



Fundamentos de Sistemas Digitais

Dispositivos, ferramentas e técnicas de implementação de
circuitos combinatórios.

Nas últimas aulas...

- ▶ Vimos um estudo de caso
 - ▶ e dele derivamos um circuito
- ▶ Navegamos por técnicas de representação e de minimização
 - ▶ Técnicas que simplificam a vida
- ▶ Entendemos o que é uma representação canônica
 - ▶ E conhecemos mais de um tipo
- ▶ Discutimos formas de implementar o circuito
 - ▶ Desenhamos bastante!



Algumas considerações

Quando assumimos que há disponibilidade indiscriminada de funções lógicas básicas e diversas para implementar o circuito de uma dada função lógica, dizemos que estamos usando ‘lógica aleatória’ (*random logic*).

Lógica aleatória permite otimizar elementos, mas seu uso prático pode ser mais caro.

Precisamos conhecer outras opções

Na aula de hoje

- ▶ Discutiremos em linhas gerais o fluxo de projeto de hardware
 - ▶ Em uma visão geral, abrangente, porém suficiente
- ▶ Veremos algumas alternativas de implementação de hardware
 - ▶ Com foco naquelas que são mais fáceis de entender
- ▶ Aprenderemos alguns formatos de arquivos e ferramentas
 - ▶ Conheceremos alguns programas acadêmicos e comerciais
- ▶ Começaremos a estudar uma HDL
 - ▶ Verilog HDL - *Hardware Description Language*

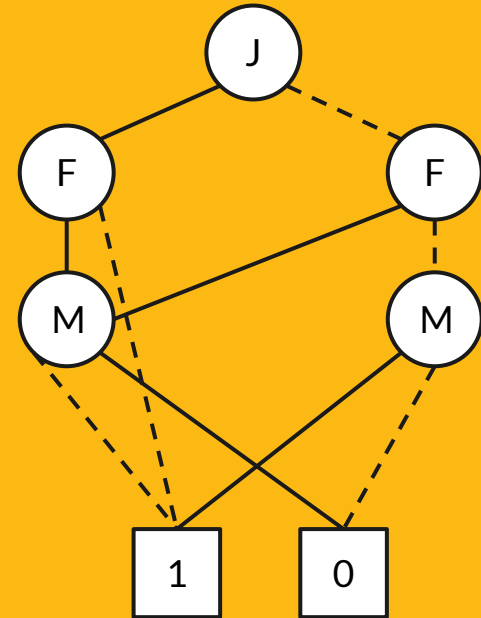


Fizeram o dever de casa?

- 1) Pesquise por: ROBDD - *Recursive Ordered Binary Decision Diagram*
 - a) *Essa estrutura é canônica?*
- 2) Entenda suas operações de redução a partir de um BDD.
- 3) Monte o diagrama ROBDD correspondente ao nosso exemplo.
- 4) Traga desenhado em seu caderno e compare com a resposta apresentada.

Fizeram o dever de casa?

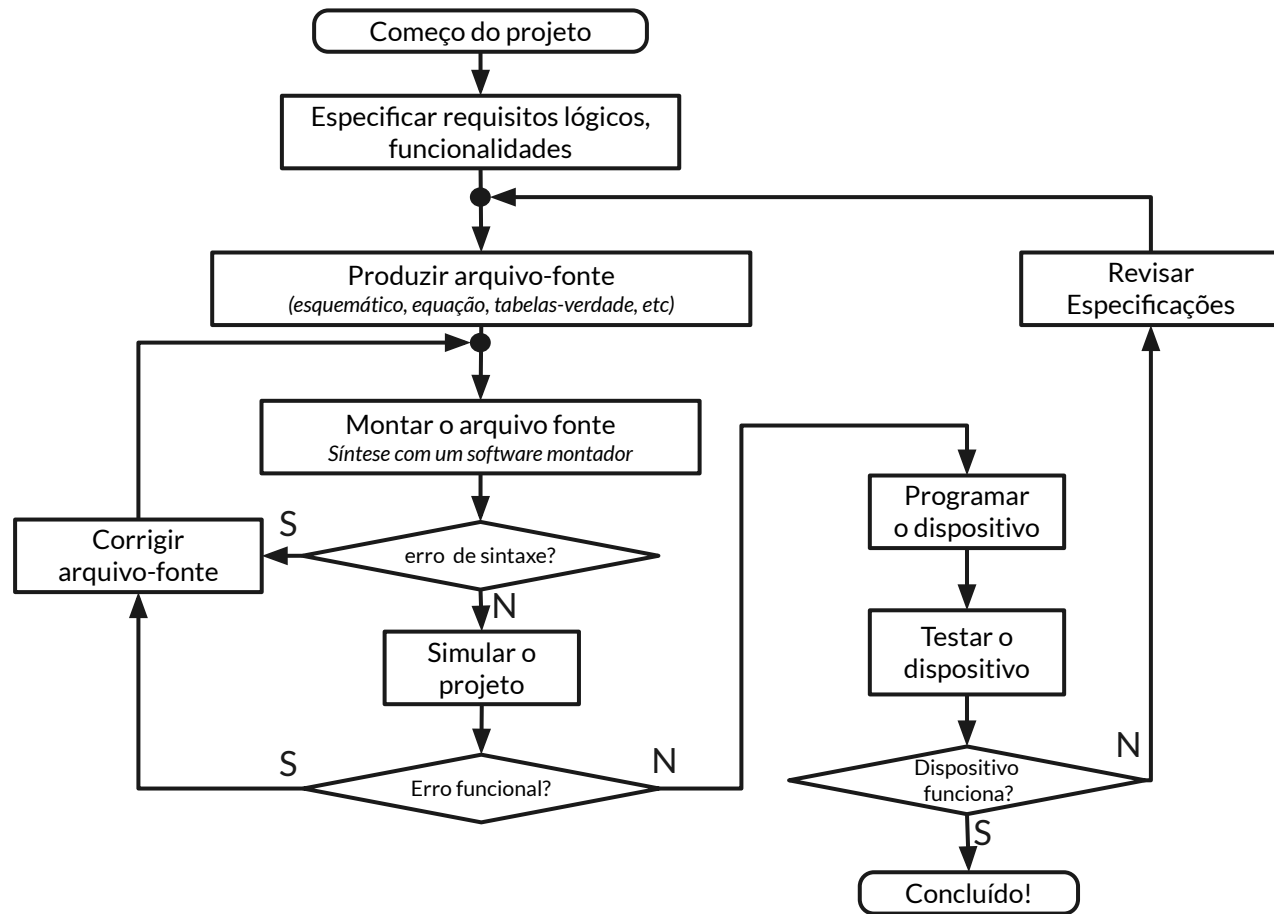
- 1) Pesquise por: ROBDD - *Recursive Ordered Binary Decision Diagram*
 - a) *Essa estrutura é canônica?*
- 2) Entenda suas operações de redução a partir de um BDD.
- 3) Monte o diagrama ROBDD correspondente ao nosso exemplo.
- 4) Traga desenhado em seu caderno e compare com a resposta apresentada.



