

Prof. Me. Renato Alves Ferreira
email: renato.ferreira@fmu.br

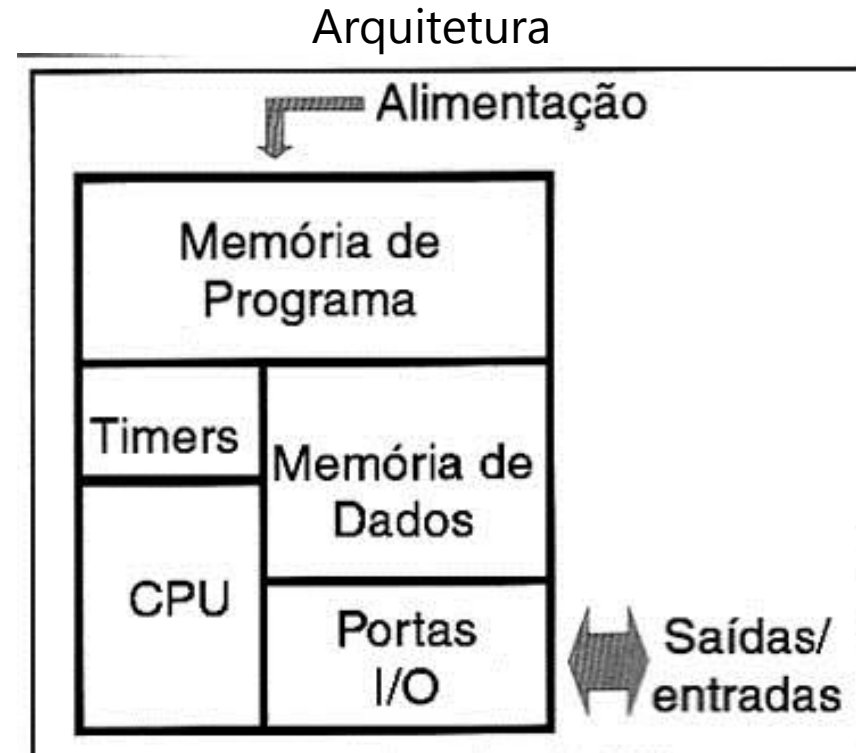
Disciplina:

MICROCONTROLADORES

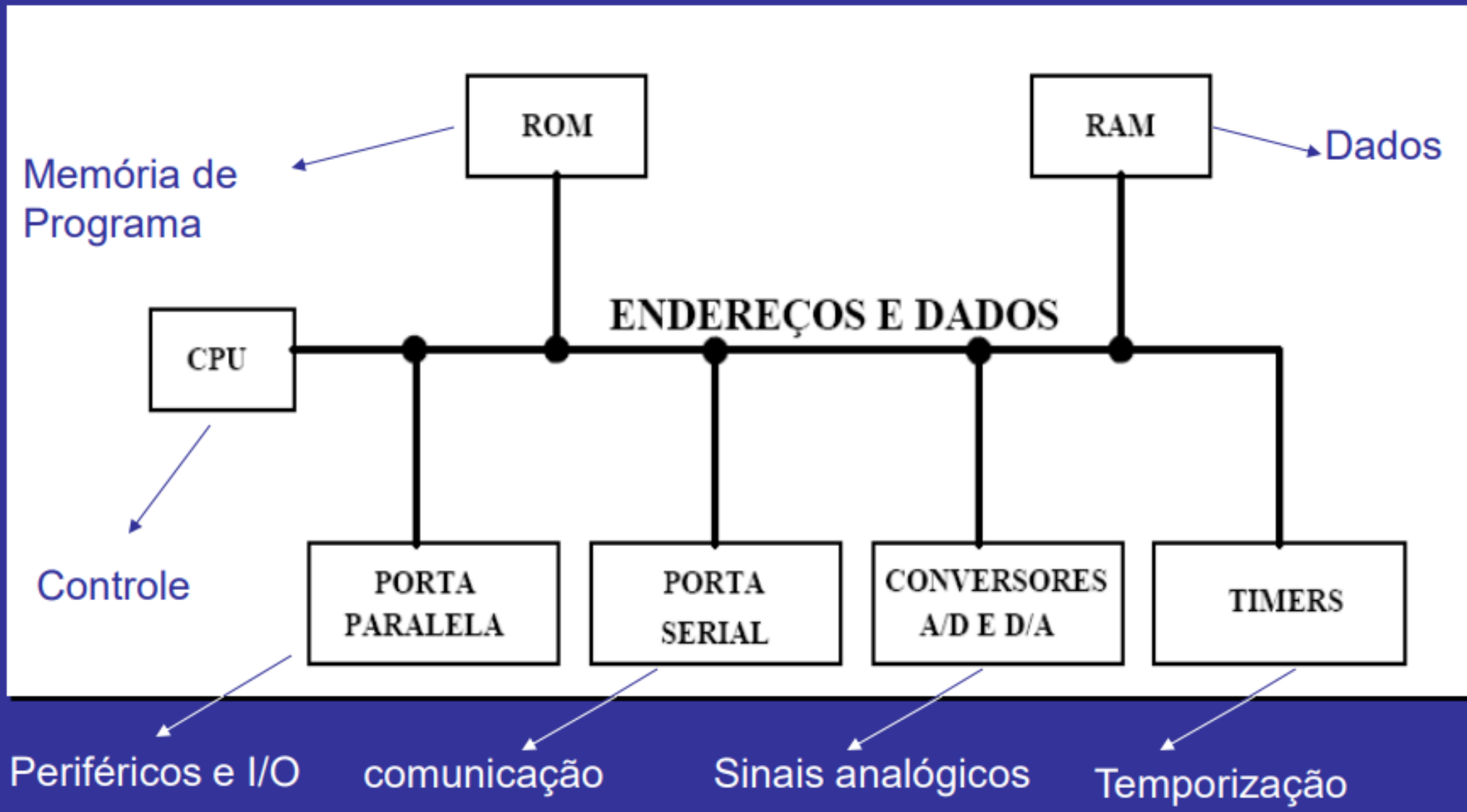
Introdução: Estudos sobre os microcontroladores, revisão / introdução ao Arduino e Atividades Práticas.

O que é um Microcontrolador

Microcontroladores são chips inteligentes que consistem num circuito processador que possui entradas, saídas e uma memória.

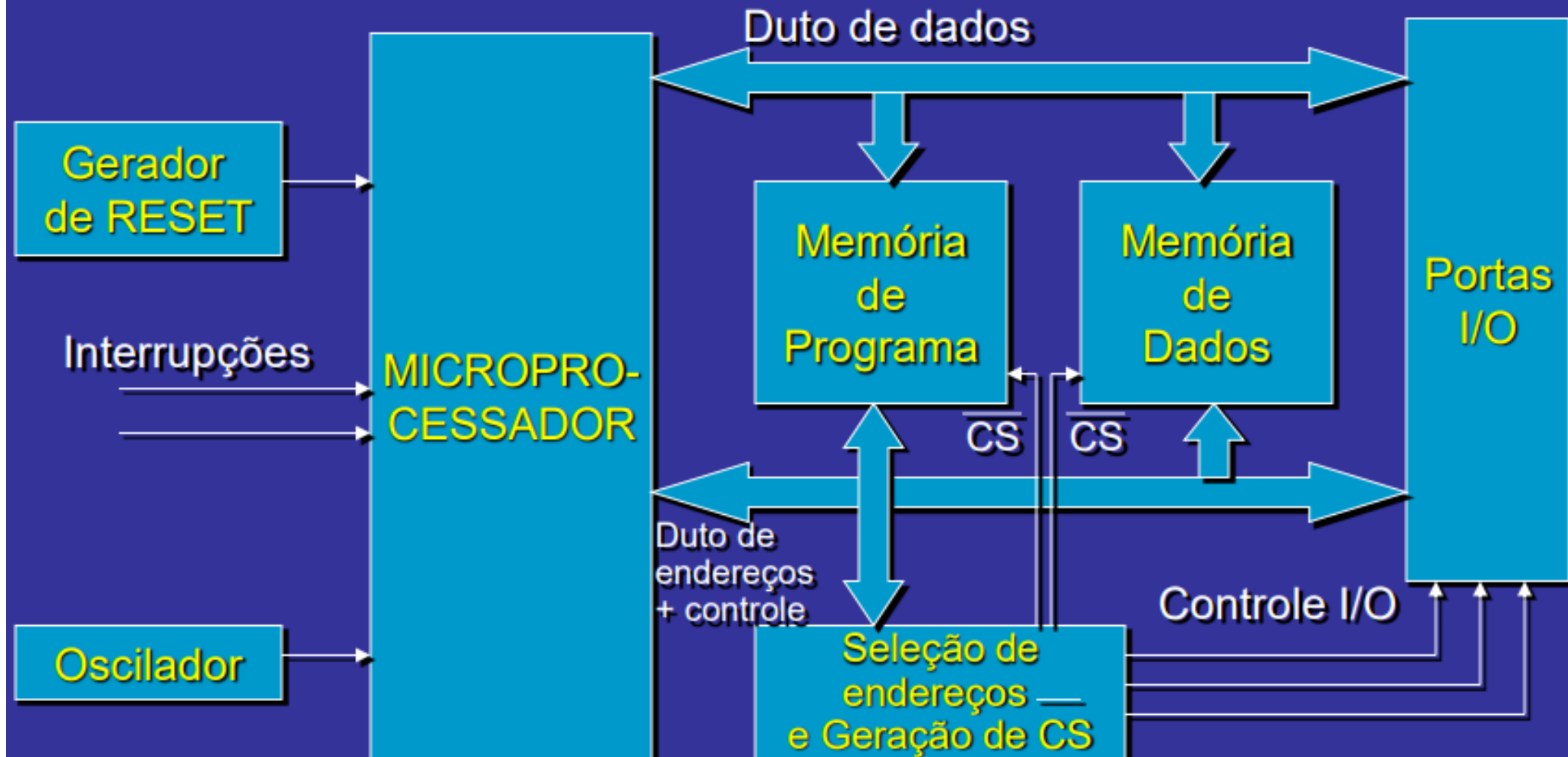


Exemplo típico da arquitetura de um microcontrolador



Microcontrolador de 8 Bits

(em um mesmo chip)



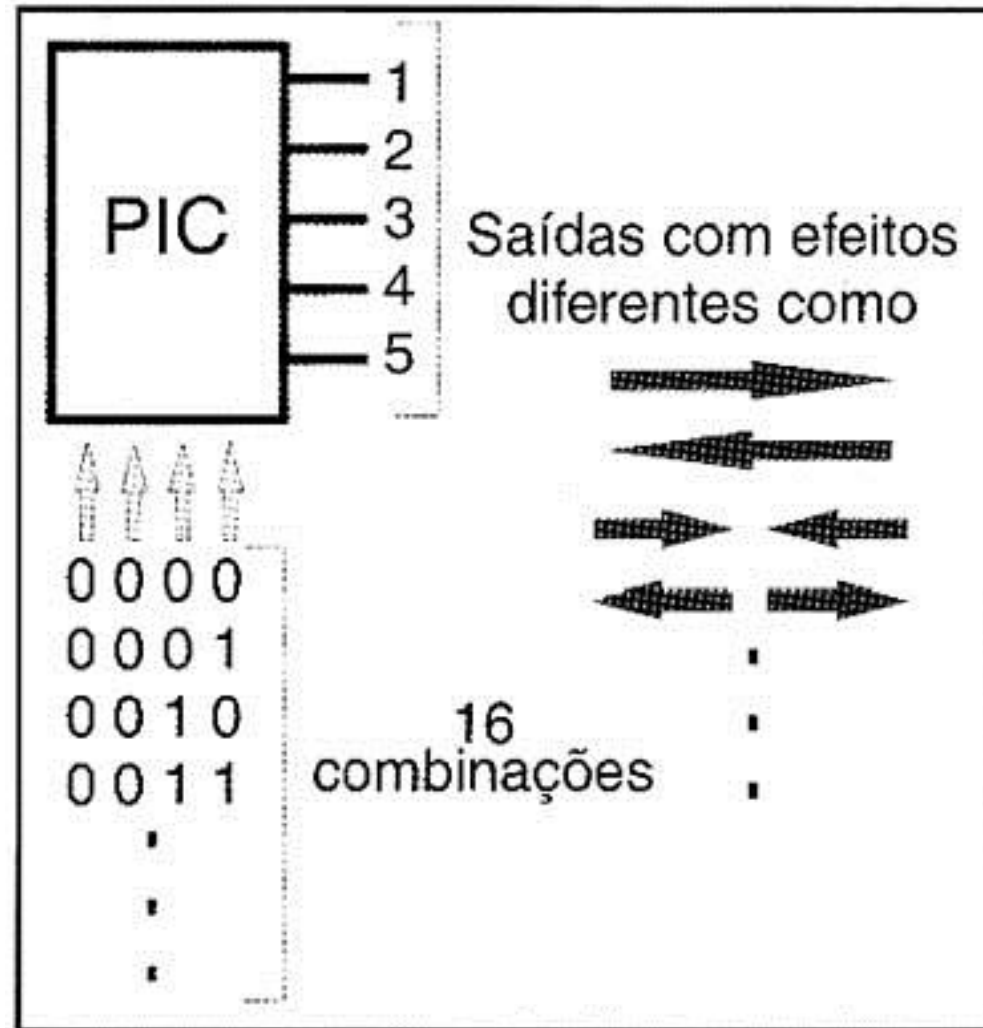
Entradas e saídas do Microcontrolador

O que o circuito do microcontrolador vai apresentar nas saídas (0 ou 1) depende do tipo de sinal que aplicamos nas entradas e(0 ou 1), orquestrado pelo programa que está gravado na sua memória.

Exemplo:

Se desejamos fazer um circuito sequencial que produza efeitos diferentes, programamos os diversos efeitos. Podemos combinar 4 entradas de modo que, com a combinação dos níveis lógicos tenhamos 16 efeitos diferentes. Veja na figura a seguir.

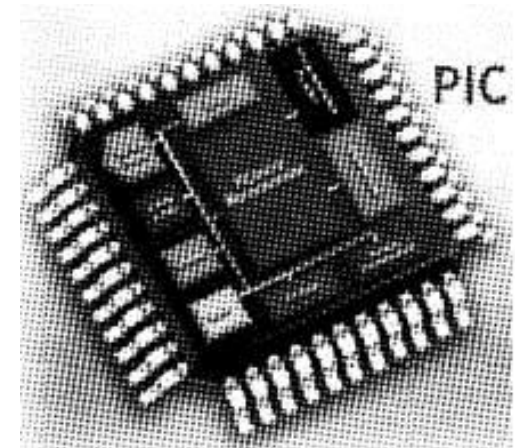
Entradas e saídas de um PIC



Diferenças entre Microcontroladores

O que diferencia os diversos tipos de microcontroladores é a sua capacidade de memória, podendo variar de algumas centenas de bytes a centenas de milhares de bytes, a arquitetura, a velocidade e a alimentação.

Não basta fazer o projeto e escolher um microcontrolador qualquer, é preciso escolher um microcontrolador que tenha as características exigidas pelo projeto.



Arquitetura dos Microcontroladores

São diversos circuitos que formam um microcontrolador com algumas variações entre eles. A forma como os diversos circuitos são interligados e ocorre seu funcionamento é chamada de Arquitetura do microcontrolador.

As arquiteturas podem variar entre RISC, CISC ou Híbridas.

RISC - Reduced Instruction Set Computer (Computador com Conjunto Reduzido de Instruções). Cada instrução pode ser executada em apenas um ciclo do clock.

CISC - Complex Instruction Set Computer (Computador com Conjunto de Instruções Complexas). São mais poderosos, porém mais lentos por precisarem de vários ciclos de clock para executar algumas instruções.

