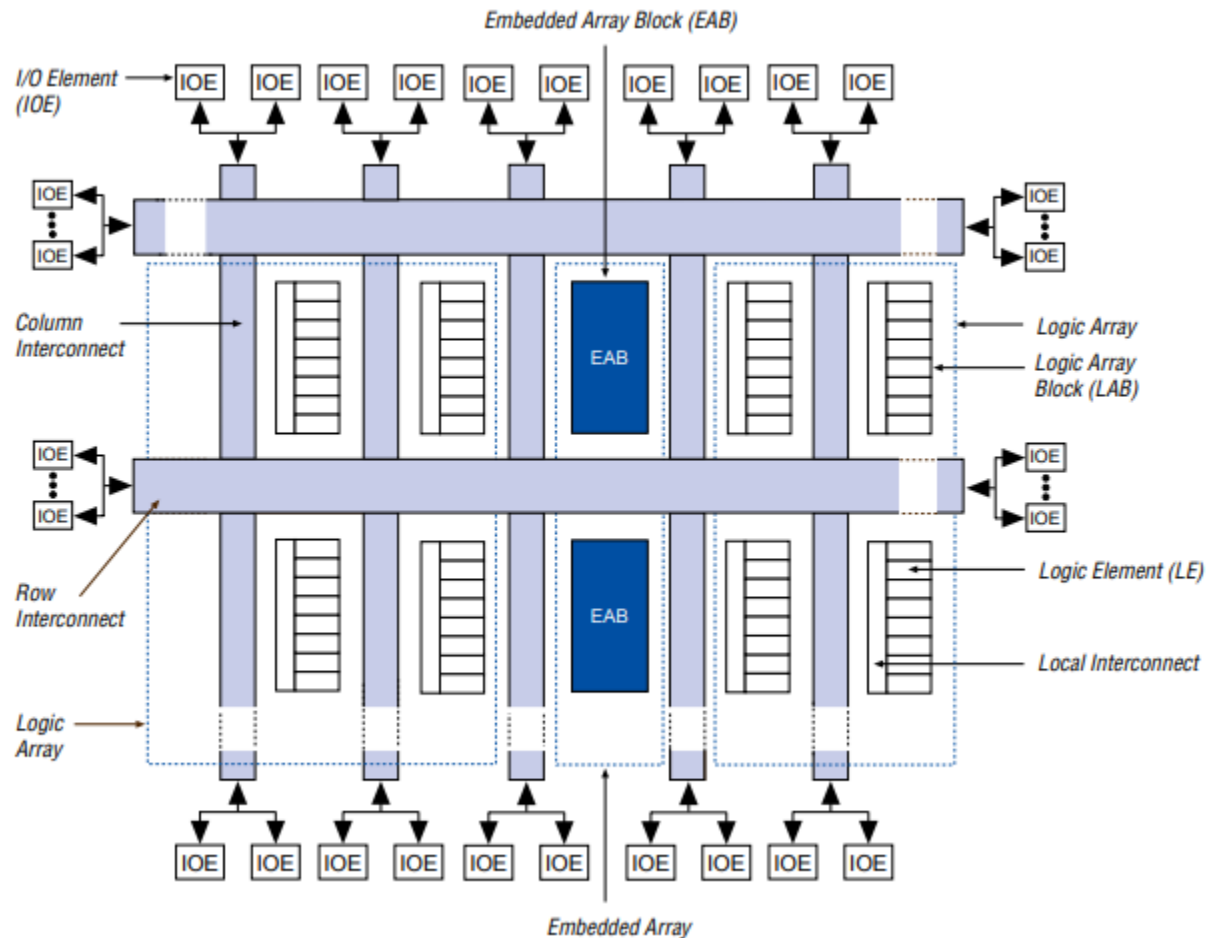


PLD – FPGA

Field programmable gate array (FPGA)

Contém uma **matriz de blocos lógicos programáveis**, e uma **hierarquia de interconexões reconfiguráveis**.