Ciclo Genérico de Desenvolvimento de Hardware



Ciclo de Desenvolvimento de Hardware Programável



Ciclo de Desenvolvimento de Hardware Programável

Essa é uma visão geral do fluxo

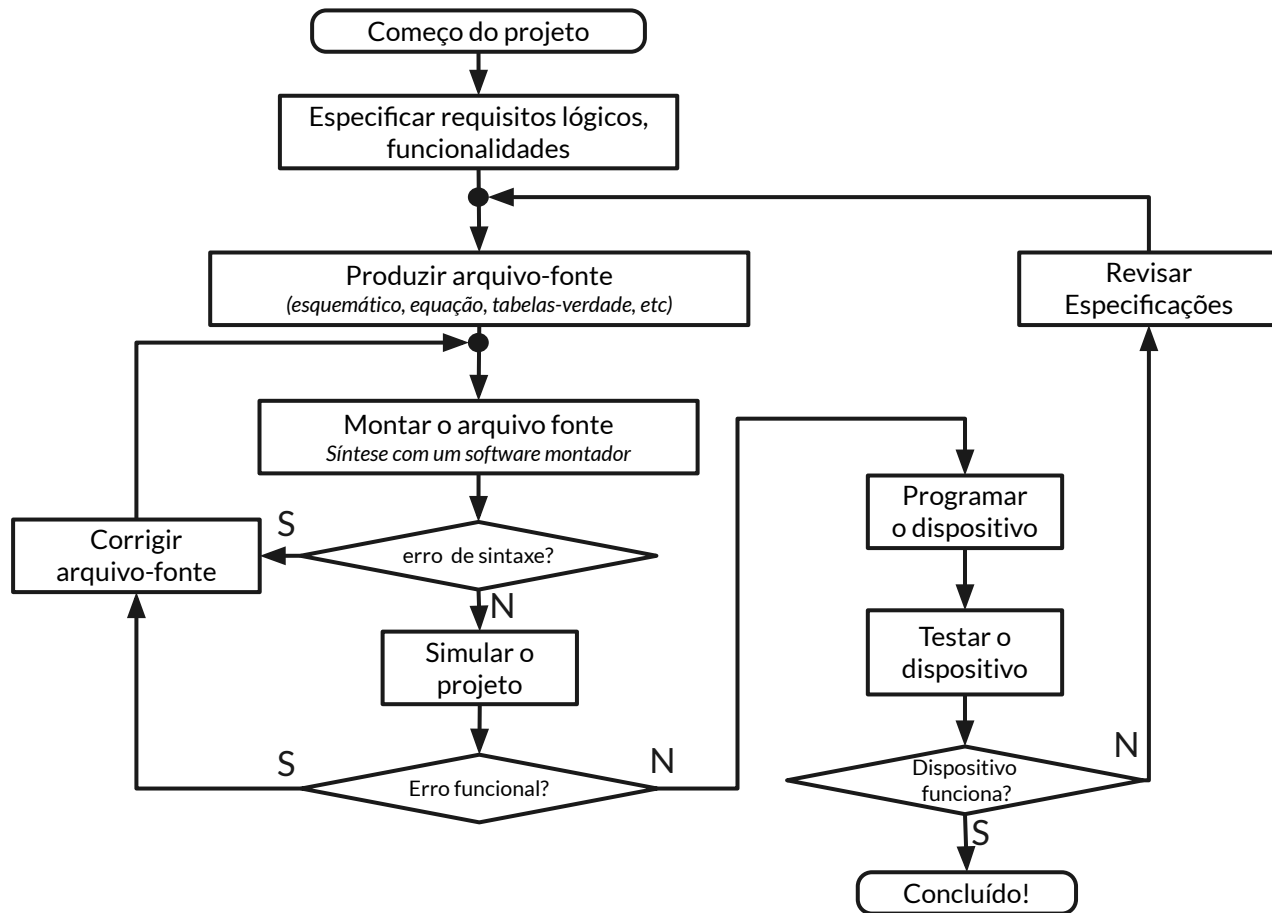
Autores usam mais ou menos detalhes.

Serve para hardware programável

Circuitos pré-fabricados

Algumas etapas se repetem em todos os fluxos

Como especificação, simulação, e teste



Ciclo de Desenvolvimento de Hardware Programável

Essa é uma visão geral do fluxo

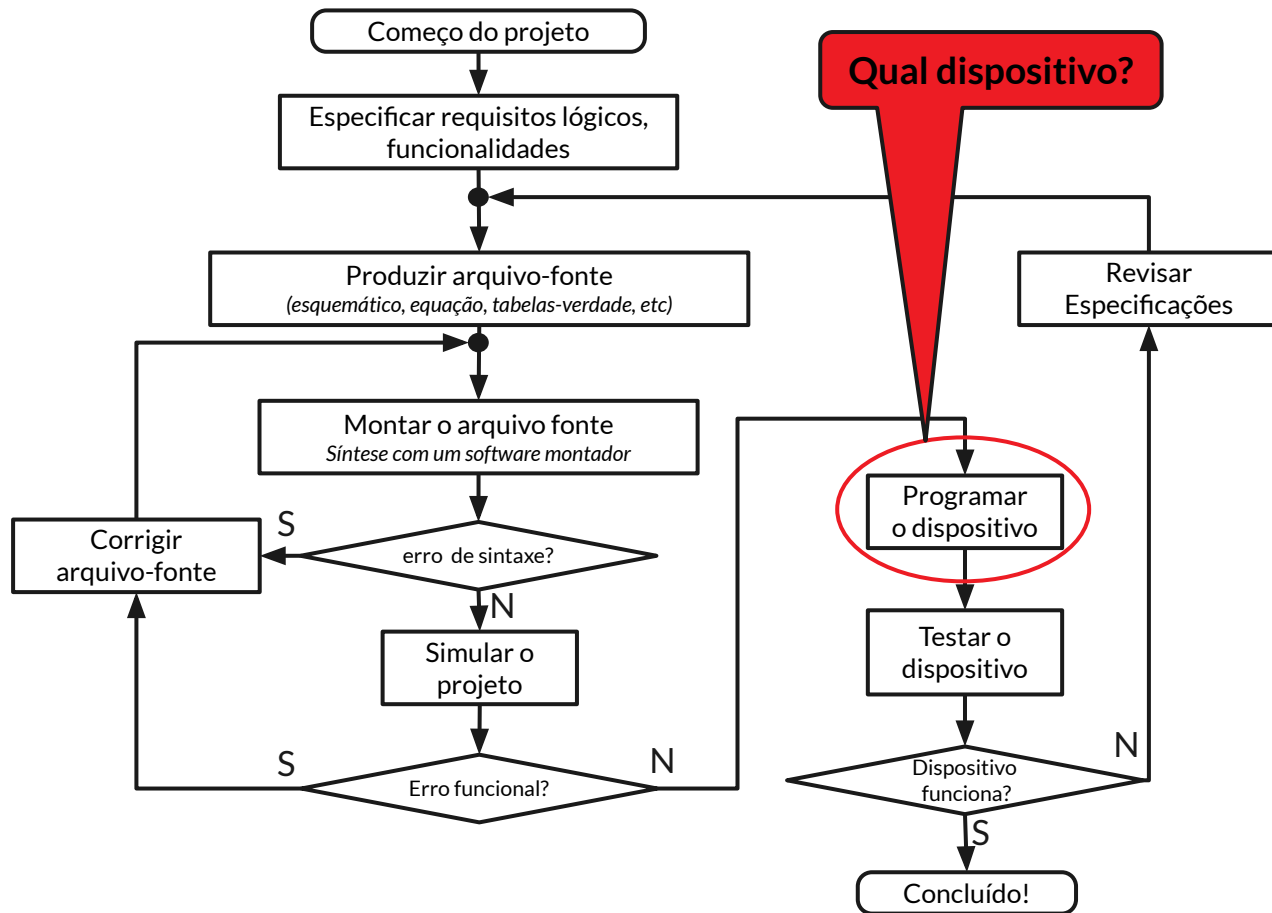
Autores usam mais ou menos detalhes.