Comparativo entre Arquitetura RISC e CISC

	RISC	CISC
Set de Instruções	Pequeno	Grande
Tempo de execução	Pequeno	Grande
Capacidade de Processamento	Grande	Grande

Instruções e Clock

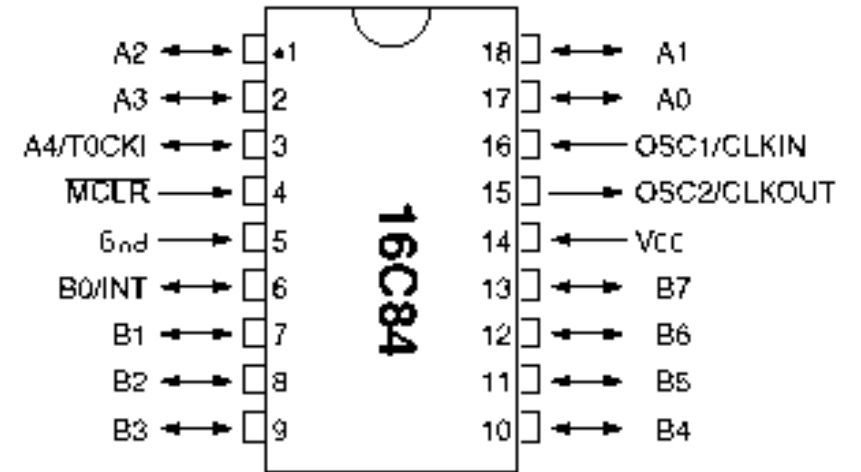
As **instruções** são as ordens que o microcontrolador obedece e em sua função executa alguma tarefa.

O **clock** é o ritmo de operação do microcontrolador. Um oscilador de clock determina a velocidade com que o microcontrolador opera.



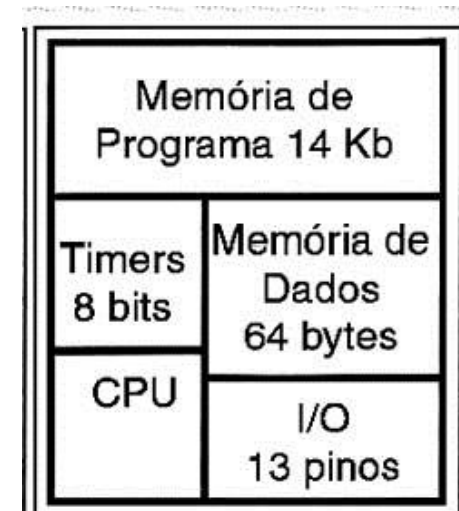
Exemplo de um PIC básico: PIC 16C84

Diagrama interno simplificado



Exemplo de um PIC básico: PIC 16C84

Detalhes sobre os 5 circuitos internos



- A **CPU** é o cérebro do microcontrolador, onde está o set de instruções que o microcontrolador reconhece.
- No **circuito de memória**, há uma parte reservada para as instruções do programa e para os dados.
- Os **timers** são usados para determina os tempos de execução das instruções. Exemplos: Quanto tempo desejamos que um sinal esteja presente numa determinada saída, ou de quanto em quanto tempo devem ser feitas as leituras de sensores ligados nas entradas.
- O **circuito I/O** faz o interfaceamento do microcontrolador com o mundo exterior.

Mais detalhes sobre Microcontroladores

Circuito de I/O – CMOS e TTL

Muitos microcontroladores possuem saídas com componentes CMOS e ou TTL.

Sendo **CMOS**, isso significa que podem tanto suprir como drenar **a mesma** corrente de saída. Por exemplo, com limites em torno de 10 mA. Isso é diferente dos dispositivos **TTL** em que se **pode drenar mais corrente** do que fornecer.

Os circuitos CMOS não consomem tanta energia quanto circuitos TTL. Com o baixo consumo exige menos capacidade da fonte, permitindo um projeto mais simples e mais barato. Porém, este consumo no circuito CMOS cresce em frequências altas quando comparado ao TTL.

Os componentes CMOS são mais propensos a sofrer dano por descarga eletrostática do que os componentes TTL.

Como gravar em um PIC – Programador/Gravador para PIC

O programador/gravador consiste numa placa com um soquete para se encaixar o PIC que será programado e um cabo conectado ao computador.



Atividade 1 – Tome nota dos principais pontos discutidos e pesquise o assunto para comparar com outros microcontroladores.

Atividade 2 – Atividade de Pesquisa - Simulador de Projetos de Circuitos Eletrônicos **Tinkercad**. Abra o PDF **Usando o Tinkercad** e tente criar o projeto com Arduino conforme o roteiro. Veja os próximos 6 slides para maiores informações

Atividade 3 - Prática com o Arduino no Tinkercad

Revisão para manipulação de **sinais de saída**.

Sugestão: Semáforo ou qualquer outro acionamento.

- Montar o projeto no protoboard (Tinkercad)
- Fazer o programa conforme o quadro ao lado

Quadro de configurações de sinais

Endereçamento e temporização		
Pinos de saída 8,9 e 10		
Pino	Led	Ação
8	Verde	3 segundos
9	Amarelo	1 segundo
10	Vermelho	5 segundos

Componentes utilizados para as atividades 2 e 3

Circuito Elétrico

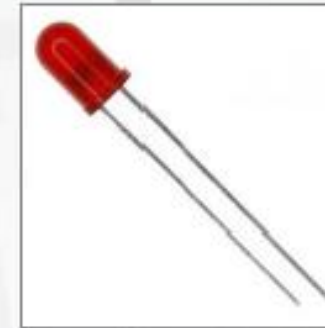
Utilize os seguintes componentes:

Arduino Uno



1 Unidade

LED



3 Unidades

Resistor 330 Ω



3 Unidades

Fio Jumper



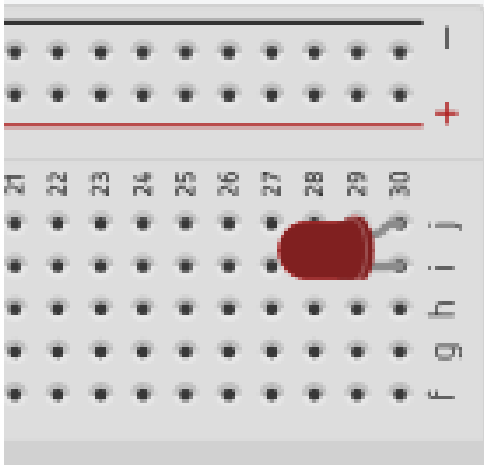
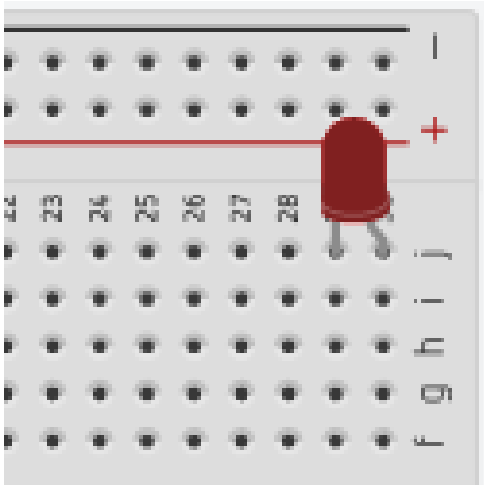
4 Unidades

Protoboard



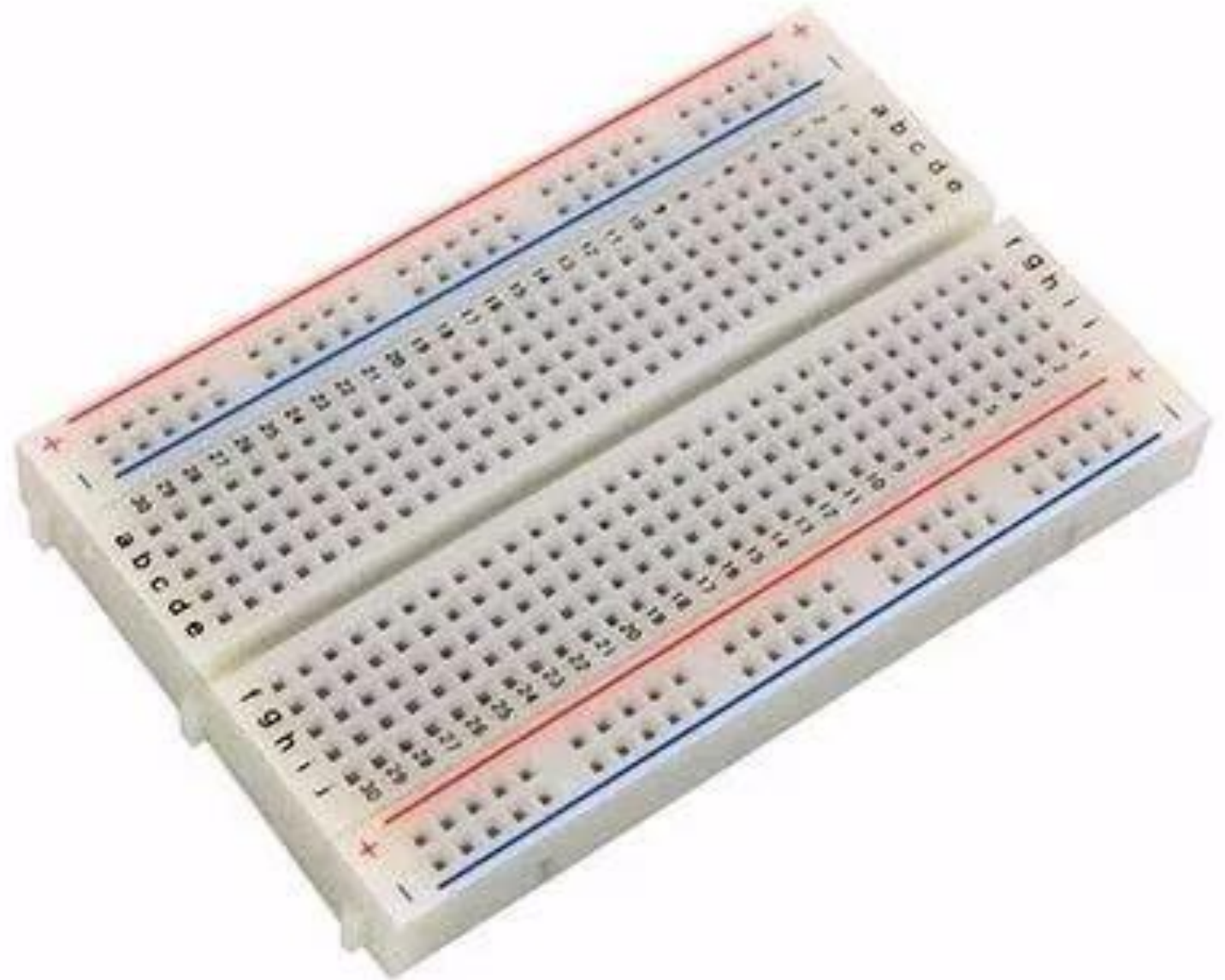
1 Unidade

Lembrete: Uso do protoboard



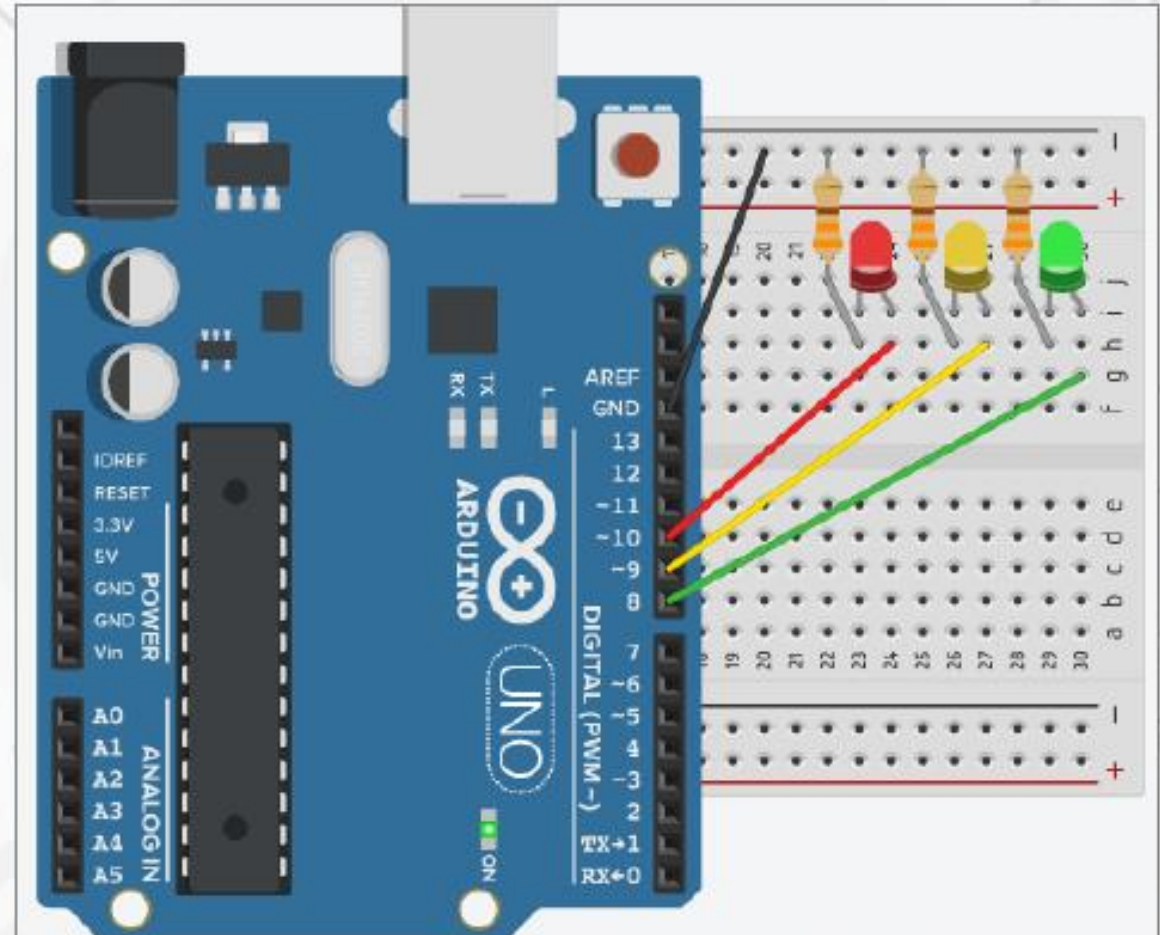
Teste você!

Monte um circuito simples qualquer em seu caderno. Posteriormente implemente no protoboard.



Exemplo de Ligação

- LED vermelho ficará acionado por 3 segundos;
- LED Amarelo ficará acionado por 1 segundo;
- LED verde ficará acionado por 5 segundos;



Exemplo de programa

```
// Exemplo de controle de sinal de saída

void setup() {
    pinMode(5, OUTPUT); // pino 5 como saída
}

void loop() {
    digitalWrite(5, 1); // Envia 1 para a porta
    delay(3000); // Aguarda 3 segundos

    digitalWrite(5, 0); // Envia 0 para a porta
    delay(2000); // Aguarda 2 segundos
}
```

Teste você!

Agora codifique o
programa de controle
do semáforo



sketch_dec07a

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```


Estudo sobre o microcontrolador 8051

Parte 1

O **Intel MCS-51**, conhecido como 8051, é um microcontrolador de 8 bits foi desenvolvido pela Intel nos anos 80, para uso em sistemas embarcados, mas outros fabricantes também o reproduziram.