Muitos FPGAs têm também, memória **RAM embutida** e outros circuitos, tais como **PLL** (phase locked loop) ou **multiplicadores**.



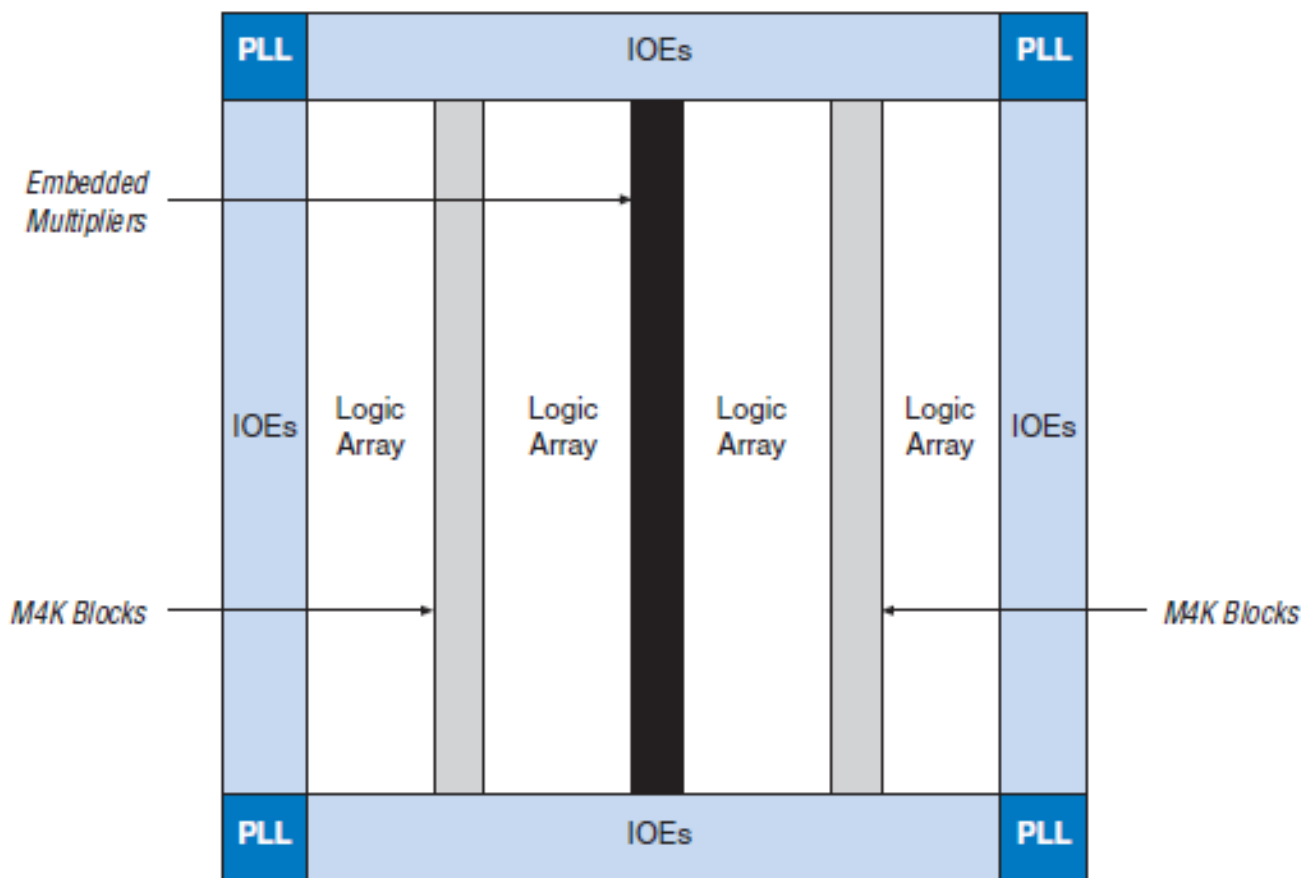
Altera Flex 10k

PLD – FPGA

Field programmable gate array (FPGA)

Contém uma **matriz de blocos lógicos programáveis**, e uma hierarquia de interconexões reconfiguráveis.