Serve para hardware programável

Circuitos pré-fabricados

Algumas etapas se repetem em todos os fluxos

Como especificação, simulação, e teste



Dispositivos Lógicos Programáveis - PLD

Programmable Logic Devices



ROM

► Read Only Memory

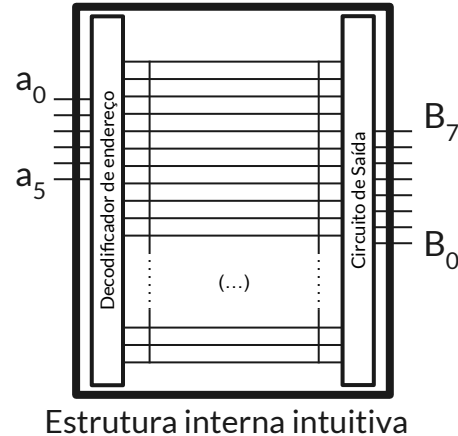
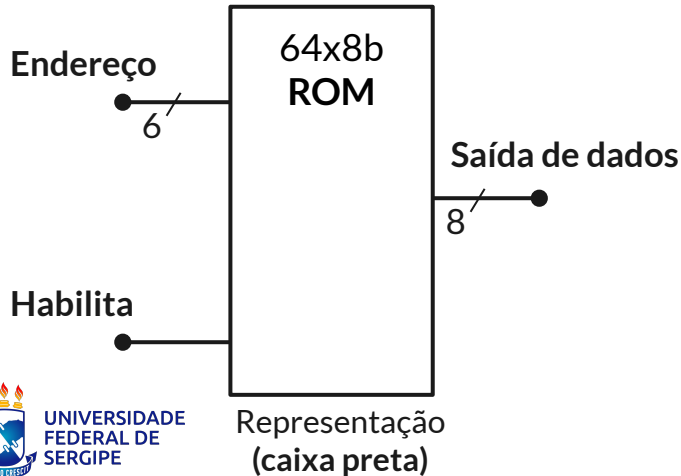
► Memória de Somente Leitura

Dispositivo de memória que vem **programado de fábrica**

Usuário tem pouca ou nenhuma capacidade de programação

O armazenamento de dados é **não-volátil**

O dado precisa estar bem depurado, sua produção costuma ser para uso em massa



Muitos consoles de videogames mais antigos usavam esses cartuchos substituíveis para suportar múltiplos jogos. Esse é um de Pokémon Silver. A ROM é o chip à direita "MX23C1603-12A".

PROM

- ▶ **Programmable Read Only Memory**
 - ▶ Memória Programável de Somente Leitura

A principal diferença para uma ROM:

ROM, o dado é armazenado na fábrica

PROM, o dado é programado **após** a fabricação

Quando usar uma PROM?

Quando o **volume requerido inviabiliza** produzir uma ROM, ou;

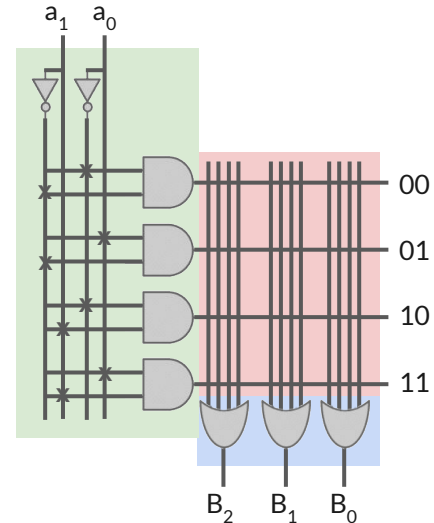
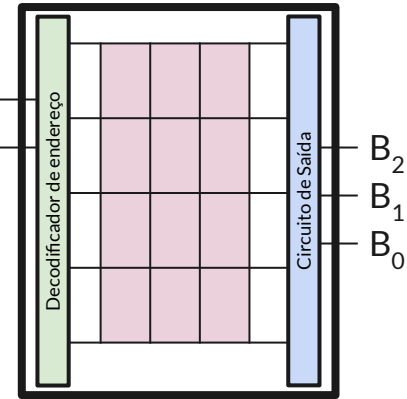
Durante o desenvolvimento de um sistema, antes de fabricar como ROM

São fabricadas com a memória vazia; empresas podem mantê-las em estoque, e programá-las somente quando necessário.

Variações/Evoluções: EPROM (*Erasable*-), EEPROM (*Electrically Erasable* -)

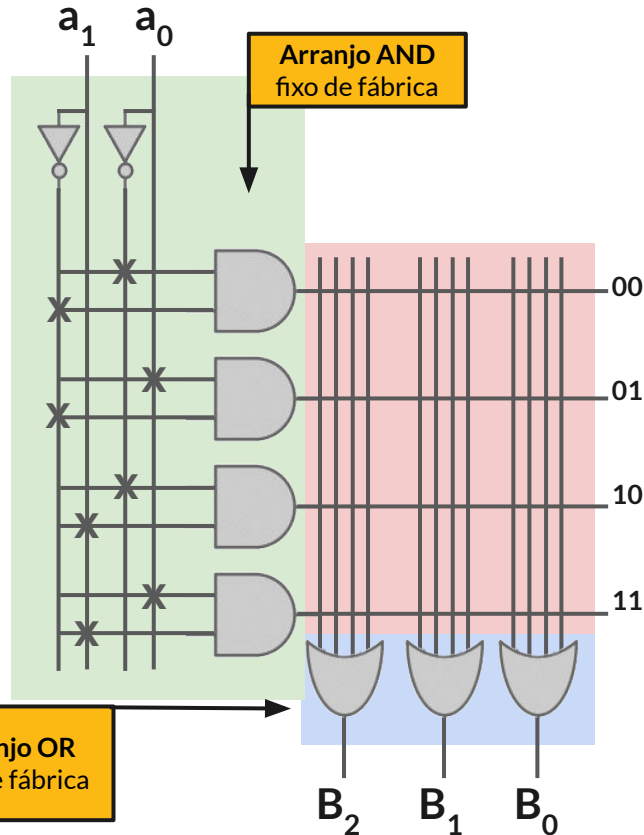
Uma forma de programação é queimando fusíveis que ligam linhas de conexão