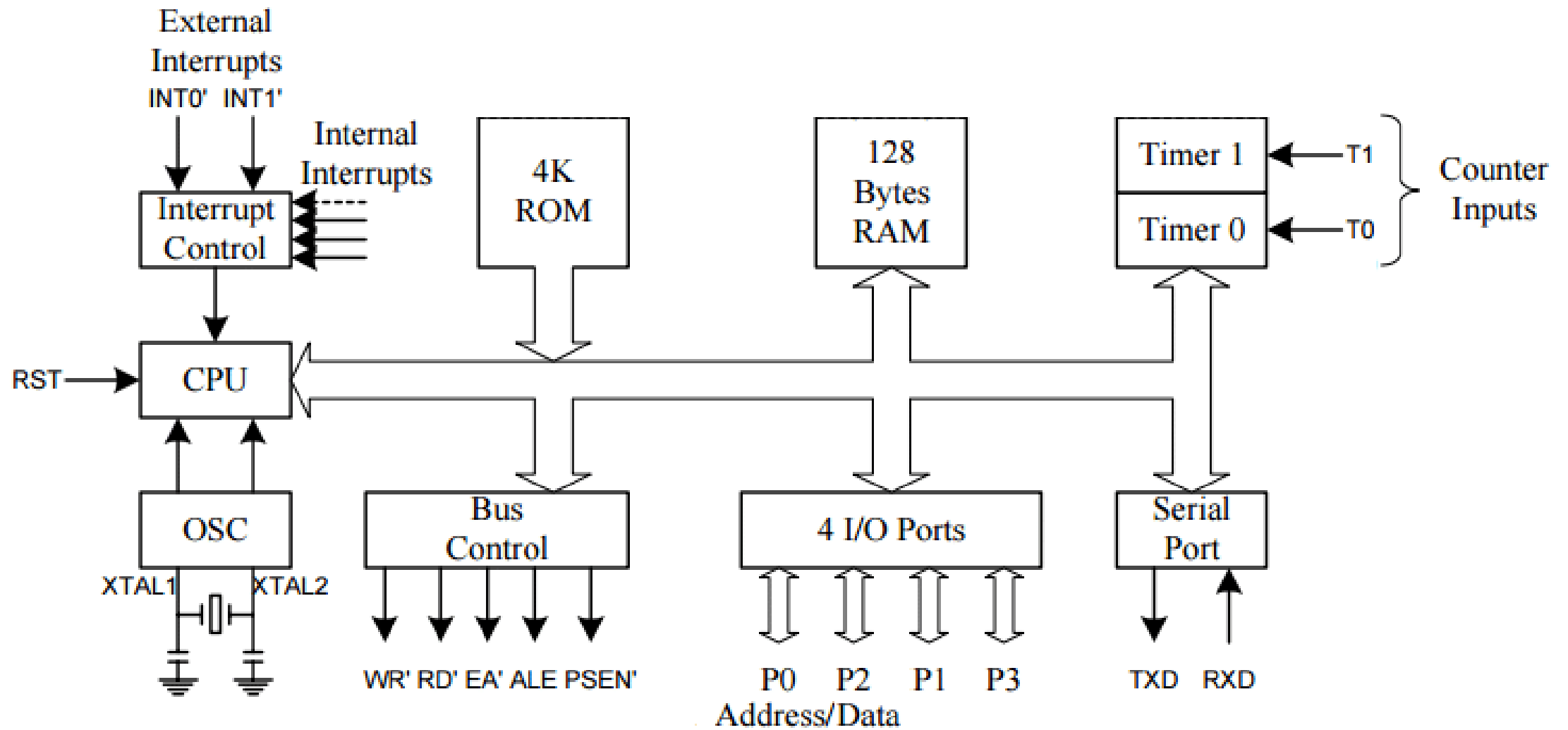
Principais características básicas de um dos modelos do 8051

- CPU de 8 bits;
- Encontrado com arquitetura Harvard e CISC;
- RAM - Memória de dados: 128 bytes;
- ROM - Memória de programa: 4KB;
- Endereçamento de 64KB para RAM e ROM externas;
- Mecanismo de interrupção;
- Quatro portas bidirecionais (input/output) de 8 bits;
- Dois Temporizadores/Contadores de 16 bits;
- Um canal de comunicação serial (UART full-duplex);
- Clock 4 a 30 Mhz / 1 a 120 Mhz

(T2) P1.0	1	40	VCC
(T2 EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
(MOSI) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)



Arquitetura interna do 8051.



Detalhes sobre as memórias no 8051 - Mapa de Memória

Como característica da arquitetura Harvard, instruções e dados são armazenados em espaços distintos. No 8051, a memória de programa pode ser acessada somente em modo de leitura. Além disso, o barramento de endereços permite acesso a memórias externas para dados e programa.

Memória de Programa

A memória de programa interna tem capacidade de armazenamento de 4KB. Essa memória pode ser expandida externamente, possibilitando acesso a 64KB.

Memória de Dados

A memória de dados interna é separada em dois segmentos: Low-RAM e High RAM. Essas duas seções de memória diferem no modo de acesso, sendo que a High-RAM não é padrão em todos os dispositivos. Além disso, os Registradores de Funções Especiais - SFR (do inglês, Special Function Registers) ocupam esse endereçamento. Assim como a memória de programa, os dados podem ser armazenados externamente.

Arquitetura e Organização do 8051 para conexões externas

O barramento de endereços do 8051 é de 16 bits, portanto pode gerar 65536 endereços.

O barramento de dados é de 8 bits e bidirecional, isso significa que a CPU pode realizar operações de leitura e escrita na memória.

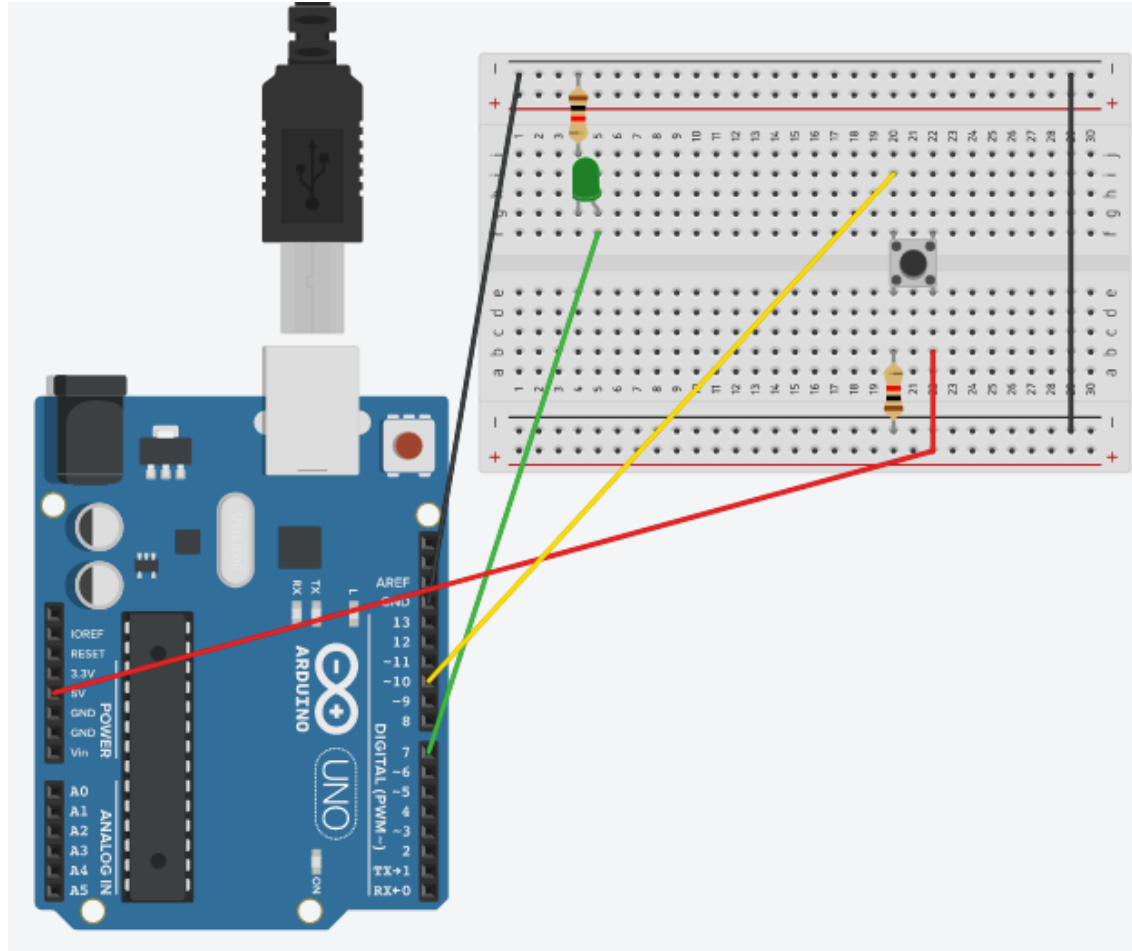
As Figuras as seguir mostram as conexões de um 8051 com o ambiente externo, onde é possível verificar ligações com os barramentos, memórias de dados e programas externos, dispositivos de I/O e atuação dos sinais de controle.

fonte: adaptada de

<https://www.embarcados.com.br/arquitetura-intel-8051/>

Laboratório : Atividade 4 no Arduino/Tinkercad

Entrada e saída de sinais via portas bidirecionais



Atividade no Tinkercad

1o) Acrescentar um led amarelo

2o) Alterar o acionamento da chave : Pressionou liga só o verde, soltou a chave, liga só o amarelo

Depois que testou.....

3o) Acrescentar mais uma chave ao projeto

4o) Ao pressionar a 2a chave, ligar os dois leds.

Estudo das Pinagem – Pinagens de controle

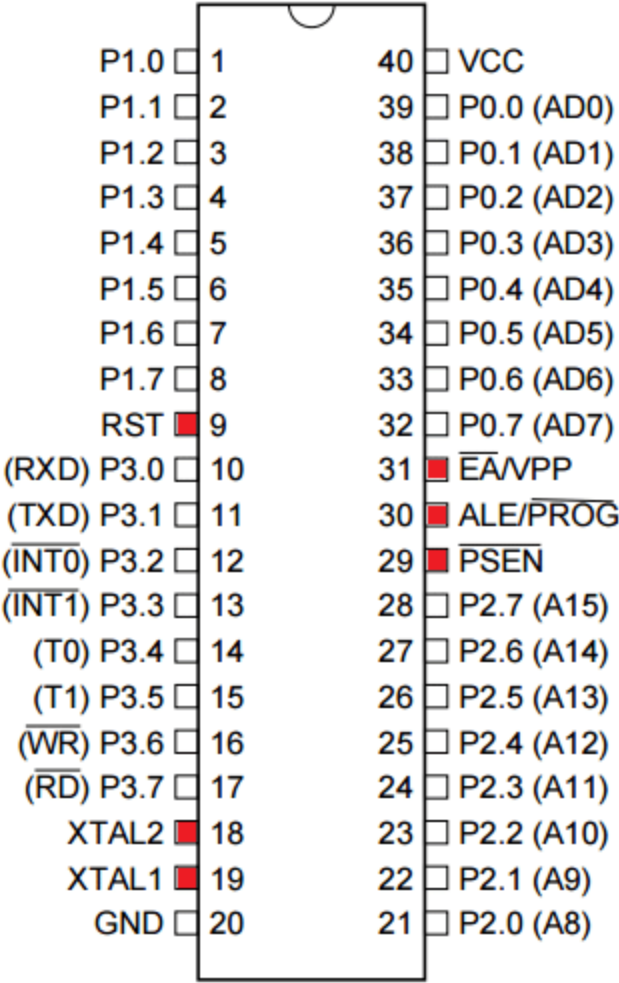
Os pinos em destaque fazem parte da interface do microcontrolador com as memórias externas.

- **EA** (External Access Enable): Determina se a memória de programa é interna ou externa;
- **ALE** (Address Latch Enable): Saída habilitadora do latch de endereços para acesso da memória externa;
- **PSEN** (Program Store Enable): Operação de leitura da memória de programa externa.

Além desses pinos temos as funções de Reset e Clock do sistemas:

- **RST**: Reset do microcontrolador;

XTAL1 e 2: Clock – Os circuitos devem operar sincronizados por um determinado sinal de frequência única, denominado Clock o qual determina quando realizar uma determinada operação. Apesar do 8051 possuir um gerador de clock interno, pode-se conectar um cristal de quartzo externo nesses pinos XTAL1 e XTAL2

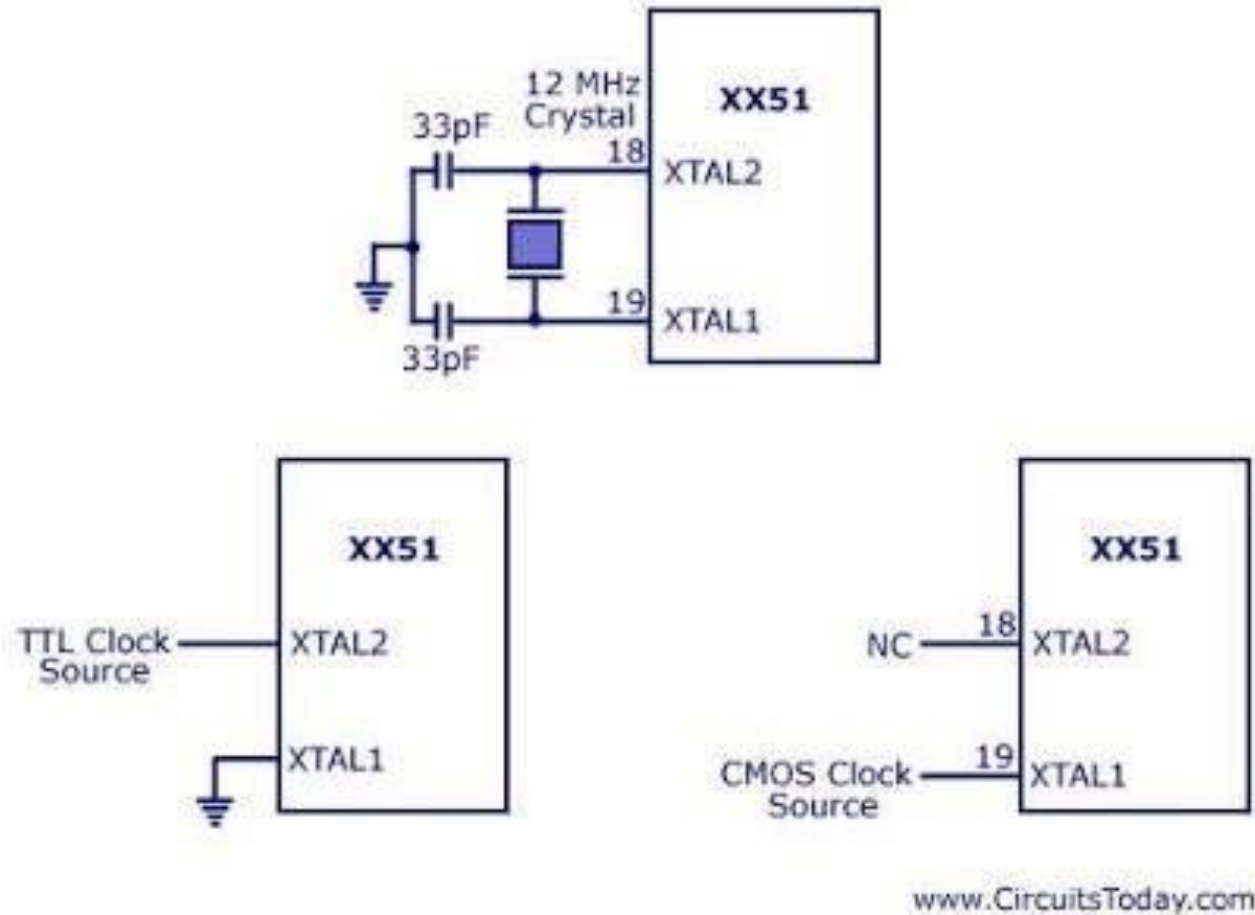


Pinos de Controle

Os pinos XTAL 1 e XTAL2

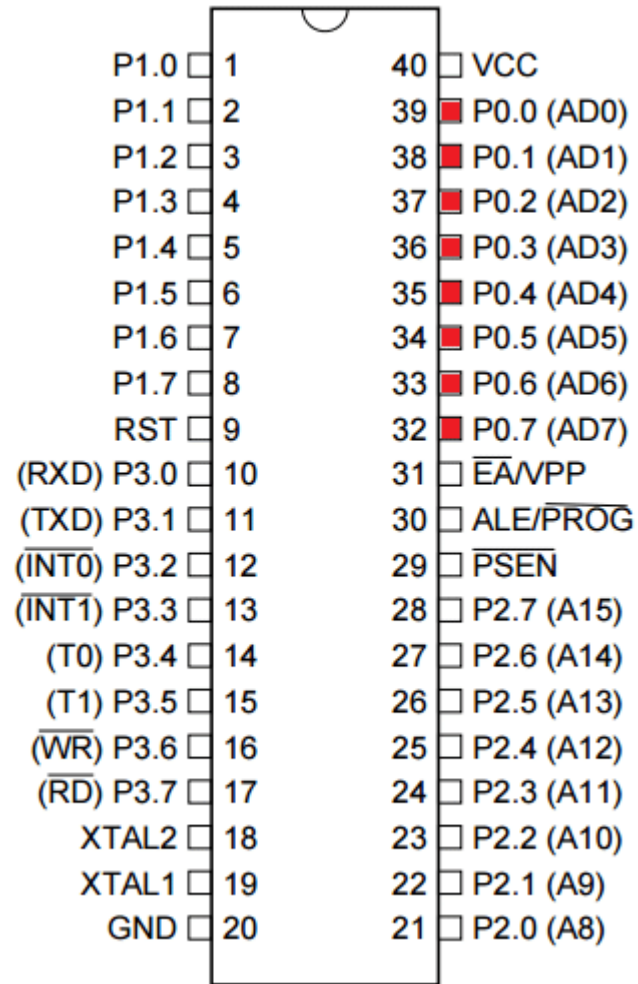
Esses dois pinos podem ser usados para conectar um cristal externo ao microcontrolador ou ter acesso ao clock interno do mesmo.