Muitos FPGAs têm também, memória RAM embutida e outros circuitos, tais como PLL (phase locked loop) ou multiplicadores.

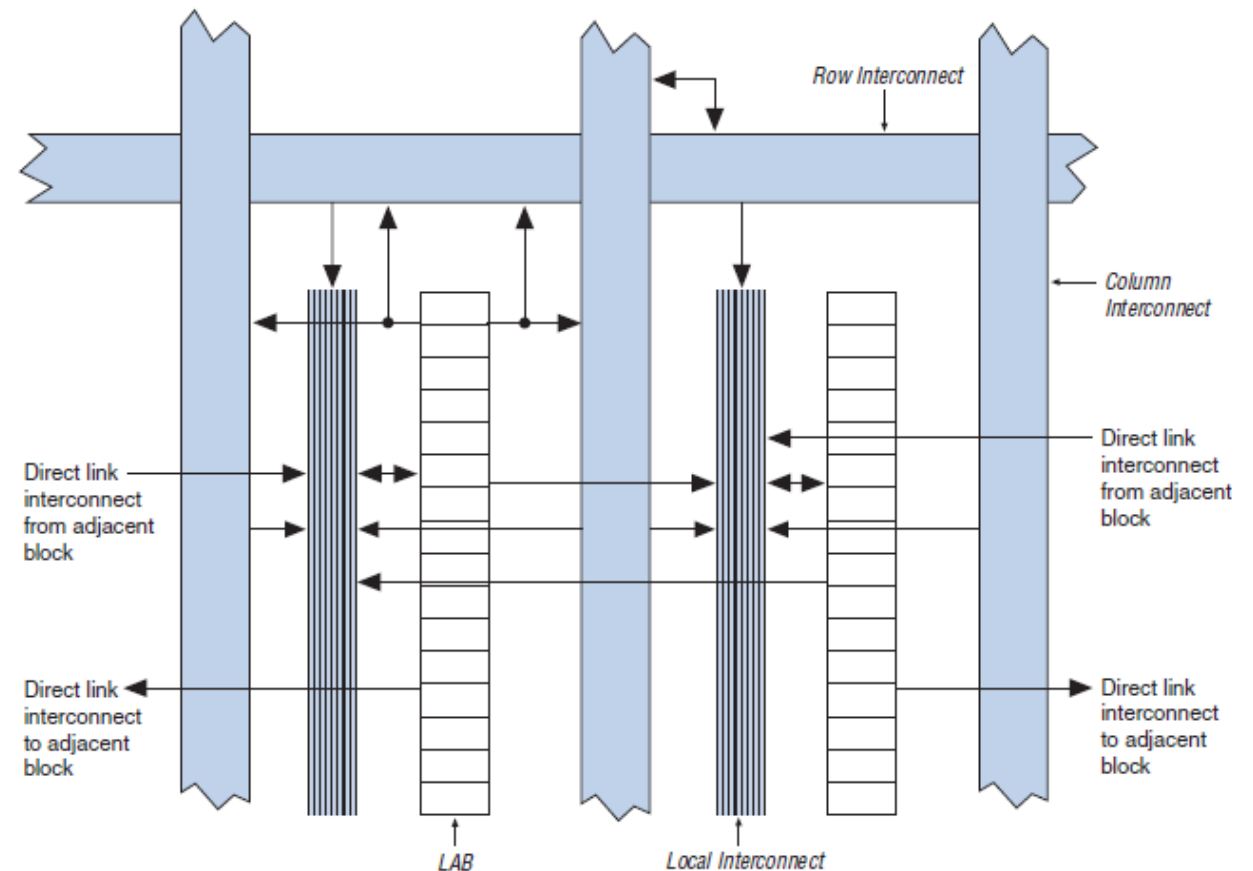


Altera Cyclone II

PLD – FPGA

Logic array block (LAB)

Composto por um conjunto de *logic elements* (LE) e um barramento de interconexão local.