Entradas		Funções Lógicas		
a_0	a_1	B_0	B_1	B_2
0	0			
0	1			
1	0			
1	1			

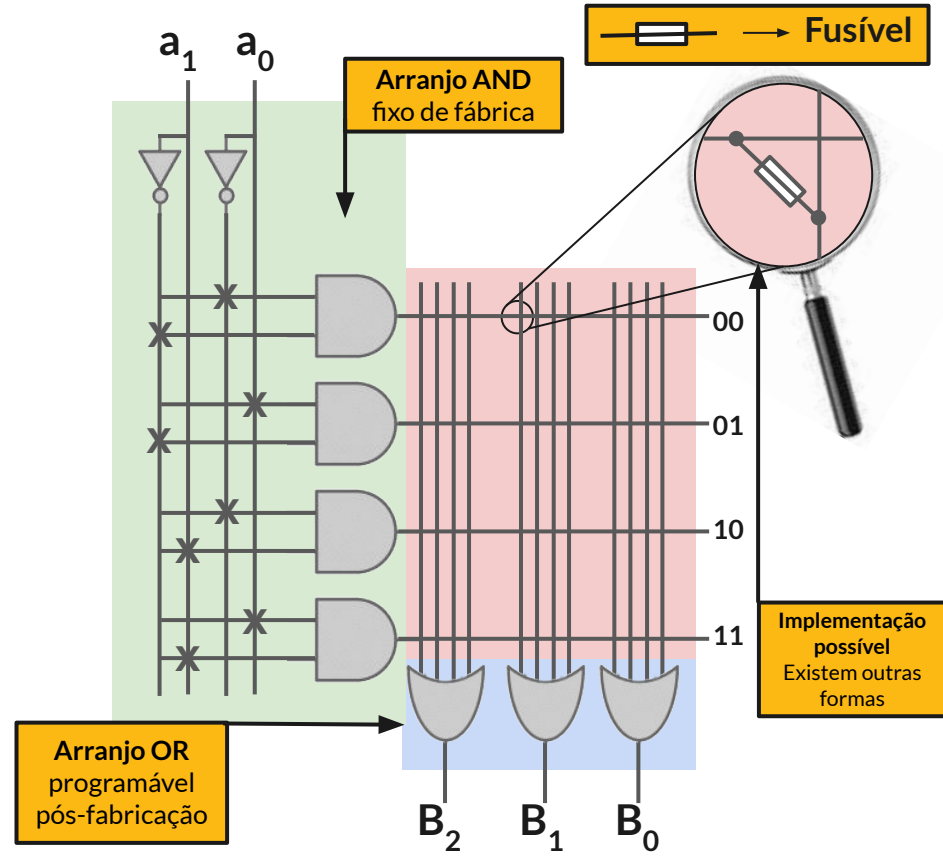


ROM x PROM: Estrutura interna

ROM - Read Only Memory

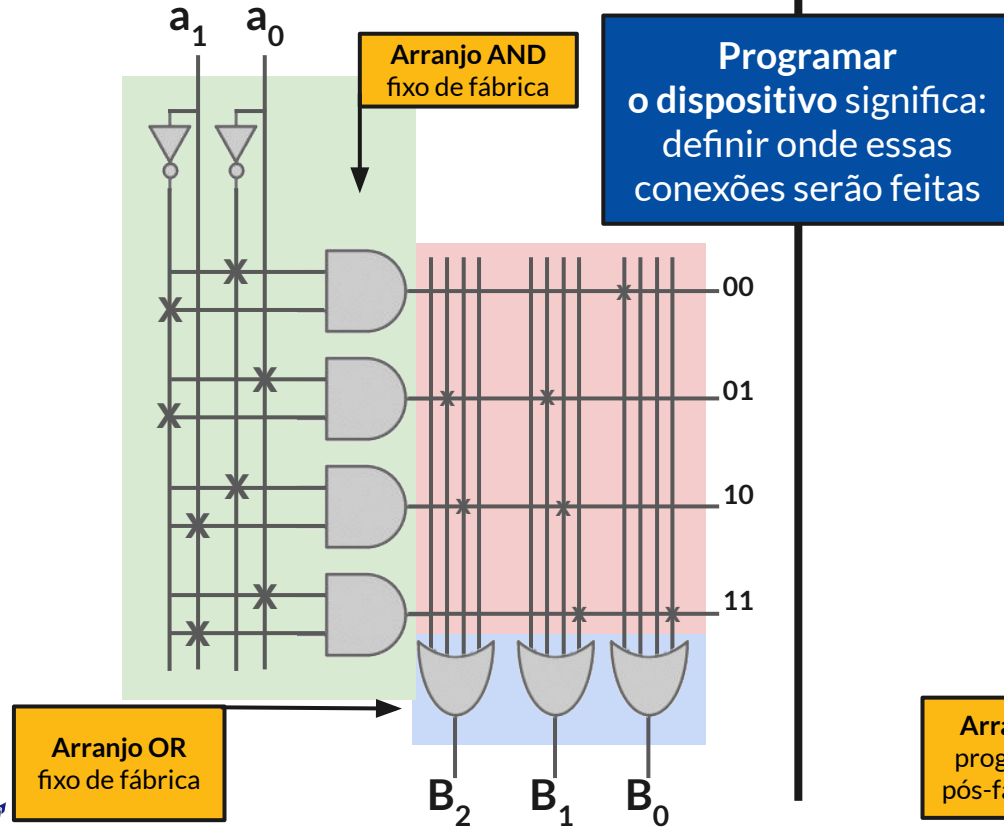


PROM - Programmable Read Only Memory

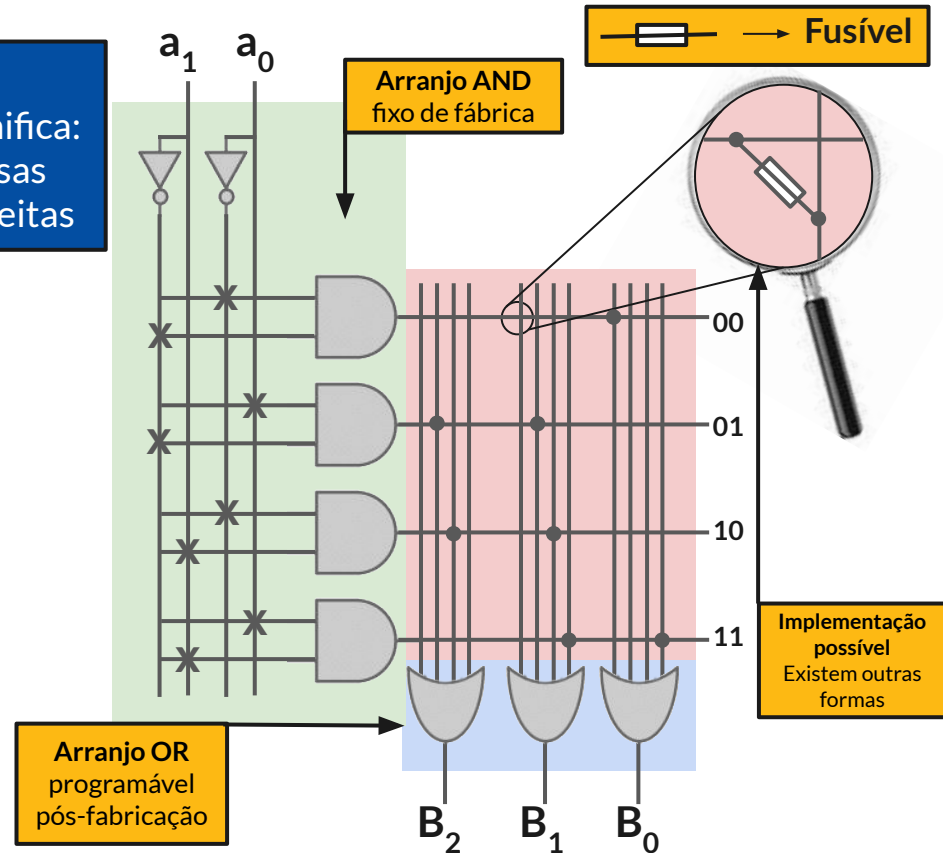


