

MICROPROCESSADORES

8080

E

8085

Hardware

VOLUME 1

9.ª Edição

ÉRICA

MICROPROCESSADORES

8080

e

8085

CIP — Brasil. Catalogação-na-Fonte
Câmara Brasileira do Livro, SP

V81^{4m}
v.1-

Visconti, Antônio Carlos José Franceschini, 1952-
Microprocessadores 8080 e 8085 / Antonio Carlos
José Franceschini Visconti. -- São Paulo : Érica,
1981-

Bibliografia.

Conteúdo: v.1. Hardware.

1. Microprocessadores - Programação I. Título.

81-1217

17. CDD-651.8

18. -001.642

Índices para catálogo sistemático:

1. Microprocessadores 8080 : Programação : Processamento de dados 651.8 (17.) 001.642 (18.)
2. Microprocessadores 8085 : Programação : Processamento de dados 651.8 (17.) 001.642 (18.)
3. Programação : Microprocessadores 8080 : Processamento de dados 651.8 (17.) 001.642 (18.)
4. Programação : Microprocessadores 8085 : Processamento de dados 651.8 (17.) 001.642 (18.)

ENG. ANTONIO CARLOS JOSÉ FRANCESCHINI VISCONTI

MICROPROCESSADORES

8080

e

8085

Hardware

VOLUME 1

9.^a Edição

1986



LIVROS ÉRICA EDITORA LTDA.

Nenhuma parte desta publicação poderá ser reproduzida, guardada pelo sistema "retrieval" ou transmitida de qualquer modo ou por qualquer outro meio, seja este eletrônico, mecânico, de fotocópia, de gravação, ou outros, sem prévia autorização por escrito desta EDITORA.

Desenhos:

DEVANIR VENEZIANO FERREIRA

Composição dos textos:

ROSELI BARBARESCO DE OLIVEIRA

Revisão

GUARACIABA MICHELETTI

LIVROS ÉRICA EDITORA LTDA.

Rua Jarinu, 594 - Tatuapé - São Paulo

Fone: 294-8686 - C.G.C. 50.268.838/0001-39

Caixa Postal 15.617

DEDICO

'A MARLENE,

LAURA E LUÍZA.

PREFÁCIO

Nos dias de hoje podemos notar o grande crescimento da utilização dos computadores para simplificar as mais árduas tarefas e solucionar problemas por mais complicados que possam ser. Este equipamento tem-se tornado uma ferramenta obrigatória para o desenvolvimento de qualquer área, exigindo o seu conhecimento pelos profissionais de todos os tipos de atividades.

A crescente popularidade do computador deve-se à evolução da eletrônica que vem caminhando a passos largos na área de sistemas digitais, principalmente após o surgimento dos microprocessadores. Este circuito especial fez com que o computador deixasse de ser privilégio de grandes instituições, pois tornou-se mais compacto, de custo bem mais acessível e com maiores capacidades.

Com o objetivo de fornecer material para o estudante ou projetista, que procura o conhecimento inicial sobre microprocessador, é que foi desenvolvida esta obra, dividida em dois volumes. O primeiro volume trata da parte de hardware e o segundo, da parte de software, o que corresponde a uma divisão natural no campo dos microprocessadores.

No primeiro volume temos noções de sistemas de computador, com o detalhamento das principais seções que o compõem; não nos prendemos a fundamentos teóricos e sim a descrições práticas.

Para cada parte, foi feito o estudo baseado em circuitos dedicados a cada uma delas, procurando-se, desta maneira, fugir a análises empíricas e aproximar-se das necessidades de conhecimento para aplicações reais. Entre o grande número de opções disponíveis, seguimos a linha do microprocessador 8080 que é, sem dúvida, a mais difundida e que formou as bases para as outras linhas de microprocessadores existentes na atualidade.

Cada seção foi analisada em capítulo separado, permitindo, desta maneira, a divisão racional para o estudo, além de possibilitar que a obra fosse posteriormente utilizada como fonte de consulta no decorrer das aplicações práticas.

No último capítulo foi desenvolvido e detalhado um circuito especialmente projetado, utilizando-se um microprocessador e demais periféricos, para aplicações gerais. Procurou-se, desta forma, englobar o que foi visto nos capítulos anteriores.

O segundo volume também foi desenvolvido sem muitas preocupações teóricas e seguindo a mesma linha do microprocessador 8080.

Este volume trata das técnicas de programação, análise de todas instruções e fundamentos da linguagem de programação assembler.

Alguns programas foram selecionados, desenvolvidos e analisados em detalhes para demonstrar as aplicações e os modos de operação do microprocessador.

Nosso intuito foi de suprir as necessidades de material didático, não só para cursos de nível técnico, como também para os de nível superior e, até mesmo, para as indústrias que iniciam a utilização de microprocessadores ou, ainda, para aquelas que o utilizam com o objetivo de entretenimento.

SUMÁRIO

VOLUME 1 - HARDWARE DE MICROPROCESSADORES

| | | | |
|---------------------|----------|---|-----------------|
| CAPÍTULO I | - | INTRODUÇÃO AOS MICROPROCESSADORES | 11 |
| I.1 | | Desenvolvimento da Eletrônica..... | 11 |
| I.2 | | Transformações sociais e industriais..... | 12 |
| I.3 | | Software..... | 13 |
| I.4 | | Hardware..... | 14 |
| CAPÍTULO II | - | APRESENTAÇÃO DOS SISTEMAS | 15 |
| II.1 | | Noções de computador..... | 15 |
| II.2 | | Memórias..... | 16 |
| II.3 | | Dispositivos de entrada e saída..... | 17 |
| II.4 | | Unidade central de processamento..... | 18 |
| II.5 | | Bus de informações..... | 20 |
| CAPÍTULO III | - | SISTEMA DE NUMERAÇÃO | 21 |
| III.1 | | Formação dos sistemas de numeração..... | 21 |
| III.2 | | Sistema binário..... | 22 |
| III.3 | | Sistema Hexadecimal..... | 24 |
| III.4 | | Conversão de sistemas de numeração..... | 26 |
| CAPÍTULO IV | - | BLOCOS LÓGICOS | 31 |
| CAPÍTULO V | - | MEMÓRIAS | 43 |
| V.1 | | Estrutura das memórias..... | 43 |
| V.2 | | Memórias a semicondutores..... | 46 |
| V.3 | | Estrutura da EPROM 2716..... | 48 |
| V.4 | | Estrutura da RAM 2114..... | 50 |
| CAPÍTULO VI | - | MICROPROCESSADOR 8080 | 53 |
| VI.1 | | Apresentação dos microprocessadores..... | 53 |

| | | |
|----------------------|---|-----------|
| VI.2 | Descrição da pinagem do microprocessador 8080... | 55 |
| VI.3 | Arquitetura interna..... | 60 |
| VI.4 | Seção de lógica e aritmética..... | 60 |
| VI.5 | Rede de registradores..... | 62 |
| VI.6 | Seção de controle..... | 64 |
| VI.7 | Configuração geral..... | 65 |
| CAPÍTULO VII | - GERADOR DE CLOCK 8224 | 67 |
| VII.1 | Generalidades..... | 67 |
| VII.2 | Descrição da pinagem..... | 67 |
| VII.3 | Arquitetura interna..... | 71 |
| VII.4 | Esquema de ligação..... | 71 |
| CAPÍTULO VIII | - CONTROLADOR DO SISTEMA 8228 | 73 |
| VIII.1 | Generalidades..... | 73 |
| VIII.2 | Descrição da pinagem..... | 73 |
| VIII.3 | Arquitetura interna..... | 75 |
| VIII.4 | Esquema de ligação..... | 76 |
| CAPÍTULO IX | - MICROPROCESSADOR 8085 | 79 |
| IX.1 | Generalidades..... | 79 |
| IX.2 | Descrição da pinagem..... | 80 |
| IX.3 | Arquitetura interna..... | 84 |
| CAPÍTULO X | - INTERFACE DE COMUNICAÇÃO PARALELA PROGRAMÁVEL 8255 | 87 |
| X.1 | Generalidades..... | 87 |
| X.2 | Descrição da pinagem..... | 87 |
| X.3 | Arquitetura interna..... | 89 |
| X.4 | Modo de operação..... | 90 |
| CAPÍTULO XI | - INTERFACE DE COMUNICAÇÃO SERIAL PROGRAMÁVEL 8251 | 97 |

| | | | |
|-----------------|-------------|--|------------|
| | XI.1 | Generalidades..... | 97 |
| | XI.2 | Descrição da pinagem..... | 100 |
| | XI.3 | Arquitetura interna..... | 104 |
| | XI.4 | Modo de operação..... | 105 |
| CAPÍTULO | XII | - CONTADOR PROGRAMÁVEL 8253 | 113 |
| | XII.1 | Generalidades..... | 113 |
| | XII.2 | Descrição da pinagem..... | 113 |
| | XII.3 | Arquitetura interna..... | 115 |
| | XII.4 | Modo de operação..... | 116 |
| CAPÍTULO | XIII | - ESQUEMA BÁSICO DE UM MICROCOMPUTADOR... | 123 |

1.1 DESENVOLVIMENTO DA ELETRÔNICA

A eletrônica é uma ciência de constante desenvolvimento, apresentando sempre novas teorias e novas técnicas. Assim, sempre temos o surgimento de novos produtos suprimindo as constantes necessidades do mercado.

Particularmente na área dos semi-condutores, a eletrônica tem evoluído a passos largos, onde podemos notar as diferentes fases, a descoberta dos transistores, o surgimento dos circuitos integrados e a recente apresentação dos microprocessadores.

No começo da década de 70, quando foram apresentados os primeiros microprocessadores, já se notou o interesse da utilização deste componente pelos diversos setores industriais e em um curto espaço de tempo, temos notado uma crescente utilização dos microprocessadores em equipamentos destinados à pesquisa, assistência, entretenimento, contabilidade e todos os setores industriais.

Para entender o porquê desta penetração tão grande e rápida, vamos voltar ao início da eletrônica dos semicondutores, onde tínhamos o transistor e o diodo que combinados com os outros componentes eletrônicos exigiam um árduo trabalho na fase de projeto e uma complexidade semelhante na fase de montagem e manutenção.

Com o progresso da tecnologia de semicondutores, surgiu o circuito integrado, que consiste de um único componente eletrônico agrupando uma série de componentes discretos, voltado a executar certas tarefas mais específicas. Na medida em que fosse aumentada a integração de um circuito, isto é, o número de componentes discretos agrupados em um único circuito integrado, mais específica torna-se a sua aplicação.

Esta evolução segue uma curva como mostra o gráfico da figura 1.1.

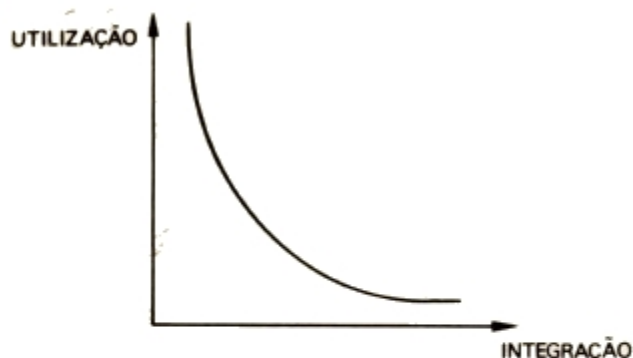


Fig. I.1

O microprocessador surgiu para quebrar esta evolução, pois o microprocessador em si é um circuito integrado que possui uma altíssima integração, mas é voltado para uma utilização bastante genérica. Isto se dá pelo fato do microprocessador ser um circuito programável que se torna específico, quando colocamos nele uma programação, podendo ser alterada sua aplicação alterando sua programação.

I.2 TRANSFORMAÇÕES SOCIAIS E INDUSTRIAIS

O microprocessador provocou uma revolução na eletrônica e no mercado, pois com ele tornou-se possível a disposição de equipamentos que antes exigiam vultuosos investimentos, ocupavam um grande volume, um longo tempo e um grande número de pessoas em seus desenvolvimentos. A confiabilidade tornou-se alta, pois foi reduzido o número de componentes e, em consequência, a complexidade dos circuitos.

Os exemplos de transformação devido à influência do microprocessador são muitos. Podemos notar que em todos os campos o microprocessador tem sido aplicado e com uma série de benefícios desde equipamentos destinados à pesquisa espacial onde a complexidade de circuitos elétricos e também dos dispositivos mecânicos foi minimizada e substituída por um único circuito integrado capaz de resolver tarefas complexas, até em simples calculadoras que, hoje, são facilmente adquiridas e que, sem o microprocessador, seriam equipamentos complexos, caros e de difícil acesso.

Assim sendo, dispomos hoje na eletrônica de um novo componente, o microprocessador, que nada mais é do que um único circuito integrado mas que se tornou uma palavra chave da eletrônica, tornando-se um passo de separação entre duas eras.

Quando tratamos da parte referente à programação do microprocessador, teremos que relacioná-la com a palavra "software"; quando tratarmos dos componentes e dos circuitos elétricos, relacionaremos com o "hardware".

Novos termos foram criados e procuraremos explicá-los e utilizá-los durante o estudo que faremos. Vamos tentar classificar o significado de software e hardware que constituem as definições das duas áreas principais dos projetos com microprocessadores.

1.3 SOFTWARE

O microprocessador é um circuito que possui a capacidade de executar diversos tipos de funções distintas. Cada função é específica e bem determinada, mas o número de funções não é unitário, como é a característica dos outros circuitos integrados digitais, pois temos que notar que os circuitos integrados digitais possuem uma função específica, função esta que com os sinais colocados em sua entrada, combinados com as suas variáveis de estados produzem uma saída específica. Por exemplo, um circuito inversor possui uma entrada, uma saída e não possui variáveis de estados. Quando colocamos na entrada de um circuito inversor o nível lógico "1", teremos na sua saída o nível lógico "0" e, quando colocamos na entrada o nível lógico "0", teremos na saída o nível lógico "1". Portanto, a função do inversor é produzir uma saída correspondente ao inverso da entrada.

Como dissemos, o microprocessador não possui uma única função, mas diversas funções, às quais damos o nome de "instrução".

Cada instrução é colocada dentro do microprocessador e, a cada instante, o microprocessador executa a instrução específica que lhe foi colocada. Quando queremos que o microprocessador execute uma tarefa, temos que criar uma série de instruções as quais ele irá executar uma a uma. A esta série de instruções da

mos o nome de programa. Portanto, para que o microprocessador execute uma tarefa devemos programá-lo.

Quando estamos desenvolvendo um programa, determinando qual a série de instruções devem ser executadas, estamos trabalhando em software do microprocessador.

1.4 HARDWARE

O microprocessador por si só não é auto suficiente, exige uma série de componentes para sua utilização. Um sistema com microprocessador tem a necessidade de possuir "portas" de entradas e saídas por onde os sinais são recebidos e enviados pelo circuito, memória onde estarão armazenados os programas e dados, controladores, buffers e demais circuitos.

O microprocessador é apenas a unidade central do processamento do circuito onde os dados são manipulados. A este conjunto de componentes interligados que formam o circuito, damos o nome de hardware do microprocessador. Assim a palavra hardware está relacionada com os circuitos elétricos.

Um sistema genérico com microprocessador pode ser representado conforme a figura 1.2.

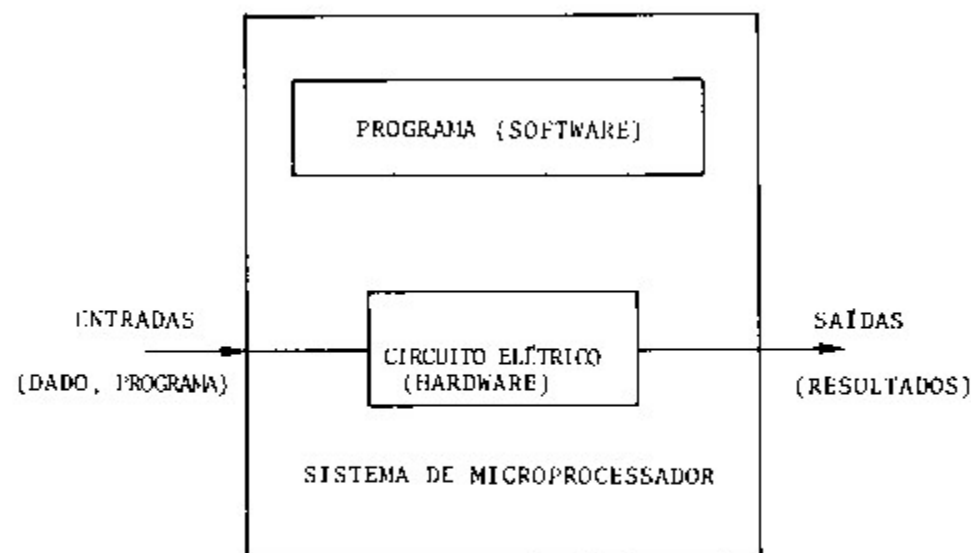


Fig. 1.2

II.1 NOÇÕES DE COMPUTADOR

Para podermos entender o circuito integrado "microprocessador", devemos, primeiramente, ter uma rápida noção de computadores, seu princípio de funcionamento.

Não iremos estudar o computador como uma máquina de processamento de dados, capaz de executar programas sofisticados com grande velocidade e grande capacidade de armazenamento de dados. Não iremos tentar estudar os programas complexos que um computador pode executar, mas vamos tentar entender do que é constituído o computador, vamos ver quais são os seus blocos e quais suas funções.

O computador pode ser definido como um sistema capaz de executar uma tarefa específica que poderá ser alterada, a qualquer momento, de acordo com as necessidades.

Isto se dá pelo fato de ser o computador um sistema programável. Para entender o que é um sistema programável, vejamos no seguinte: temos um problema que precisamos solucionar; encontrar a solução deste problema é uma tarefa árdua, então tomamos uma máquina para nos auxiliar na solução deste problema. Para isto devemos ditar quais as funções que esta máquina deve executar, introduzir estas funções na máquina e receber a solução para a nossa análise. Portanto, o computador necessita que ditemos quais as funções que deve executar e isto nada mais é que sua programação.

Para visualizar melhor a utilização de um computador, apresentamos na figura II.1 um diagrama de blocos mostrando os passos que devem ser executados para uma aplicação genérica.

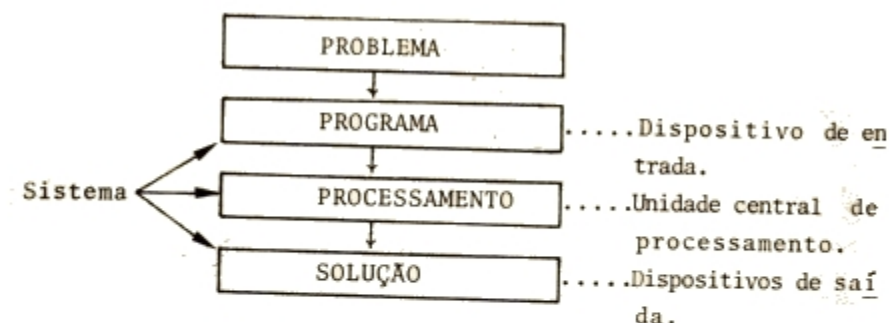


Figura II.1 Utilização clássica de um computador.

Já começamos a perceber quais são os blocos básicos que podem ser isolados dentro do sistema de computador. Esta estrutura é apresentada na figura II.2.



Fig. II.2 Sistema básico de computador.

Procuraremos prender-nos ao estudo destes blocos que constituem o sistema básico de computador com uma rápida descrição de cada um deles e um estudo mais aprofundado nos próximos capítulos.

II.2 MEMÓRIAS

A memória do computador é onde iremos armazenar os dados que devem ser manipulados pelo computador (o que chamamos de memória de dados) e também onde estará armazenado o programa

do computador (o que chamamos de memória de programa). Aparentemente não existe uma diferença física para o sistema entre as memórias de dados e as memórias de programas, apenas podemos utilizar memórias fixas para armazenar dados fixos ou programas e memórias que podem ser alteradas pelo sistema para armazenarmos dados que podem variar no decorrer do programa.

Existem diversos tipos de memória que podem ser utilizadas pelo computador: memórias de núcleo magnético, fita magnética, discos magnéticos e até memória de semicondutor em forma de circuito integrado. As memórias de semicondutor, que são as mais utilizadas em sistemas de microprocessador, serão melhor estudadas em suas estruturas e tipos no capítulo V.

11.3 DISPOSITIVOS DE ENTRADA E SAÍDA

Os dispositivos de entrada e saída são os dispositivos responsáveis pela interligação entre o homem e a máquina, são os dispositivos por onde o homem pode introduzir informações na máquina ou por onde a máquina pode enviar informações ao homem. Como dispositivos de entrada podemos citar os seguintes exemplos: leitor de fita magnética, leitor de disco magnético, leitor de cartão perfurado, leitor de fita perfurada, teclado, painel de chaves, etc. Estes dispositivos tem por função a transformação de dados em sinais elétricos para a unidade central de processamento.

Como dispositivos de saída podemos citar os seguintes exemplos: gravador de fita magnética, gravador de disco magnético, perfurador de cartão, perfurador de fita, impressora, video, display, etc. Todos eles tem por função a transformação de sinais elétricos em dados que podem ser manipulados mais tarde ou dados que são imediatamente entendidos pelo homem.

DISPOSITIVOS DE ENTRADA E SAÍDA.

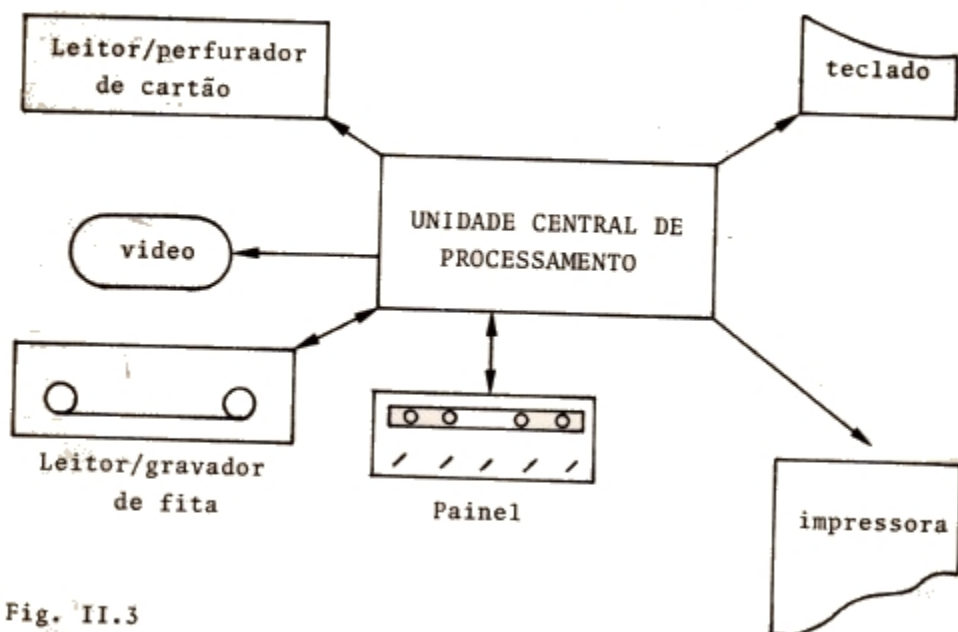


Fig. II.3

Estes dispositivos são conectados à unidade central de processamento por intermédio de "portas" que são interfaces de comunicação dos dispositivos de entrada e saída.

Algumas interfaces especiais serão estudadas nos capítulos IX e X.

II.4 UNIDADE CENTRAL DE PROCESSAMENTO.

Esta é a parte do computador responsável em coordenar todas as tarefas e executar os cálculos, ela também pode ser chamada de processador e pode ser dividida em três partes básicas: unidade lógica-aritmética (ULA), unidade de controle e rede de registradores.

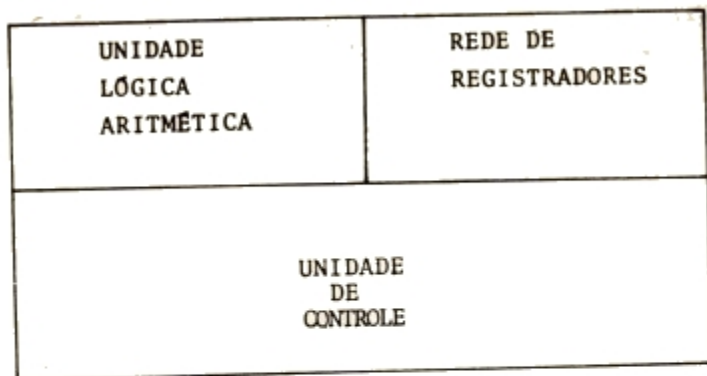


Fig. II.4 Unidade central de processamento.