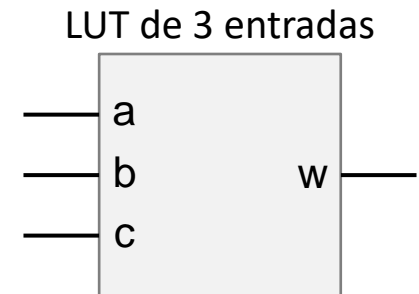
PLD – FPGA

Look up table (LUT)

Diferentemente dos SPLDs e CPLDs, os **FPGAs não têm planos AND ou OR.**

As **funções de lógica combinacional são geradas** através de uma tabela de consulta (**LUT – lookup table**), que permite criar **qualquer função lógica**, limitada apenas pelo número de entradas na LUT.

Os dados de configuração da LUT ficam armazenados em uma **memória do tipo SRAM**, portanto, **volátil**. Assim, os dispositivos desse tipo são ligados a uma **memória de configuração externa, do tipo E²PROM**, que carrega automaticamente a configuração logo após a energização do circuito.



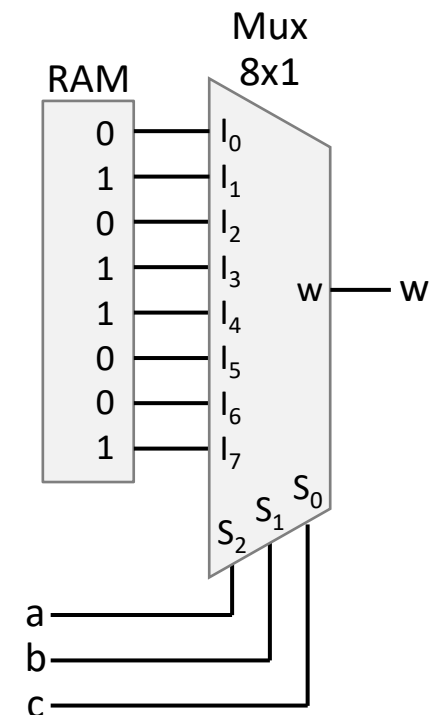
PLD – FPGA

Look up table (LUT)

Diferentemente dos SPLDs e CPLDs, os FPGAs não têm planos AND ou OR.

As funções de lógica combinacional são geradas através de uma tabela de consulta (LUT – *lookup table*), que permite criar qualquer função lógica, limitada apenas pelo número de entradas na LUT.

Os dados de configuração da LUT ficam armazenados em uma memória do tipo SRAM, portanto, volátil. Assim, os dispositivos desse tipo são ligados a uma memória de configuração externa, do tipo E²PROM, que carrega automaticamente a configuração logo após a energização do circuito.