ROM x PROM: Estrutura interna

ROM - Read Only Memory



PROM - Programmable Read Only Memory



Desafio

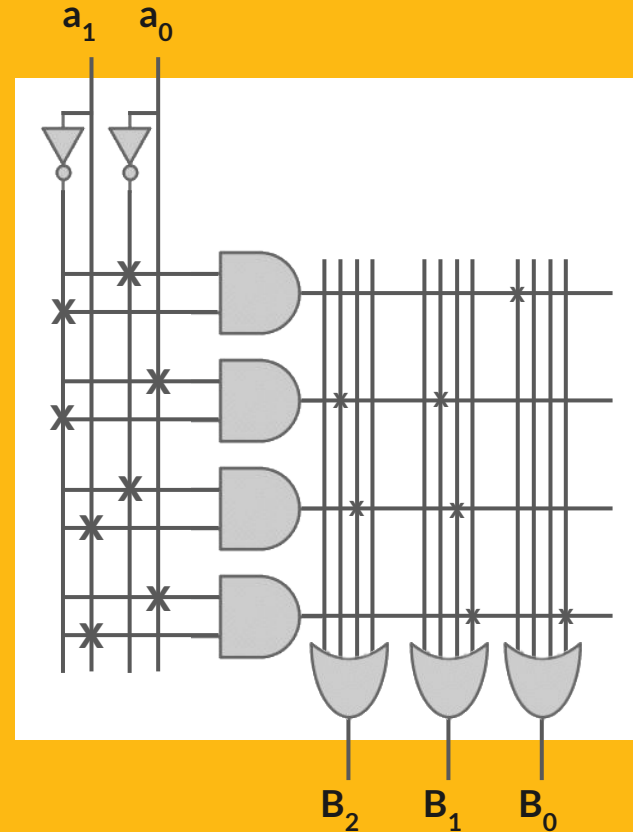
Quais as funções lógicas implementadas em:

$B_0 =$

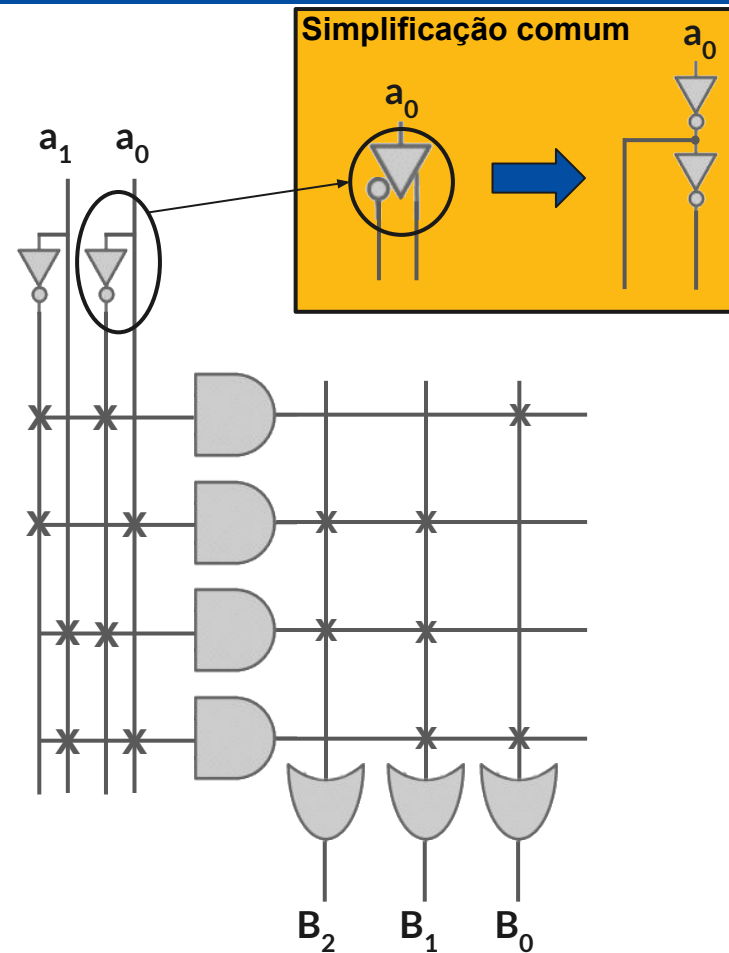
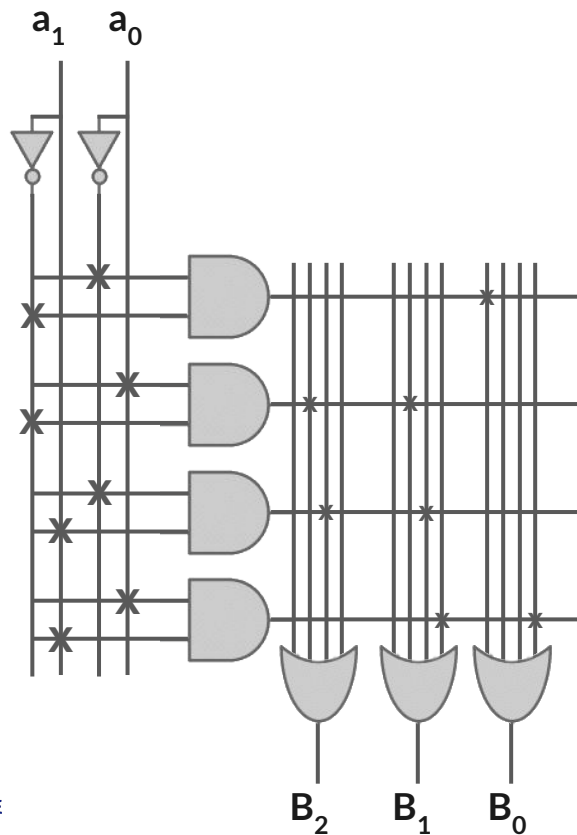
$B_1 =$