A unidade aritmética é a responsável pela execução dos cálculos com os dados. Por exemplo, as operações lógicas AND, OR, NOR, EXCLUSIVE OR, etc., entre dois dados ou então as operações aritméticas com a soma, subtração, multiplicação e divisão.

A unidade de controle é a responsável por gerar os sinais de controle para os sistemas, sinais estes que podem ser, por exemplo, de leitura de memória, de escrita de memória, de leitura de periférico, de sincronização de interface de comunicação de interrupção, enfim, todos os sinais de controle necessários para o sistema.

A rede de registradores é constituída por uma série de registradores que são utilizados em uso geral onde são armazenados, temporariamente, dados que estão sendo manipulados pela unidade central de processamento, ou registrador utilizado como contador de programa, ou, ainda, registrador utilizado como armazenador de endereços, etc.

Com o desenvolvimento da eletrônica de semicondutores foi possível a construção da unidade central de processamento em um único circuito integrado e apesar de haver algumas reduções nas capacidades desta unidade central de processamento, elas apresentam um grande número de utilizações e uma grande difusão. Por ser um processador em um único circuito integrado o que apresenta dimensões bastante diminutas em relação aos processadores anteriores estes circuitos recebem o nome de microprocessadores.

Portanto, o microprocessador nada mais é do que uma unidade central de processamento de dimensões reduzidas.

Um estudo mais detalhado sobre as partes da unidade central de processamento é feito no capítulo VI.

II.5 BUS DE INFORMAÇÕES

Pudemos perceber até aqui que, entre a unidade central de processamento, a memória e os dispositivos de entrada e saída deve haver uma transferência de sinais elétricos para transmissão de informação.

Estas informações podem ser classificadas como dados ou endereços ou ainda em sinais de controle. Assim sendo, temos três tipos de linhas diferentes para a transmissão dos sinais elétricos, a estas linhas damos o nome de barramento ou "bus". Portanto, dentro de um sistema temos o bus de dados, o bus de endereços e o bus de controle como podemos ver representados na figura II.5.

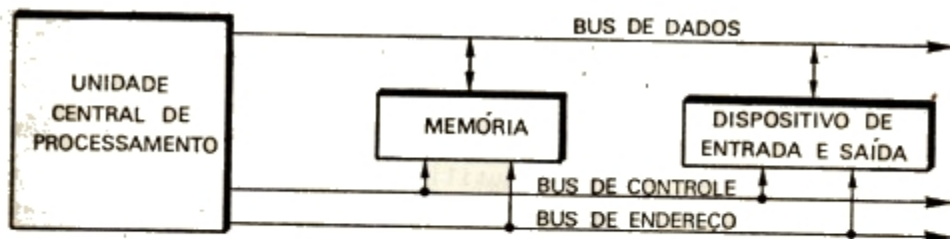


Fig. II.5

III.1 FORMAÇÃO DOS SISTEMAS DE NUMERAÇÃO

Para podermos entender melhor o funcionamento de um sistema com microprocessador, precisamos entender como são os sinais utilizados para troca de informações entre os vários circuitos que compõem o sistema, isto é, como um dado é codificado para ser armazenado na memória ou como uma instrução é codificada para ser recebida pela unidade central de processamento para ser processada.

Para que possamos entender isto, devemos também conhecer alguns sistemas de numeração diferentes daqueles que estamos habituados.

Vamos primeiramente estudar o sistema de numeração decimal. Estamos bastante familiarizados com este que é o utilizado no dia a dia e é composto por dez algarismos (0 à 9), com os quais podemos representar qualquer quantidade através de uma lei de formação. Este sistema tem sido, durante séculos, bastante eficiente para o homem e sua adoção tornou-se universal. Mas não é a única maneira existente para se expressar uma quantidade, pelo contrário, qualquer outro sistema composto por diferente número de algarismos traria a mesma eficiência para o homem, apenas seria necessário nos habituarmos com o mesmo.

Já para a computação e mesmo para qualquer circuito digital este sistema de numeração decimal não é o mais apropriado, devido ao fato de que uma informação deveria ter dez maneiras diferentes de ser expressa, representando cada um dos dez algarismos do sistema de numeração decimal. Para um circuito elétrico é conveniente a apresentação de uma informação em apenas dois estados lógicos diferentes, por ser mais simples termos somente duas situações distintas como a presença ou não de um nível de tensão.

Isto poderia ser exemplificado como uma chave que possui duas situações distintas: aberta ou fechada, onde em sua

saída teremos a presença ou não de um nível de tensão.

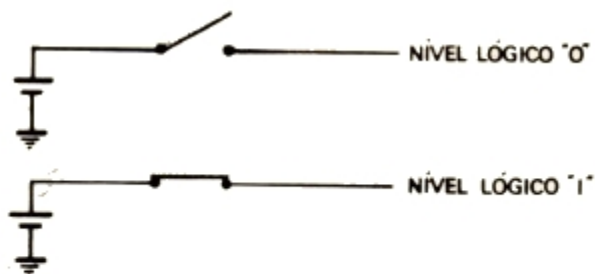


Fig. III.1

Para um circuito digital que possui normalmente como nível de tensão de referência 5 Volts, é convencionado que, quando tivermos na sua saída o nível de tensão 5 Volts, temos presente o nível lógico "1" e quando o nível de tensão for zero Volts, temos o nível lógico "0" (para circuitos com tecnologia TTL temos em média que para o nível lógico "0", a tensão máxima é de 0,5 Volts e para o nível lógico "1" tensão mínima de 3,5 Volts).

III.2 SISTEMA BINÁRIO

Desta maneira, se utilizarmos apenas dois estados lógicos distintos, o sistema de numeração utilizado é o sistema binário que é composto por dois algarismos (0 e 1).

Para representarmos uma quantidade o sistema binário, devemos utilizar o mesmo princípio de formação usado no sistema decimal. No sistema decimal colocamos os números um ao lado do outro; o algarismo mais à direita representa o número de vezes a base na potência zero, seguido pelo algarismo que representa o número de vezes a base na potência 1 e assim por diante. Por exemplo o número 702 no sistema decimal significa:

$$\begin{array}{c|c|c} 10^2 & 10^1 & 10^0 \\ \hline 7 & 0 & 2 \end{array} \Rightarrow$$

$$7 \times 10^2 + 0 \times 10^1 + 2 \times 10^0 = 702_{10} \text{ (le-se 702 na base 10)}$$

Da mesma maneira podemos representar um número no sistema binário com a diferença que no caso a base é dois e os algarismos são apenas o "0" e o "1". Por exemplo o número 1101 no sistema binário significa:

| | | | | |
|-------|-------|-------|-------|----|
| 2^3 | 2^2 | 2^1 | 2^0 | => |
| 1 | 1 | 0 | 1 | |

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

Assim, para representarmos a quantidade zero, utilizamos o algarismo "0", para representarmos a quantidade um, utilizamos o algarismo "1", para representarmos a quantidade 2 utilizamos o algarismo "1" seguido pelo algarismo "0", isto é "10", etc, o que nos trará a seguinte tabela de correspondência:

| |
|-------------------|
| $0_{10} = 0_2$ |
| $1_{10} = 1_2$ |
| $2_{10} = 10_2$ |
| $3_{10} = 11_2$ |
| $4_{10} = 100_2$ |
| $5_{10} = 101_2$ |
| $6_{10} = 110_2$ |
| $7_{10} = 111_2$ |
| $8_{10} = 1000_2$ |

Para os circuitos elétricos, portanto, o sistema numérico utilizado é o binário e cada dígito é conhecido pelo nome de BIT (do inglês binary digit). Podemos ter circuitos com apenas um bit de saída como o caso de uma porta AND onde a saída poderá apenas apresentar o nível lógico "0" ou "1", temos outros como os contadores de quatro dígitos onde suas saídas poderão ser de "0000" à "1111".

Os microprocessadores são classificados como microprocessadores de 4 ou 8 ou 16 e até já começam a aparecer os de 32 bits.

Isto significa que estes números referem-se ao comprimento dos códigos das instruções utilizadas por estes circuitos, assim o microprocessador de 4 bits tem como código de instrução os números de "0000" à "1111"; o de 8 bits; os números de "0000 0000" à "1111 1111", etc.

Como atualmente são mais usados microprocessadores e demais circuitos com 8 bits, convencionou-se chamar-se ao grupo de 8 bits pelo nome de byte. Assim um byte pode ser por exemplo o seguinte número: 1011 0010.

III.3 SISTEMA HEXADECIMAL

O sistema binário de numeração é bastante conveniente para ser empregado em circuitos elétricos e ele é utilizado desde circuitos simples até grandes computadores. Isto trouxe um inconveniente devido ao grande número de dígitos necessários para se expressar um número.

Por exemplo, um programa para um microprocessador de 8 bits é composto por uma série de instruções cada uma delas composta por 8 dígitos. Para facilitar isto, é utilizado pelo programador o sistema numérico hexadecimal que é composto por dezesseis algarismos que são:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A representação de uma quantidade no sistema hexadecimal é feita utilizando o mesmo princípio de formação dos outros sistemas, porém, neste caso, a base é dezesseis e os algarismos utilizados são só indicados acima (de "0" à "F").

A correspondência entre o sistema hexadecimal e o decimal pode ser visto pela tabela a seguir:

| | | |
|-----------|-----|-----------|
| 0_{16} | $=$ | 0_{10} |
| 1_{16} | $=$ | 1_{10} |
| 2_{16} | $=$ | 2_{10} |
| 3_{16} | $=$ | 3_{10} |
| 4_{16} | $=$ | 4_{10} |
| 5_{16} | $=$ | 5_{10} |
| 6_{16} | $=$ | 6_{10} |
| 7_{16} | $=$ | 7_{10} |
| 8_{16} | $=$ | 8_{10} |
| 9_{16} | $=$ | 9_{10} |
| A_{16} | $=$ | 10_{10} |
| B_{16} | $=$ | 11_{10} |
| C_{16} | $=$ | 12_{10} |
| D_{16} | $=$ | 13_{10} |
| E_{16} | $=$ | 14_{10} |
| F_{16} | $=$ | 15_{10} |
| 10_{16} | $=$ | 16_{10} |
| 11_{16} | $=$ | 17_{10} |
| \vdots | | \vdots |

Por exemplo, o número $8E3_{16}$ significa:

$$\begin{array}{c|c|c} 16^2 & 16^1 & 16^0 \\ \hline 8 & E & 3 \end{array} \Rightarrow 8 \times 16^2 + E \times 16^1 + 3 \times 16^0$$

e para encontrarmos o seu correspondente no sistema decimal (sendo $E = 14$) temos:

$$\begin{aligned} 8E3_{16} &\Rightarrow 8 \times 16^2 + 14 \times 16^1 + 3 \times 16^0 = 2048_{10} + 224_{10} + 3_{10} = \\ &= 2275_{10} \end{aligned}$$

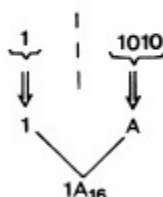
Sabemos que os sistemas digitais utilizam o sistema binário de numeração e com uma análise mais profunda podemos mostrar que a cada grupo de quatro dígitos de um número do sistema binário corresponde um dígito do sistema hexadecimal. Para exemplificar isto vamos analisar o seguinte $n^\circ 11010_2$.

O seu correspondente no sistema decimal é:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 26_{10}$$

portanto temos que: $11010_2 = 1A_{16}$

mas se tomarmos o nosso n° no sistema binário e o dividirmos da direita para a esquerda em grupos de quatro dígitos temos:



Sempre existirá apenas um dígito do sistema hexadecimal para representar quatro dígitos do sistema binário.

A razão da utilização deste outro sistema de numeração, para quem trabalha com microprocessador ou qualquer circuito digital, é que o binário corresponde diretamente aos sinais em cada ponto do circuito e o hexadecimal, por ser uma condensação do sistema binário, facilita a escrita e a leitura dos números, torna a tarefa de programação mais simples e possibilita a conversão dígito a dígito, direta para o sistema binário.

III.4 CONVERSÃO DE SISTEMAS DE NUMERAÇÃO

Vamos ver algumas regras práticas para a conversão de sistemas. A conversão de uma quantidade do sistema binário ou hexadecimal para decimal já foi por nós exemplificada quando da apresentação destes sistemas, basta multiplicarmos o equivalente decimal de cada dígito do sistema por sua base elevada à potência correspondente à localização do dígito e somarmos o resultado de todas as multiplicações.

Exemplo: para o sistema binário:

$$\begin{aligned} 1001101_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + \\ &0 \times 2^1 + 1 \times 2^0 = 64 + 8 + 4 + 1 = 77_{10} \end{aligned}$$

ou de maneira mais simples, basta escrevermos acima de cada digito o valor correspondente a sua localização e somarmos estes valores onde temos o dígito "1".

$$\begin{array}{ccccccc} 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array} \quad l_2 = 64 + 8 + 4 + 1 = 77_{10}$$

de maneira análoga para o sistema hexadecimal temos:

$$3F5 = 3 \times 16^2 + 15 \times 16^1 + 5 \times 16^0 = 3 \times 256 + 15 \times 16 + 5 \times 1 = 1013_{10}$$

Para a conversão de decimal para binário sugerimos a seguinte regra prática: subtraímos do n° a ser convertido em binário a quantidade de 2 elevado ao maior expoente desde que não ultrapasse o valor de nosso n° , por exemplo para o n° 211_{10} o valor a ser subtraído é 128 (2^7), $211 - 128 = 83$, a seguir escrevemos em uma linha de maneira decrescente o maior expoente encontrado até o zero e colocamos o dígito 1 abaixo do maior expoente.

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

Procedemos da mesma maneira com o resto da subtração até termos zero como resultado. Assim sendo:

$$83 - 64 (2^6) = 19$$

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & & & & & & \end{array}$$

$$19 - 16 (2^4) = 3$$

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & & 1 & & & & \end{array}$$

$$3 - 2 (2^1) = 1$$

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & & 1 & & 1 & & \end{array}$$

$$1 - 1 (2^0) = 0$$

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & & 1 & & & 1 & 1 \end{array}$$

preenchemos o restante das casas com zero e temos o n° final:

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

$$211_{10} = 1101\ 0011_2$$

vamos fazer o outro exemplo:

$$\begin{array}{l} 78_{10} \\ 78 - 64 (2^6) = 14 \\ 14 - 8 (2^3) = 6 \\ 6 - 4 (2^2) = 2 \\ 2 - 2 (2^1) = 0 \\ \begin{array}{ccccccc} 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \\ 78_{10} = 100\ 1110_2 \end{array}$$

Para convertermos um número de decimal para hexadecimal sugerimos a conversão deste número para binário, depois o agrupamento deste número em grupos de quatro dígitos e a conversão de cada grupo diretamente para hexadecimal. Vamos fazer a conversão para hexadecimal dos números decimais vistos anteriormente.

$$211_{10} = 1101\ 0011_2$$

sendo

$$1101_2 = 13_{10} = D_{16}$$

$$0011_2 = 3_{10} = 3_{16}$$

portanto

$$211_{10} = 1101\ 0111_2 = D\ 3_{16}$$

$$78_{10} = 0100\ 1110_2$$

sendo

$$0100_2 = 4_{10} = 4_{16}$$

$$1110 = 14_{10} = E_{16}$$

portanto

$$78_{10} = 0100\ 1110_2 = 4E_{16}$$

Vamos fazer alguns exercícios englobando todas as con
versões vistas anteriormente:

1 - Conversão de binário para decimal.

$$0100\ 1101\ 1110_2 =$$

$$0 \times 2^{11} + 1 \times 2^{10} + 0 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1024 + 128 + 64 + 16 + 8 + 4 + 2 = 1246_{10}$$

2 - Conversão de hexadecimal para decimal.

$$78A5_{16} =$$

$$7 \times 16^3 + 8 \times 16^2 + 10 \times 16^1 + 5 \times 16^0 =$$

$$7 \times 4096 + 8 \times 256 + 10 \times 16 + 5 =$$

$$28672 + 2048 + 160 + 5 = 30885_{10}$$

3 - Conversão de decimal para binário.

$$452_{10}$$

$$452 - 256 (2^8) = 196$$

$$2^8\ 2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$

$$1$$

$$196 - 128 (2^7) = 68$$

$$2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$1 \quad 1$$

$$68 - 64 (2^6) = 4$$

$$2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$1 \quad 1 \quad 1$$

$$4 - 4 (2^2) = 0$$

$$2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0$$

$$452_{10} = 1 \ 1100 \ 0100_2$$

4 - Conversão de decimal para hexadecimal.

$$924_{10} =$$

$$11 \ 1001 \ 1100_2 = 39C_{16}$$

5 - Conversão do hexadecimal para binário.

$$784_{16}$$

$$7_{16} = 0111_2$$

$$8_{16} = 1000_2$$

$$4_{16} = 0100_2$$

$$784_{16} = 0111 \ 1000 \ 0100_2$$

6 - Conversão de binário para hexadecimal.

$$1001 \ 1111 \ 0010 \ 0110_2$$

$$1001_2 = 9_{16}$$

$$1111_2 = F_{16}$$

$$0010_2 = 2_{16}$$

$$0110_2 = 6_{16}$$

$$1001 \ 1111 \ 0010 \ 0110_2 = 9F26_{16}$$

O objetivo deste capítulo é o de relembrar os principais blocos lógicos, suas funções de transferência, bem como exemplos de aplicações dos mesmos, para que possamos, sem qualquer dúvida, compreender o que será desenvolvido nos capítulos a seguir.

Outra importante observação é a de que os níveis lógicos aqui representados por "A", "B" ou "Z", correspondem a níveis reais de tensão, onde nível "1" poderia por exemplo assumir o valor de 5 Volts DC ou ainda nível "0" o valor de zero Volts DC.

Feitas estas considerações, passemos aos blocos, com alguns exemplos de aplicações também.

Bloco OR (OU)Função de TransferênciaTabela da verdade

$$Z = A + B$$

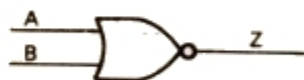
| A | B | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Como podemos verificar através da tabela, teremos "Z" = "0", apenas quando as duas entradas "A" e "B" estiverem também a nível lógico zero, caso contrário teremos "Z" = "1" em quaisquer outras situações.

Exemplo:

Realizar através de blocos "OR" de duas entradas, a função $Z = A + B + C + D$.



Bloco NOR (NOU)Função de TransferênciaTabela da verdade

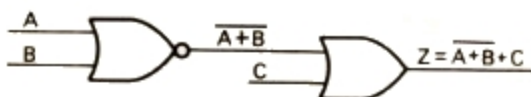
$$Z = \overline{A + B}$$

| A | B | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Como podemos notar através da tabela, teremos "Z" = "1" somente se "A" = "B" = "0". Outrossim, o bloco apresentado realiza a função inversa a do "OR".

Exemplo:

Realizar através de blocos "NOR" e "OR" de duas entradas, a função $Z = \overline{A + B} + C$

Bloco AND (E)Função de TransferênciaTabela da Verdade

$$Z = A.B$$

| A | B | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Da tabela apresentada, verificamos que teremos "Z" = "1" apenas se as duas entradas estiverem no nível lógico "1".

Exemplo:

Realizar através de blocos "AND" de duas entradas, a função $Z = A.B.C.D$



Bloco NAND (NE)

Função de Transferência

Tabela da verdade



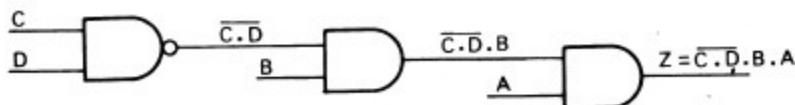
$$Z = \overline{A \cdot B}$$

| A | B | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Podemos notar pela tabela apresentada, que o bloco "NAND" realiza a função inversa ao "AND".

Exemplo:

Realizar através de blocos lógicos "NAND" e "AND" de duas entradas, a função $Z = A \cdot B \cdot \overline{C} \cdot \overline{D}$



Bloco Inversor

Função de Transferência

Tabela da verdade



$$B = \overline{A}$$

| A | B |
|---|---|
| 0 | 1 |
| 1 | 0 |

Da tabela da verdade, notamos que o bloco inversor apresenta na saída um nível lógico oposto ao da entrada.

Obs.:

Podemos utilizar como bloco inversor os blocos "NOR" e "NAND" já apresentados, ligados conforme as ilustrações a seguir.



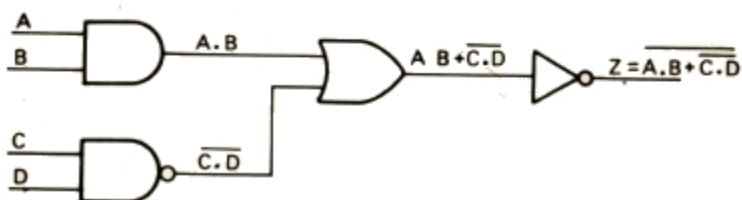
Bloco NAND
como inversor



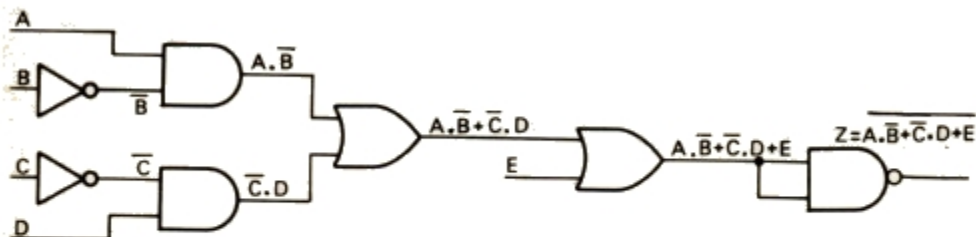
Bloco NOR
como inversor

Exemplo:

1) Realizar a função $Z = \overline{A \cdot B + \overline{C} \cdot D}$



2) Realizar a função $Z = \overline{A \cdot \overline{B} + \overline{C} \cdot D + E}$



Obs.:

Foi utilizado para inverter a função " $A \cdot \overline{B} + \overline{C} \cdot D + E$ " um bloco "NAND" de duas entradas ligado como inversor.

Bloco EXCLUSIVE OR
(OU EXCLUSIVO)



$$Z = A \oplus B$$

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

Tabela da Verdade

| A | B | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Bloco Coincidência



$$Z = A \odot B$$

$$= \bar{A}\bar{B} + AB$$

Função de Transferência

Tabela da verdade

| A | B | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Bloco Biestável

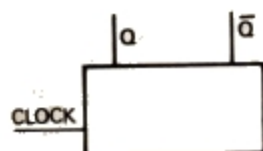
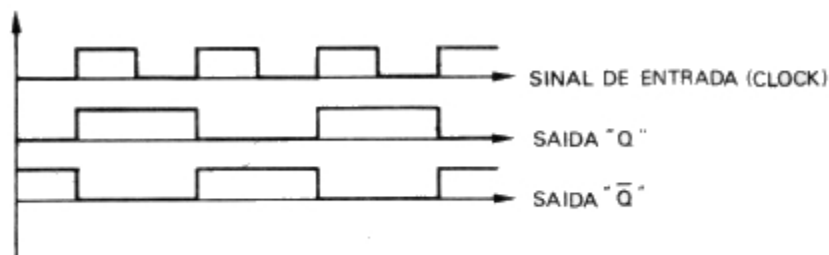


Tabela da Verdade

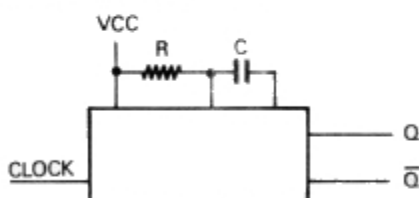
| Clock | Q | \bar{Q} | Observações |
|-------|---|-----------|--------------------|
| X | 0 | 1 | Estado inicial |
| | 1 | 0 | Transição de clock |
| | 0 | 1 | 2ª Trans. de clock |

Pela tabela da verdade notamos que, no estado inicial, a saída "Q" está no nível zero e " \bar{Q} " no nível 1. Na primeira transição de clock (nível zero para nível um), temos que a saída "Q" vai para nível um, cuja saída é estável, alterando-se em correspondência com as variáveis da entrada (clock).

Diagrama de Tempos de bloco Biestável

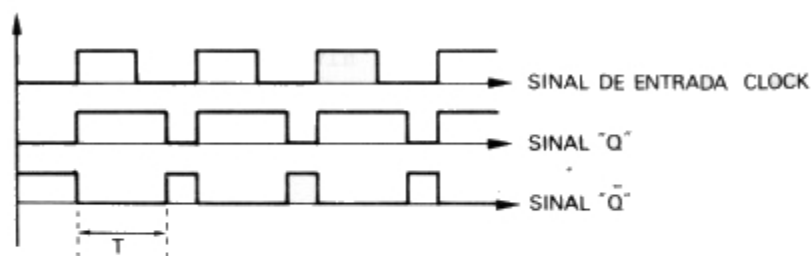


Bloco Monoestável



O monoestável é um circuito cuja saída é alterada com a variação do sinal de entrada porém, após um intervalo de tempo pré-determinado pelo circuito, volta ao estado inicial anterior a transição.