8051 Clock Circuit



Estudo das Pinagem – PORT 0

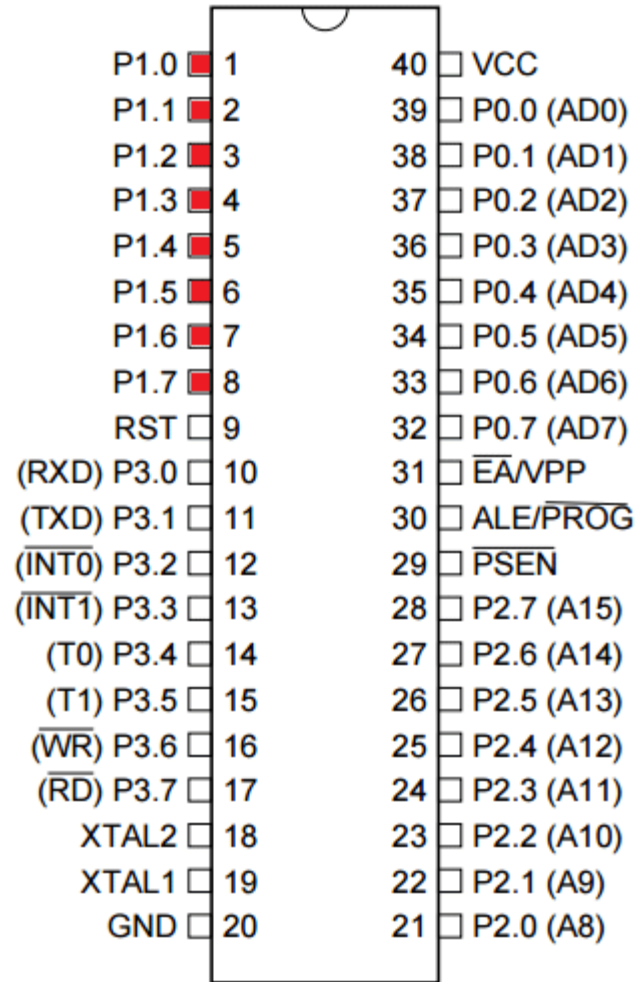
Porta de 8 bits bidirecional. Quando a memória externa é utilizada, essa porta determina os bits de endereço AD0~AD7. Além disso, é utilizada como interface para o barramento de dados.



Porta P0 do 8051

PORT 1

Porta bidirecional de 8 bits de propósito geral. Os pinos possuem pull-up interno.



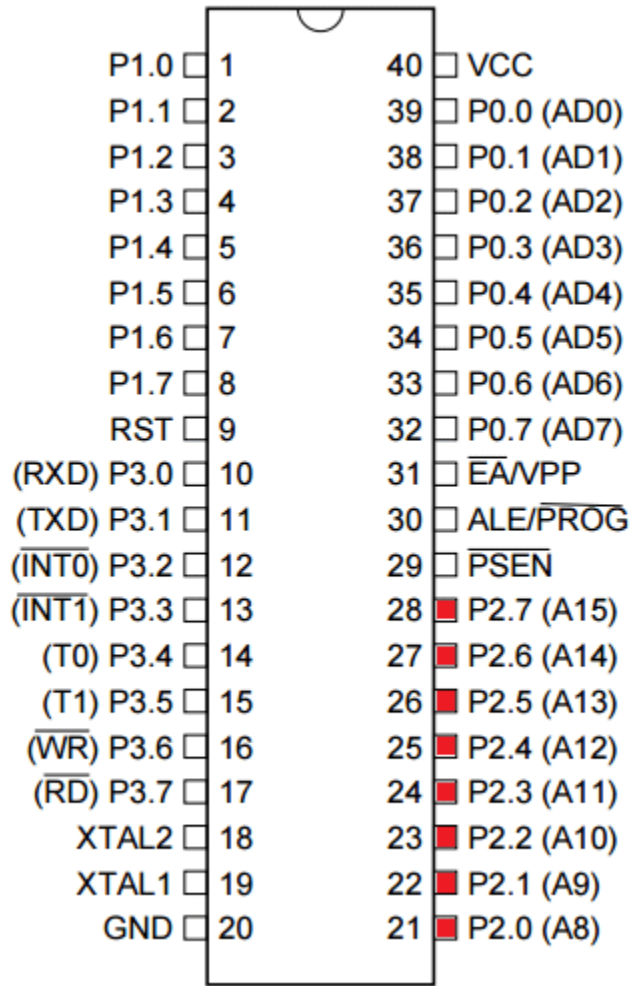
Porta P1

Nota

O pull-up ou pull-down é basicamente um resistor que fica ligado junto ao sinal que você deseja ler. Ele serve para garantir que um determinado sinal, ou uma determinada tensão, seja lida enquanto o pino não recebe nenhum sinal.

Um conceito importante é o de “pino flutuando”, que é o estado que o pino fica sem receber nada. Isso ocorre, pois o pino não está ligado a nada e não há como saber qual sinal ele está recebendo, se é sinal alto (2~5v) ou sinal baixo (0~1v).

Porta de 8 bits bidirecional com pull-up interno. Essa porta também é utilizada quando o acesso à memória externa requer um endereço de 16 bits. Nesse caso, não deve ser utilizada como I/O.

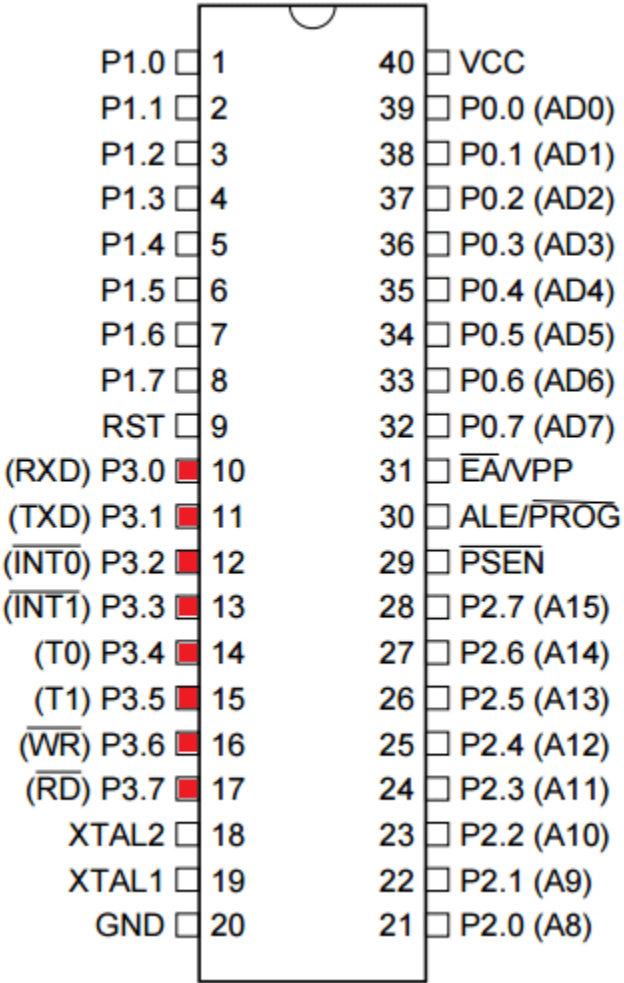


Porta P2 do 8051.

PORT 3

Porta de 8 bits bidirecional com pull-up interno. Essa porta também é utilizada para funções especiais da família 8051:

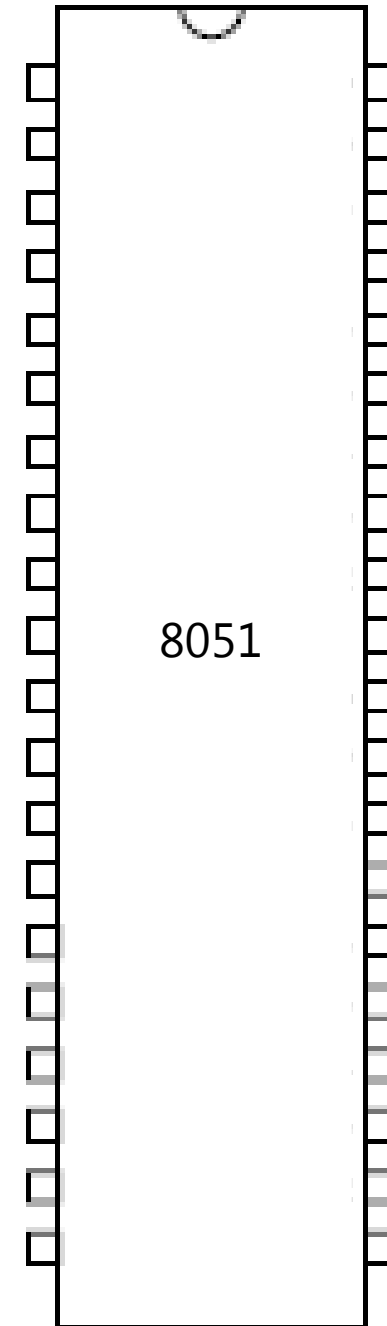
- RXD e TXD: Transmissor/Receptor de dados da UART;
- INT0: Interrupção externa 0;
- INT1: Interrupção externa 1;
- T0: Entrada do Timer0;
- T1: Entrada do Timer1;
- WR e RD: Sinais de controle para acesso da memória de dados externa.



Porta P3 do 8051.

Atividade 4

Desenhe o microcontrolador com toda a pinagem, com sua numeração, sigla ou nome e descreva todas as funções de cada pino, de acordo com as características estudadas do 8051.



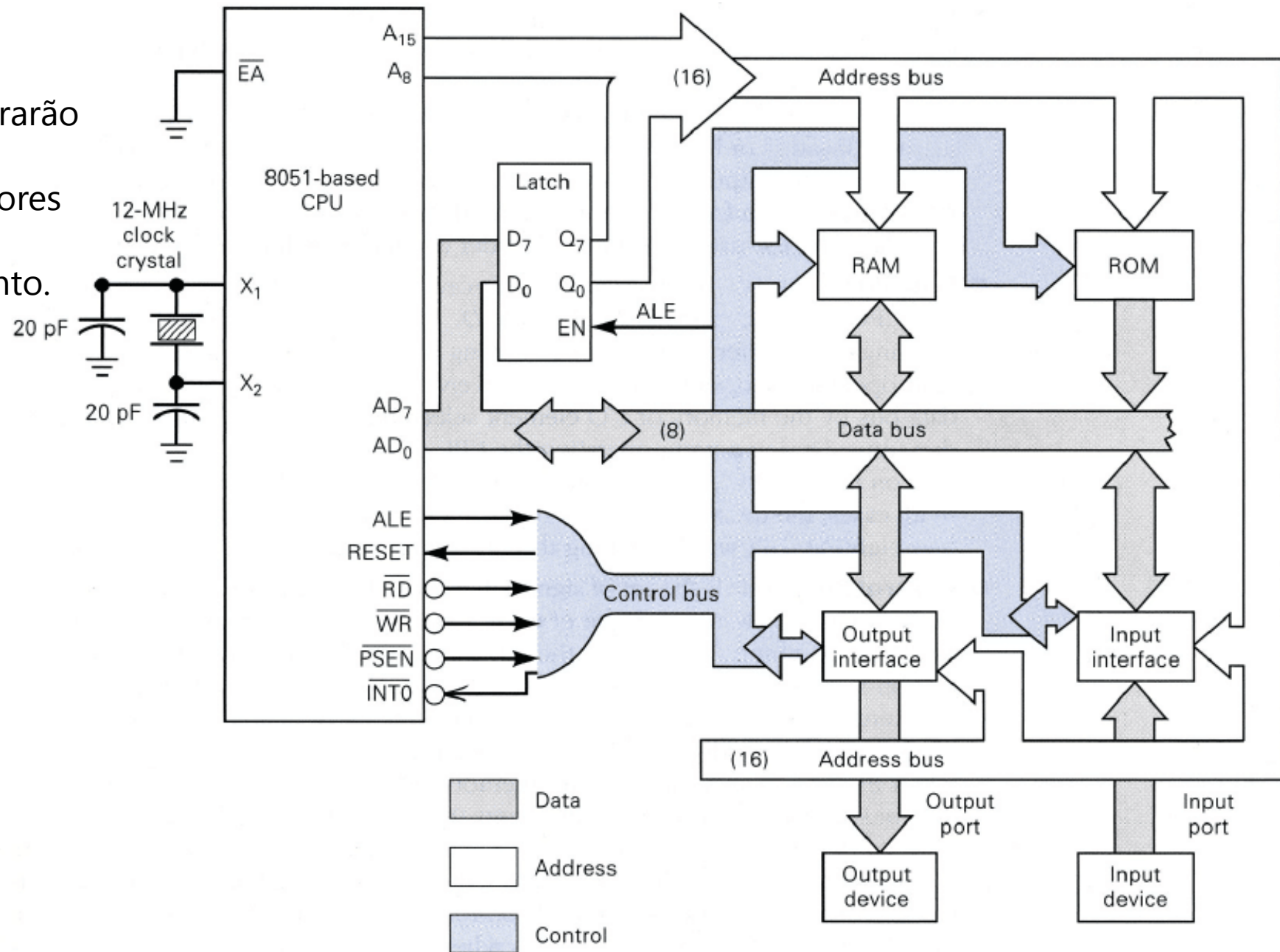
Laboratório : Atividade no Arduino/Tinkercad

Estudo sobre o microcontrolador 8051

Parte 2

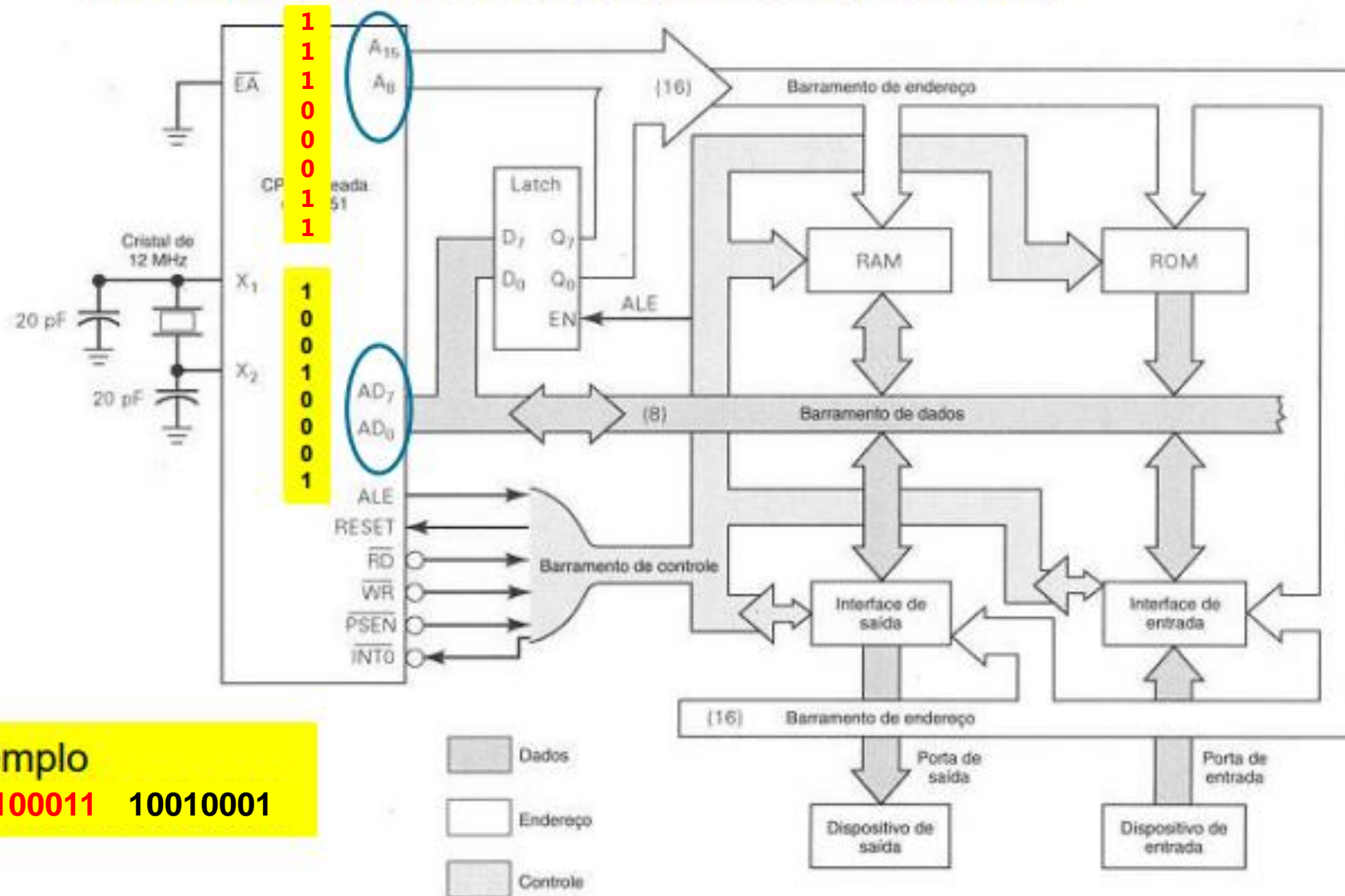
Visão Geral

Os slides seguintes mostrarão algumas partes com maiores detalhes do funcionamento.