$B_2 =$

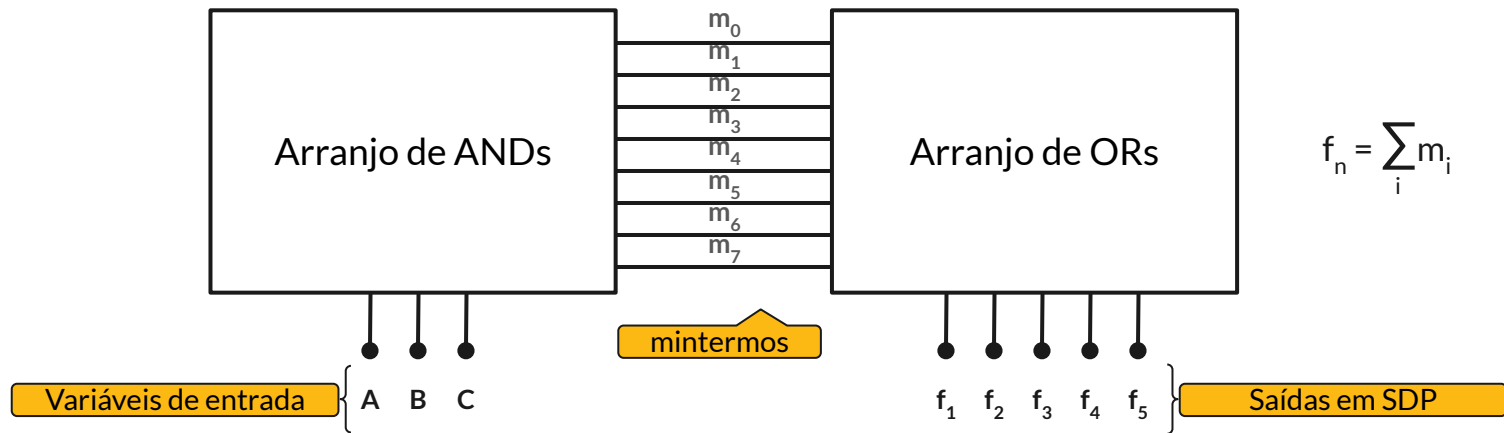
Acompanhe a resposta no quadro!



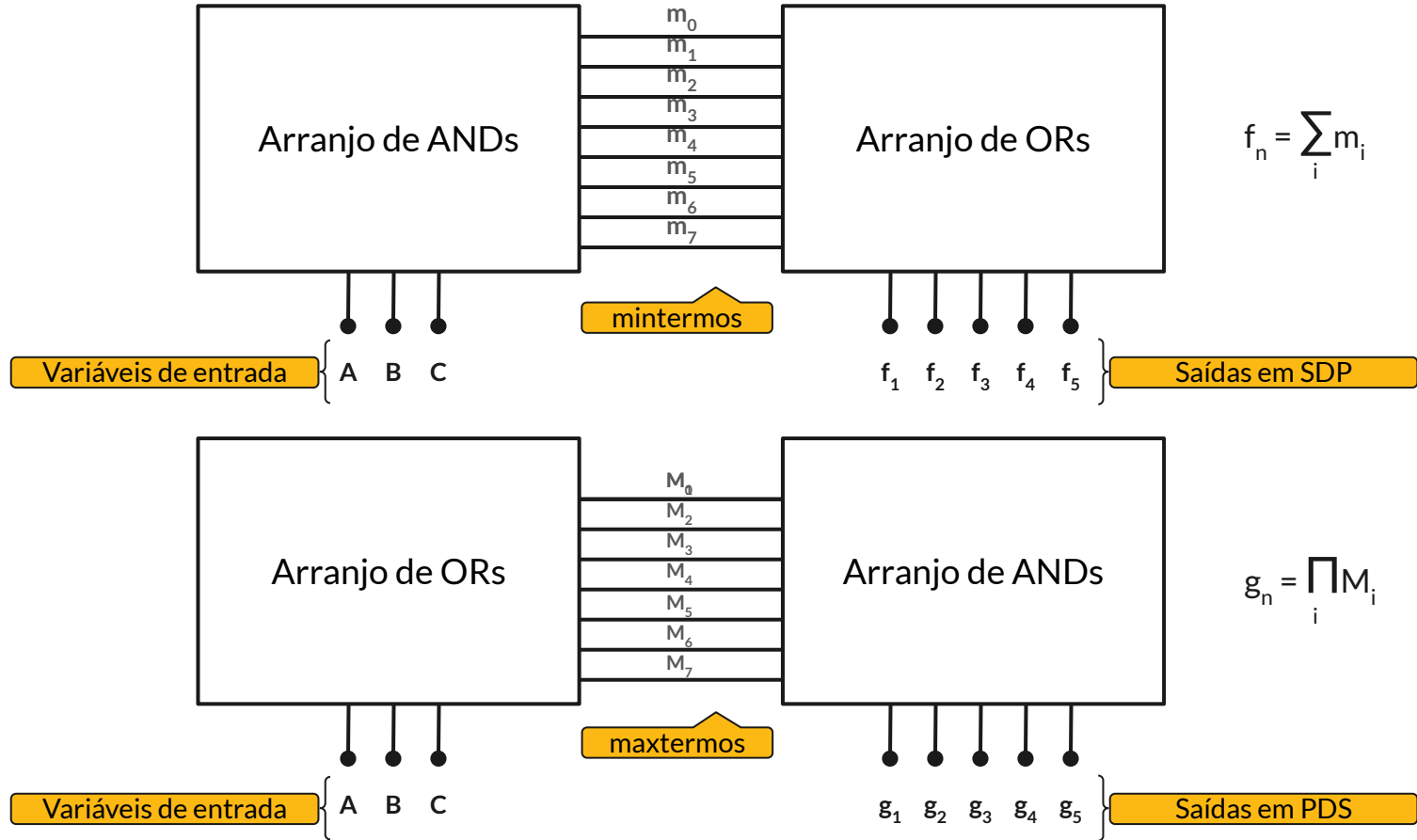
Vamos simplificar a representação?