Diagrama de Tempos de Bloco Monoestável



O intervalo de tempo em que a saída "Q" passa de nível zero a nível um é determinado pela expressão $T = 0,69.R.C$,

onde T é o valor desejado e, fixando-se "R" ou "C" determinaremos "C" ou "R" respectivamente.

Bloco decodificador



Decodificador é o circuito que transforma uma informação de entrada (por exemplo binária) em uma diferente informação de saída (por exemplo hexadecimal).

Exemplo:

Decodificador para display de 7 segmentos.

Tabela da Verdade

| Entrada | | | | | Saída | | | | | | | Hexadecimal | |
|---------|---|---|---|--|-------|---|---|---|---|---|---|----------------|--|
| A | B | C | D | | a | b | c | d | e | f | g | Correspondente | |
| 0 | 0 | 0 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | |
| 0 | 0 | 0 | 1 | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | |
| 0 | 0 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 | |
| 0 | 1 | 0 | 0 | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 4 | |
| 0 | 1 | 0 | 1 | | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 | |
| 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 6 | |
| 0 | 1 | 1 | 1 | | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 7 | |
| 1 | 0 | 0 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 | |
| 1 | 0 | 0 | 1 | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 9 | |
| 1 | 0 | 1 | 0 | | 1 | 1 | 1 | 0 | 1 | 1 | 1 | A | |
| 1 | 0 | 1 | 1 | | 0 | 0 | 1 | 1 | 1 | 1 | 1 | B | |
| 1 | 1 | 0 | 0 | | 1 | 0 | 0 | 1 | 1 | 1 | 0 | C | |
| 1 | 1 | 0 | 1 | | 0 | 1 | 1 | 1 | 1 | 0 | 1 | D | |
| 1 | 1 | 1 | 0 | | 1 | 0 | 0 | 1 | 1 | 1 | 1 | E | |
| 1 | 1 | 1 | 1 | | 1 | 0 | 0 | 0 | 1 | 1 | 1 | F | |

Exemplo:

$$\text{N}^{\circ} 2 \begin{cases} a & b & c & d & e & f & g \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{cases}$$

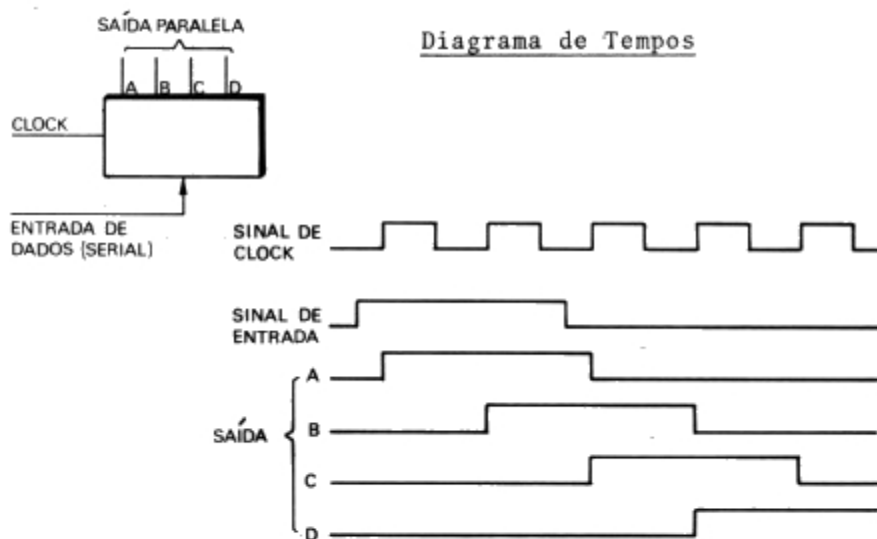
$$\begin{array}{c|c} a & \\ \hline g & b \\ \hline e & \\ \hline d & \end{array}$$

Shift Register (registrador de deslocamento)

Existem dois tipos básicos de Registradores de deslocamento, que são o Registrador Serial e o Paralelo.

Registrador de Deslocamento Serial

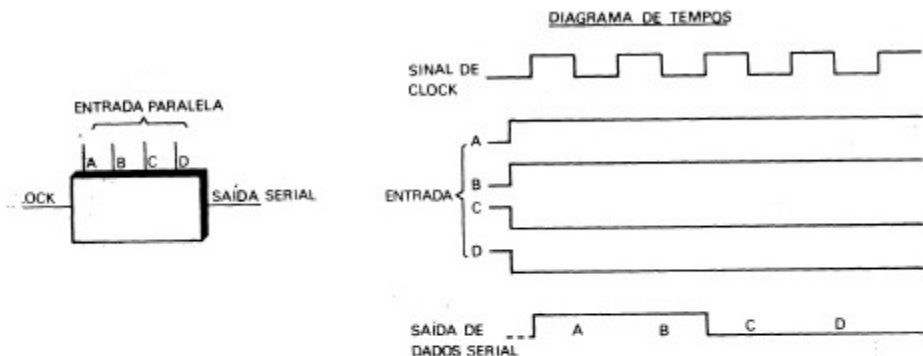
Este elemento transfere, através da variação do sinal de clock, um sinal presente na entrada para uma das saídas. À medida que o sinal de clock varia (nível zero para nível um, por exemplo), o sinal passa de uma das saídas para a outra e, assim, sucessivamente.



Registrador de Deslocamento Paralelo

Este elemento recebe dados paralelamente e os desloca serialmente à saída, operação comandada através das transições

de clock.

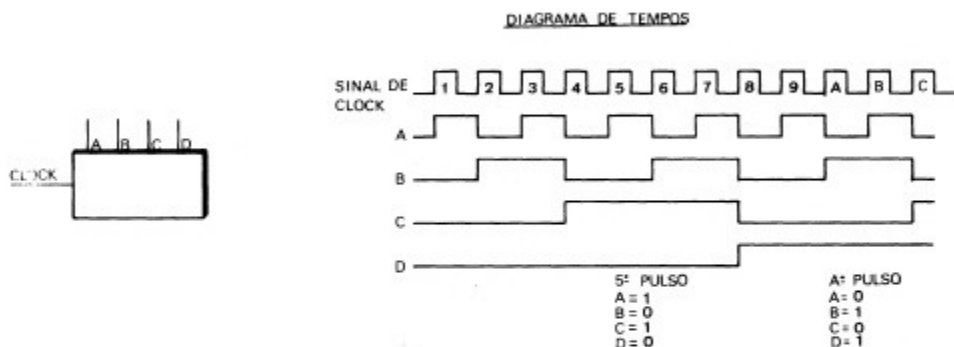


Sinal de Entrada = 1 1 0 0

Sinal de Saída = 1 1 0 0

Contadores

Os contadores, ou divisores de frequência, são elementos utilizados na contagem de pulsos. Normalmente, como sabemos, necessitamos de 4 bits para expressarmos um número hexadecimal. Dessa forma, os contadores aparecem em blocos de 4 bits cada, para que possamos expressar um dado número na base 16 (hexadecimal).



Como podemos notar no diagrama apresentado, na saída

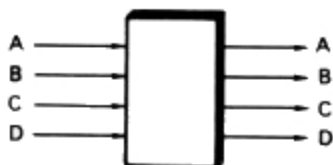
"A" teremos o sinal de clock dividido por 2, na saída B dividido por 4, na saída C dividido por 8 e na D por 16. Se traçarmos por um dos pulsos de clock perpendicular como o fizemos no 5º pulso, por exemplo, iremos notar que os correspondentes valores de A, B, C e D nos fornecerão em binário o correspondente ao número de pulsos de clock.

Exemplo:

5º Pulso

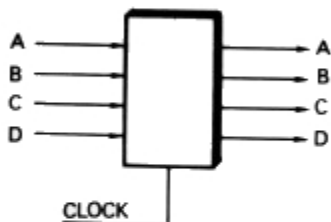
| | | |
|-------|---------|---------------|
| A = 1 | D C B A | } = 5 em hexa |
| B = 0 | 0 1 0 1 | |
| C = 1 | | |
| D = 0 | 8 4 2 1 | |

Buffer



Os buffers são circuitos que simplesmente colocam os dados de suas entradas em suas saídas. Os buffers são utilizados quando um circuito não possui capacidade de corrente suficiente para acionar os circuitos ligados a suas saídas. Neste caso, intercalamos entre estes dois circuitos o buffer.

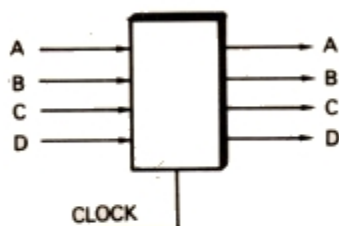
Latch



Os latches são circuitos que quando recebem um sinal de clock, transferem os dados de suas entradas para as saídas e mantêm estas saídas fixas até que haja um outro sinal de clock,

mesmo que as entradas variem.

Porta



As portas são circuitos integrados que combinam as funções de buffer e latch, isto é, transferem as informações das entradas para as saídas quando recebem um sinal de clock e possuem nível de corrente de saída suficiente para aplicar informações em vários outros circuitos.

V.1 ESTRUTURA DAS MEMÓRIAS

O conjunto das memórias forma um dos blocos básicos do sistema de computador, onde são guardadas as informações para serem processadas. Estas informações são codificadas na forma binária, para seu armazenamento.

Vamos fazer uma apresentação de uma memória hipotética que possui dois bits para seu endereçamento (A_1 , A_0) e um bit de dado (D_0). Com dois bits de endereçamento podemos selecionar quatro endereços diferentes, conforme as seguintes variações possíveis:

| A_0 | A_1 | |
|-------|-------|-------------|
| 0 | 0 | 1º endereço |
| 0 | 1 | 2º endereço |
| 1 | 0 | 3º endereço |
| 1 | 1 | 4º endereço |

Desta maneira, em cada endereço, temos uma informação armazenada. Supondo-se que nos dois primeiros endereços temos armazenada a informação "1" e, nos dois seguintes, a informação "0", podemos representar esta memória conforme a figura V.1.

| | |
|------------------|------------------|
| (00) 1 | (01) 1 |
| (10) 0 | (11) 0 |

Fig. V.1 Memória de 4 bits.

Portanto, se quisermos ler o conteúdo da memória em um determinado endereço, basta fornecer o endereço desejado para a memória através do bus de endereço (A_1 , A_0) e teremos disponível no bus

de dados (D_0) o seu conteúdo. Esta será uma memória de 4×1 (4 endereços por 1 dado).

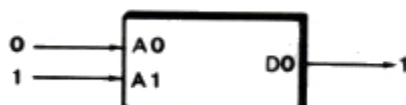


Fig. V.2 Leitura de memória.

Para entender como uma memória retém uma informação vamos representá-la conforme a estrutura da figura V.3.

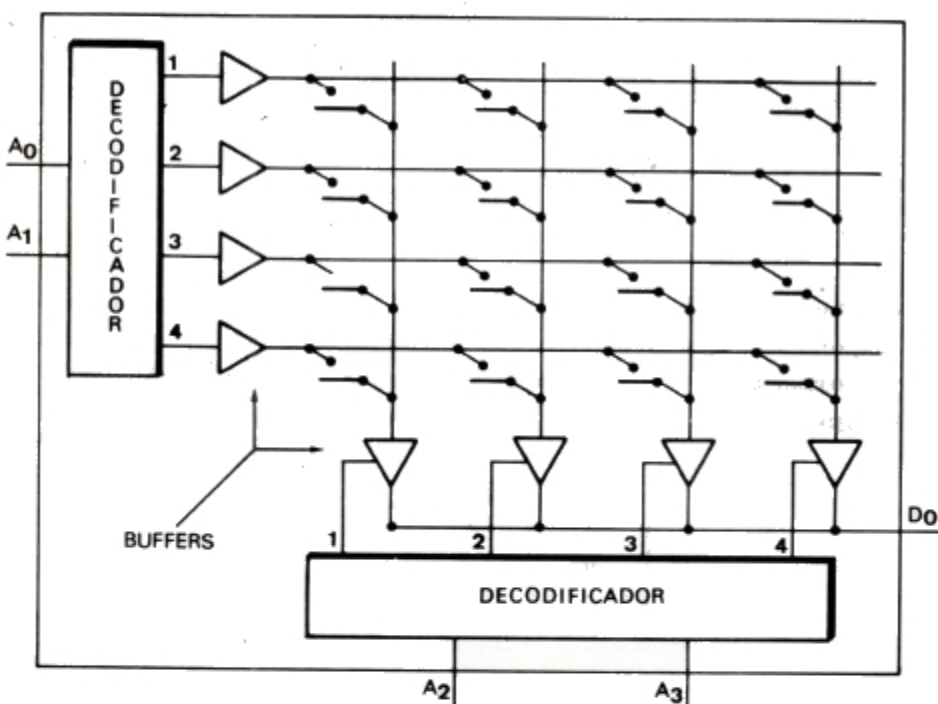


Fig. V.3 Memória de 16×1 .

Cada posição da memória pode ser representada por uma chave que pode estar aberta, para quando a informação armazenada for "zero" ou fechada, para quando a informação armazenada for "um". O decodificador colocará um em uma de suas saídas e zero nas demais conforme a configuração de suas entradas que são os bits de endereço. Assim quando as entradas forem: $A_1 = 1$, $A_0 = 0$ teremos "um" na saída "2" e zero nas demais, e assim por diante.

Quando tivermos uma certa configuração dos bits A_1 e A_0 , teremos uma determinada linha selecionada e com uma certa configuração dos bits A_3 e A_2 , temos uma determinada coluna selecionada. Caso a chave da intersecção da linha com a coluna selecionada esteja fechada, teremos na saída D_0 a informação "um" e, se esta chave estiver aberta, teremos a informação "zero". Desta maneira podemos escrever informações nesta memória fechando ou abrindo as chaves nas posições desejadas e quando quisermos ler o conteúdo de determinada posição, basta configurarmos os bits de entrada para selecionarmos a posição e o seu conteúdo estará disponível no bit de saída.

Tomando-se duas dessas memórias que acabamos de representar e interligando-se os bits de entrada (A_0 de uma com A_0 da outra, A_1 de uma com A_1 de outra, assim por diante), teremos para cada endereço selecionado dois bits de saída, um de cada memória. No total temos uma memória de 32 bits ou 16×2 , isto é, 16 posições diferentes com 2 informações por posição. Caso desejemos uma memória com mais posições de armazenamento de informações, basta aumentarmos o número de bits de entrada e, se desejarmos mais informações por posição, basta aumentarmos o número de planos desta memória.

Uma memória genérica de x endereços ($m + n$) por y informações por endereço pode ser representada conforme a figura V.4. O tamanho desta memória é $2^x \times y$ endereços.

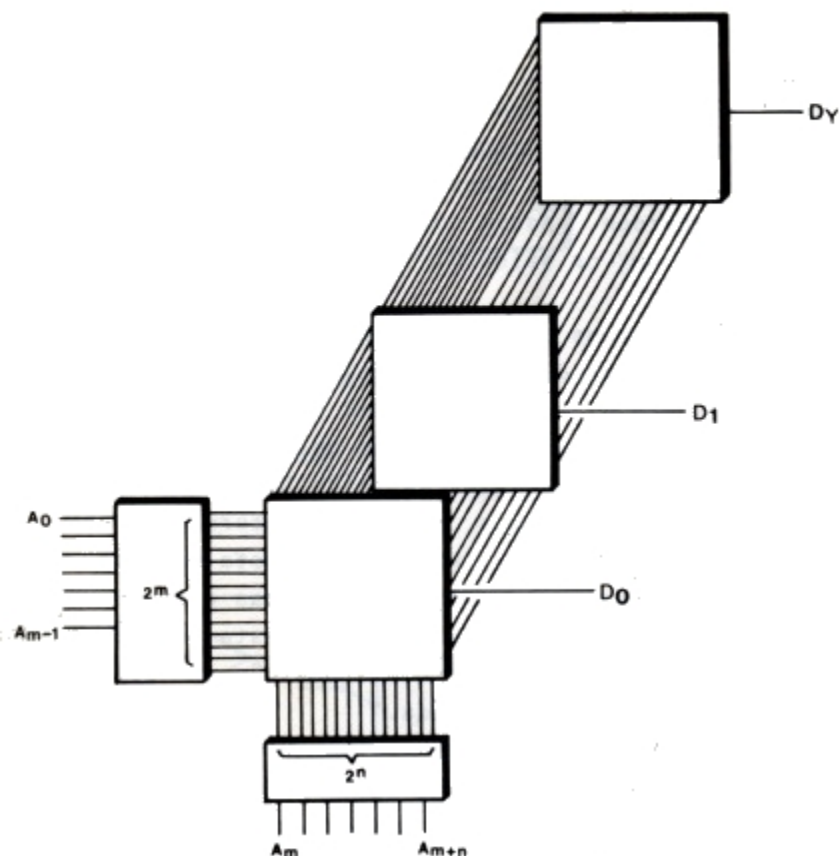


Fig. V.4 Memória de $2^x \times y$.

2.2 MEMÓRIAS À SEMICONDUTORES.

As memórias à semicondutores podem ser divididas em dois grupos diferentes:

- Memória ROM (read only memory) memória apenas de leitura.
- Memória RAM (random access memory) memória de acesso aleatório.

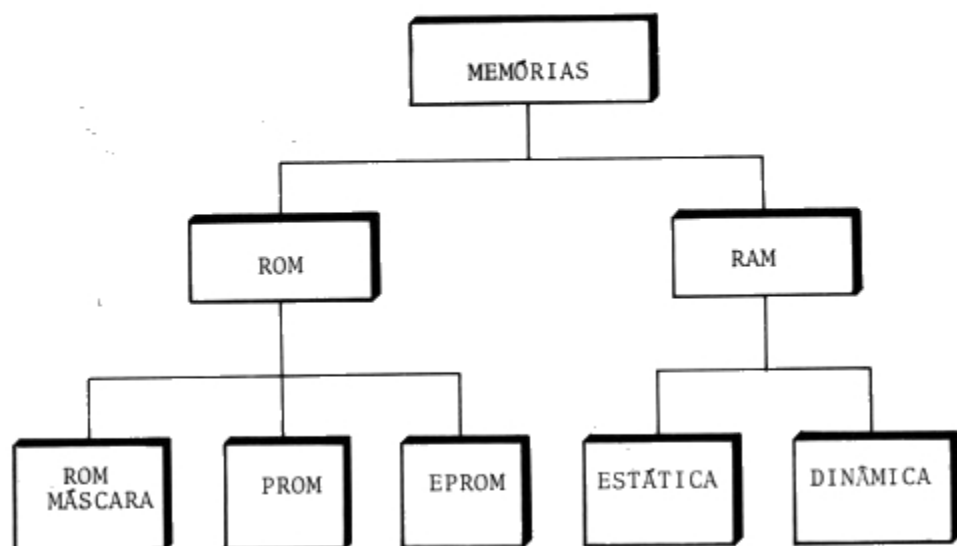


Figura V.5 Tipos de memórias.

As memórias ROM são designadas como memórias de programa por serem memórias que não podem ser alteradas pelo programa, porém têm a vantagem de não perderem as suas informações mesmo quando é desligada sua alimentação. As memórias ROM são utilizadas para armazenar os programas ou dados que não necessitam de ser alterados.

Entre os principais tipos de memórias ROM podem ser destacados os seguintes:

- ROM mascara são memórias nas quais as informações são gravadas na sua fabricação, por isto o nome de ROM mascara. O conteúdo de cada posição da memória ROM mascara é determinado antes de sua fabricação e cada posição é programada na sua fabricação, não podendo mais ser alterada.

- PROM (programmable read only memory) memória apenas de leitura programável; são memórias que podem ser eletricamente programáveis, porém, após programadas, o seu conteúdo não pode ser mais alterado.

- EPROM (erasable programmable read only memory) memória apenas de leitura programável apagável; são memórias que podem ser eletricamente programáveis e podem ser apagadas mais tarde, para serem reutilizadas com uma nova programação. Para serem apagadas estas memórias devem ser expostas à luz ultravioleta.

As memórias RAM são designadas como memórias de dados e são memórias que podem ser lidas ou gravadas pelo programa. As memórias RAM são utilizadas para armazenar temporariamente dados que são alterados no decorrer do programa. Qualquer informação que temos na unidade central de processamento, podemos escrever em uma memória RAM e, mais tarde, quando necessitarmos, basta lermos esta informação nessa memória.

As memórias RAM podem ser divididas em dois grupos:

Dinâmicas: são memórias nas quais as informações vão gradativamente desaparecendo, portanto após um certo tempo necessitam ser regravadas. Existem circuitos integrados especiais que de tempo em tempo lêem estas memórias e as regravam. Estes circuitos integrados são chamados de circuitos de refresh.

Estáticas: são memórias que retêm suas informações enquanto permanecer a sua alimentação, não sendo necessário que suas informações sejam regravadas de tempo em tempo.

Para que tenhamos um contato mais direto com algumas memórias, vamos descrever duas memórias que consideramos como sendo as mais usadas. Estas memórias escolhidas são a RAM estática 2114 e a EPROM 2716. Existe, porém, um grande número de tipos de memórias diferentes que são escolhidas de maneira diferente, dependendo da aplicação, por terem capacidade de armazenamento e custos diferentes.

V.3 ESTRUTURA DA EPROM 2716

Como vimos anteriormente, as memórias EPROM são gra

vadas eletricamente e podem ser apagadas por exposição à luz ultravioleta para serem reaproveitadas com novas informações. Entre as diferentes memórias EPROM vamos analisar a 2716.

Esta memória possui 16384 bits formados por 11 bits de endereço contendo cada um 8 bits de dados, porém, para simplificarmos esta informação, vamos representar cada 1024 bits ($=2^{10}$) por 1k, desta maneira o tamanho desta memória é de 16 k ou ainda 2k x 8 bits. O tempo máximo para que após estabilizadas as informações de endereço até que as informações de dados estejam estabilizadas é de 450 ns ($=450 \times 10^{-9}$ segundos).

O tempo para a programação dos 16384 bits é de 100 segundos e a tensão de programação é de nível TTL (isto é +5Volts).

Resumindo, a 2716 possui as seguintes características:

Tamanho: 2k x 8 bits

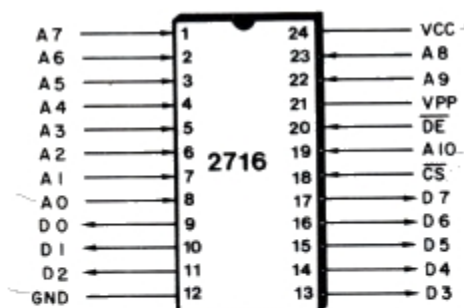
Alimentação: fonte única de +5 Volts.

Tempo de acesso: 450ns máximo.

Tempo de programação total: 100 segundos.

Nível de tensão de programação: nível TTL

Descrição da pinagem:



V_{CC}, GND

Alimentação

V_{CC} = + 5 Volts ± 5%

GND = referência 0 Volts

A₁₀ - A₀ (entrada)

Bus de endereço

Bus de dados

CS (entrada)

Seleção de circuito, um sinal neste pino habilita a comunicação entre a memória e a unidade central de processamento.

DE (entrada)

Habilitação do bus de dados. Para uma leitura da 2716 devemos aplicar a configuração dos endereços desejado no bus de endereço, aplicarmos "0" no pino CS para acionamento elétrico de seleção do circuito integrado e, finalmente, aplicarmos "0" no pino DE para habilitarmos os buffers do bus de dados, após isto as informações estarão disponíveis para leitura.

V_{pp}

Tensão de programação: durante a programação da 2716 um nível de tensão de + 25 Volts deve ser aplicado a este pino. Em operação normal devemos ter este pino conectado diretamente ao pino V_{CC}.

V.4 ESTRUTURA DA RAM 2114

As memórias RAM são memórias que podem ser lidas ou gravadas a qualquer momento pela unidade central de processamento sem que seja necessário níveis de tensão especiais para a gravação ou qualquer sistema especial para apagamento. O único inconveniente maior destas memórias é que seu conteúdo é perdido quando desligamos a fonte de alimentação. Entre as memórias RAM selecionamos para análise mais detalhada a 2114. Esta memória possui apenas um pino para diferenciar a operação de leitura ou escrita e usa pinos comuns para entrada ou saída de dados.

As principais características da 2114 são:

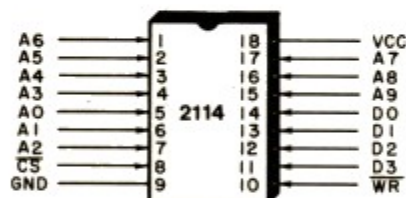
Tamanho: 1k x 4

Tempo de acesso: 450 ns máximo

Alimentação: fonte única de + 5 Volts

Tipo: memória estática.