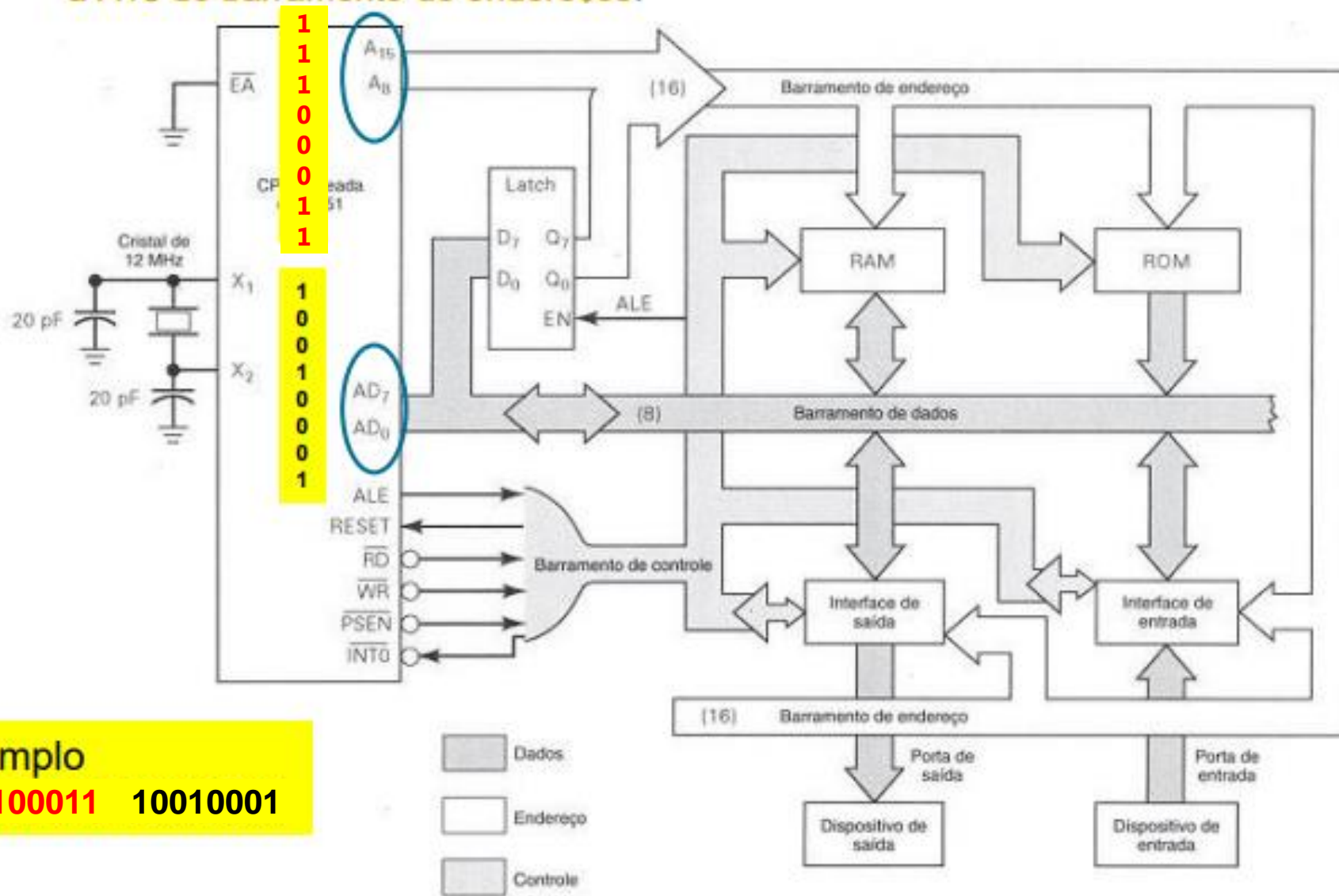
■ Barramento de endereços

- É unidirecional (16 BITS), porque a informação flui apenas em uma direção, da CPU para a memória ou para os elementos de E/S.



■ Barramento de endereços

- Quando a CPU quer ler de ou escrever em uma certa posição de memória ou dispositivo de E/S, ela coloca um código de endereço de 16 bits nos pinos de saída A0 a A15 do barramento de endereços.

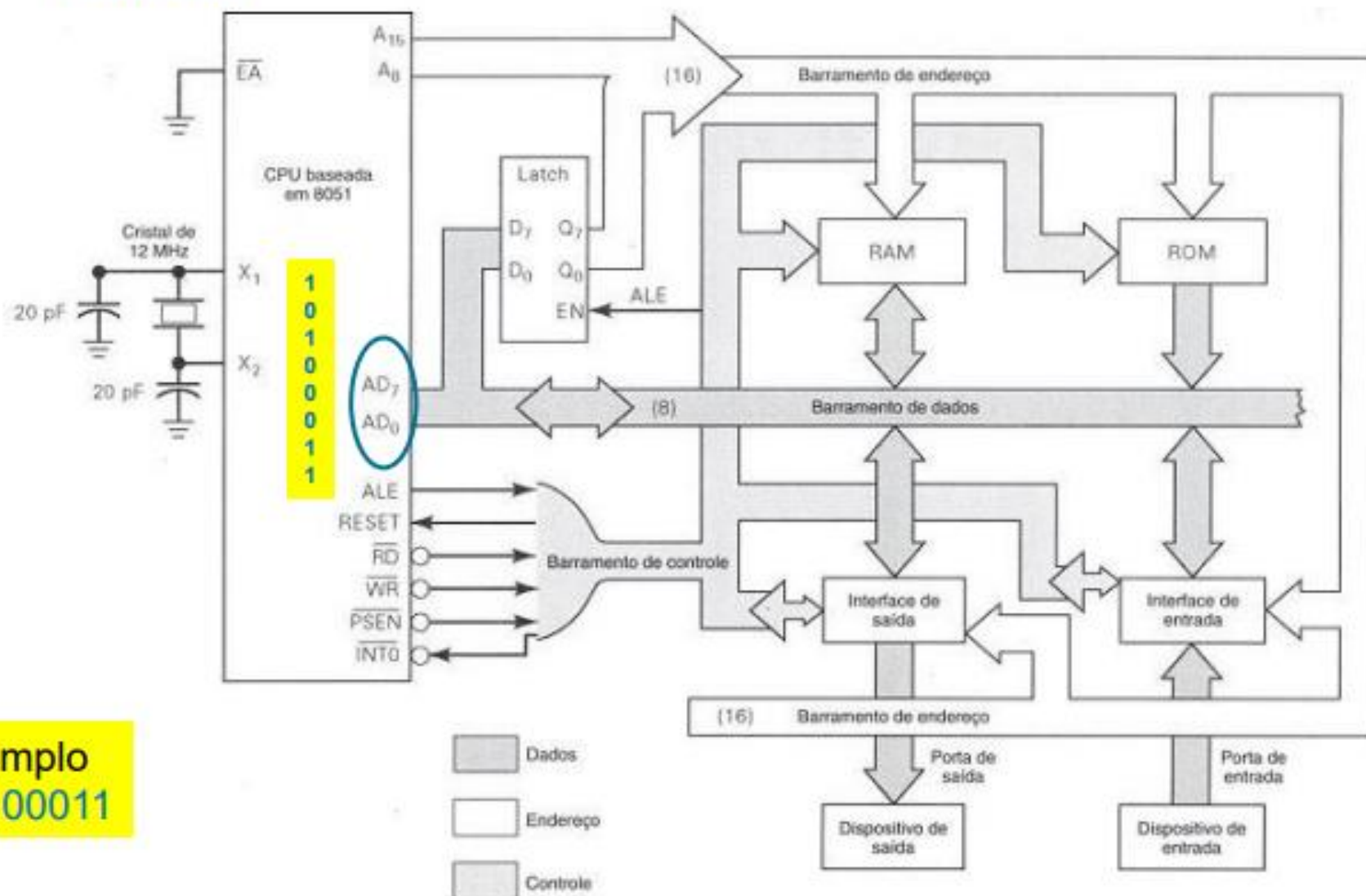


exemplo

11100011 10010001

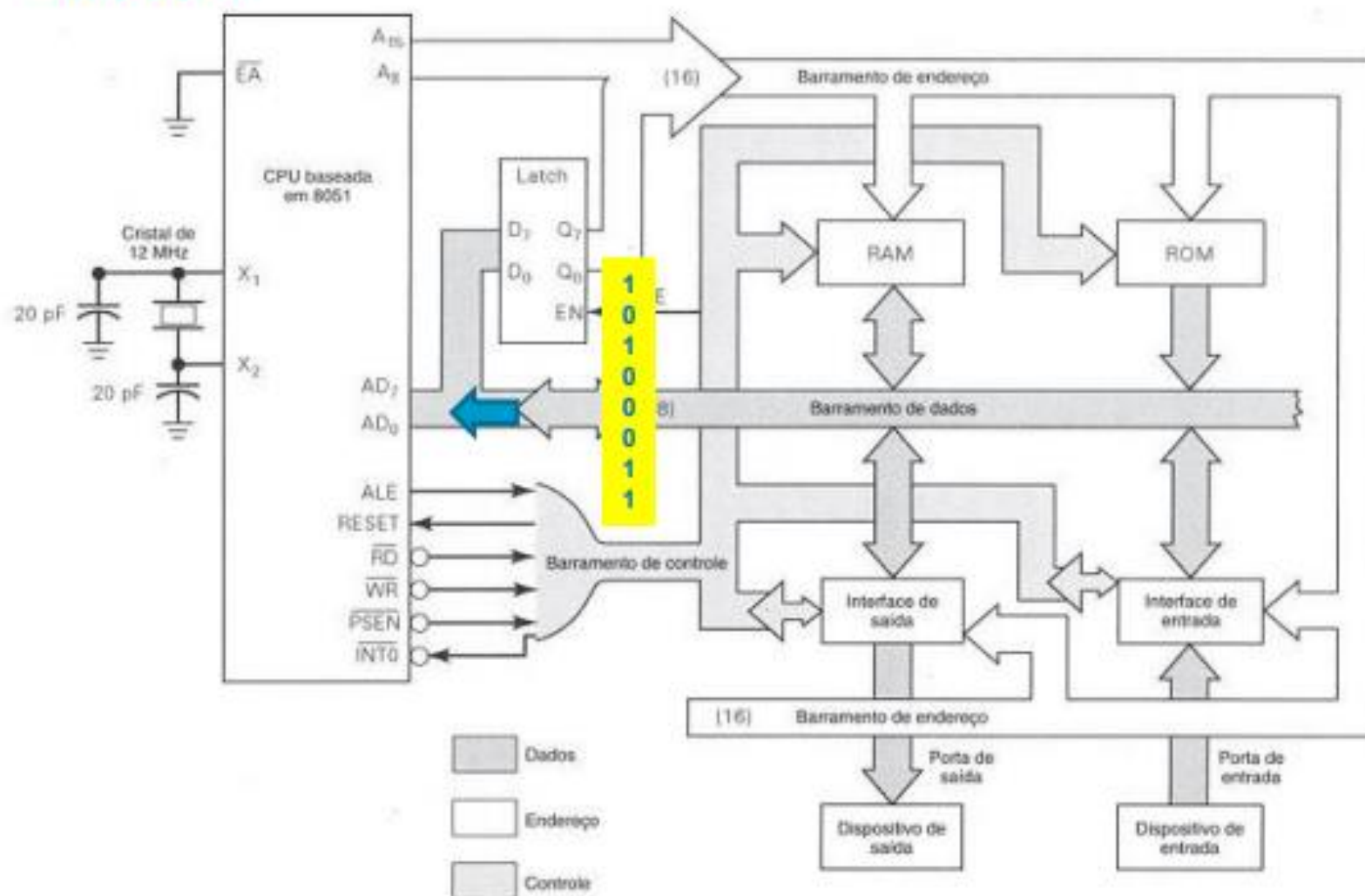
■ Barramento de dados

- Os oito pinos de dados da CPU, pinos AD0 a AD7 podem ser tanto entradas quanto saídas (bidirecional), dependendo se a CPU está realizando uma operação de leitura ou de escrita.



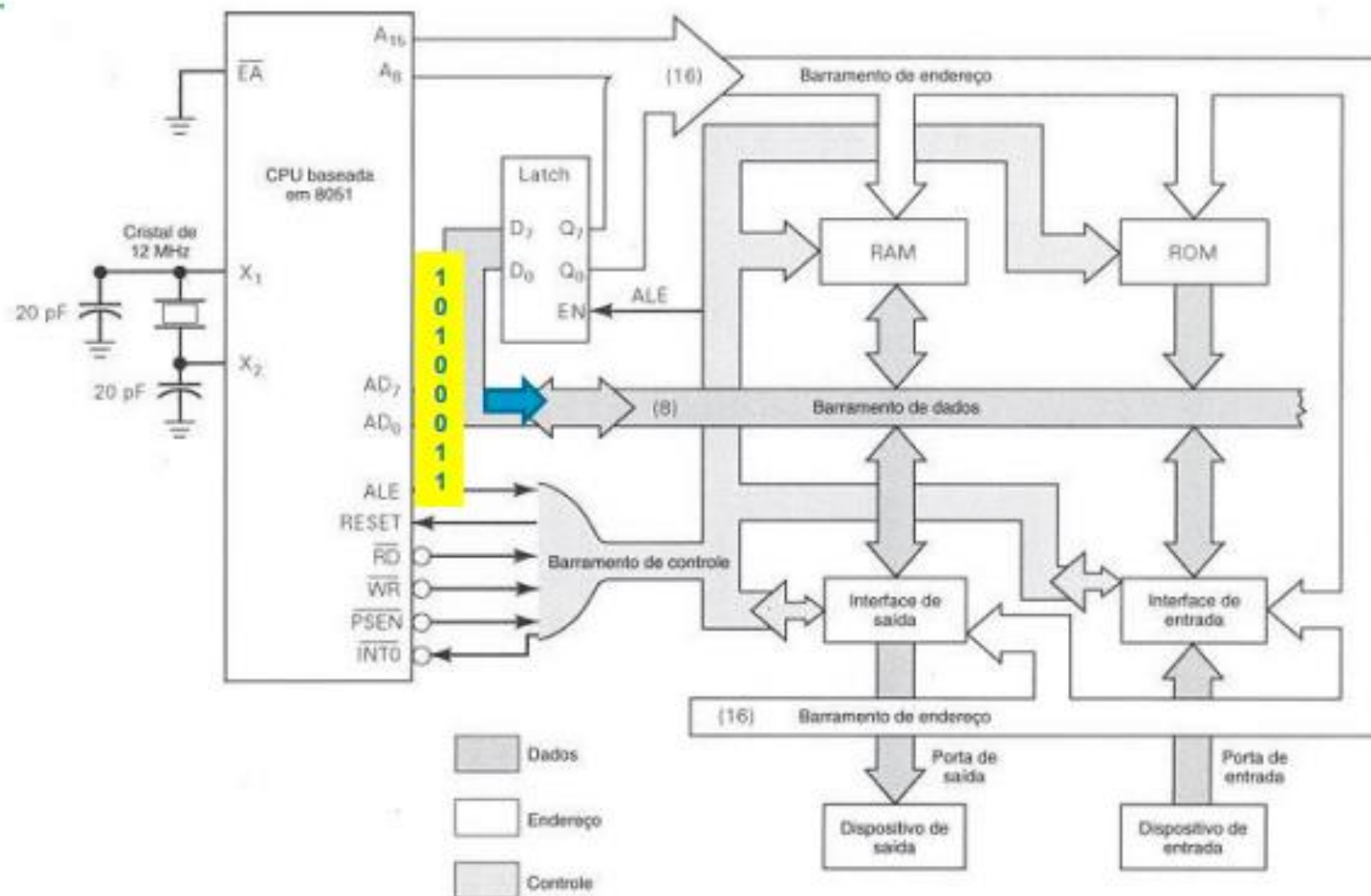
■ Barramento de dados

- Operação de leitura: os pinos de dados da CPU agem como entradas e recebem dados que foram colocados no barramento pela memória ou por um dispositivo de E/S selecionado pelo endereço colocado no barramento de endereço.