PLD – FPGA



Look up table (LUT) – Granularidade

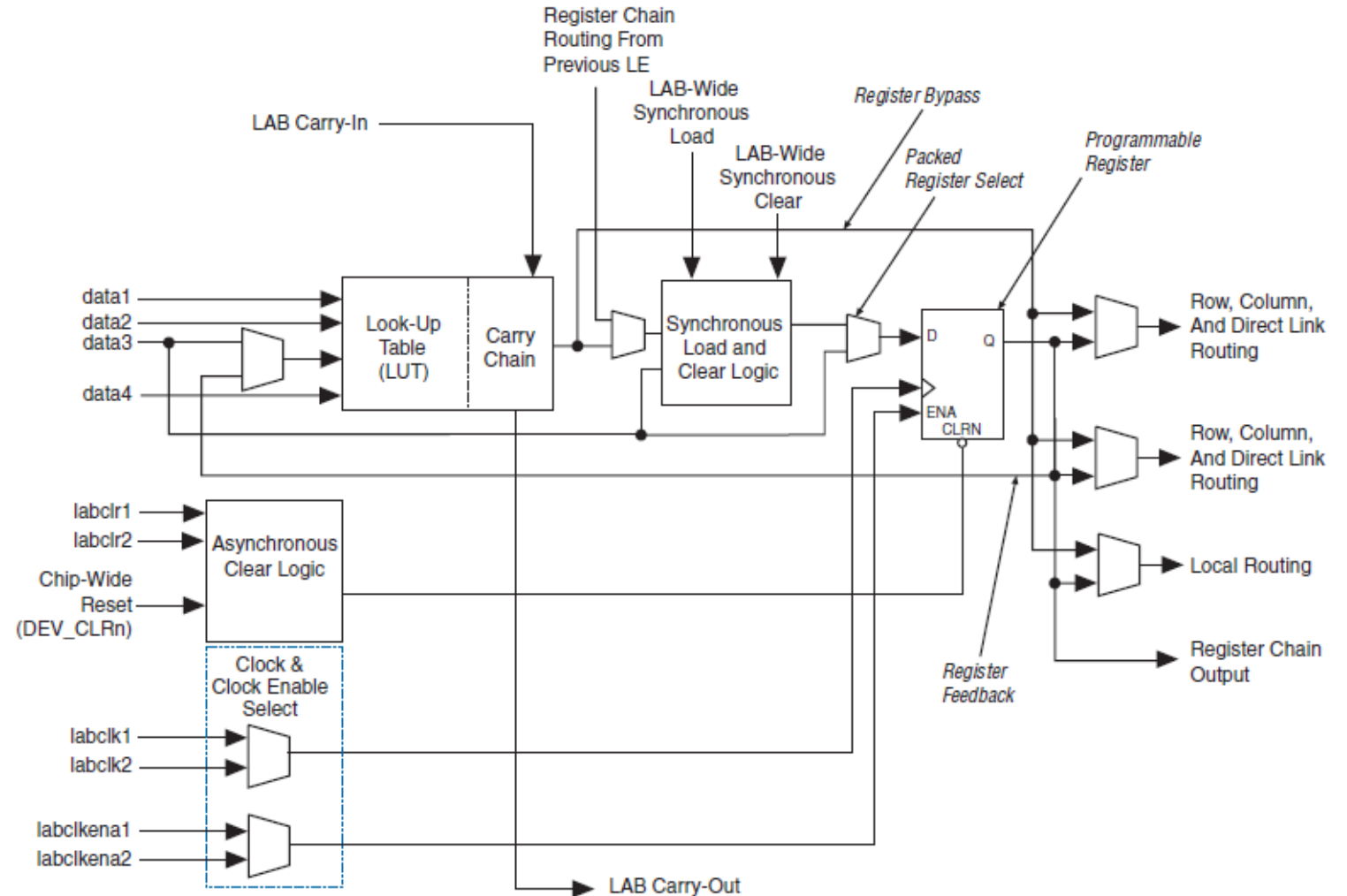
Um elemento lógico com uma LUT de muitas entradas é capaz de realizar funções lógicas mais complexas em um só LE. Porém, para funções simples, esse LE ficará subutilizado.

Um LE com uma LUT com poucas entradas terá uma utilização mais plena com funções com poucas variáveis. Por outro lado, para funções mais complexas, haverá a necessidade do uso de mais de um LE.

Na família Cyclone II da Altera, os LEs têm uma LUT de 4 entradas e um flip-flop., além de multiplexadores programáveis que permitem configurar o LE, de maneira semelhante à macrocélula de um CPLD, porém com maior número de opções.