Descrição da pinagem:



V_{CC}, GND

Alimentação

V_{CC} = + 5 Volts \pm 5%

GND = referência 0 Volts

A₉ - A₀ (entrada)

Bus de endereço

D₃ - D₀ (entrada/saída)

Bus de dados bidirecional

CS (entrada)

Seleção do circuito: um sinal neste pino habilita a comunicação entre a memória e a unidade central de processamento.

WR (entrada)

Escrita: Para leitura da memória colocamos nível "1" neste pino e para escrita colocamos nível "0".

VI.1 APRESENTAÇÃO DOS MICROPROCESSADORES

Vamos estudar a arquitetura do microprocessador, isto é, de que é constituído e como é sua estrutura interna.

Para simplificar este estudo vamos nos restringir a um microprocessador específico, isto porque achamos ser mais didático do que um estudo de forma genérica e mais fácil uma aplicação prática.

Não acreditamos ser esta particularização inconveniente devido ao elevado número de microprocessadores disponíveis atualmente no mercado, pois entendendo-se um microprocessador específico, torna-se bastante simples o entendimento de outros através apenas de manuais dos fabricantes ou outros tipos de literaturas, pois as diferenças são pequenas e as estruturas básicas semelhantes. O que também difere de um microprocessador para outro são as instruções, cada fabricante possui o seu conjunto específico com uma estrutura básica semelhante, apenas diferindo por algumas instruções especiais, devido fundamentalmente as diferenças de arquitetura.

Atualmente dispomos de um número incalculável de microprocessadores, desde aqueles de propósito geral até aqueles de propósito específico, também um grande número de fornecedores para um mesmo microprocessador. Assim podemos resumir, apresentando os microprocessadores de uso geral e seus fornecedores mais populares encontrados hoje no mercado.

| Fabricante | Microprocessador (número do modelo) |
|------------|-------------------------------------|
| Intel | 8080A, 8085, 8086 |
| Zilog | Z80, Z 8000 |
| Motorola | 6800, 6802, 6809 |
| Signetics | 2650A |

Tabela VI.1

Cada modelo de microprocessador apresenta características particulares que podem representar alternativas para diferentes tipos de aplicação, algumas destas características podem ser vistas na tabela a seguir.

| 8004A | 8085 | 8086 | 8080 | 8080 | 8082 | 8086 | 286 | TECNOLOGIA |
|-----------|----------|----------|--------|-------|-------|-------|---------|---|
| N MOS | N MOS | N MOS | N MOS | N MOS | N MOS | N MOS | N MOS | voltagens requeridas |
| 5, 12, -5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | encapsulamento (nº de pinos) |
| 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | máxima frequência de clock (MHz). |
| 2,6 | 5 | 5 | 4 | 2 | 2 | 2 | 1.2 | tempo de instrução (menor/menor) us. |
| 1.5/3.75 | 1.3/5.85 | 0.4/40.4 | 1/5.75 | 1/2.5 | 2/5 | 2/5 | 4.8/9.6 | tamanho de palavra (dado/instrução) |
| 8/8 | 8/8 | 16/16 | 8/8 | 8/8 | 8/8 | 8/8 | 8/8 | capacidade de endereçamento direto (palavras). |
| 64k | 64k | 1M | 64k | 64k | 64k | 64k | 32k | capacidade de acesso direto de memória |
| sim | sim | sim | sim | sim | sim | sim | sim | número de instruções básicas |
| 78 | 80 | 133 | 150 | 89 | 89 | 100 | 75 | número de registradores de propósito, geral internos. |
| 8 | 8 | 14 | 0 | 0 | 0 | 0 | 7 | |

Tabela VI.2

Entre todos os microprocessadores, escolhemos como objetivo de nosso estudo os microprocessadores 8080 e 8085, desenvolvidos pela Intel e fabricados por outros como: AEG, ADVANCED Micro Devices, National, NEC, Phillips, SGS-ATES, Siemens, Signetics, Texas e outros. A escolha destes modelos particulares não se deve ao fato de os considerarmos os melhores, pois acreditamos já existirem outros mais complexos, ou como intuito de propaganda, pois este não é nosso objetivo; mas por serem os mais popu

lares, por serem basicamente os pioneiros dos microprocessadores e por não apresentarem sofisticacões extremas de difícil compreensão.

Para estudarmos o 8080 não vamos estudar apenas o circuito integrado 8080, mas também os outros circuitos integrados necessários para a utilização do 8080 ou ainda, suas interligações para que, conjuntamente, possam formar um sistema completo. O microprocessador 8085 será examinado em maiores detalhes no capítulo IX.

VI.2 DESCRIÇÃO DA PINAGEM DO MICROPROCESSADOR 8080

O microprocessador 8080 é uma unidade central de processamento em um único circuito integrado de 40 pinos fabricado em tecnologia MOS de canal N. Sua configuração dos pinos pode ser vista na figura VI.1.



Fig. VI.1 Configuração dos pinos.

V_{SS}, V_{BB}, V_{CC}, V_{DD}

Alimentação: Este microprocessador requer para seu funcionamento três níveis de tensão de alimentação e ainda o sinal de terra.

V_{SS} → referência 0 Volts
 $V_{BB} = - 5 \pm 5 \%$ Volts
 $V_{CC} = + 5 \pm 5 \%$ Volts
 $V_{DD} = +12 \pm 5 \%$ Volts

ϕ_1, ϕ_2 (entrada)

Clock: O 8080 é um dispositivo dinâmico que necessita de um sinal de clock (relógio de referência) e trabalha com duas diferentes fases que são geradas externamente (ver gerador de clock 8224 no capítulo VII) e não são compatíveis com o nível lógico TTL.

$A_{15} - A_0$ (saída)

Bus de endereço: são dezesseis saídas three-state (três estados), isto é, possuem os níveis lógicos "0" ou "1" ou então podem estar desativadas apresentando em cada pino uma alta impedância. Estas dezesseis saídas formam dezesseis linhas que constituem o bus de endereço. (A palavra bus pode ser traduzida para o português como "via", mas usaremos o termo em inglês "bus" pois acreditamos ser mais difundido). Estas saídas são utilizadas para endereçar memórias ou dispositivos de entrada e saída (I/O).

O máximo de memória capaz de ser endereçável é de 64k (65536) e o máximo de dispositivos de entrada e saída, é de 512, pois só é possível utilizar-se de 8 bits.

A_0 é o bit de endereço menos significativo e A_{15} é o mais significativo.

$D_7 - D_0$ (entrada/saída)

Bus de dados: são oito pinos de entrada e saída three-state que formam as oito linhas do bus de dados. O bus de dados forma uma via de comunicação bi-direcional entre o microprocessador e as memórias ou os dispositivos de entrada e saída para a transferência de dados ou instruções.

Além disso, durante o primeiro ciclo de clock de cada ciclo de máquina, estes pinos D₇ - D₀ possuem informações que geram uma palavra de controle, palavra esta que será interpretada pelo controlador do sistema (ver 8228 capítulo VIII) que por sua vez produzirá os sinais de controle.

Para podermos entender porque em determinado instante estes pinos de entrada e saída de dados são utilizados para informações de controle, recordemos o capítulo II onde foi exposto que a unidade central de processamento possui três vias diferentes de comunicação com as memórias e os dispositivos de entrada e saída: o bus de endereço, bus de dados e bus de controle. Assim sendo, deveríamos ter pinos específicos para cada um destes bus, o que sem dúvida traria um número muito elevado de pinos em um único circuito integrado.

O artifício utilizado pelo fabricante para eliminar este problema é termos nestes pinos D₇- D₀ em determinado instante dados ou instruções e em outro instante variáveis de controle que serão enviadas a outro circuito integrado e neste outro circuito integrado teremos o bus de controle.

SYNC (saída)

Sinal de sincronização: sinal para indicar aos circuitos externos o início de cada ciclo de máquina.

DBIN (saída)

Bus de dados como entrada: sinal para indicar aos circuitos externos que o bus de dados está no modo entrada.

WAIT (saída)

Espera: sinal que indica que o microprocessador está em estado de espera, aguardando comando externo. Isto é necessário para que os circuitos externos tenham conhecimento de quando a unidade central de processamento terminou a tarefa atual e está disponível para uma nova tarefa.

WR (saída)

Escrita: sinal que é usado para controle de escrita em memória ou dispositivo de entrada e saída. O dado no bus de dados permanece estável, enquanto o sinal WR estiver no nível lógico "0".

HOLD (entrada)

Retenção: Um sinal no pino HOLD faz com que a unidade central de processamento fique no estado de retenção, isto é, tão logo o 8080 termine seu atual ciclo de máquina este colocará o bus de dados e o bus de endereço no estado de alta impedância, enviará pelo pino HLDA um sinal, significando que o 8080 admite, a partir deste sinal, o estado de retenção e permite desta maneira que um dispositivo externo adquira controle do bus de dados e do bus de endereço.

Isto pode ser útil, por exemplo, quando existir um dispositivo para controle de acesso direto à memória por um dispositivo de entrada e saída.

HLDA (saída)

Indicação de estado de retenção: sinal que indica aos dispositivos externos que o 8080 está em estado de retenção. Este sinal aparece em resposta ao sinal recebido pelo pino HOLD e indica que o bus de dados e o bus de endereço encontra-se em estado de alta impedância.

READY (entrada)

Pronto: um sinal no pino READY indica para o 8080 que um dado proveniente de memória ou um dispositivo de entrada e saída está disponível no bus de dados para ser lido pelo 8080.

Quando o 8080 enviar um endereço pelo bus de endereço e não receber um sinal de READY, este entrará em estado de WAIT e permanecerá enquanto a entrada READY estiver em nível bai

xo. Tão logo o dado esteja disponível a entrada READY irá para nível alto e o 8080 poderá ler este dado. Este sinal é usado para sincronizar a unidade central de processamento com memórias ou dispositivos de entrada e saída lentos.

INTE (saída)

Interrupção habilitada: sinal que indica que a unidade central de processamento está habilitada a receber uma interrupção. A interrupção poderá ser habilitada ou desabilitada através de software por meio de instruções apropriadas e será desabilitada sempre que a unidade central de processamento aceitar uma interrupção ou receber um sinal de RESET.

INT (entrada)

Requisição de interrupção: um sinal no pino INT faz com que uma interrupção na unidade central de processamento ocorra, tão logo a instrução atual que está sendo processada termine. Caso a unidade central de processamento esteja em estado de HOLD ou INTE esteja em nível baixo a interrupção não será aceita.

RESET (entrada)

Reset: um sinal no pino RESET faz com que o conteúdo do contador de programa seja zerado fazendo com que o programa recomece a partir da posição de memória zero. Este sinal também faz com que a unidade central de processamento saia do estado de HOLD e desabilita a possibilidade de interrupção. As saídas INTE e HLDA irão para o nível baixo. Note que o sinal de reset não altera os registradores, o acumulador, os flags e o stack pointer.

Fizemos até aqui uma rápida análise de um por um dos pinos do 8080. Uma série de termos novos apareceram, mas todos e les serão melhor entendidos no decorrer de nosso estudo. Apenas tivemos uma visão do bus de endereço, bus de dados e das variáveis de controle do 8080. Estas variáveis de controle não serão manipuladas uma a uma pelo programador do microprocessador, e

sim serão utilizadas para controlar os outros circuitos que fazem parte do nosso sistema. Apenas os sinais de RESET e INT podem ser atuados pelo operador do sistema.

Vamos agora passar para uma análise em termos de blocos do microprocessador 8080.

VI.3 ARQUITETURA INTERNA

Podemos dividir o microprocessador em geral em três blocos básicos que são: seção de lógica e aritmética
rede de registradores
seção de controle

Estes três blocos formam o que é chamado de unidade central de processamento (figura II.4) que junto às memórias e os dispositivos de entrada e saída formam o sistema básico de computador (figura II.2).

VI.4 SEÇÃO DE LÓGICA E ARITMÉTICA

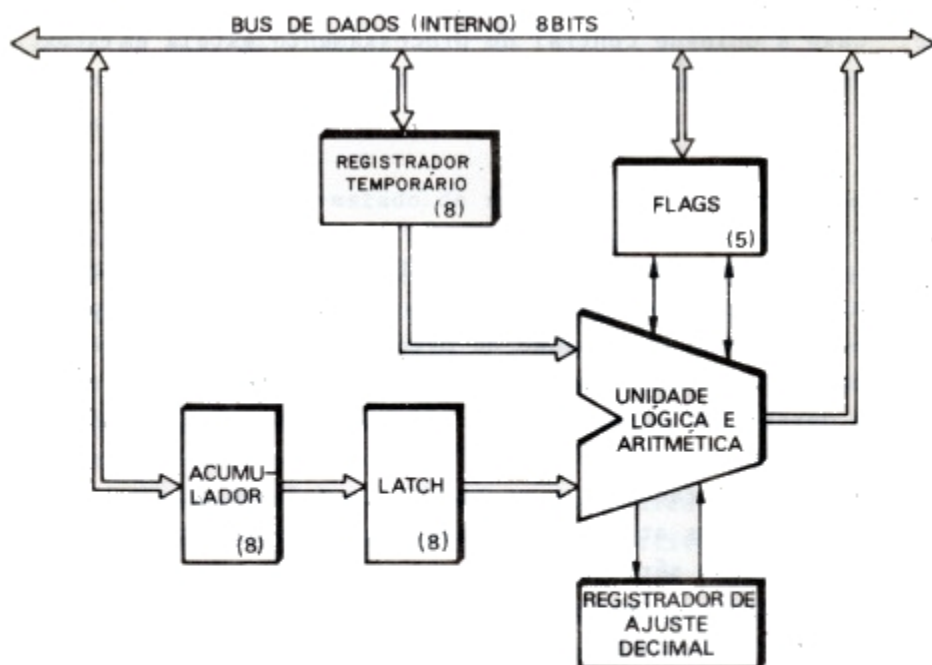


Fig. VI.2 Seção de lógica e aritmética do 8080.

A unidade central de processamento possui um bus de dados interno por onde passam os dados para serem usados pela seção lógica e aritmética, estes dados irão se alojar ou no registrador temporário ou no registrador acumulador.

Nestes registradores estão os dois operandos da operação lógica ou aritmética. Sempre o resultado da operação lógica ou aritmética irá ser armazenado no acumulador. Assim sendo, por exemplo, para somar dois números por uma instrução de soma, primeiramente deve ser armazenado um desses números no acumulador e o outro em um registrador da rede de registradores. O que a seção central de processamento fará quando receber a instrução de soma será transferir o número do acumulador para o latch do acumulador, buscar o outro número, que estará no registrador da rede de registradores, especificado na instrução de soma e transferi-lo para o registrador temporário, executar a operação de soma destes dois números na unidade lógica-aritmética e transferir o resultado para o acumulador.

Os flags são flip-flops que serão posicionados conforme o resultado da operação. Os flags do 8080 são os seguintes.

- Z: Zero - quando o resultado de uma operação for zero este flag será posicionado com o nível lógico "1", caso o resultado seja diferente de zero, será posicionado com "0".
- S: Sinal - quando o resultado de uma operação for negativa este flag será posicionado com o nível "1" e quando positivo, com o nível "0".
- P: Paridade - quando o resultado de uma operação apresentar um número par de bits com o nível lógico "1" este flag será posicionado com "1" e quando tivermos um número ímpar será posicionado com "0".
- C: Carry - quando o resultado de uma operação apresentar um "vai um" no bit mais significativo este flag será posicionado com o nível "1", caso

contrário será posicionado com "0".

AC: Auxiliar carry - quando o resultado de uma operação apresentar um "vai um" no seu terceiro bit este flag será posicionado com o nível "1" caso contrário será posicionado em "0".

Podemos através de instruções especiais em nosso programa testar estes flags para analisarmos o resultado da operação lógica ou aritmética, apenas o auxiliar carry (AC), não pode ser testado diretamente pelo nosso programa, pois ele será usado somente pelo registrador de ajuste decimal.

O registrador de ajuste decimal é um registrador para auxiliar a seção de lógica e aritmética na transformação de um byte hexadecimal em dois dígitos decimais.

VI.5 REDE DE REGISTRADORES

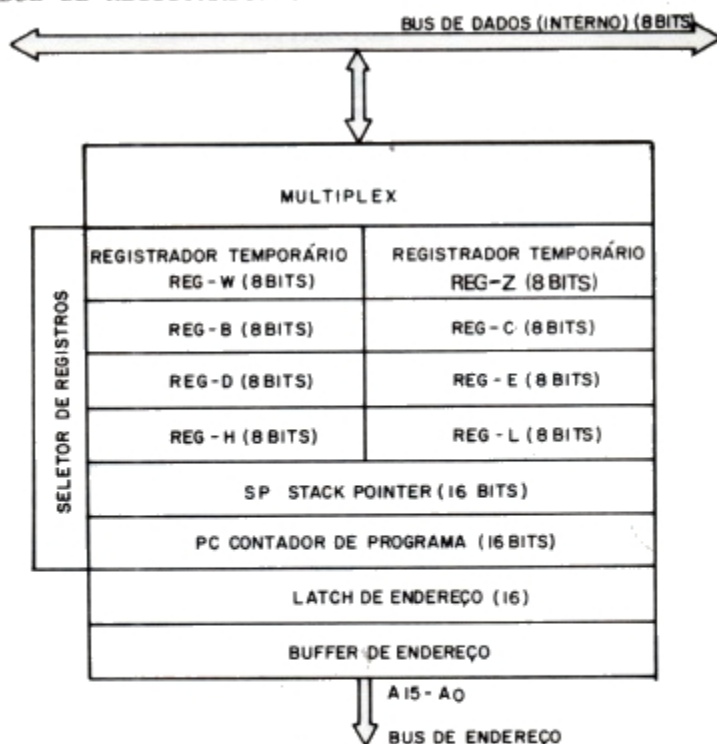


Fig. VI.3 Registradores do 8080.

Internamente ao microprocessador existe uma série de registradores que são utilizados para armazenar informações para posterior utilização. Estes registradores são os seguintes:

B, C, D, E, H, L são seis registradores de 8 bits para uso geral com acesso direto pelo programa. Estes seis registradores podem ser endereçados separadamente ou em pares formando desta maneira registradores de 16 bits.

Estes seis registradores possuem acesso direto à memória e também à seção de lógica e aritmética, permitindo assim a transferência direta de seus conteúdos para uma destas áreas ou vice-versa. Podemos considerar estes registradores como um espaço onde guardamos um dado, uma constante ou um endereço de uma posição de memória para utilização futura não perdendo desta maneira a informação. Estes registradores são utilizados para armazenar um dos operandos (o outro estará no acumulador) de uma operação lógica ou aritmética.

W,Z São dois registradores de 8 bits para uso geral, mas sem acesso pelo programa. Estes dois registradores são utilizados apenas pelo microprocessador para executar certas instruções, sendo assim é como se não existissem para o programa.

SP Stack pointer ("indicador de pilha" em português, mas esta tradução é pouco utilizada). Registrador de 16 bits. O microprocessador tem a possibilidade de utilizar uma parte de uma memória externa como uma pilha que opera na maneira de "último a entrar/primeiro a sair", onde podemos armazenar o conteúdo dos demais registradores (exceto W,Z). O registrador stack pointer possui o endereço da posição de memória onde está armazenado o último dado enviado para esta pilha.

PC Contador de programa, registrador de 16 bits onde está armazenado o endereço de memória que contém a próxima instrução a ser executada. Quando a unidade central de processamento for procurar a próxima instrução a ser executada, ela pega o seu endereço neste registrador.

VI.6 SEÇÃO DE CONTROLE

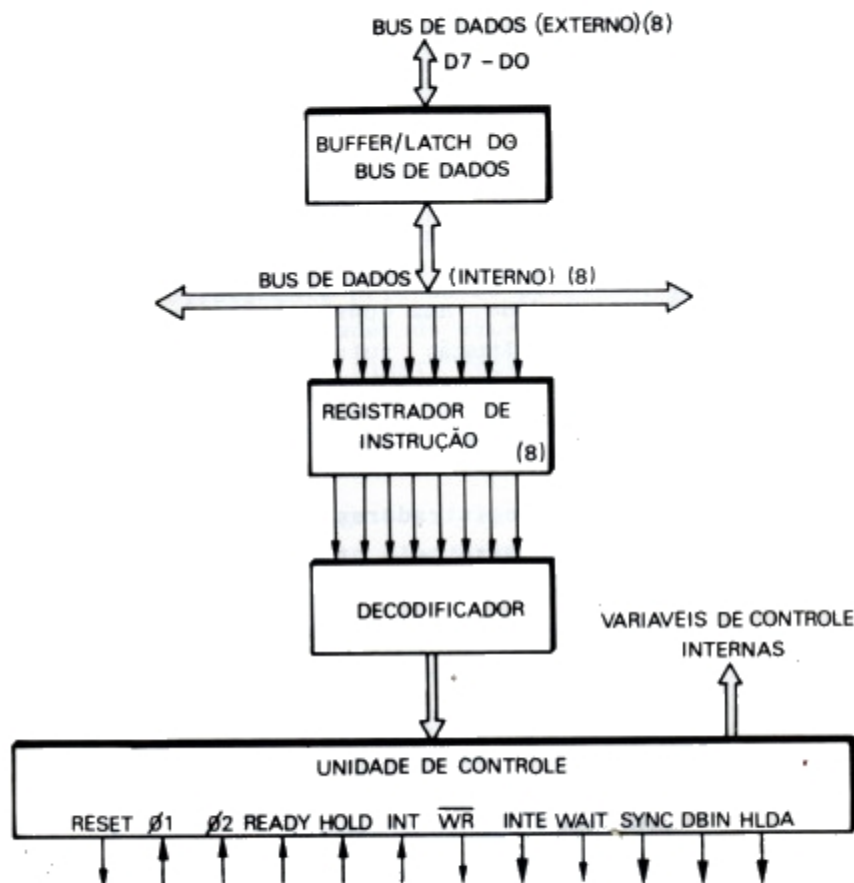


Fig. VI.4 Seção de controle do 8080.

A seção de controle fornece as variáveis de controle para o funcionamento do sistema. Existem dois grupos de variáveis de controle: as variáveis de controle internas que são os sinais que acionam os latches, buffers, registradores, multiplex e demais circuitos internos da unidade central de processamento e as variáveis de controle externas que são os sinais descritos

na seção VI.2 (WR, DBIN, INTE, INT, HLDA, HOLD, WAIT, READY, SYNC, ϕ_1 , e RESET).

Para a geração destas variáveis de controle temos, primeiramente, a entrada na unidade central de processamento, através do bus de dados externo, de uma instrução. Esta instrução é armazenada no registrador de instrução e por meio do decodificador é decodificada para a unidade de controle, a qual gerará as variáveis de controle. Assim sendo, cada instrução gera uma configuração diferente de variáveis de controle, variáveis estas que permitem a execução desta instrução específica.

VI.7 CONFIGURAÇÃO GERAL

Na figura VI.5 está representado o microprocessador 8080 onde, cortando as linhas pontilhadas estão representados cada um dos seus 40 pinos e, internamente às linhas pontilhadas, está representada por blocos, a arquitetura interna formada pela seção de lógica e aritmética, rede de registradores e a seção de controle.

Para a formação da unidade central de processamento com o 8080, precisamos de outros circuitos integrados que são utilizados para geração de clock (8224) e o controlador do sistema (8228) que serão estudados nos próximos capítulos.