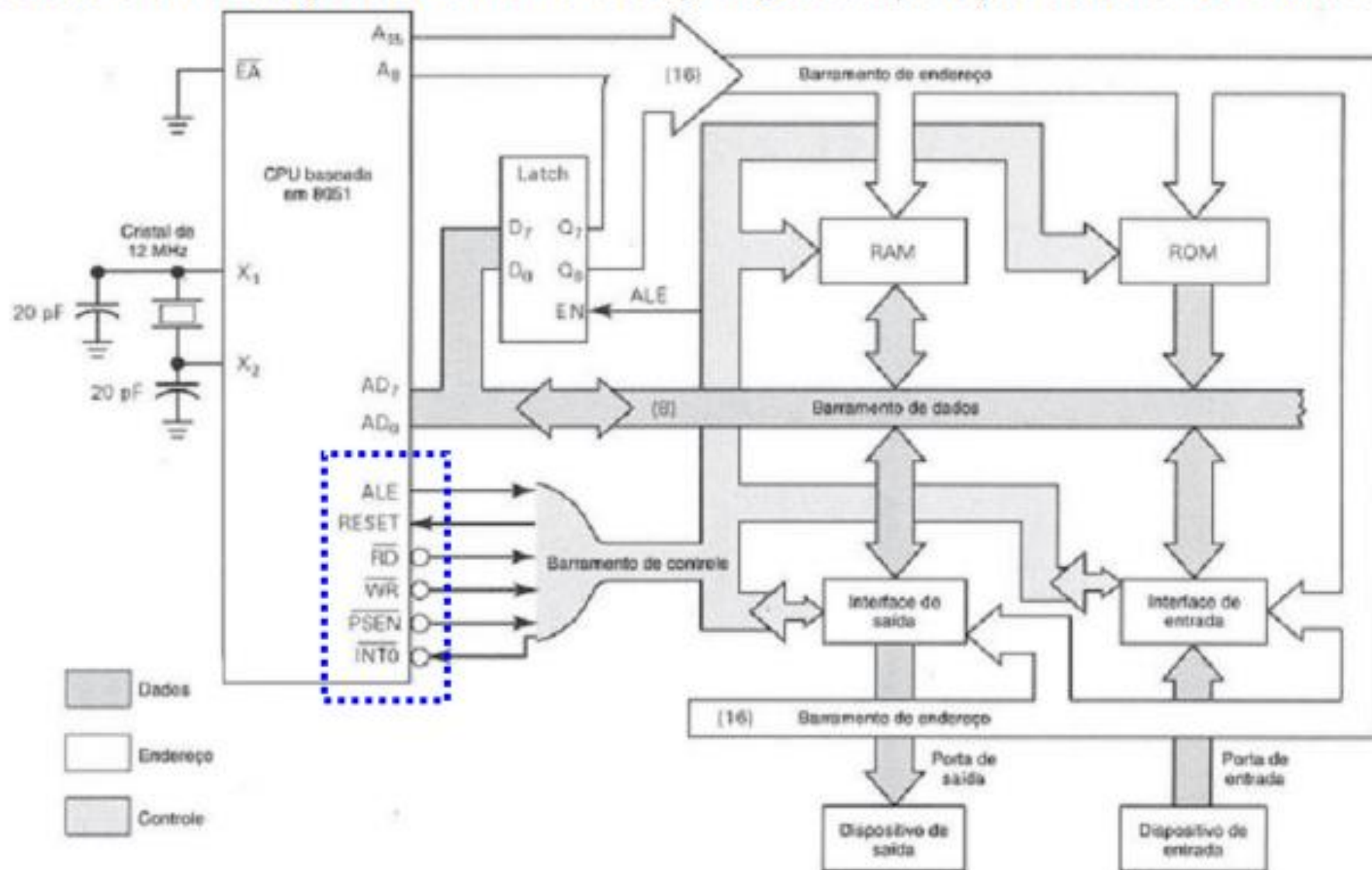
■ Barramento de dados

- Operação de escrita: os pinos de dados da CPU agem como saídas e colocam os dados no barramento de dados que são enviados para a memória ou elemento de E/S selecionado (de acordo com o endereço colocado no barramento de endereço).



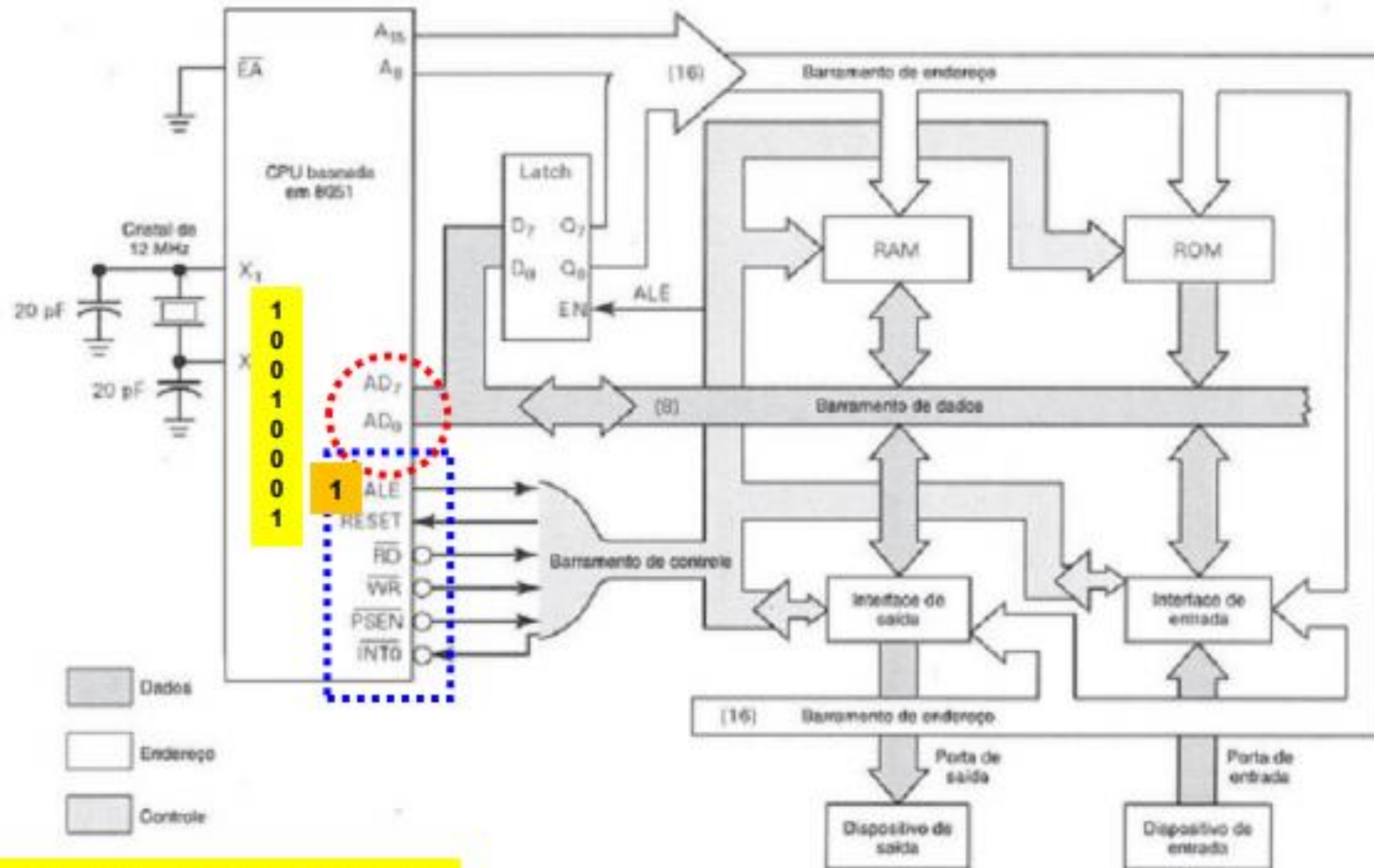
Barramento de controle

- Usado para sincronizar as atividades dos elementos do microcomputador. Alguns destes sinais de controle como **ALE**, **/PSEN**, **/RD** e **/WR** são enviados pela CPU para outros elementos para informar a eles que tipo de operação esta sendo executada .



Barramento de controle

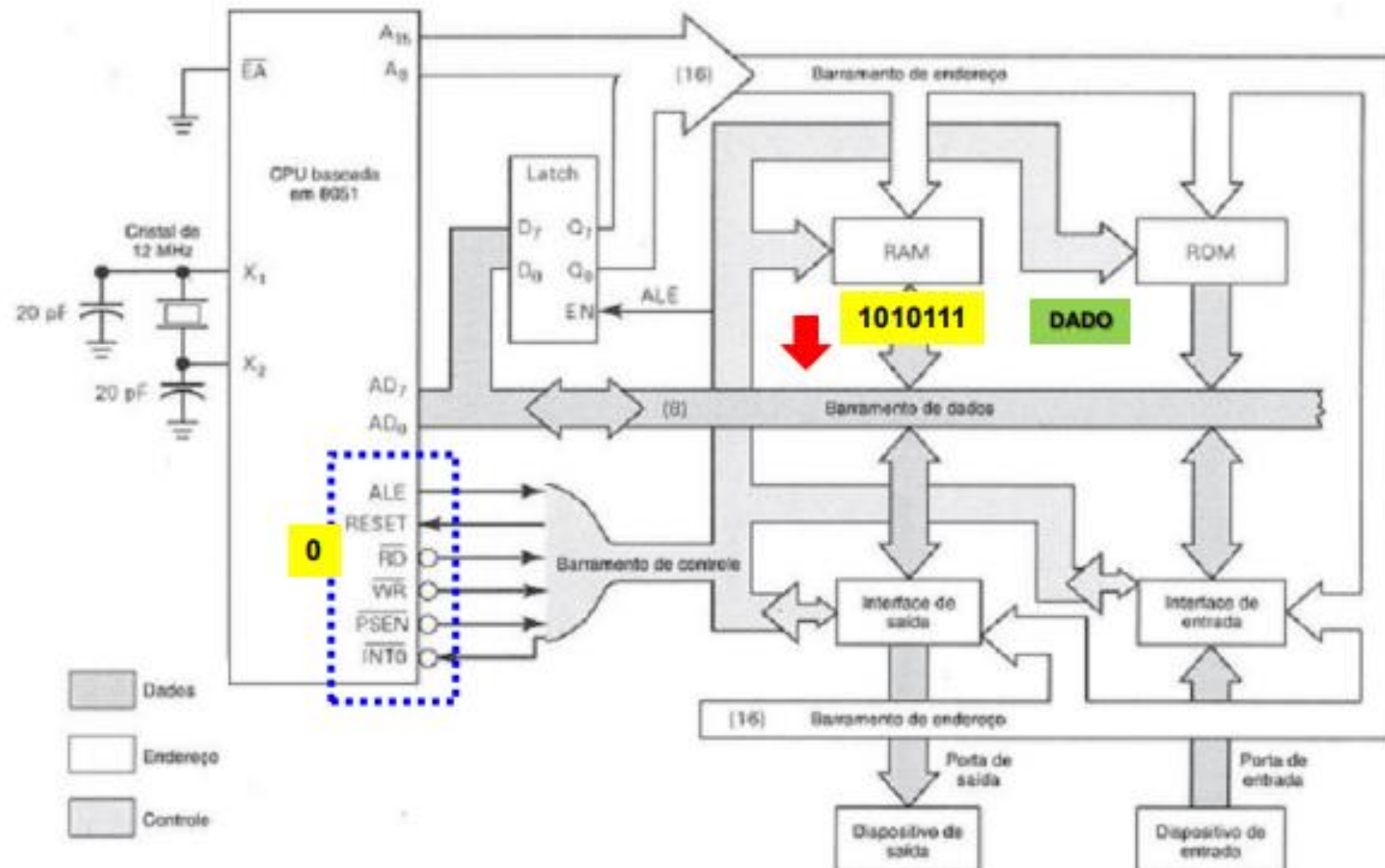
- O ALE (Address Latch Enable - habilitador do latch de endereço): está em nível ALTO (1) enquanto a CPU está colocando a parte baixa do endereço em AD_0 a AD_7 . Este sinal habilita o latch de endereço a armazenar a parte baixa do endereço durante este intervalo.



Exemplo : 1001001010010001

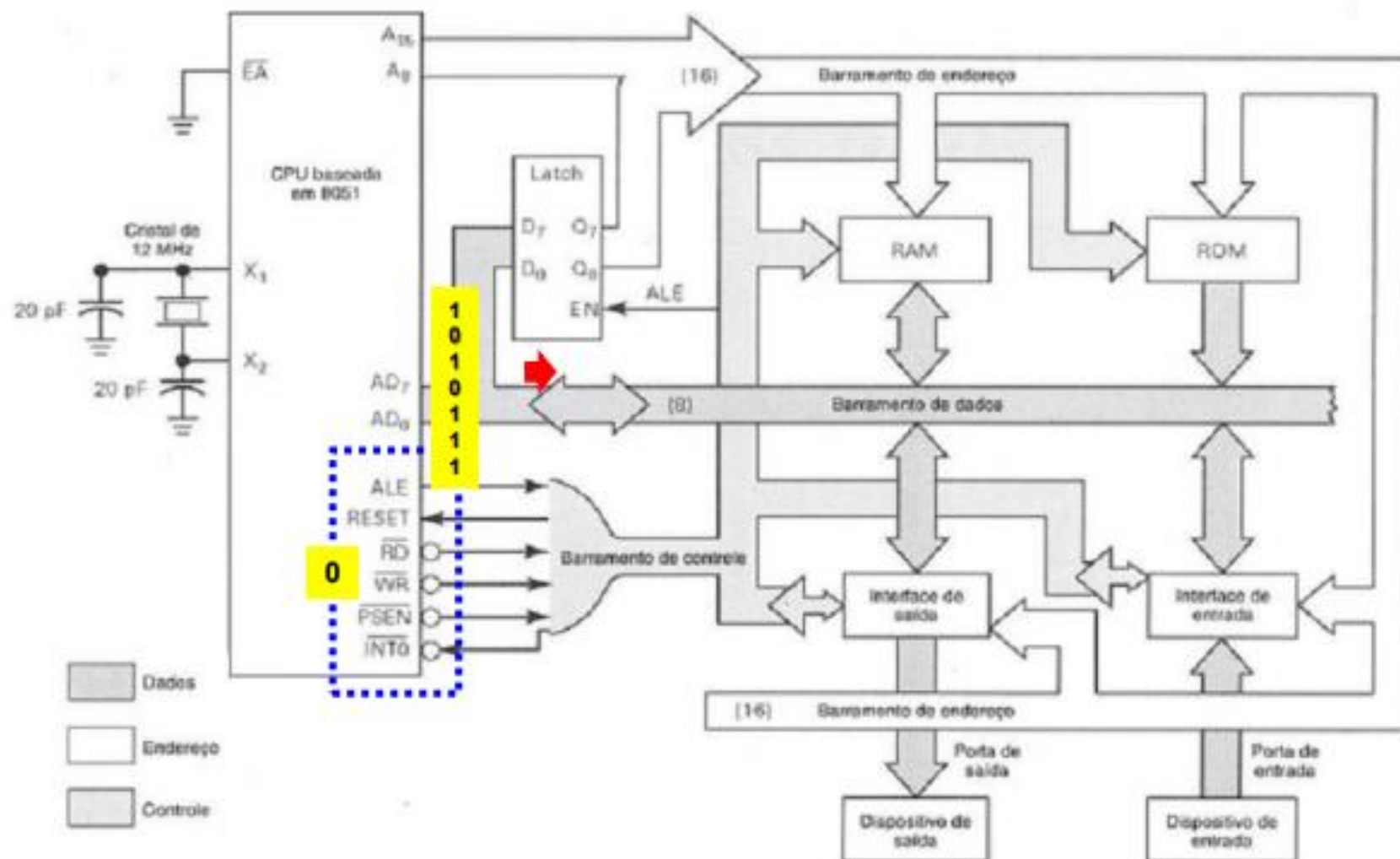
■ Barramento de controle

- 0 signal /RD está em BAIXO quando a CPU quer que a memória ou uma porta de entrada externa coloque um dados no barramento de dados.