Arranjos Lógicos



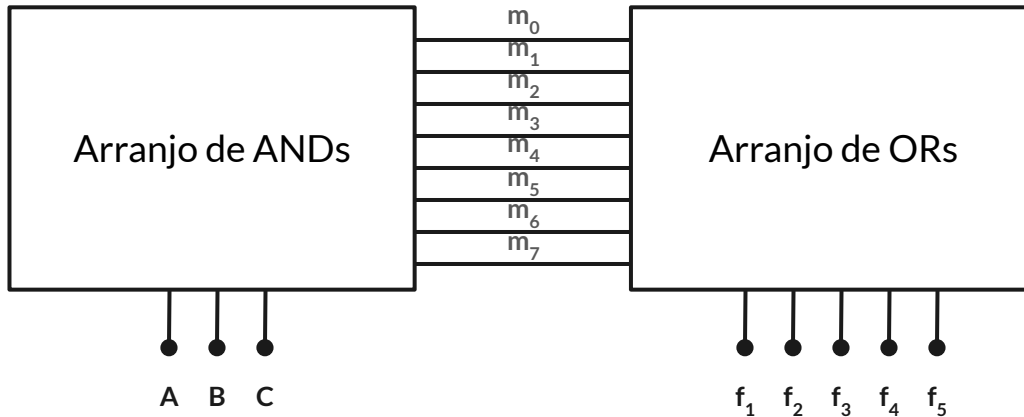
Arranjos Lógicos



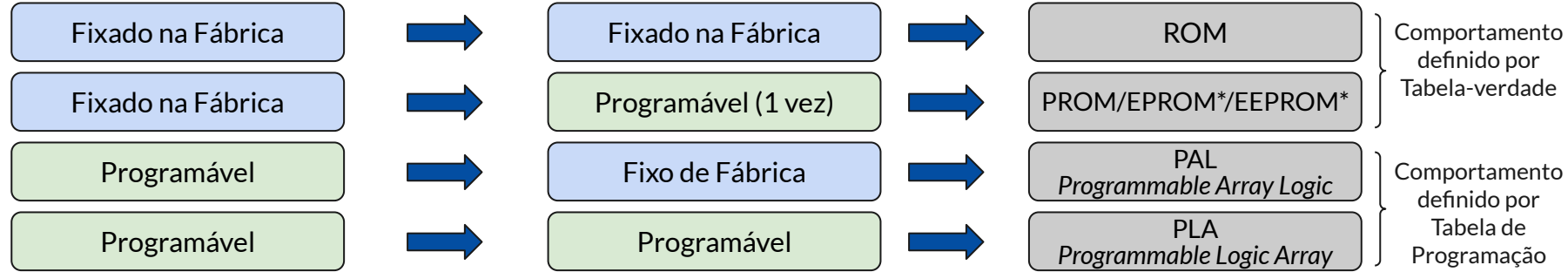
Arranjos Lógicos



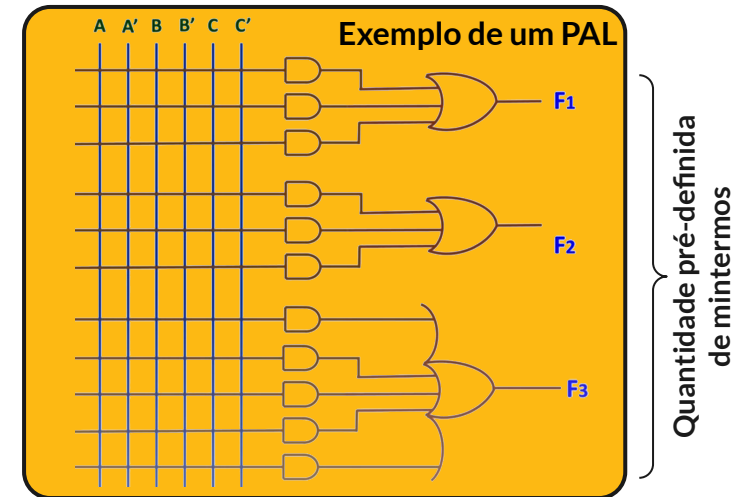
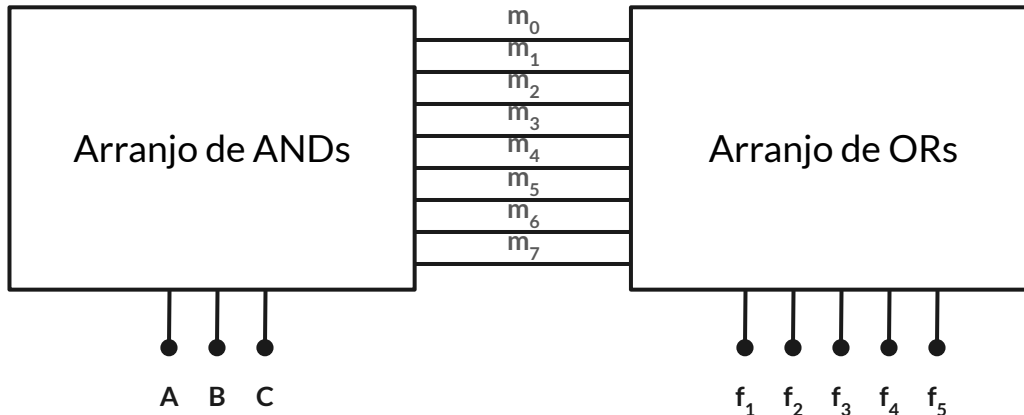
* EPROMs e EEPROMs permitem ciclos de reprogramação. Assume-se que são menos frequentes que as leituras, por isso ainda ROMs



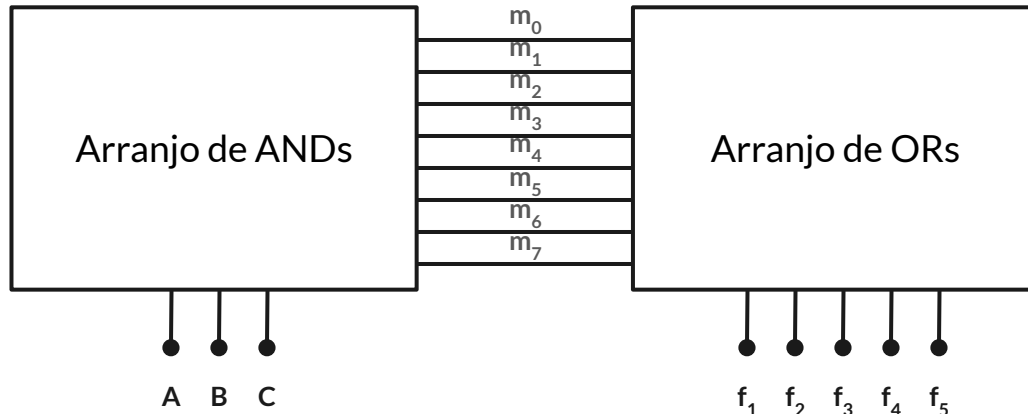
Arranjos Lógicos



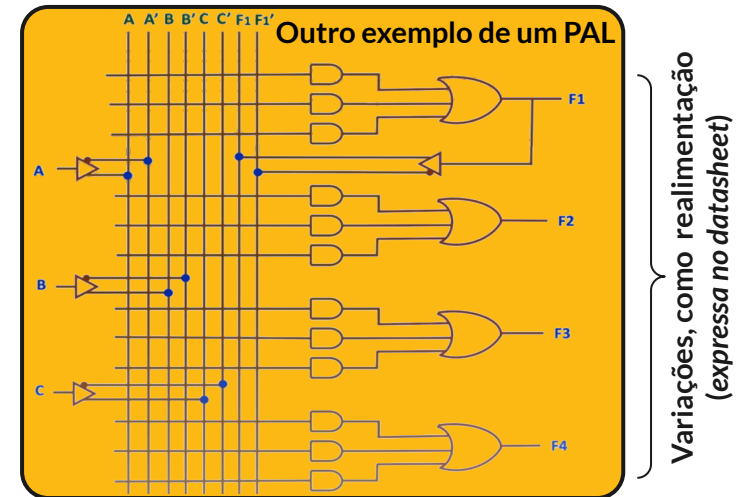
* EPROMs e EEPROMs permitem ciclos de reprogramação. Assume-se que são menos frequentes que as leituras, por isso ainda ROMs