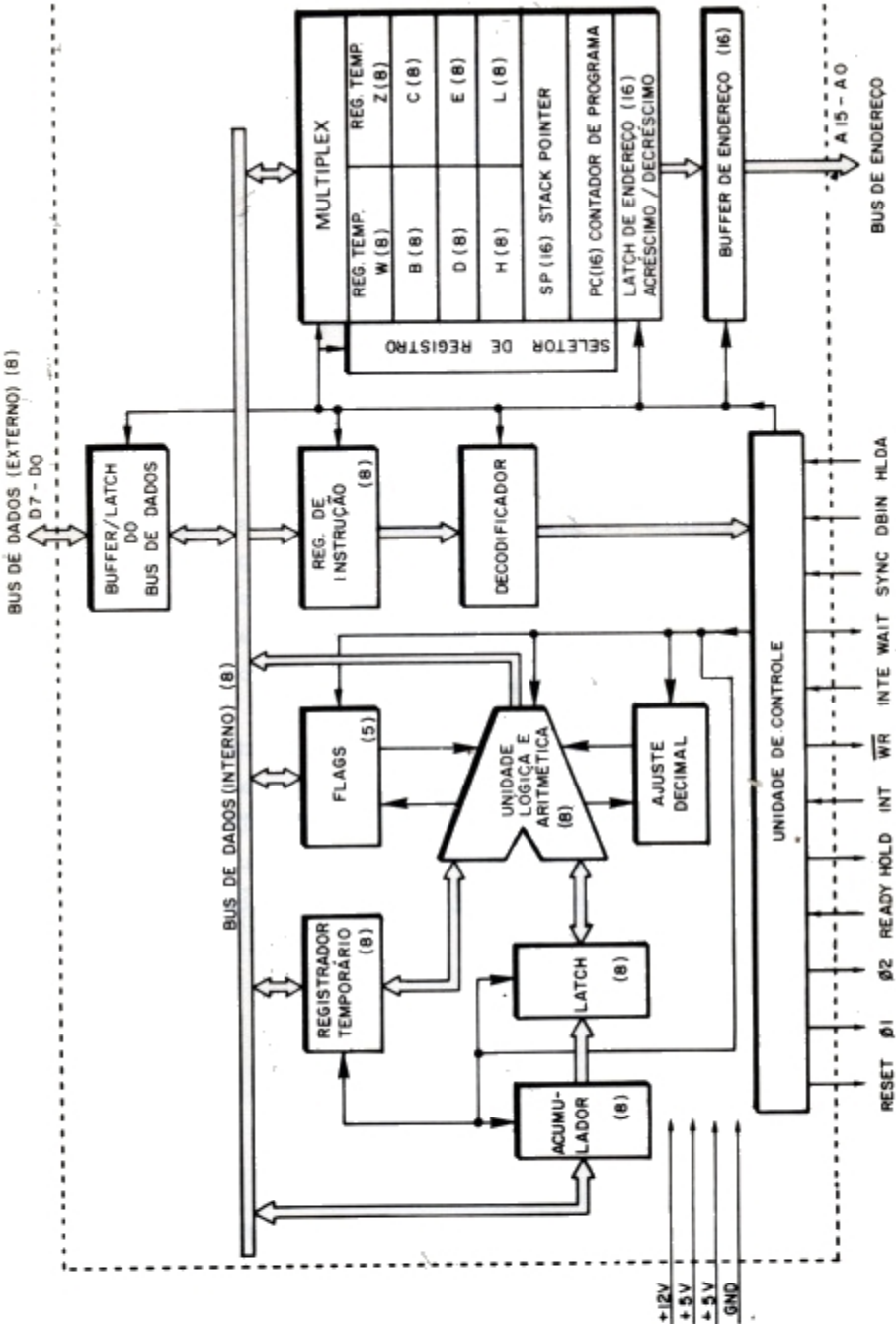


Fig. VI.5 Configuração Geral do 8080.

VII.1 GENERALIDADES

Todo microprocessador é um dispositivo dinâmico que necessita de um sinal de relógio de referência, sinal este que irá sincronizar seu funcionamento e que é chamado de sinal de clock. A cada variação deste sinal teremos as mudanças das variáveis de estado do microprocessador que irão causar a execução de suas funções.

Para a geração deste sinal de clock usa-se normalmente um cristal de quartz que, devidamente polarizado, oscila numa frequência estável e bem determinada.

Para a polarização do cristal e também sua adequação de sinal com o microprocessador 8080, foi desenvolvido um circuito integrado especial o 8224 que também é usado para algumas outras funções como veremos a seguir.

VII.2 DESCRIÇÃO DA PINAGEM

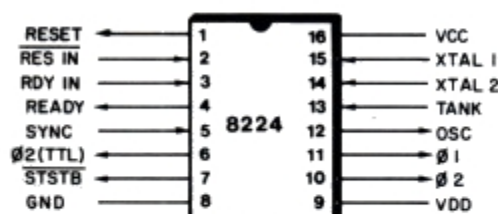


Fig. VII.1 Configuração dos pinos.

V_{CC}, V_{DD}, GND

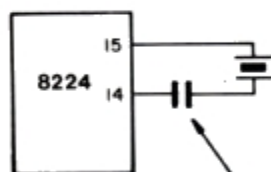
Alimentação:

$$V_{CC} = + 5 \pm 5\% \text{ Volts}$$

$V_{DD} = + 12 \pm 5\%$ Volts
 GND \rightarrow referência 0 Volts

XTAL₁, XTAL₂ (entrada)

A frequência de operação para o circuito oscilador é proveniente de um cristal externo que é conectado nestes dois pinos.



Necessário apenas para frequência acima de 10MHz.

Fig. VII.2

TANK (entrada)

No caso de utilizar-se um cristal que fornece uma frequência harmônica de oscilação, teremos um ganho bem menor do que de um cristal de frequência fundamental.

Neste caso, deve ser usado um oscilador externo formado por um circuito LC, sintonizado na frequência de cristal para reforçarmos o sinal e este circuito é conectado na entrada TANK.

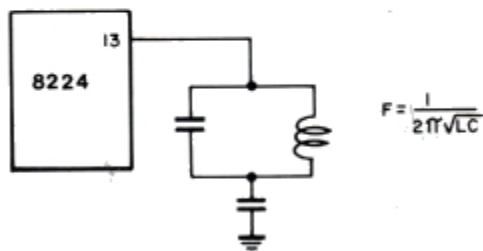


Fig. VII.3

OSC (saída)

Esta saída fornece um sinal na mesma frequência do cristal para uso geral.

ϕ_1, ϕ_2 (saída)

O sinal do cristal passa por um circuito divisor por nove interno ao 8224 que gera as duas formas de onda necessárias para o 8080, que são denominadas de ϕ_1 e ϕ_2 .

ϕ_2 (TTL) (saída)

Esta saída fornece o sinal ϕ_2 de nível compatível com TTL para uso geral.

SYNC (entrada)

Por onde é recebido um sinal gerado pelo 8080 para sincronização com o início de cada ciclo de máquina.

RES IN (entrada)

Entrada do sinal de reset: nesta entrada é conectado um circuito RC externo por onde é gerado o sinal de reset para o 8080.

Este sinal de reset é gerado toda vez que a fonte for ligada ou por uma chave externa.

Internamente ao 8224, temos um circuito Schmitt Trigger que transforma a transição lenta do sinal externo, num pulso de subida rápida que por sua vez é enviado a um flip-flop tipo D, sincronizado com o clock interno, que gera o sinal reset, com nível de saída correto.

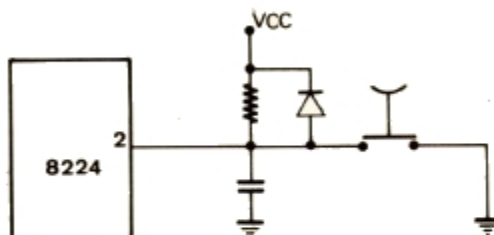


Fig. VII.4

RESET (saída)

Reset: por onde é enviado o sinal de reset para o 8080.

READY IN

Entrada do sinal de pronto: por esta entrada é recebido o sinal de READY de memória ou dispositivos de entrada e saída lentos que é internamente conectado a um flip-flop tipo D sincronizado com o clock interno e gera o sinal READY com nível de saída correto.

READY (saída)

Pronto: por onde é enviado o sinal de READY para o 8080.

STSTB (saída)

Indicação de estado: é originado pelo recebimento de um sinal de sincronização pelo 8080 (SYNC) ou por um sinal de RESET e é usado para diferenciar a informação contida no bus de dados.

Como já vimos na descrição do bus de dados do 8080 (capítulo VI), podemos ter neste bus: dados ou informações de controle. No início de um ciclo de máquina, as informações neste bus são de controle, neste instante o sinal de SYNC é gerado pelo 8080 e recebido pelo 8224, o qual gera um sinal de STSTB, permitindo ao controlador de sistema 8228, identificar que neste

instante as informações contidas no bus de dados são informações de controle.

VII.3 ARQUITETURA INTERNA

Internamente o 8224 é constituído de um circuito oscilador, um circuito divisor para gerar as frequências e de flip-flops tipo D, como pode ser visto na figura VII.5.

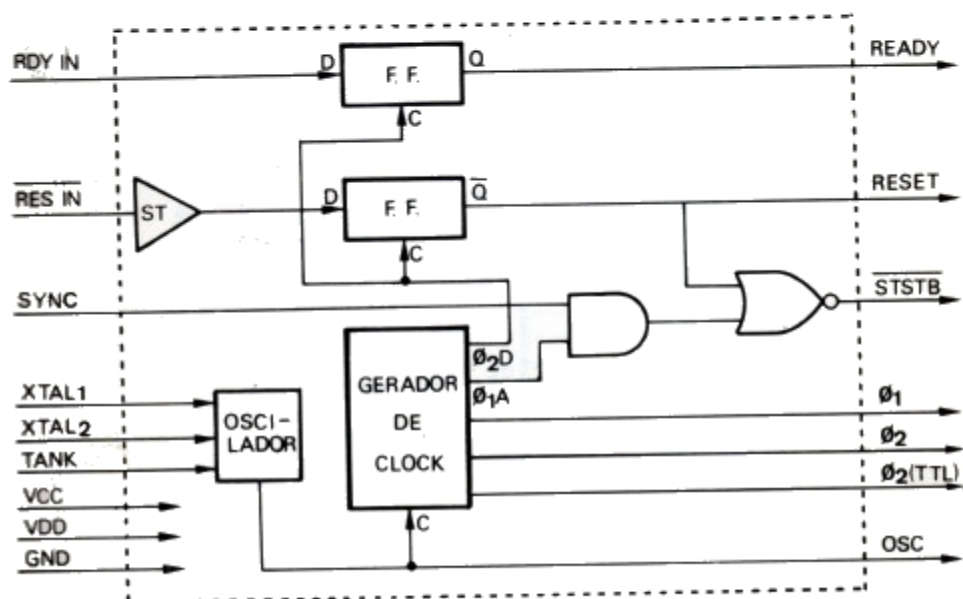


Fig. VII.5 Diagrama de blocos do 8224.

VII.4 ESQUEMA DE LIGAÇÃO

Na figura VII.6 temos o esquema de ligação do 8224 e dentro dos tracejados são indicados os circuitos opcionais.

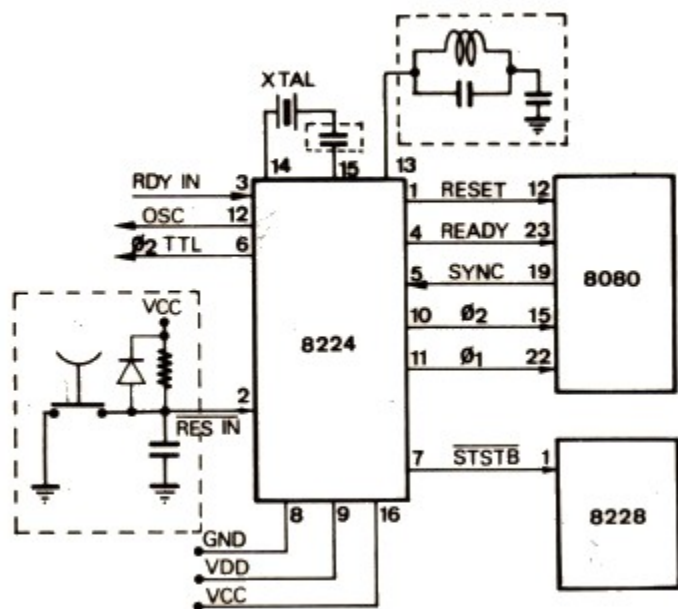


Fig. VII.6 Esquema de ligação de 8224.

VIII.1 GENERALIDADES

O 8228 é um circuito integrado usado para controlar o sistema de microprocessador demultiplexando o bus de dados do 8080, gerando o bus de dados para o sistema e o bus de controle. Este circuito faz a ligação da unidade central de processamento com as memórias e os dispositivos de entrada e saída.

VIII.2 DESCRIÇÃO DA PINAGEM



Fig. VIII.1 Configuração dos pinos.

V_{CC}, GND

Alimentação:

$$V_{CC} = + 5 \pm 5\% \text{ Volts}$$

GND → referência 0 Volts.

D₇ - D₀ (entrada/saída)

Bus de dados: são oito pinos de entrada e saída que for

mam o bus para a transferência de dados com a unidade central de processamento (ver capítulo VI).

DB₇ - DB₀ (entrada/saída)

Bus de dados do sistema: são oito pinos de entrada e saída que formam o bus para a transferência de dados com as memórias e o dispositivo de entrada e saída. Estes pinos podem, ainda, operar em three-state para possibilitar acesso direto à memória por um dispositivo de entrada e saída.

STSTE (entrada)

Indicação de estado: este sinal é proveniente do 8224 (ver capítulo VII) a cada início de um ciclo de máquina e serve para indicar ao 8228 que as informações contidas no bus de dados (D₇ - D₀) são informações de controle.

DBIN (entrada)

Bus de dados como entrada: este sinal é proveniente da unidade central de processamento e faz com que o controlador de sistema coloque o bus de dados no modo entrada.

WR (entrada)

Escrita: este sinal é proveniente da unidade central de processamento e é usado pelo controlador do sistema para gerar as variáveis de escrita ou leitura de memórias ou dispositivos de entrada e saída.

HLDA (entrada)

Indicação de estado de retenção: este sinal é proveniente da unidade central de processamento e é usado para indicar ao controlador de sistema que se encontra em estado de retenção.

BUSEN (entrada)

Habilitação dos bus: um sinal nesta entrada força os buffers de saída do bus de dados e os buffers do bus de controle em three-state. Se esta entrada estiver em nível "zero", temos operações normais nos bus de dados e de controle.

I/O W, I/O R, MEM W, MEM R, INTA (saída)

Estes sinais formam o bus de controle, os quatro primeiros servem para controle de leitura e escrita em memórias ou dispositivos de entrada e saída. INTA é um sinal de controle que serve para indicar para os outros dispositivos que uma interrupção foi aceita pela unidade central de processamento.

O 8080 tem a possibilidade de reconhecer até oito níveis diferentes de interrupção, (instrução RST0 à RST7) controlados por circuitos integrados especiais (8214 ou 8259). Caso desejemos apenas um nível de interrupção, basta ligarmos a saída INTA por um resistor de 1k Ω a fonte de + 12 Volts. Nesta condição, quando houver uma solicitação de interrupção, automaticamente o controlador de sistema colocará no bus de dados a instrução RST7.

VIII.3 ARQUITETURA INTERNA

Internamente o 8228 é constituído de um driver bidirecional para o bus de dados, um latch para as variáveis de dados, controlado pelo sinal STSTB e um decodificador para a geração das variáveis de controle, como pode ser visto na figura VIII.2.

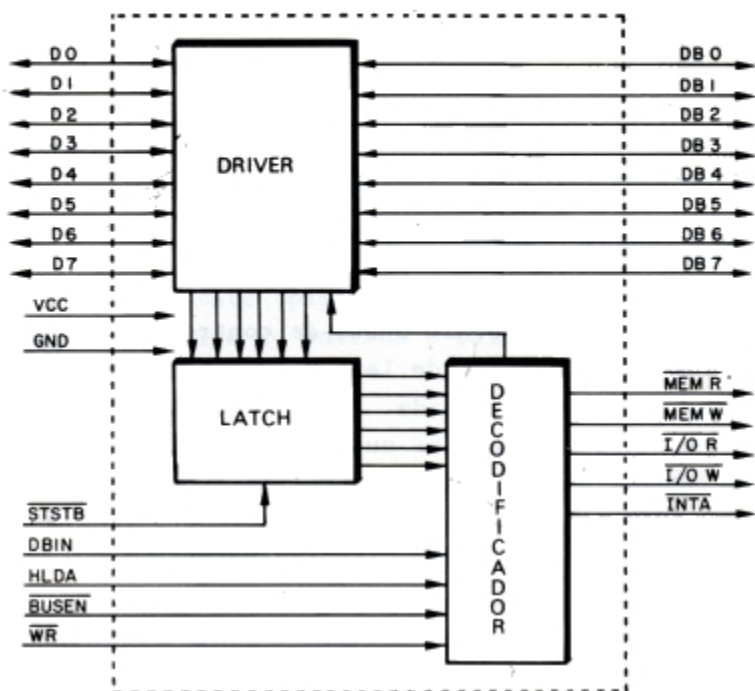


Fig. VIII.2 Diagrama de blocos do 8228.

VIII. 4 ESQUEMA DE LIGAÇÃO

Na fig. VIII.3, temos o esquema de ligações do 8228. Estes três circuitos juntos formam o mínimo indispensável para o funcionamento da unidade central de processamento com o micro processador 8080.

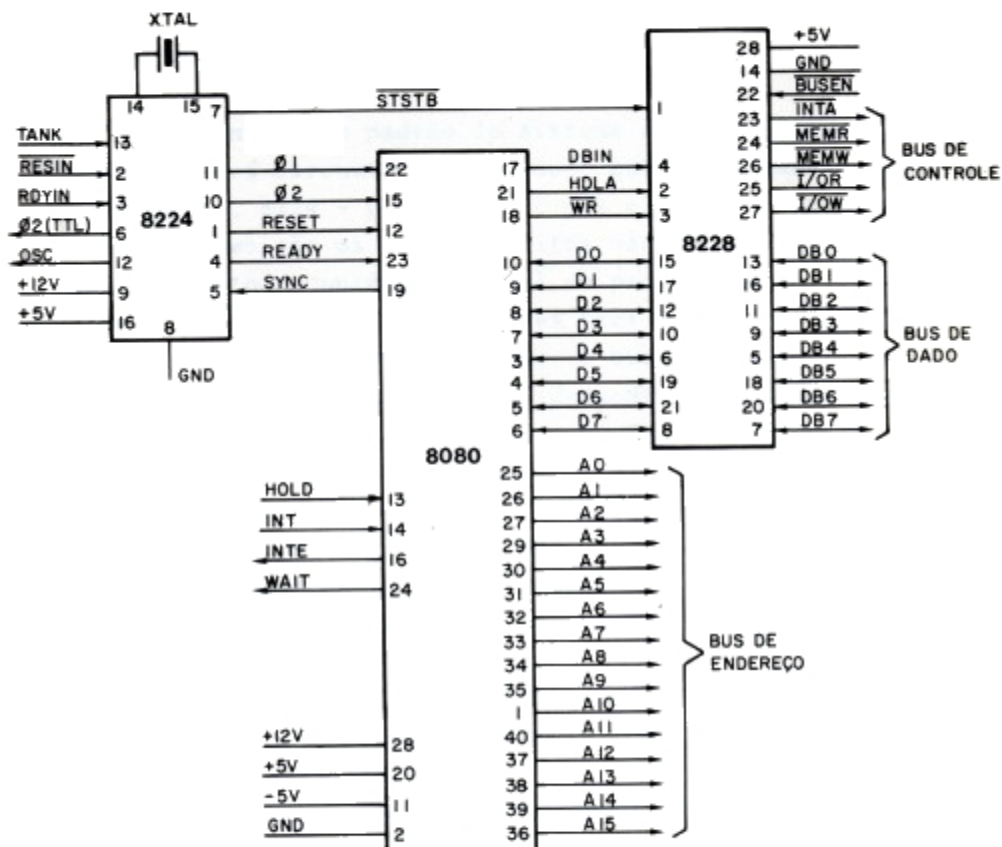


Fig. VIII.3 Interface padrão do sistema 8080.

IX.1 GENERALIDADES

A interface padrão do sistema de microprocessador 8080 é formada pelos circuitos integrados: 8080 - unidade central de processamento, 8224 - gerador de clock e 8228 - controlador do sistema. Estes são os três circuitos básicos necessários para o funcionamento da unidade central de processamento formado pelo 8080. Com a interligação destes três circuitos são formados o bus de endereço, o bus de dados, o bus de controle para os periféricos e as demais variáveis de controle; sinais estes necessários para o funcionamento de um sistema básico de microprocessador.

A partir destes três circuitos já estava disponível a geração de microprocessador de oito bits com capacidade suficiente para utilização geral. O próximo passo seria conseguir agrupar estes três circuitos em apenas um e também conseguir algumas melhorias adicionais.

A concorrência entre fabricantes, gerada pelo interesse neste campo de circuitos integrados, apresentou rapidamente soluções melhores.

Para estudo de um destes novos microprocessadores, escolhemos o 8085 que, por ter sido desenvolvido pelo mesmo fabricante e utilizar basicamente a mesma nomenclatura e arquitetura do 8080, facilita o estudo.

O 8085 é um microprocessador de 8 bits que possui o seu conjunto de instruções completamente compatível com o 8080, isto é, um programa, processado por um sistema constituído pelo 8080 pode ser processado por um sistema com o 8085. O 8085 é mais vantajoso porque dispensa o uso de gerador de clock externo (8224) e de controlador do sistema (8228) pois, estes já estão agregados ao mesmo. Uma das dificuldades encontradas foi tentar obter todos os sinais necessários com um circuito integrado de 40 pinos. A maneira encontrada para contornar esta dificuldade foi utilizar os pinos do bus de dados também como par

te do bus de endereço. Assim, em um determinado instante, nestes pinos, são fornecidas informações de endereço que podem ser agrupadas em um latch e, em outro instante, são fornecidas informações de dados.

Além desta maior integração e conjunto de instruções compatível com o 8080, existem outras melhorias, como fonte de alimentação simples (+ 5V e GND), maior velocidade e mais níveis de interrupção.

IX.2 DESCRIÇÃO DA PINAGEM

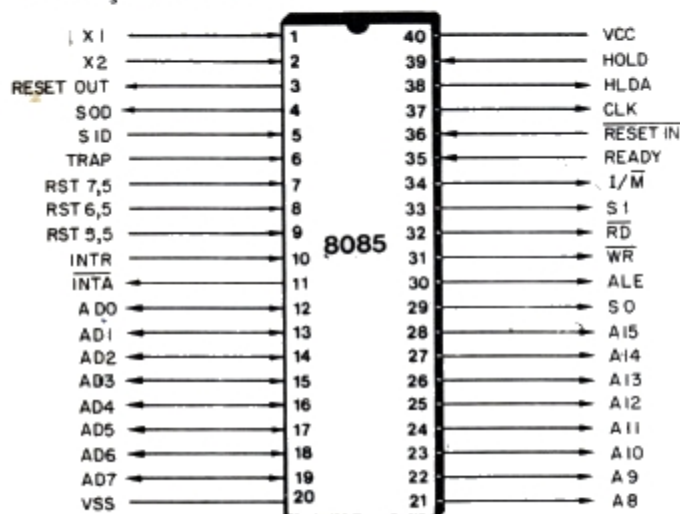


Fig. IX.1 Configuração dos pinos.

V_{CC} , V_{SS}

Alimentação: o 8085 requer uma única fonte de alimentação.

V_{SS} + referência 0 Volts

V_{CC} = + 5 ± 5% Volts

X_1 , X_2

Clock: como o 8080, o microprocessador 8085 é um dispositivo dinâmico que necessita de um sinal de clock, apresenta, porém, a vantagem de não necessitar de um circuito controlador

externo, podendo ser seu sinal gerado diretamente entre estes dois pinos e de uma das seguintes maneiras:

- por um cristal de quartz,
- por uma rede RC,
- por um circuito ressonante LC e
- por um sinal externo proveniente de um outro circuito oscilador.

CLK (saída)

Saída de clock: sinal de clock que é a metade da frequência em X_1 e X_2 para uso geral.

A₁₅ - A₈ (saída)

Bus de endereço: são os 8 bits mais significativos (dos 16), que formam as saídas three-state do bus de endereço.

AD₇ - AD₀ (entrada/saída)

Bus de dados/bus de endereço: são os 8 bits menos significativos de bus de endereço durante o primeiro ciclo das instruções e, nos demais ciclos, constitui o bus de dados.

ALE (saída)

Habilitação do latch de endereço: sinal de sincronização, gerado no primeiro ciclo da instrução simultaneamente com as informações de endereçamento, é utilizado para habilitar os latches de endereço a fim de reter a atual informação dos pinos AD_7 - AD_0 .

SID (entrada)

Entrada de dados serial: utilizado para entrada de dados de forma serial. A utilização deste pino está relacionada com uma instrução especial RIM.

SOD (saída)

Saída de dados serial: utilizado para a saída de da dos de forma serial. A utilização deste pino está relacionada com uma instrução especial SIM.

INTR (entrada)

Requisição de interrupção: possui a mesma função que o pino INT do 8080.

INTA (saída)

Indicação de interrupção: possui a mesma função que o pino INTA do 8228.

RST 5,5; RST 6,5; RST 7,5 (entradas)

Requisição de interrupção: possuem a mesma característica do pino INTR, exceto que, quando alguma destas interrupções ocorrem, o contador de programa será carregado com um endereço fixo que depende apenas de quais destes pinos receberam o sinal de interrupção. Estas interrupções podem ser habilitadas ou desabilitadas por programa.

TRAP (entrada)

Requisição de interrupção: reconhece a interrupção da mesma maneira que INTR, RST 5,5, RST 6,5, RST 7,5, porém não po de ser desabilitado por programa. Todas estas interrupções pos suem um nível de prioridade, sendo que o de maior prioridade é o TRAP seguido por RST 7,5, RST 6,5, RST 5,5 e finalmente INTR que possui a menor prioridade.