■ Barramento de controle

- O sinal /WR está em BAIXO quando a CPU está colocando um dado no barramento de dados (para ser escrito na memória ou em uma porta de saída externa).



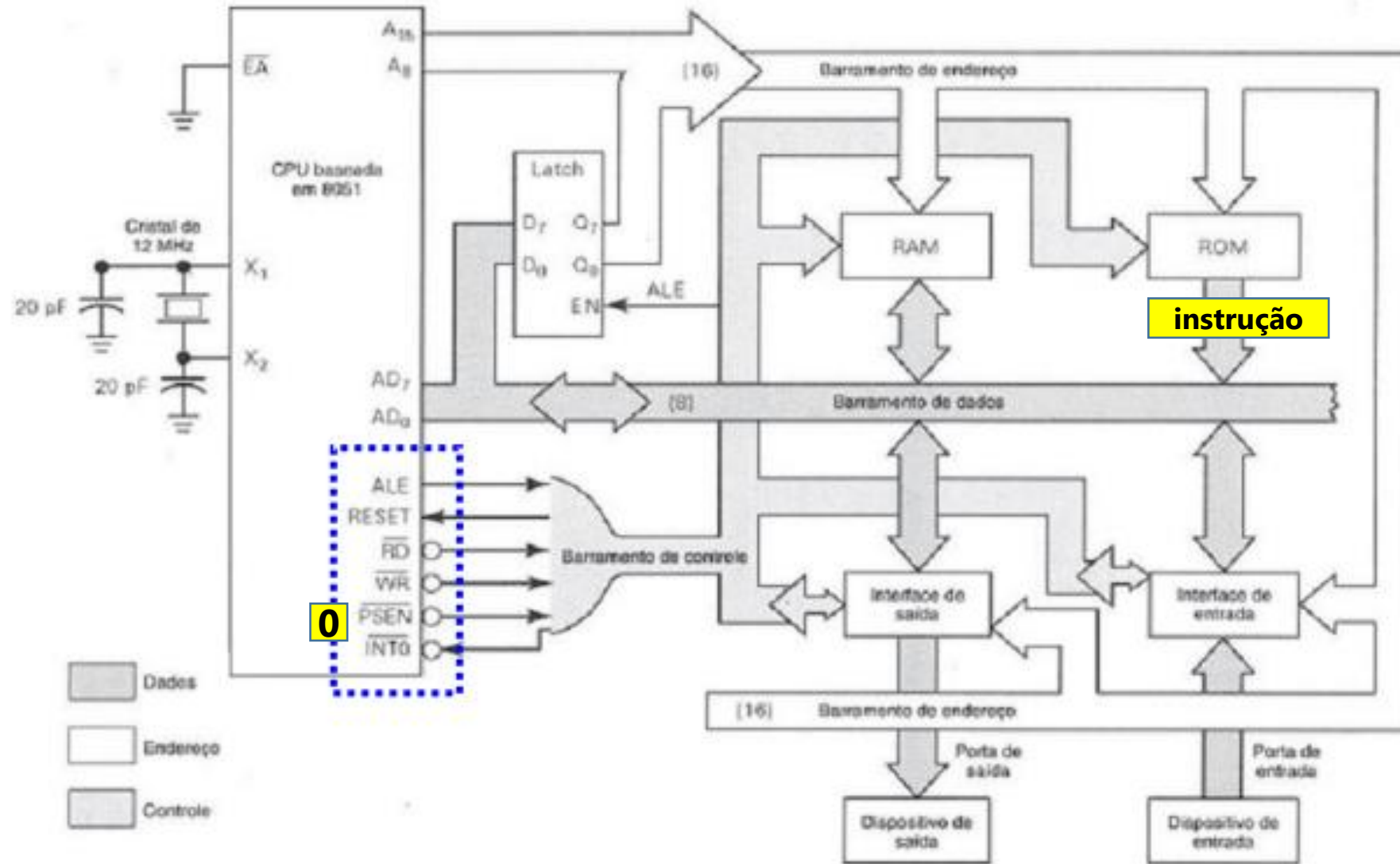
Laboratório : Atividade no Arduino/Tinkercad

Estudo sobre o microcontrolador 8051

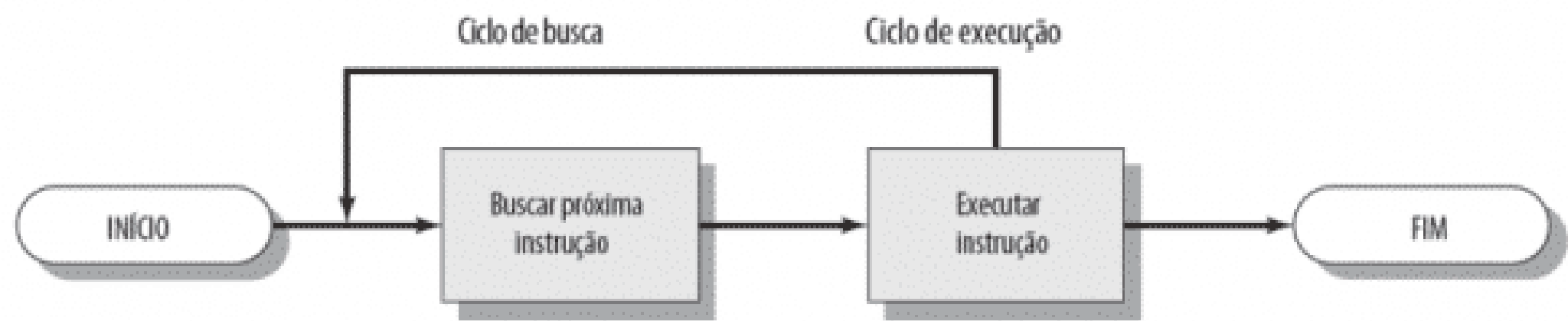
Parte 3

Barramento de controle

- O sinal /PSEN (Program Store Enable - habilitador da memória de programa) é BAIXO quando a CPU habilita a memória ROM p colocar um dado no barramento de dados.

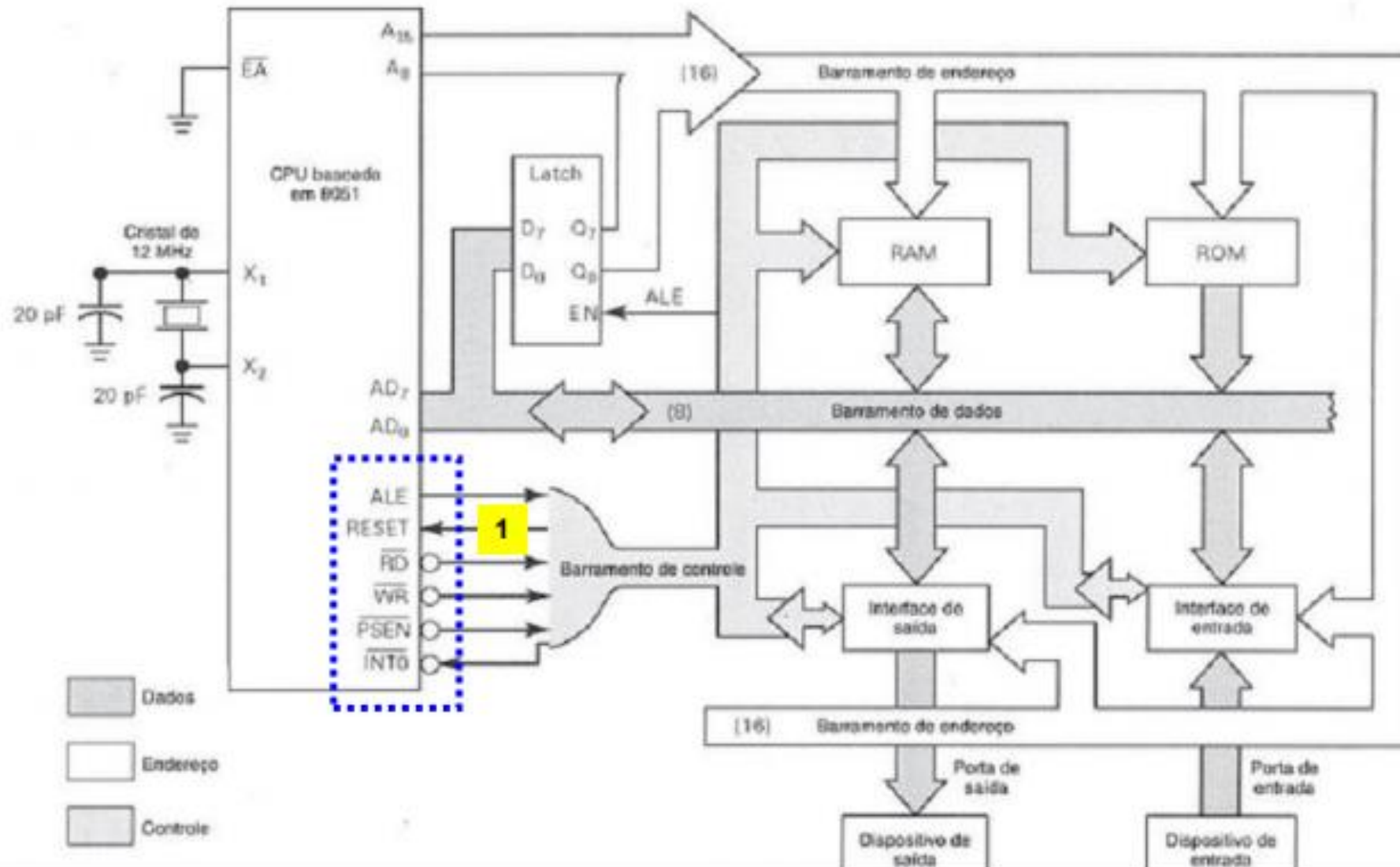


Simplificando o ciclo de Execução



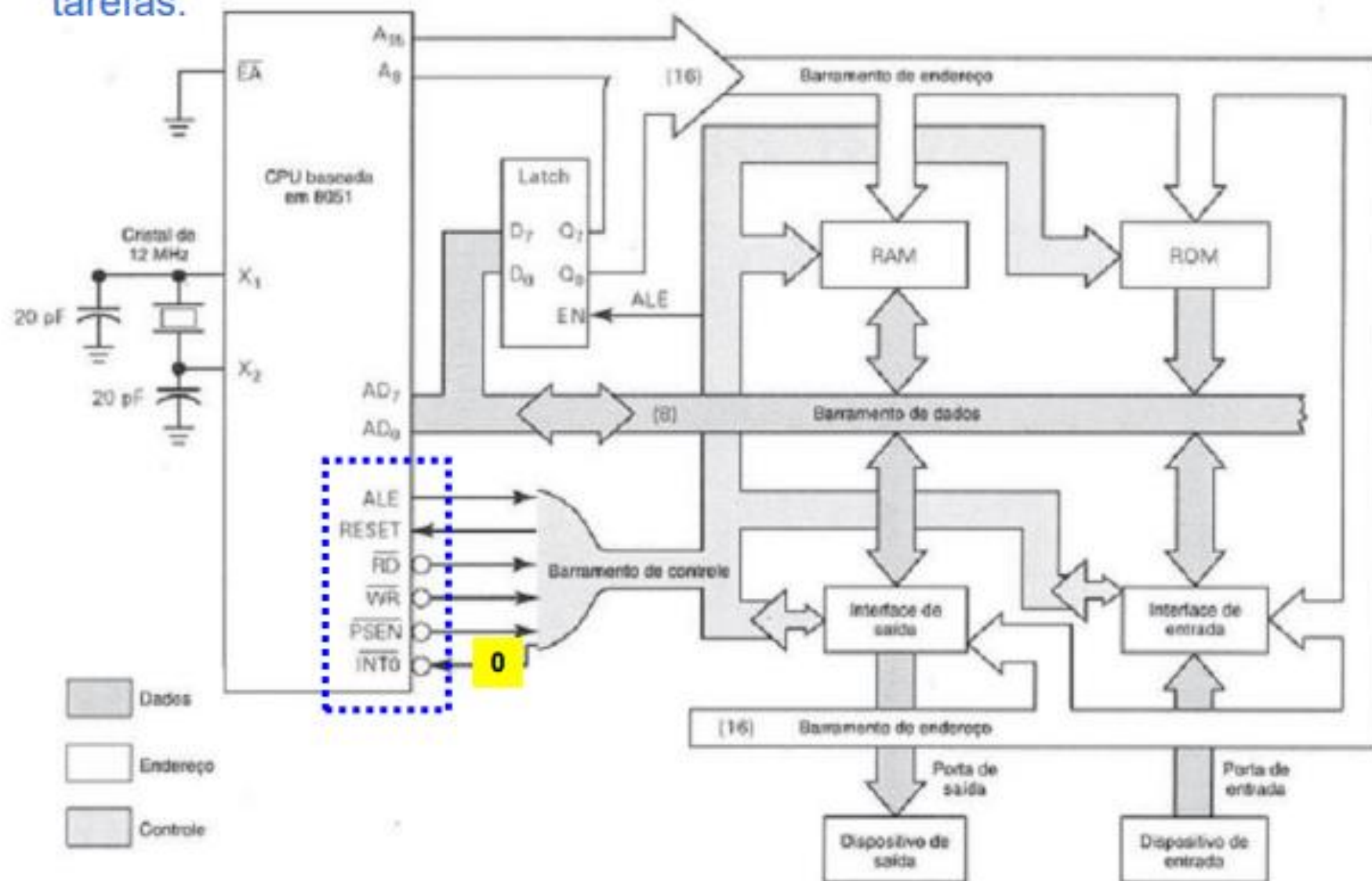
■ Barramento de controle

- Os elementos de E/S podem enviar sinais de controle para a CPU. Um exemplo é a entrada de RESET (RST) da CPU, que quando em nível ALTO faz com que a CPU vá para um estado inicial.