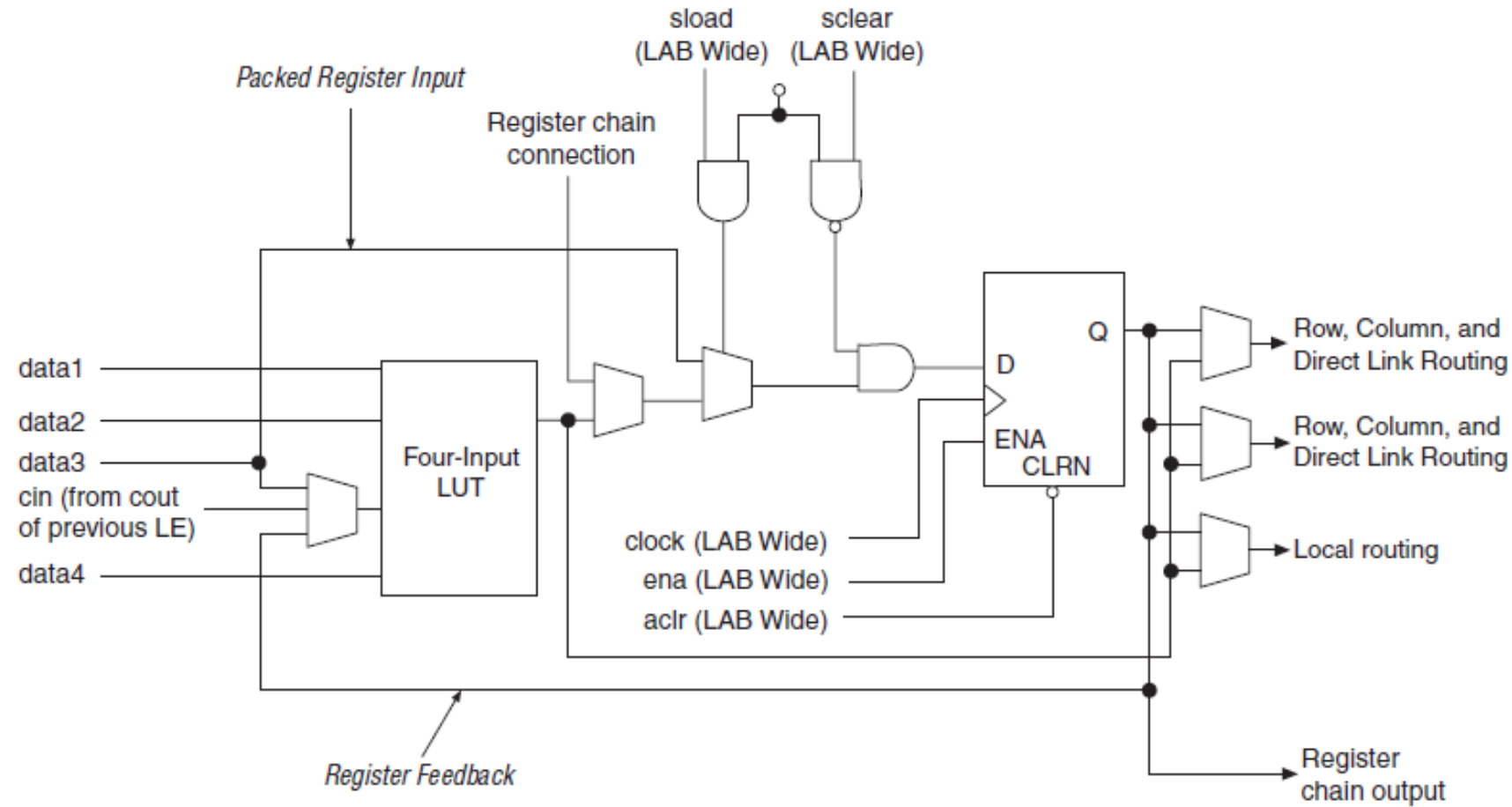
PLD – FPGA

Logic element (LE)