HOLD (saída)

HLDA (entrada)

READY (entrada)

Possuem as mesmas funções dos pinos de mesma nomenclatura

tura do 8080.

RESET IN (entrada)

Entrada de reset: possui a mesma função do RESET do 8080.

RESET OUT (saída)

Saída de reset: indica que a 8085 recebeu um sinal de reset e pode ser usado como sinal de reset dos outros circuitos integrados do sistema.

S₁ , S₀ (saída)

Status: indicam o estado atual em que se encontra o microprocessador.

| S ₁ | S ₀ | estado |
|----------------|----------------|--------------|
| 0 | 0 | retenção |
| 0 | 1 | escrita |
| 1 | 0 | leitura |
| 1 | 1 | interrompido |

I/M (saída)

Dispositivo de entrada e saída ou memória: o nível lógico "1" neste pino indica que a comunicação de dados do microprocessador será feita com um dispositivo de entrada e saída. O nível lógico "0" indica que a comunicação de dados será feita com as memórias.

RD (saída)

Controle de leitura: indica que o dispositivo de entrada e saída ou a memória selecionada será lida e que o bus de dados está posicionado como entrada.

WR (saída)

Controle de escrita: indica que a informação do bus de dados é para ser escrita no dispositivo de entrada e saída ou memória selecionada.

IX.3 ARQUITETURA INTERNA

A arquitetura interna do microprocessador 8085 (ver figura IX.2) é basicamente a mesma do 8080, constituída pela seção de lógica e aritmética, rede de registradores e seção de controle.

A seção lógica e aritmética é formada pelo registrador acumulador, um registrador para armazenamento temporário de operandos e a unidade que processará as operações lógicas e aritméticas dos dados contidos no acumulador e no registrador temporário. Além disso, possui o registrador para ajuste decimal e o registrador com 5 flags que são os mesmos flags do microprocessador 8080; Z: zero, S: sinal, P: paridade, C: carry, AC: auxiliar carry.

A rede de registradores também é idêntica a do microprocessador 8080, formada por dois registradores temporários de 8 bits (W, Z) sem acesso por programa, seis registradores de 8 bits para uso geral (B, C, D, E, H, L) que podem ser utilizados por programa aos pares formando três registradores de 16 bits para uso geral (BC, DE, HL), um registrador de 16 bits para indicação da pilha (stack pointer) e outro registrador de 16 bits para armazenamento dos endereços das instruções do programa (contador de programa). As informações contidas nestes registradores podem ser utilizadas como endereço e, neste caso, os 8 bits mais significativos de endereço são, diretamente enviados ao bus de endereço externo do microprocessador e, os 8 bits menos significativos são enviados ao bus de dados interno para serem enviados ao bus de dados ou endereço externo. As informações destes registradores também podem ser dados que, por intermédio do bus de dados interno, podem ser enviadas ao bus de dados ou ao endereço externo ou ao acumulador ou, ainda, ao registrador temporário.

Finalmente, completando a arquitetura do microprocessador 8085, temos a seção de controle formada pelo registrador de instruções que recebe as instruções, provenientes da memória de programa, pelo bus de dados ou endereço externo e envia estes 8 bits que podem ser uma instrução completa ou parte de uma instrução, ao decodificador que irá acionar na unidade de controle as variáveis de controle externas que são usadas para controle dos outros circuitos integrados do sistema e, também, as variáveis de controle internas para o controle interno de cada bloco do microprocessador. Completando a seção de controle, existem ainda outros dois controladores, um para as interrupções que decodifica os sinais externos, examinando as prioridades e alterando o conteúdo do registrador contador de programa, conforme a interrupção recebida, e o outro controlador, destinado à transmissão ou recepção serial de dados.

Por esta sua configuração, este microprocessador tem capacidade de formar a unidade central de processamento de um sistema de microcomputador, podendo ser diretamente conectado às memórias e interface para dispositivos de entrada e saída.

Alguns dos circuitos integrados especiais para formação das interfaces serão analisados nos próximos capítulos.

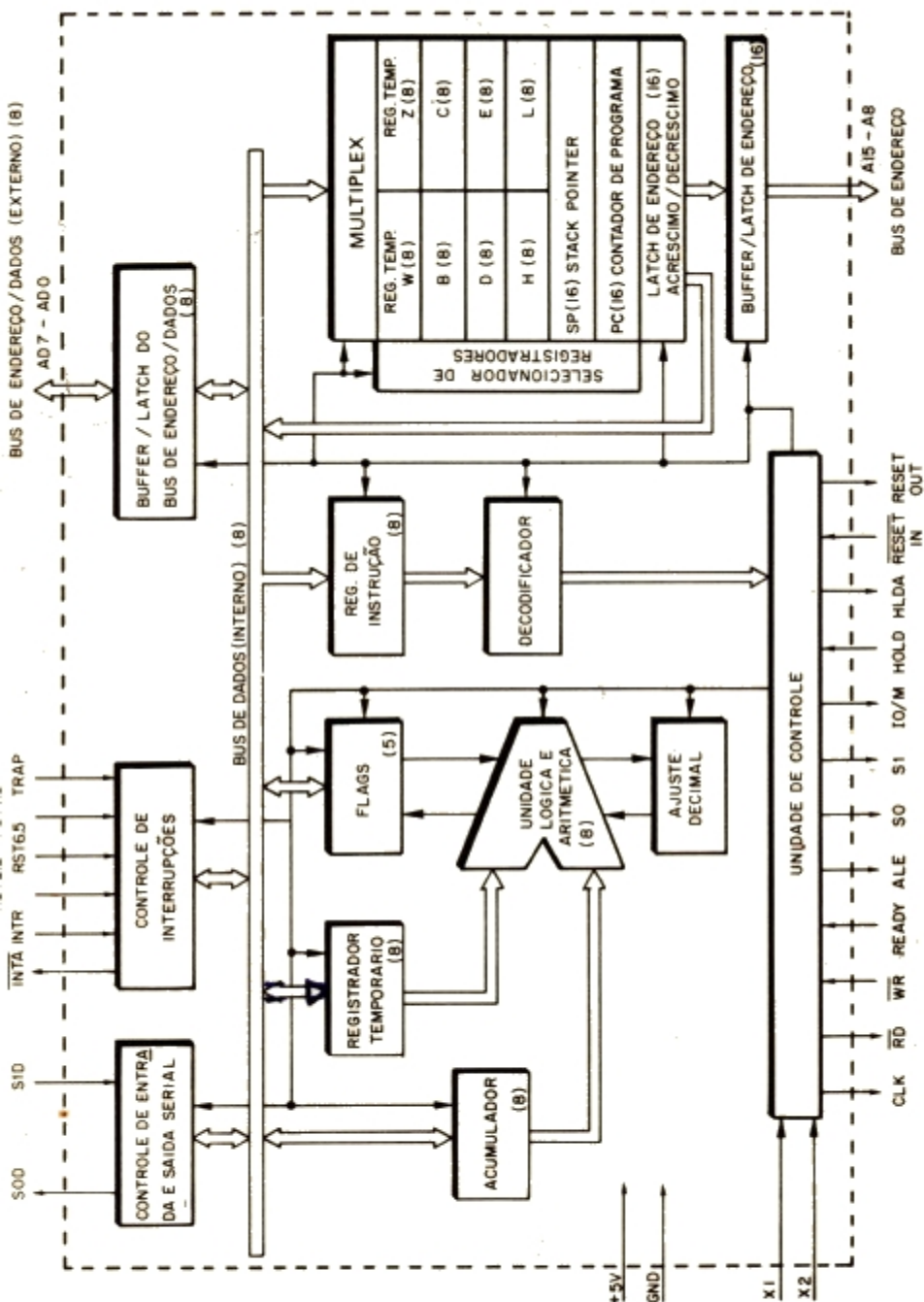


fig. IX.2 Configuração geral do 8085.

X.1 GENERALIDADES

Para a comunicação de dados entre a unidade central de processamento e os dispositivos de entrada e saída, os dados devem passar através de circuitos especiais que são as interfaces de entrada e saída. Nestes circuitos, um dado proveniente de um dispositivo de entrada e saída é armazenado para posterior leitura pela unidade central de processamento ou vice-versa. Certos dispositivos ou certas aplicações necessitam de interfaces de comunicação paralela, isto é, os dados de comunicação são formados por diversos bits e são transferidos todos os bits simultaneamente. Entre as interfaces paralelas, vamos analisar mais detalhadamente a 8255 que é uma interface de propósito geral, programável, projetada para trabalhar com o 8080 ou 8085 podendo, entretanto, ser utilizada em sistemas com outros microprocessadores como por exemplo o Z-80.

X.2 DESCRIÇÃO DA PINAGEM



Fig. X.1 Configuração dos pinos.

V_{CC}, GND

Alimentação:

V_{CC} = + 5 Volts ± 5%

GND = referência 0 Volts

D₇ - D₀ (entrada/saída)

Bus de dados bidirecional: estes pinos são conectados diretamente ao bus de dados do microprocessador, bus este utilizado para transferência de dados entre o microprocessador e a interface.

PA₇ - PA₀ (entrada/saída) : Porta A

PB₇ - PB₀ (entrada/saída) : Porta B

PC₇ - PC₀ (entrada/saída) : Porta C

Cada uma destas portas forma um grupo de 8 bits para transferência de dados entre os dispositivos de entrada e saída, e a interface. Cada um destes 24 bits pode ser, individualmente, monitorado pela unidade central de processamento e o sentido de entrada ou saída é selecionado por programa, como pode ser visto na descrição de modo de operação (X.4).

A₁ - A₀ (entrada)

Bus de endereço: conectados diretamente ao bus de endereço do microprocessador e são utilizados para selecionar uma das três portas ou o registrador da palavra de controle da 8255

RESET (entrada)

RESET: o mesmo sinal de reset recebido pelo microprocessador e demais circuitos do sistema é enviado a 8255 para sincronização geral do sistema. Um sinal de reset na 8255 limpa

todos seus registradores internos, incluindo a palavra de controle, e coloca as três portas no modo entrada.

CS (entrada)

Seleção do circuito: habilita a comunicação entre a interface e o microprocessador.

RD (entrada)

Leitura: habilita a interface a enviar dados ao microprocessador através do bus de dados.

WR (entrada)

Escrita: habilita a interface a receber dados ou palavras de controle do microprocessador através do bus de dados.

X.3 ARQUITETURA INTERNA

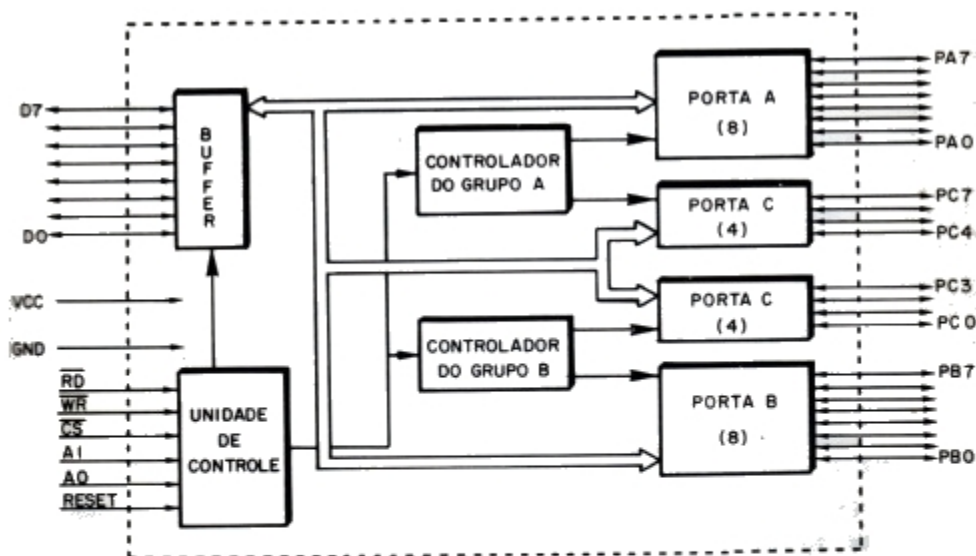


Fig. X.2 Arquitetura interna da 8255.

As três portas de comunicação da interface paralela 8255 são divididas em dois grupos: o grupo A, formado pela porta A e os quatro bits mais significativos da porta C ($PC_7 - PC_4$), e o grupo B, formado pela porta B e os quatro bits menos significativos da porta C ($PC_3 - PC_0$). Cada controlador destes grupos recebe os comandos da unidade de controle e os envia para as portas associadas a ele.

A unidade de controle interna decodifica as variáveis de controle do sistema e gera os comandos para o buffer do bus de dados e para os controladores de grupo. A programação é feita por meio da palavra de controle fornecida pelo sistema.

Os dados enviados pela unidade central de processamento serão recebidos pela interface paralela através do bus de dados e enviados aos dispositivos de entrada/saída por uma das portas conforme selecionada pela unidade de controle ou, de maneira inversa, os dados recebidos através da porta selecionada serão enviados à unidade central de processamento pelo bus de dados, fazendo, assim, esta interface a interligação da unidade central de processamento com os dispositivos de entrada/saída.

X.4 MODO DE OPERAÇÃO

A interface de comunicação paralela 8255 é um dispositivo programável pelo microprocessador e pode operar em três modos distintos:

Modo 0 - Neste modo, suas portas estarão programadas para serem basicamente entradas ou saídas de dados.

Modo 1 - Neste modo, apenas as portas A e B serão entradas ou saídas de dados, porém elas são sincronizadas por meio de sinais de controle especiais ligados a porta C.

Modo 2 - Neste modo, a porta B não é usada, a porta constituirá um bus de dados bidirecional de entrada e saída e os bits da porta C são usados para sinais de controle de maneira semelhante ao modo 1.

A interface paralela 8255 pode operar em um destes modos conforme sua programação, ou, ainda, ela pode trabalhar co

combinação destes modos, isto é, uma porta pode trabalhar em um modo e outra porta em outro modo, simultaneamente.

Iremos aqui examinar mais detalhadamente o modo de operação que julgamos ser o mais geral e também o mais utilizado, pois os outros modos, ou as combinações de modos de operação são utilizadas apenas para aplicações especiais como interface com terminais de vídeo, flop-disc, impressores, teclados ou outros dispositivos especiais, que não trazem dificuldades para a utilização da interface, porém requerem um estudo detalhado dos sinais de sincronização destes dispositivos de entrada ou saída.

Para trabalhar com a 8255, a unidade central de processamento necessita apenas de duas instruções: uma para enviar um dado através do bus de dados para a interface e outra para receber um dado. Porém, este dado pode ser enviado para o registrador da palavra de controle; neste caso, o dado enviado será uma palavra para a programação da interface, ou, então, o dado pode ser enviado ou recebido para qualquer uma das três portas e, neste caso, o dado será enviado ou proveniente diretamente de um dispositivo externo.

Vamos analisar como a unidade central de processamento seleciona uma das três portas ou o registrador da palavra de controle. Isso é feito por meio dos dois bits de endereço que são conectados da unidade central de processamento à interface. Os registradores, selecionados conforme as combinações dos bits de endereço, podem ser vistos na tabela da figura X.3.

| A ₁ | A ₀ | registrador |
|----------------|----------------|-------------|
| 0 | 0 | Porta A |
| 0 | 1 | Porta B |
| 1 | 0 | Porta C |
| 1 | 1 | controle |

Fig. X.3 Endereçamento da 8255.

As variáveis de controle \overline{RD} e \overline{WR} definirão se a operação é de leitura ou escrita, respectivamente, na interface.

Assim sendo a comunicação da unidade central de proces

samento com a interface é feita por meio das duas instruções de enviar ou receber dados da interface, sendo que as próprias instruções irão acionar adequadamente as variáveis de leitura ou escrita; estas instruções deverão conter o endereço, conforme o registrador desejado.

A programação da 8255 é feita através de uma palavra de controle que é enviada da unidade central de processamento para o registrador da palavra de controle. Esta palavra de controle é formada por oito bits ($D_7 - D_0$) e a definição de cada bit está mostrado na tabela da figura X.4.

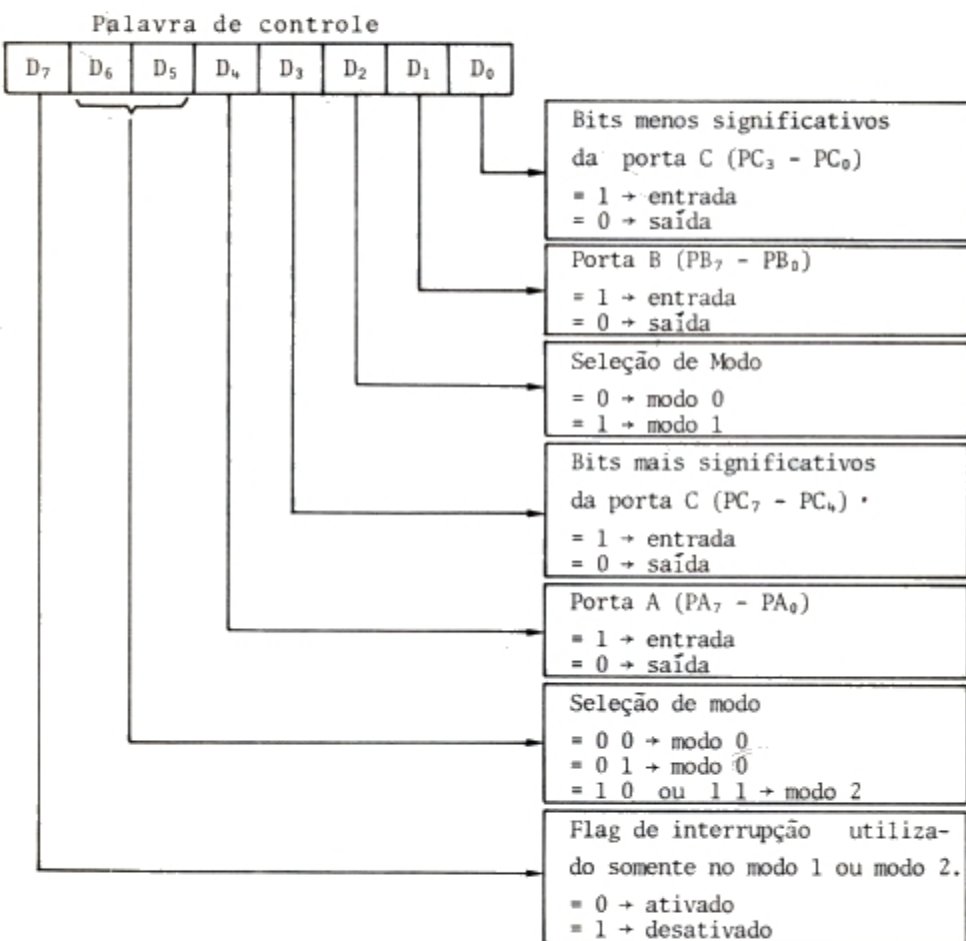
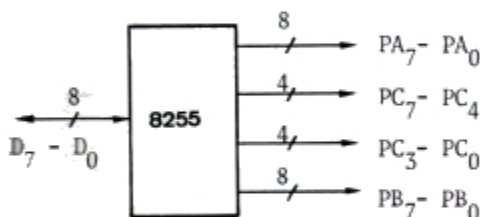


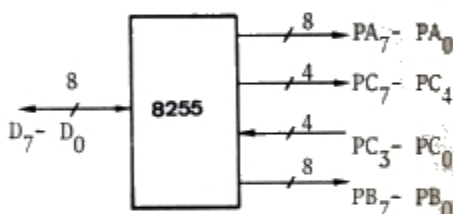
Fig. X.4 Definição da palavra de controle.

Conforme as variações possíveis da palavra de controle le podemos ter as configurações mostradas na figura X.5.

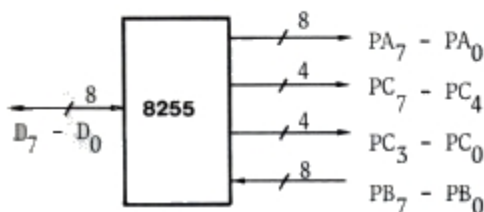
palavra de controle = 80



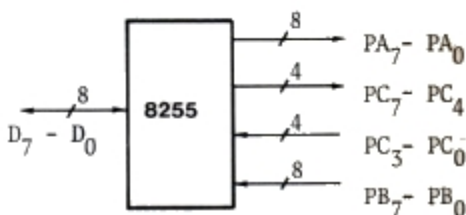
palavra de controle = 81



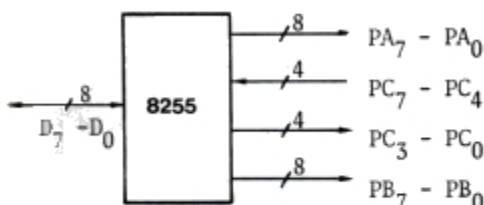
palavra de controle = 82



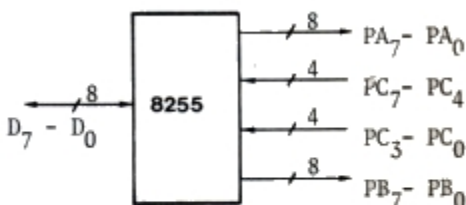
palavra de controle = 83



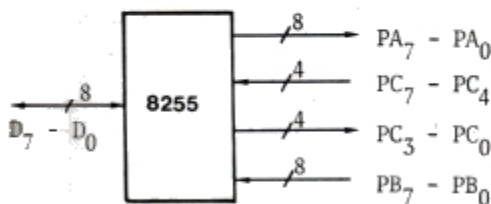
palavra de controle = 88



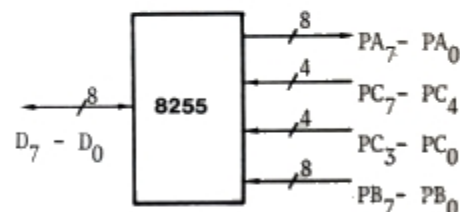
palavra de controle = 89



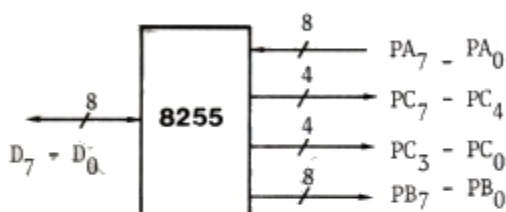
palavra de controle = 8A



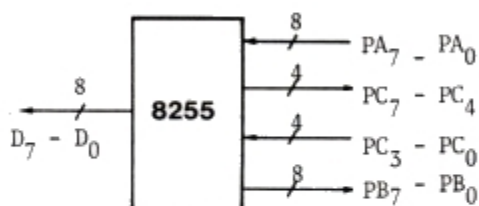
palavra de controle = 8B



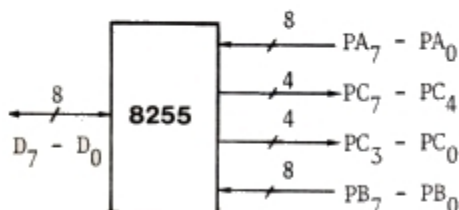
palavra de controle = 90



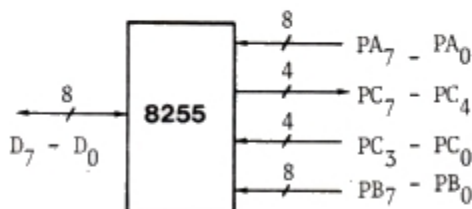
palavra de controle = 91



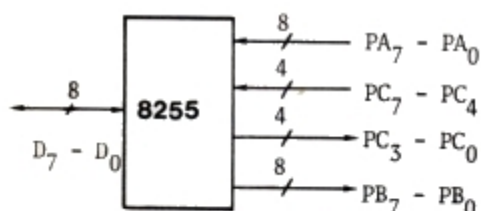
palavra de controle = 92



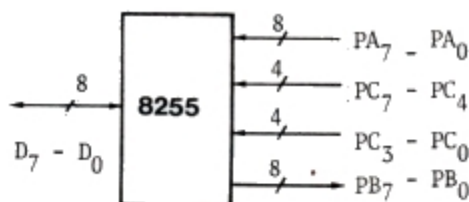
palavra de controle = 93



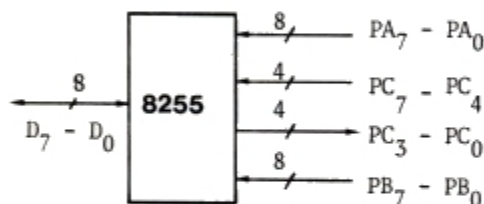
palavra de controle = 98



palavra de controle = 99



palavra de controle = 9A



palavra de controle = 9B

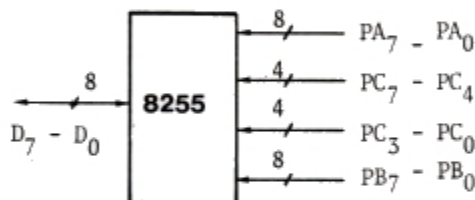


Fig. X.5 Combinações possíveis da 8255.