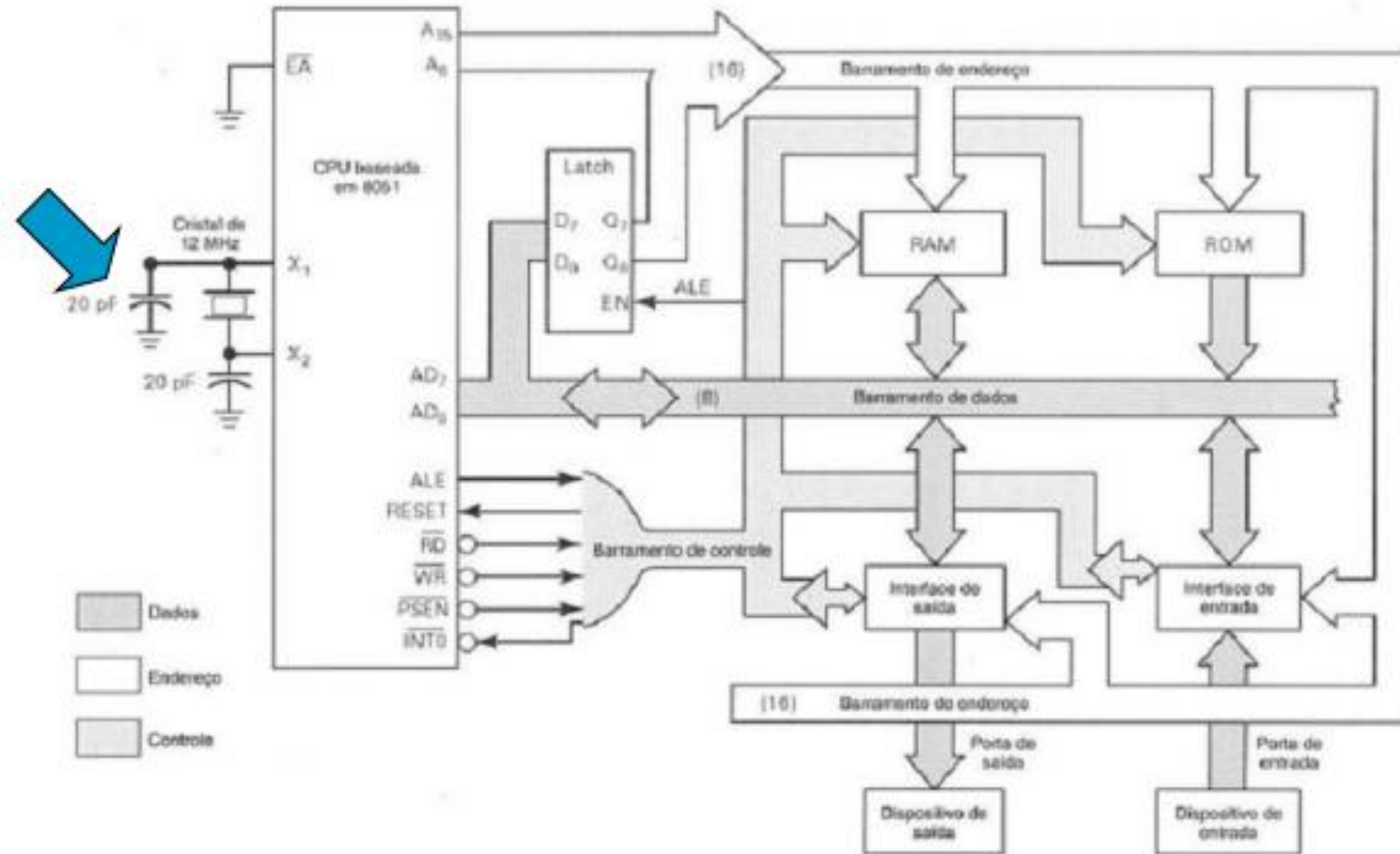
■ Barramento de controle

- Um outro exemplo é a entrada de interrupção (INT0), usada pelos dispositivos de E/S para chamar a atenção CPU quando ela está executando outras tarefas.

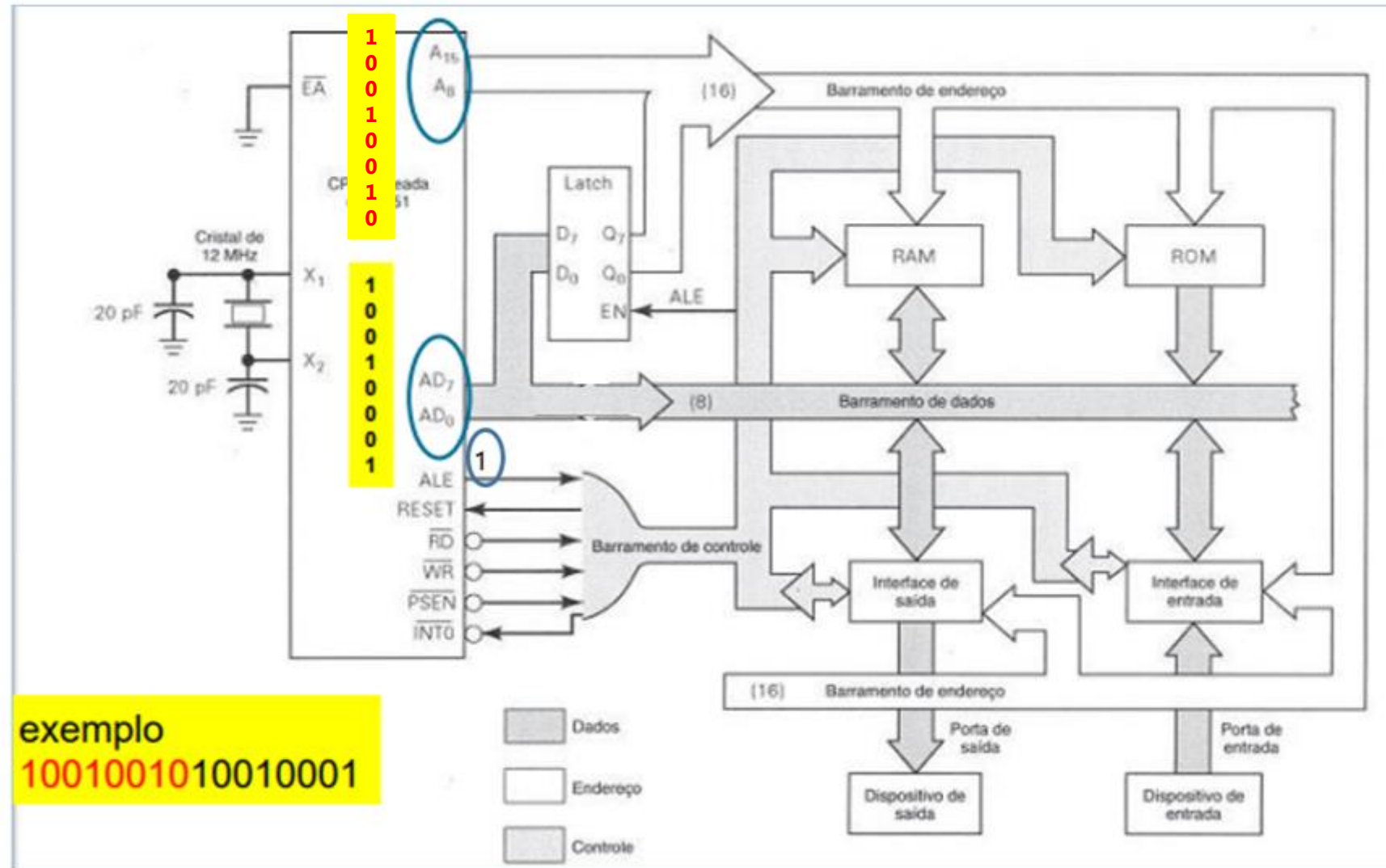


Sinal de Clock

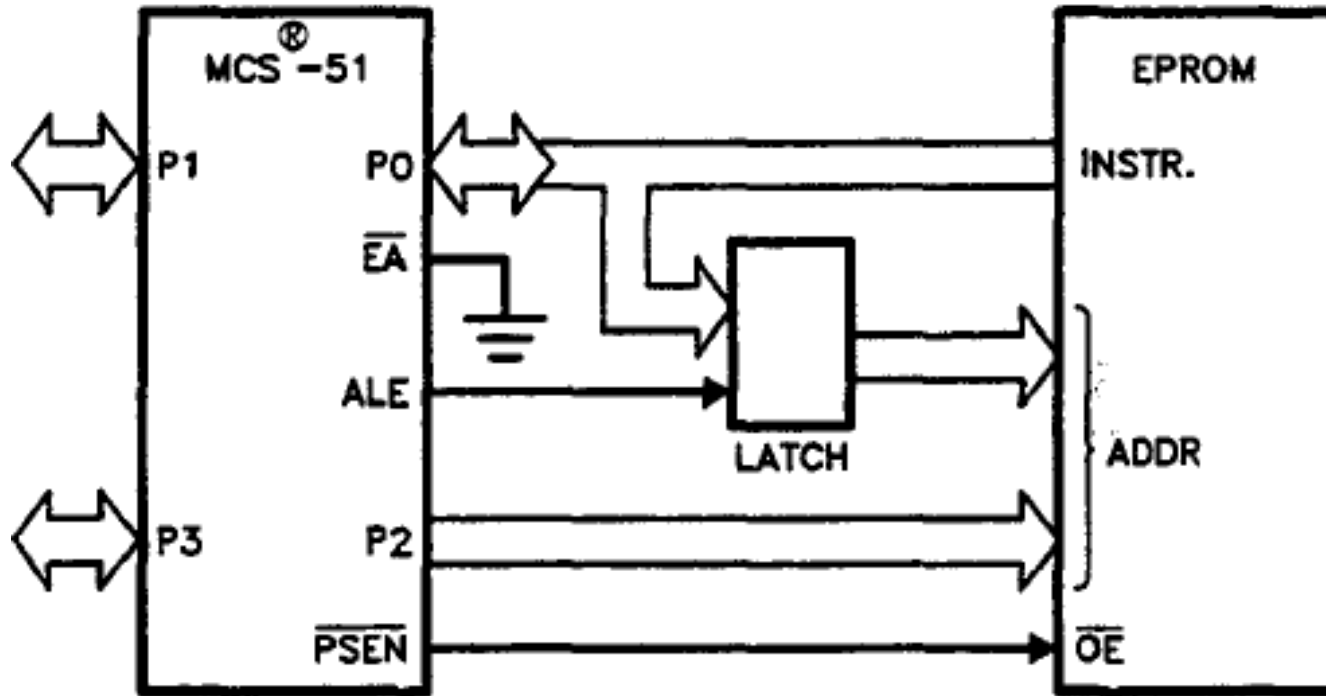
- O 8051 possui um oscilador no próprio chip que gera o sinal de clock e determina a frequência de trabalho para temporizar/sincronizar todas as suas operações. Pode-se ainda utilizar um oscilador externo para assumir essa função.



Atividade: O que está ocorrendo na imagem abaixo ?



Resumindo: Interface com Memória Externa



Conexão do 8051 com a memória de programa externa.

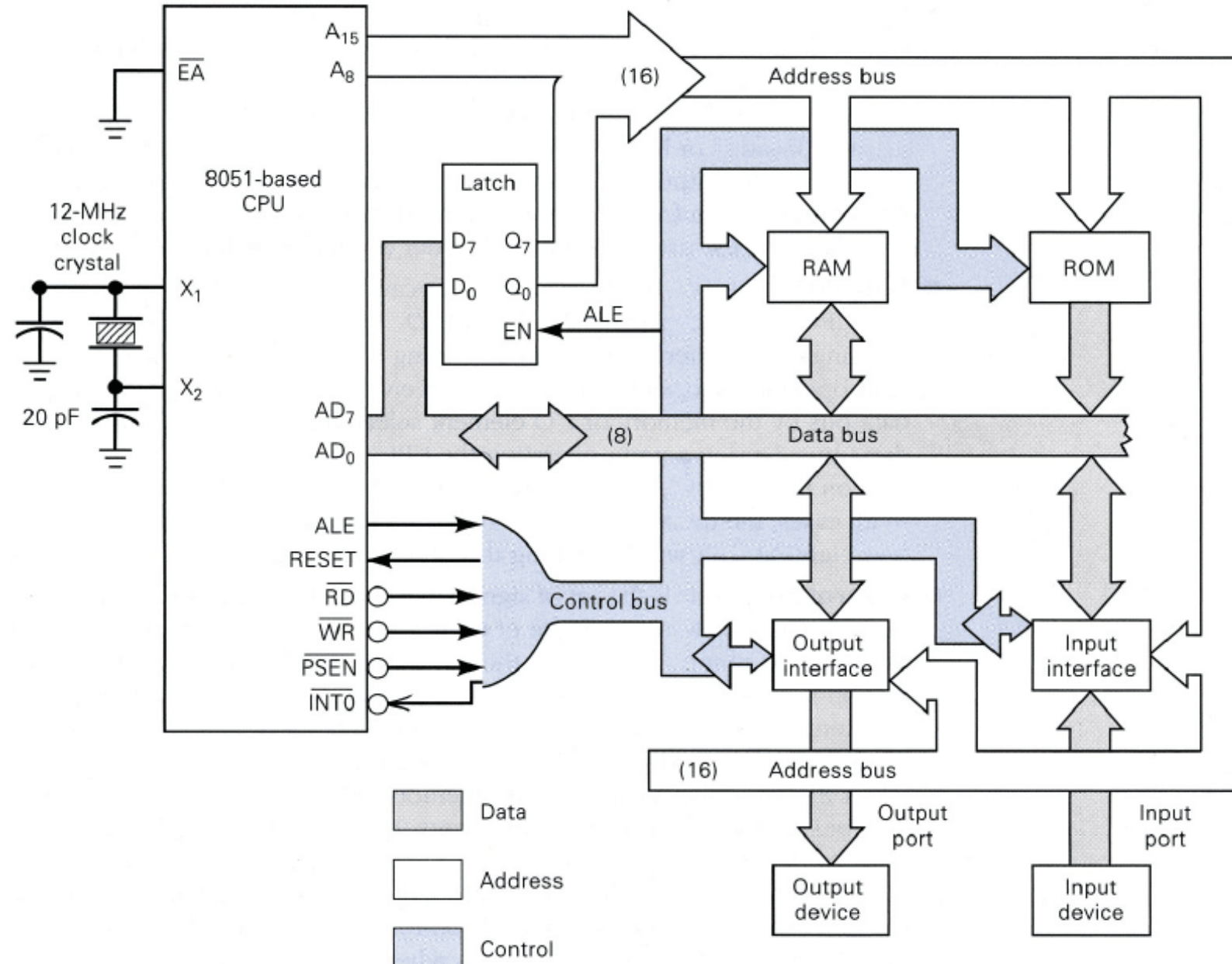
O processo de execução, a CPU emite sinais para realizar as operações de leitura e escrita das memórias. Quando a memória externa (RAM e/ou ROM) é utilizada, um circuito latch deve ser adicionado para compartilhar o PORT 0 do microcontrolador. Isso se faz necessário pois a mesma porta que é utilizada para determinar o endereço acessado também serve para as operações do barramento de dados. Além disso, o pino EA é utilizado para indicar se a memória de programa será interna ou externa. Já os sinais PSEN, RD e WR são utilizados para as operações nas memórias externas e, nos dois casos, o sinal ALE (Address Latch Enable) é usada para memória de programa externa.

Atividade 5- Analisando os slides estudados até o momento, resolva a situação abaixo

2a- Um programa situado na memória externa, pegará o dado **10101100**, endereçado por **1111000001010101** e colocará no barramento de dados.

2b- Posteriormente o dado será enviado a uma posição de memória diferente, endereçada por **1000101001011111**.

Explique em forma de algoritmo, completando a sequência e os sinais de controle envolvidos, destacando seus estados (0 ou 1), desde a leitura do dado em memória externa até seu envio à outra posição de memória.



Resolução da Atividade – primeira parte

2a

- 1 – O sinal ____ deve receber o nível **0** para habilitar o acesso a memória de _____ (**ROM**) externa.
- 2 – O endereço **11110000** é colocado no Port ____ (pinos **A_ ao A_**).
- 3 – O endereço **01010101** é colocado no Port ____ (pinos **A_ ao A_**).
- 4 – O sinal _____ recebe o bit **1**, habilitando o _____ e então o endereço **01010101** é direcionado ao barramento de _____.
- 5 – O sinal _____ em nível lógico **0**, ativa a busca de uma instrução na memória de programa _____.
- 6 – Com o sinal _____ em nível lógico **0**, o dado **10101100** é lido e colocado no barramento de _____.

Resolução da Atividade – segunda parte : O barramento de endereço deverá receber o novo endereçamento para que o dado seja colocado na nova posição da memória.

2b

7 –

8 –

9 -

10 -

11 -

12 - Com o sinal ____ em nível lógico 0, o dado **10101100** é escrito na posição de memória _____ no endereço _____.

Laboratório : Atividade no Arduino/Tinkercad

Proposta para APS – Atividade Prática Supervisionada

Desenvolvimento de projeto prático em grupo para Arduino / IoT

- *Distinguir as funções do Arduino das demais funções dos microcontroladores*
- *Interpretar as funções do Arduino*
- *Programar com Arduino / IoT*
- *Testar os códigos desenvolvidos no Arduino*
- *Projetar circuitos microcontrolados com Arduino / IoT*

Te espero na próxima aula!