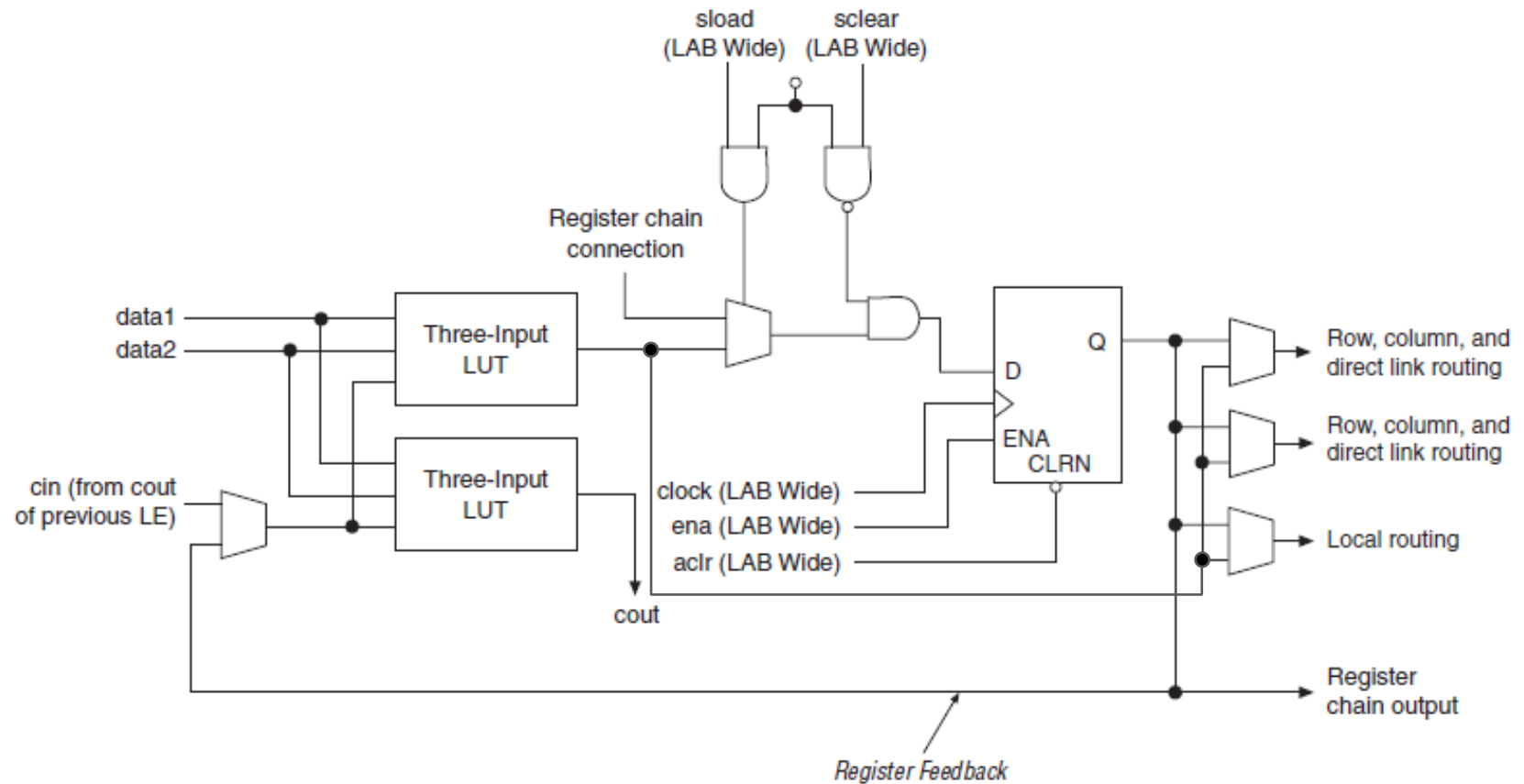
PLD – FPGA

- LE no modo normal



PLD – FPGA

- LE no modo aritmético





Fim

Até a próxima aula!