Vejamos um exemplo de aplicação da interface de comu

nicação paralela: a interface devem estar acopladas quatro chaves (que normalmente têm nível "0" e, quando acionadas, nível "1") e quatro leds (que podem ser acionadas diretamente pelas portas da 8255).

Quando uma chave é acionada pelo operador, uma tarefa diferente será executada pelo microprocessador e, após a execução de uma tarefa, o led correspondente a esta será ligado.

Entre os 24 pinos diferentes que podemos usar da 8255 vamos, por exemplo, ligar os leds nos quatro bits mais significativos da porta A e, as chaves, nos quatro bits menos significativos da porta C. Neste caso, a porta A deverá estar programada como saída de dados e, os quatro bits menos significativos da porta C, como entrada de dados. Para isso a palavra de controle de programação da 8255 poderá ser por exemplo 81.

Desta maneira a configuração do nosso sistema poderá ser como o mostrado na figura X.6.

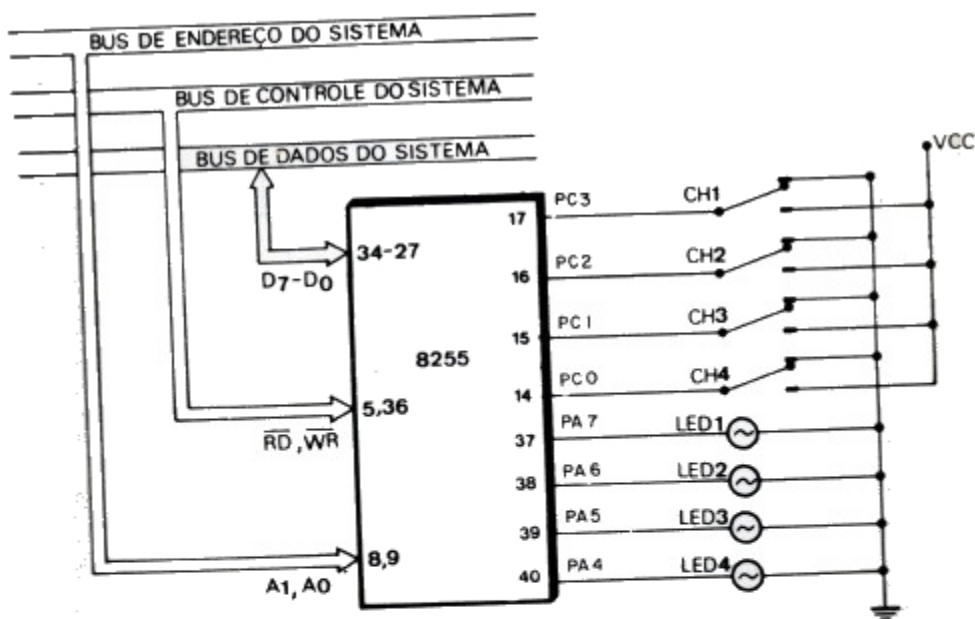


Fig. X.6 Exemplo de aplicação da 8255.

XI.1 GENERALIDADES

Certos periféricos utilizados como dispositivos de entrada/saída de dados de sistemas de microprocessadores possuem transmissão e recepção de dados serial, isto é, os bits de um dado são transmitidos um após o outro por uma única linha de transmissão e cada bit é transmitido durante um intervalo de tempo básico pré-fixado. Para melhor entendermos como este processo funciona, vamos analisar um exemplo da transmissão de um dado de 4 bits do sistema de microprocessador para um periférico:

Primeiramente, todos os bits do dado a ser transmitido são transferidos pelo bus de dados para o registrador interno da interface por uma operação de escrita (figura XI.1).

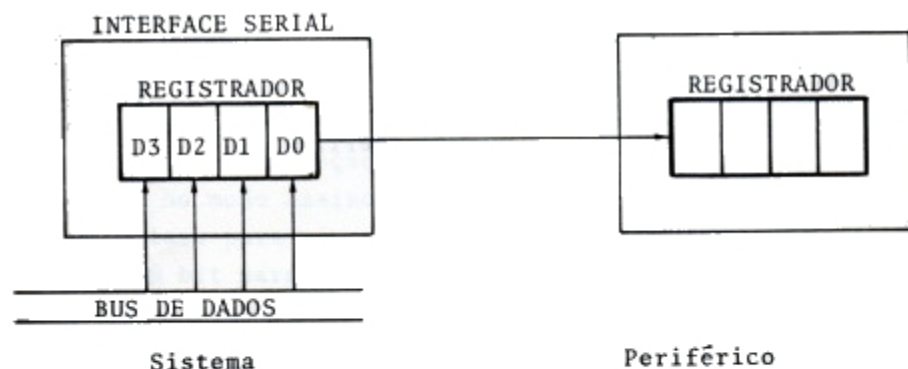
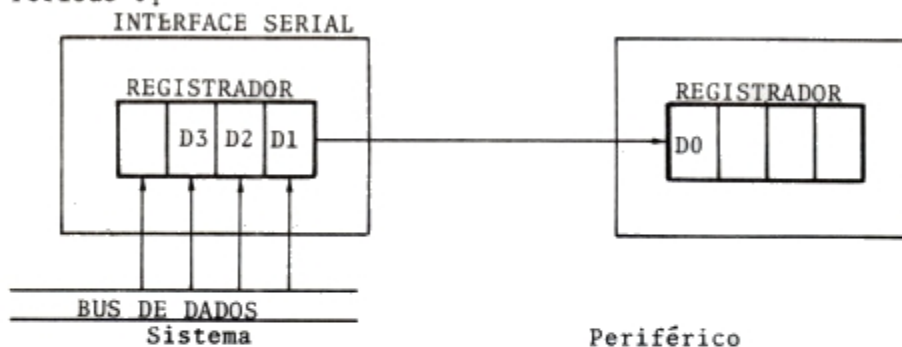


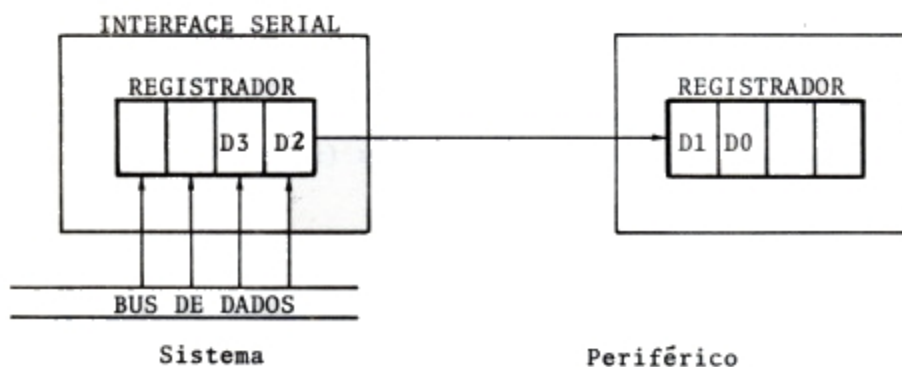
Fig. XI.1

Após o dado ter sido carregado no registrador da interface serial, ele é transmitido bit por bit para o periférico a cada intervalo de tempo (figura XI.2).

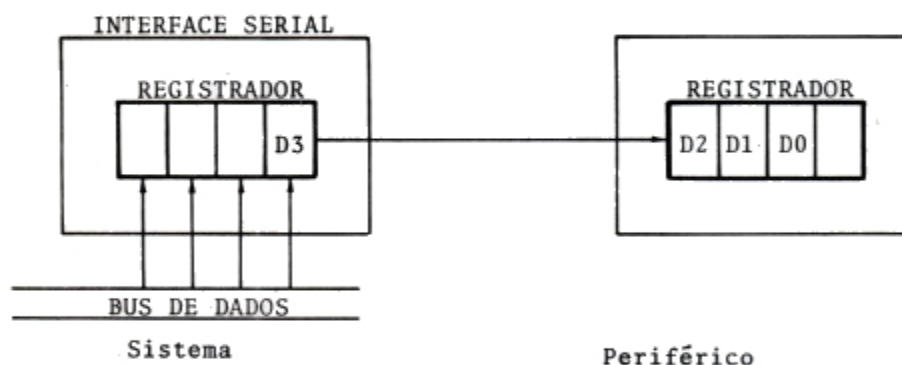
Período T_1



Período T_2



Período T_3



Período T_4

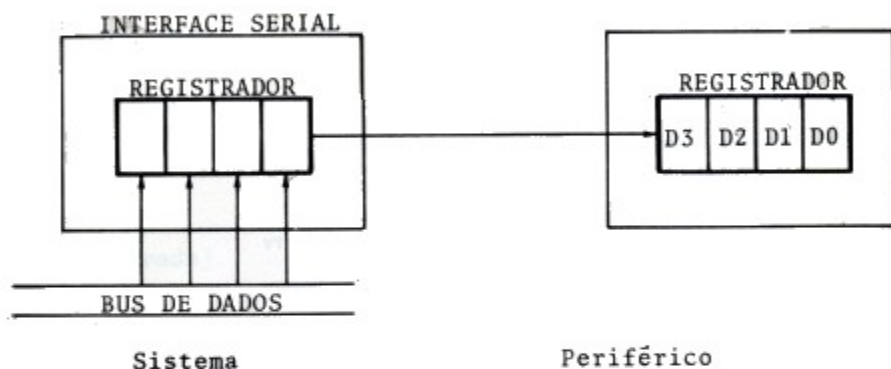


Fig. XI.2

Desta maneira, após os quatro intervalos de tempo o dado está disponível no periférico para ser processado.

Para a transmissão/recepção de dados seriais são possíveis dois modos diferentes: modo síncrono ou assíncrono.

No modo síncrono temos a transmissão de caracteres especiais para sincronização do emissor com o receptor de dados e, após são enviadas os dados um em seguida ao outro sem sinais especiais para separação de cada caracter.

No modo assíncrono não existe a transmissão de caracteres especiais para sincronização, porém, em cada caracter é adicionado um bit para indicação de início do mesmo (start bit), um ou dois bits para indicação de fim (stop bit) e um bit para verificação da transmissão (parit bit).

XI.2 DESCRIÇÃO DA PINAGEM

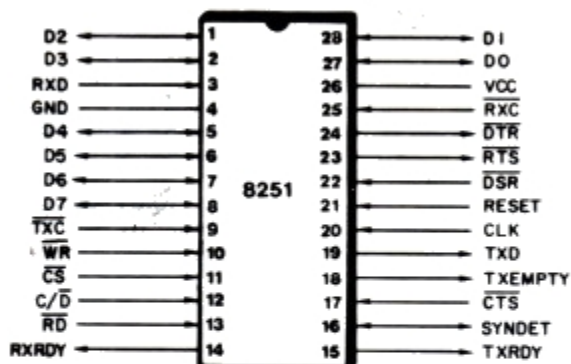


Fig. XI.3 Configuração dos pinos.

V_{CC}, GND

Alimentação

V_{CC} = + 5 Volts \pm 5%

GND = referência 0 Volts

D₇ - D₀ (entrada/saída)

Bus de dados bidirecional: estes pinos são conectados diretamente ao bus de dados do microprocessador, bus este utilizado para transferência de dados entre o microprocessador e a interface.

CS (entrada)

Seleção de circuito: habilita a comunicação entre a interface e o microprocessador.

RESET (entrada)

Reset: o mesmo sinal de reset recebido pelo microprocessador e demais circuitos do sistema é enviado a 8251 para sincronização geral do sistema. Um sinal de reset na 8251 coloca a no estado de inativação e a interface permanecerá neste estado.

do até nova programação.

RD (entrada)

Leitura: habilita a interface a enviar dados ou informações do seu status ao microprocessador através do bus de dados.

WR (entrada)

Escrita: habilita a interface a receber dados ou palavras de controle de microprocessador através do bus de dados.

C/D (entrada)

Controle ou dado: este pino indica para a interface se o conteúdo do bus de dados é uma palavra de controle ou um dado.

CLK (entrada)

Clock: a 8251 é um dispositivo dinâmico que necessita de um sinal de relógio de referência externo para sua sincronização, este sinal de relógio externo é conectado a este pino.

A frequência de clock deve ser mais que trinta vezes a frequência de transmissão ou recepção no modo síncrono ou cinco vezes a frequência de transmissão ou recepção no modo assíncrono.

DSR (entrada)

Gerador de dados pronto: quando é utilizado um modulador para transmissão de dados, um sinal nesta entrada é utilizado para informar a interface que o modulador está ativado.

DTR (saída)

Terminal de dados pronto. Quando é utilizado um modu

lador para transmissão de dados, um sinal nesta saída é utilizado para informar ao modulador que a interface está ativada.

RTS (saída)

Solicitação para enviar dados: quando é utilizado um modulador para a transmissão de dados, a interface envia um sinal ao modulador por este pino para saber de sua disponibilidade, para que a interface possa enviar-lhe dados. Caso o modulador esteja disponível para receber dados, ele responde através de um sinal que é recebido pela interface pelo pino CTS.

CTS (entrada)

Habilitado para enviar dados: um sinal neste pino informa à interface que o modulador está habilitado para receber dados.

Obs.:

Estes quatro últimos pinos analisados (DSR, DTR, RTS, CTS) são utilizados normalmente quando é utilizado um modulador para a transmissão dos dados. Caso não seja utilizado um modulador para o funcionamento normal da interface, basta conectarmos o pino DSR com o pino DTR e também o pino RTS com o pino CTS.

TxD (saída)

Transmissor de dados: por este pino são enviados os dados em série da interface para o periférico.

TxRDY (saída)

Transmissor pronto: indica ao microprocessador que o transmissor da interface está disponível para receber um dado, e tão logo um dado a ser transmitido é carregado na interface, este sinal é automaticamente ressetado.

TxEMPTY (saída)

Transmissor ocupado: um sinal indica ao microprocessador que o transmissor está ocupado, transmitindo um dado e que a interface está no modo de transmissão. Enquanto este sinal estiver ativo, o microprocessador não pode ler nem escrever na interface.

TxC (entrada)

Clock de transmissão: sinal de clock para transmissão de dados e a cada novo período deste sinal de clock, um novo bit é transmitido pela interface.

RxD (entrada)

Receptor de dados: por este pino são recebidos pela interface os dados em série do periférico.

RxRDY (saída)

Receptor pronto: indica ao microprocessador que o receptor da interface possui um dado transmitido e disponível para ser lido pelo microprocessador e tão logo este dado seja lido pelo microprocessador, este sinal será automaticamente resettado.

RxC (entrada)

Clock de recepção: sinal de clock para recepção de dados e a cada novo período deste sinal de clock, um novo bit é lido pela interface.

SYNDET (entrada/saída)

Deteção do caracter SYNC: este pino pode ser usado como entrada ou saída, dependendo da maneira em que tenha sido

programado, e é utilizado somente em modo síncrono. Para a transmissão do modo síncrono, antes da transmissão dos dados é transmitido um ou dois caracteres especiais (SYNC) para sincronização da comunicação. Quando o pino de SYNDT é programado como saída um sinal neste pino, indica que interface possui um caracter SYNC no seu buffer de recepção.

Quando é programado como entrada, um sinal neste pino faz com que a interface inicie a montagem de caracteres de dados a partir do próximo período de relógio.

XI.3 ARQUITETURA INTERNA

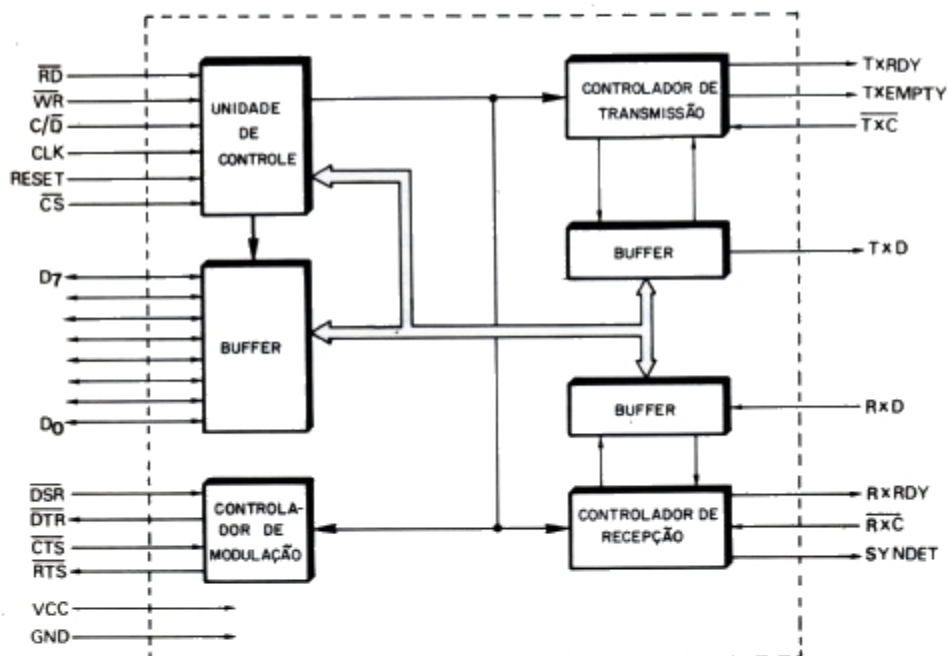


Fig. XI.4 Arquitetura interna da 8251.

As portas de transmissão e recepção da 8251 são controladas por controladores específicos que geram as variáveis de controle internas para os buffers de transmissão e recepção; os dados destes buffers são enviados através de um bus de dados interno para o buffer de dados, o qual está conectado diretamente ao bus de dados do sistema. A unidade de controle da interface decodifica os sinais de controle do sistema e gera os comandos para os controladores, conforme a programação da interface feita por meio das palavras de controle, fornecidas pelo sistema. O controlador de modulação envia e recebe os sinais de controle para o modulador e demodulador externo, quando este for utilizado.

XI.4 MODO DE OPERAÇÃO

A interface de comunicação serial é um dispositivo programável pelo microprocessador por meio de palavras de controles que são enviadas pelo microprocessador para a interface. Existem, basicamente, duas instruções do microprocessador para a interface serial: uma de escrita e outra de leitura. Para a escrita de um dado a ser transmitido ou leitura de um dado a ser recebido, esta interface é praticamente transparente para o microprocessador, pois para a transmissão de um dado após a programação de modo que a comunicação é desejada, basta escrevermos o dado na interface e esta providenciará a transformação do dado paralelo recebido por ela em uma sequência de pulsos adequados e sincronizados para a transmissão. Da mesma maneira, após uma recepção ter sido efetuada pela interface, esta transforma a sequência de pulsos recebida em um dado paralelo, disponível para ser lido pelo microprocessador no instante que este desejar, apenas com uma instrução de leitura. A interface encarrega-se de gerar os sinais para o periférico, informando que está ou não disponível para receber dados, informa o microprocessador que o dado está disponível ou não para ser lido, etc.

Para a programação da interface, no modo de comunicação desejado, basta usarmos a mesma instrução de escrita na interface para enviarmos as palavras de controle, diferenciando nesta instrução apenas o endereço no qual a informação será escrita.

ta. Para que se possa fazer esta diferenciação de endereço, basta conectarmos um bit do bus de endereço no pino C/D.

Desta forma, apenas variando este bit de endereço, podemos informar para interface se a informação que estamos mandando é uma palavra de controle ou um dado. Da mesma forma procederemos na leitura, sendo que pode ser feita a leitura de um dado da interface ou, então, a leitura do status atual, conforme o endereço selecionado.

A programação da interface é feita por duas palavras de controle diferentes. Uma que é chamada instrução de modo e outra que é a instrução de comando.

A instrução de comando define as características de operação atual da interface. Após uma instrução de modo, as outras palavras de controle enviadas para a interface serão interpretadas como instrução de comando.

Antes da análise das instruções de modo e comando, vamos analisar algumas características das comunicações para que possamos entender melhor estas instruções.

As comunicações podem ser feitas de dois modos distintos: síncrono ou assíncrono.

No modo síncrono, antes de enviarmos os dados a serem transmitidos, devemos mandar um ou dois caracteres especiais (dependendo do modo de operação do periférico) para a sincronização da transmissão. No modo assíncrono, antes de cada bit do dado, devemos ter um bit de inicialização que é de nível zero ou alguns bits de término do dado que são de nível um. Estes bits são automaticamente inseridos pela interface.

Para ambos os modos de comunicação, devemos definir o comprimento do dado que pode variar de 5 até 8 bits; também a velocidade de transmissão deve ser definida para ambos os modos. No caso da 8251 temos as seguintes velocidades de transmissão.

Modo síncrono:

- 1 bit por período de clock.

Modo assíncrono:

- 1 bit por período de clock,
- ou - 1 bit por 16 períodos de clock,
- ou - 1 bit por 64 períodos de clock.

Também durante uma transmissão, podemos ter a introdução de mais um bit que é utilizado para checar se não houve nenhum problema de transmissão alterando o dado. Este bit é chamado de bit de paridade e funciona da seguinte maneira: caso a paridade selecionada seja par, durante a transmissão de um dado, se o número total de bits "um" for par, este bit de paridade será "zero" e, caso, o número total de bits "um" for ímpar, este bit será "um". Assim, se o receptor receber um número par de bits "um", significa que, muito provavelmente, a transmissão foi correta, e se o número de bits "um" recebidos for ímpar, certamente houve deterioração do dado durante a transmissão. De maneira análoga é efetuada a checagem para paridade ímpar.

Vamos agora ver o formato da instrução de modo.

Para comunicação síncrona o formato da instrução de modo está mostrada na figura XI.5.

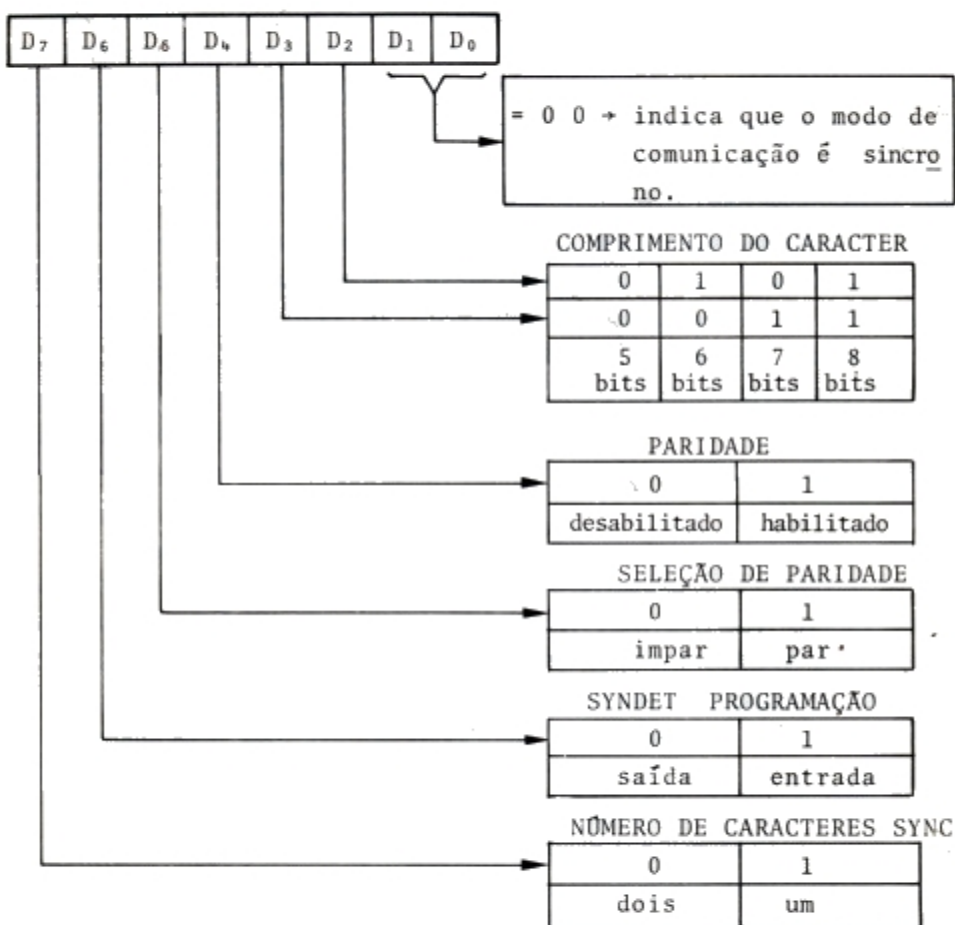


Fig. XI.5 Formato de instrução de modo para modo síncrono.