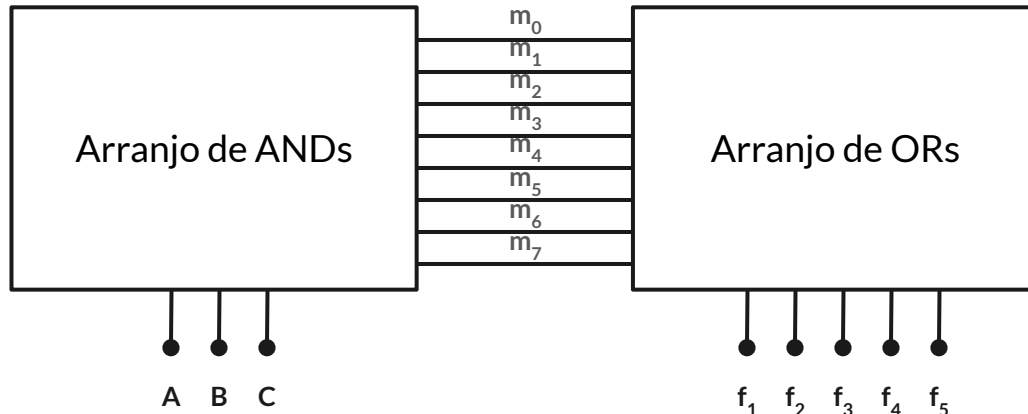
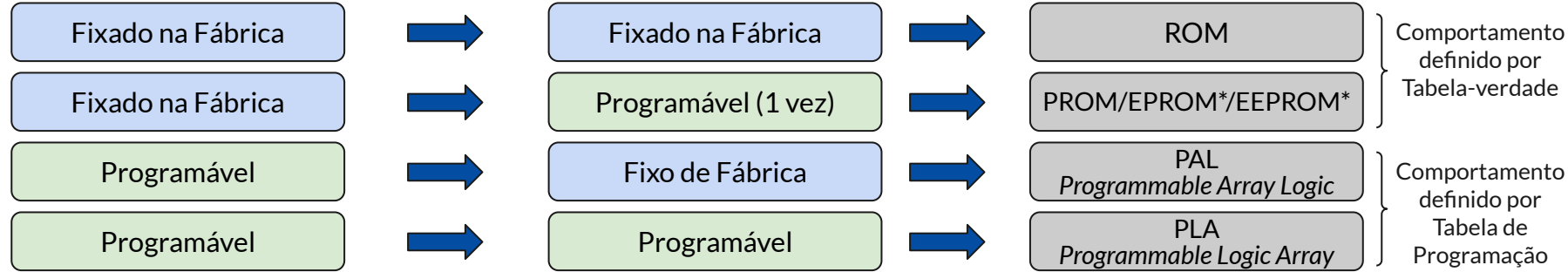
Arranjos Lógicos



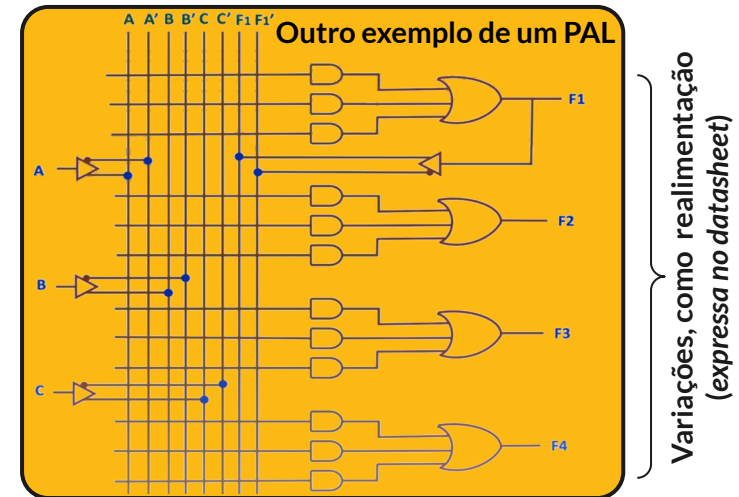
* EPROMs e EEPROMs permitem ciclos de reprogramação. Assume-se que são menos frequentes que as leituras, por isso ainda ROMs



Arranjos Lógicos



* EPROMs e EEPROMs permitem ciclos de reprogramação. Assume-se que são menos frequentes que as leituras, por isso ainda ROMs



Informação

Essas tecnologias ainda estão disponível para uso. Evite chamá-las de obsoletas.

Usá-las (ou não) é meramente uma decisão de projeto.

Às vezes os produtos de consumo que usamos as usa em sua implementação e nem sabemos.

- ▶ PROMs (e suas variações), PALs e PLAs
 - ▶ São chamados **Combinational** Programmable Logic Devices
 - ▶ Todos o circuitos estudados até agora são **Combinacionais**, assim chamados por não possuírem elementos de memória.
 - ***Aprofundaremos o conceito no futuro.***
 - ▶ Úteis para circuitos combinacionais pequenos.
- ▶ SPLD, CPLDs e FPGAs
 - ▶ Permitem a implementação de circuitos **Sequenciais**, assim chamados por possuírem elementos programáveis de memória.
 - ***Aprofundaremos o conceito no futuro.***
 - ▶ SPLD - *Simple (or Sequential) Programmable Logic Devices*
 - ▶ CPLD - *Complex Programmable Logic Device*
 - ▶ FPGA - *Field Programmable Gate Array*
 - ▶ Suportam maior complexidade. Daqui pra frente, focaremos nossas práticas em FPGA, usando uma HDL (*Hardware Description Language*) chamada **Verilog**.

Indicação de Leitura

Para conhecer melhor a estrutura dos CPLDs e FPGAs, leiam o artigo a seguir. Um clássico! Disponibilizei para vocês no AVA da disciplina.

S. Brown and J. Rose, "FPGA and CPLD architectures: a tutorial," in IEEE Design & Test of Computers, vol. 13, no. 2, pp. 42-57, Summer 1996, doi: 10.1109/54.500200.

Representação de funções booleanas

Dever de Casa

- 1) Como você implementaria um Somador Completo usando uma PROM 3x4
- 2) Hora trabalho: faça o Exercício Avaliativo disponibilizado no AVA

Referências



UYEMURA, J. P., Sistemas Digitais: uma abordagem integrada. Editora Thompson-Pioneira, Brasil, 2008

Broesch, J.D., Practical programmable circuits: a guide to PLDs, state machines, and microcontrollers. Academic Press. 2012

Hora-Trabalho de Hoje

Leia no capítulo 3 do livro do Uyemura as seções pertinentes. Responda às questões nos slides e na folha de exercícios disponível no AVA.

Dúvidas?

Na próxima aula...

Circuitos combinatórios e aritmética digital (somadores, subtratores, multiplicadores, divisores). Projeto Hierárquico.

Não falte! 😊

Obrigado pela atenção