Para a comunicação assíncrona, o formato da instrução de modo está mostrado na figura XI.6.

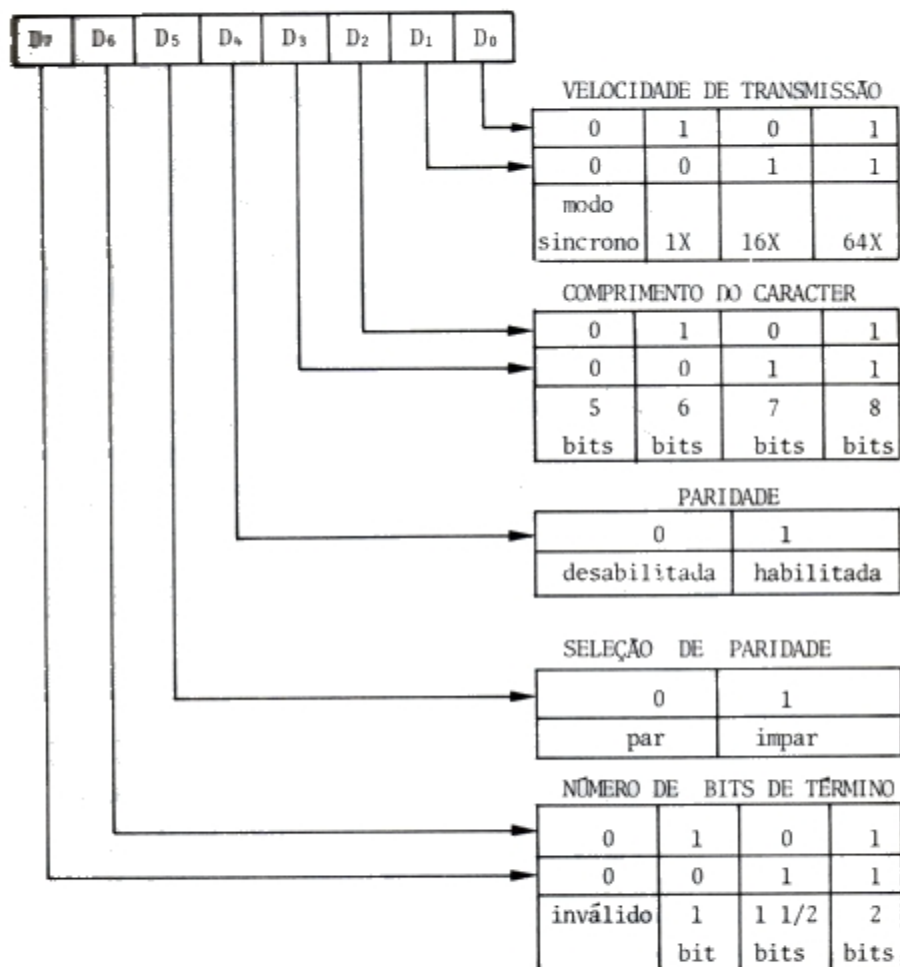


Fig. XI.6 Formato da instrução de modo para modo assíncrono.

Para ambos os modos de transmissão o formato da instrução de comando está mostrado na figura XI.7.

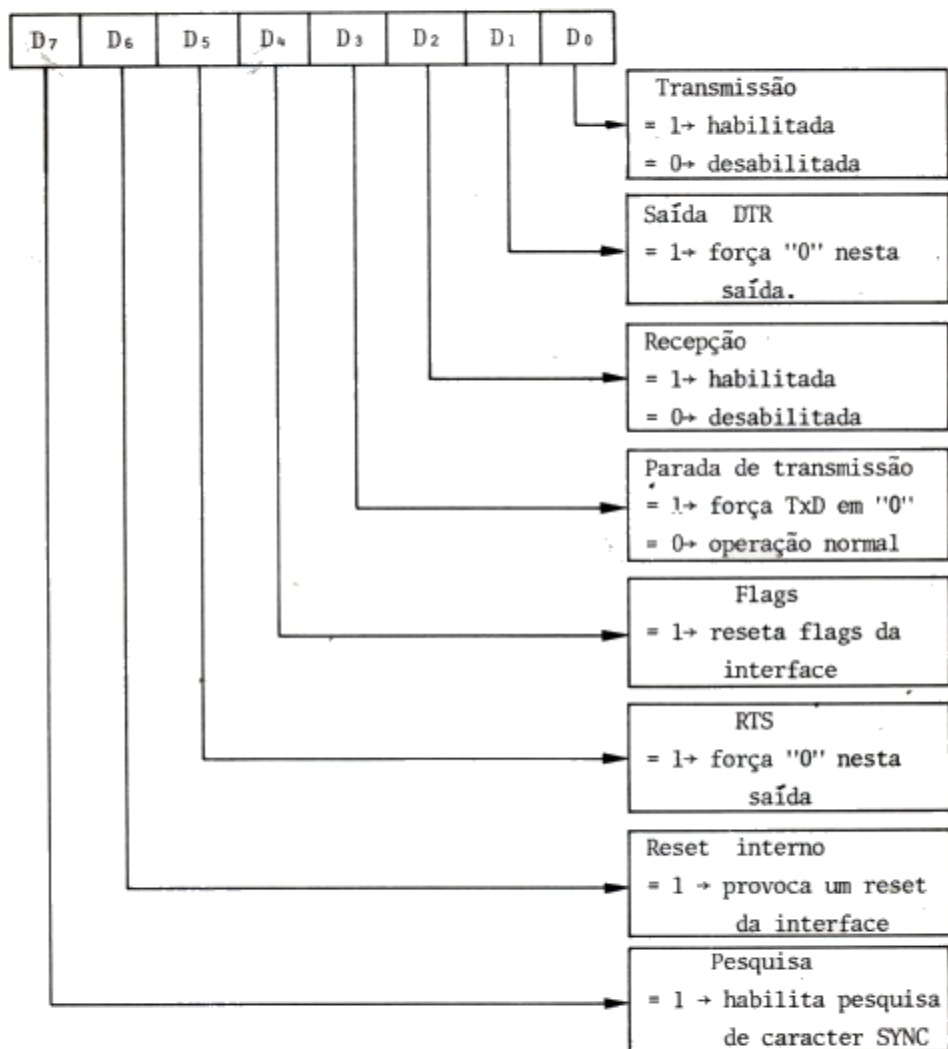


Fig. XI.7 Formato da instrução de comando.

Para a programação da 8251, é necessário que escrevamos, no endereço da palavra de controle da interface, um dado no formato da instrução de modo e, após, dados no formato da instrução de comando, conforme a necessidade.

Caso desejamos saber a qualquer instante qual é a situação atual da interface, efetuamos a leitura da palavra de status da interface. Esta palavra de status possui algumas informações idênticas às dos pinos de controle da interface além das informações das flags internas da interface. As flags da interface podem ser habilitadas ou não e também ressetadas pela instrução de comando. Nenhuma flag bloqueia o funcionamento da interface, porém elas podem ser testadas pelo programa para de terminação de alguma condição especial.

As flags disponíveis na 8251 são as seguintes:

- Flag de erro de paridade, esta flag é ligada quando é detetado um erro de paridade.

- Flag de sobreposição, quando o microprocessador não leu o dado recebido pela interface e esta começou a receber um novo dado.

- Flag de formação, utilizado apenas no modo assíncrono e é ligada quando um bit de término não é constatado no final de um dado.

Estas flags mais os outros bits que formam a palavra de status do 8251 estão dispostos como mostrado na figura XI.8.

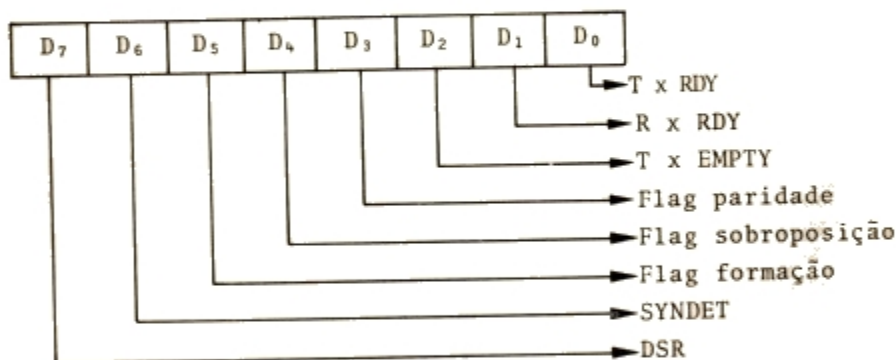


Fig. XI.8 Formato da palavra de status.

XII.1 GENERALIDADES

Com o crescimento da utilização dos microprocessadores, alguns circuitos integrados especiais estão sendo desenvolvidos para facilitar a execução de algumas tarefas do microprocessador ou, ainda, aumentar a sua capacidade. Estes circuitos especiais recebem o nome de periféricos.

Os circuitos periféricos executam tarefas, tais como: controlar as prioridades de interrupções, permitir um acesso direto de um dispositivo de entrada ou saída com a memória, executar contagens, etc.

É comum ter-se acoplado ao microprocessador sistemas mecânicos, tais como: impressores, motores passo a passo para controle de deslocamentos precisos, etc. Para acionamento destes sistemas é necessário a utilização de contagens ou controle de intervalos de tempos que exigem rotinas especiais e estabelecimento de loops de programa os quais prendem o microprocessador, impedindo-o de executar outras funções além de complicar e exigir maior trabalho do software, programando rotinas especiais para estas funções.

O contador programável 8253 é um circuito integrado do periférico especialmente projetado para estas aplicações, constituído por três contadores independentes e de modo de contagem programável.

XII.2 DESCRIÇÃO DA PINAGEM

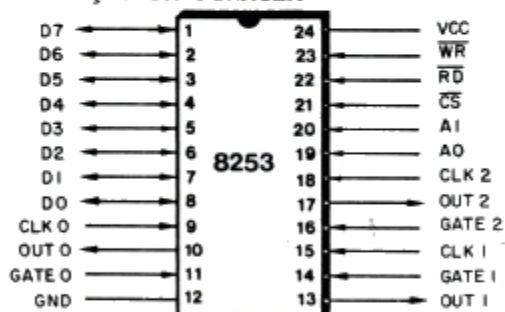


Fig. XII.1 Configuração dos pinos.

V_{CC}, GND

Alimentação

V_{CC} = + 5 Volts ± 5%

GND → referência 0 Volts

D₇, D₀ (entrada/saída)

Bus de dados bidirecional: conectados diretamente ao bus de dados do microprocessador. Os dados que são trocados entre o periférico e o microprocessador podem ser um valor para ser carregado em um dos contadores ou o valor atual de um dos contadores ou, ainda, uma palavra de controle do periférico.

A₁, A₀ (entrada)

Bus de endereço: conectado diretamente ao bus de endereço do microprocessador e são usados para selecionar um dos três contadores ou o registrador da palavra de controle, conforme a seguinte configuração.

| A ₁ | A ₀ | |
|----------------|----------------|---------------------|
| 0 | 0 | contador n° 0 |
| 0 | 1 | contador n° 1 |
| 1 | 0 | contador n° 2 |
| 1 | 1 | palavra de controle |

CLK₀, CLK₁, CLK₂ (entrada)

Sinais de clock para os contadores: cada contador tem entrada de clock independente, podendo-se, desta maneira, usar diferentes sinais para cada contador. O período deste sinal será o período de cada contagem.

GATE₀, GATE₁, GATE₂ (entrada)

Gates: cada contador pode ser disparado por software

ou hardware através de um sinal externo de gatilhamento aplica do nestes pinos.

OUT₀, OUT₁, OUT₂ (saída)

Saída dos contadores.

WR (entrada)

Escrita: habilita o periférico a receber dados ou palavra de controle do microprocessador através do bus de dados.

RD (saída)

Leitura: habilita o periférico a enviar o valor do contador selecionado ao microprocessador através do bus de dados.

CS (entrada)

Seleção do circuito: habilita a comunicação entre o contador programável e o microprocessador.

XII.3 ARQUITETURA INTERNA

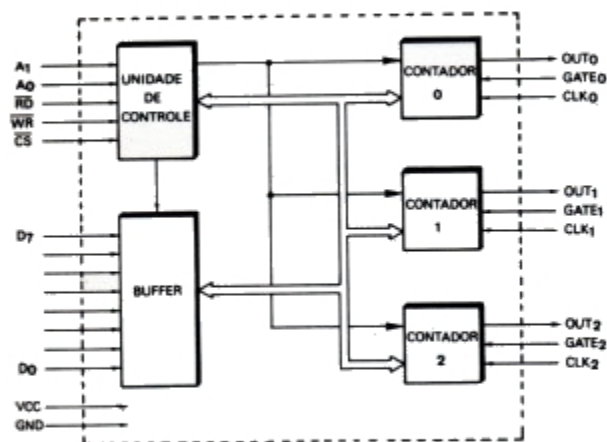


Fig. XII.2 Arquitetura interna 8253.

Basicamente o contador programável é composto por três contadores de 16 bits independentes entre si, uma unidade de controle interna que gera as variáveis de controle para os contadores e seleciona o modo de operação e mais um buffer para o bus de dados.

XII.4 MODO DE OPERAÇÃO

O contador programável 8253 é constituído de três contadores independentes entre si, de contagem decrescente, isto é, os contadores iniciam a contagem por um valor que é determinado por um programa e a cada pulso de clock decrementam este valor até o zero. Os valores dos contadores são decrementados na transição de "1" para "0" ^{transição de nível do clock} do nível do sinal de clock. Cada contador possui 16 bits e a sua contagem pode ser em modo decimal ou binário, dependendo de como foi programado. Caso qualquer um dos contadores seja programado para contar na forma decimal, o maior valor que pode ser usado para início da contagem é 9999, desta forma de 9999 a 0000, o número máximo de contagem é de 10.000 e quando programado para contar na forma binária o número máximo de contagem é 65.536, de FFFF a 0000.

Após um valor carregado no contador por meio de um sinal no pino \overline{WR} , a contagem só é iniciada na primeira transição de "1" para "0", do nível do sinal de clock, (após o sinal de write e gate estarem a nível "1") (ver fig. XII.3).

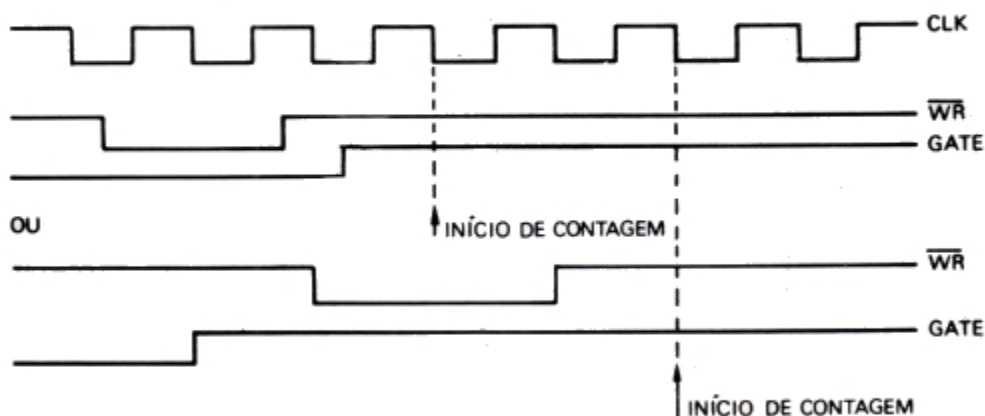


Fig. XII.3

Cada contador é individualmente programado, e uma série de palavras de controle devem ser enviadas do microprocessador para o periférico para a seleção do modo de contagem e informação do valor inicial de contagens.

Os modos de contagem são de cinco tipos diferentes como definido a seguir.

Modo 0 (figura XII.4) - após a seleção deste modo, a saída do contador estará no nível lógico "0". Para os demais modos, após a seleção a saída do contador estará no nível lógico "1".

A saída permanecerá com o nível "0" até que a contagem termine, neste instante a saída mudará para nível "1" e permanecerá com este nível até que um novo valor para a contagem seja carregado ou um novo modo de operação seja selecionado.

Caso o nível do gate de contador mude para "0" durante a contagem, esta para e só continua quando o nível do gate voltar para "1".

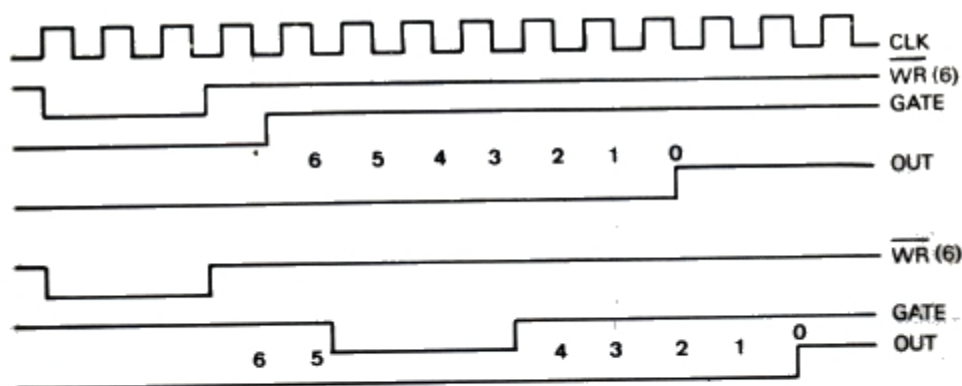


Fig. XII.4 Modo 0.

Modo 1 (figura XII.5) - Ao ser iniciada a contagem a saída do contador vai para o nível "0" e assim permanece até que termine a contagem, retornando neste instante para o nível "1".

Um pulso no gate provoca o reinício da contagem a partir do valor inicial.

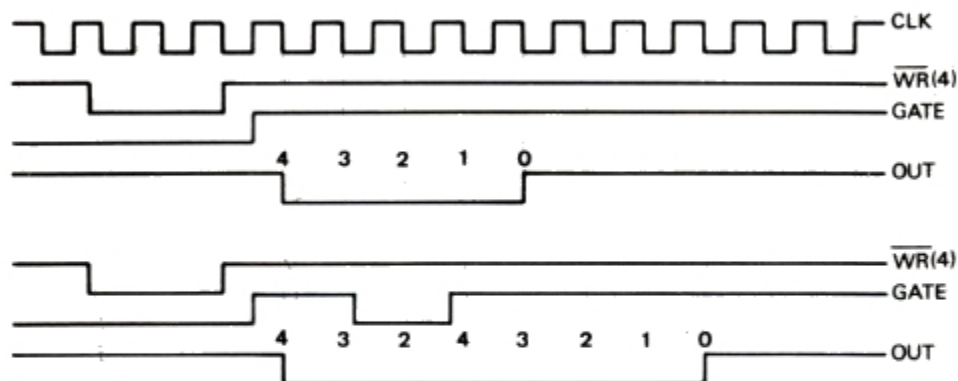


Fig. XII.5 Modo 1.

Modo 2 (figura XII.6) - no último período da contagem, a saída do contador terá nível lógico "0", após o que a contagem é reiniciada.

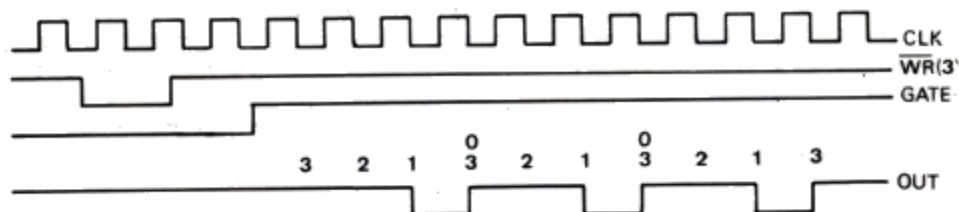


Fig. XII.6 Modo 2.

Modo 3 (figura XII.7) - O contador é decrementado duas vezes a cada transição do sinal de clock a menos da primeira vez, quando o valor da contagem é ímpar. A saída do contador inverterá o nível lógico a cada nova contagem.

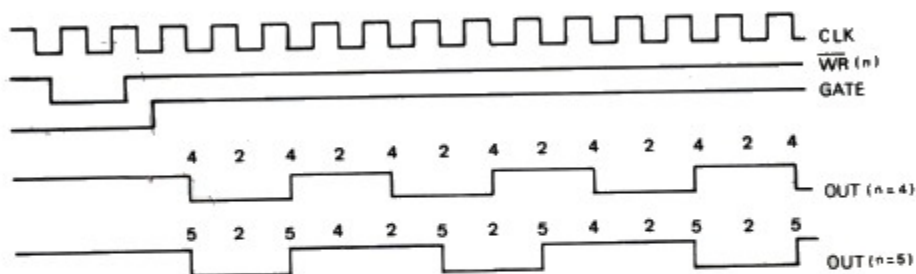


Fig. XII.7 Modo 3.

Modo 4 (figura XII.8) - A saída do contador terá nível lógico "0" após o término da contagem, durante um período do sinal de clock e só inicia novamente quando o valor da contagem é recarregado.

Caso, no meio de uma contagem, o sinal de gate passe a nível lógico "0", a contagem é inibida e recomeça do valor inicial após a volta do sinal de gate para o nível lógico "1".

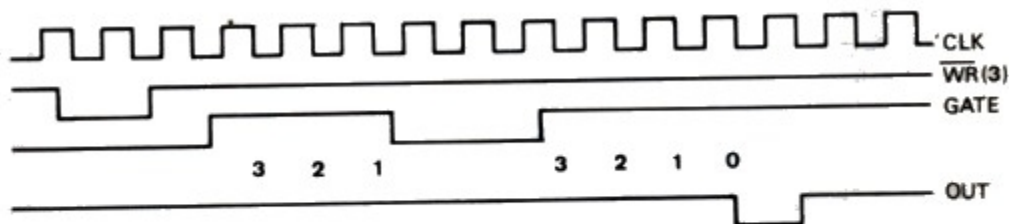


Figura XII.8 Modo 4.

Modo 5 (figura XII.9) - A saída do contador terá nível lógico "0", após o término da contagem, durante um período do sinal de clock e só inicia novamente quando o valor da contagem é recarregado. Caso, no meio de uma contagem, o sinal de gate passe para o nível lógico "0", a contagem não é inibida, en

tretanto se o sinal do gate retornar ao nível lógico "1", a contagem será reiniciada.

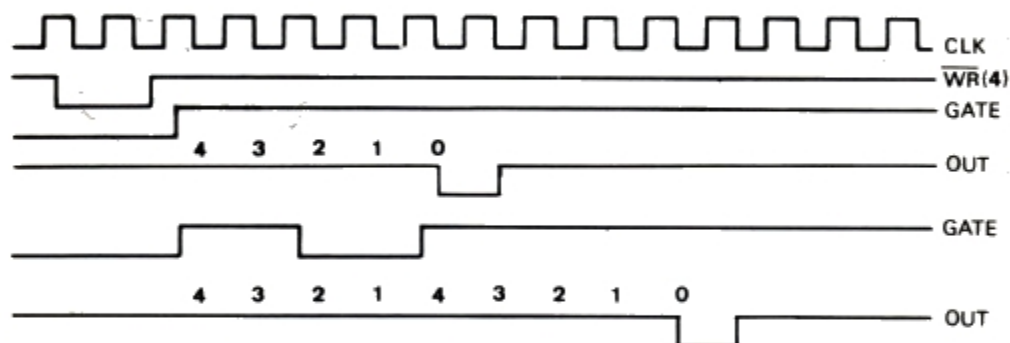


Fig. XII.9 Modo 5.

O modo de operação e a quantidade inicial de cada contador podem ser programados independentemente. Para a programação o microprocessador tem que enviar ao periférico uma palavra de controle que descreva o contador, seu modo de operação, o byte que será enviado posteriormente, para estabelecer o valor inicial da contagem (o mais significativo, o menos significativo ou o menos significativo seguido do mais significativo) e o modo de contagem (binário ou decimal). A palavra de controle e o valor inicial da contagem devem ser escritos no periférico antes da contagem ser iniciada.

A palavra de controle também pode ser utilizada para a leitura de um contador a qualquer instante. Para isto quando a operação de ler é desejada, a palavra de controle, apropriadamente organizada, é enviada ao periférico. O valor do contador, neste instante, é armazenado no buffer para posterior leitura sem que a contagem seja interrompida ou alterada.

Tanto a leitura ou escrita do valor do contador podem ser programadas de três maneiras diferentes:

- Leitura ou escrita do byte mais significativo apenas,
- leitura ou escrita do byte menos significativo apenas e,

- leitura ou escrita primeiramente do byte menos significativo e após o byte mais significativo.

Após a programação, a leitura ou escrita são executadas através dos comandos normais gerados pelas instruções de leitura ou de escrita.

A palavra de controle é formada por 8 bits. Seu formato e função de cada bit está mostrado na figura XII.10.

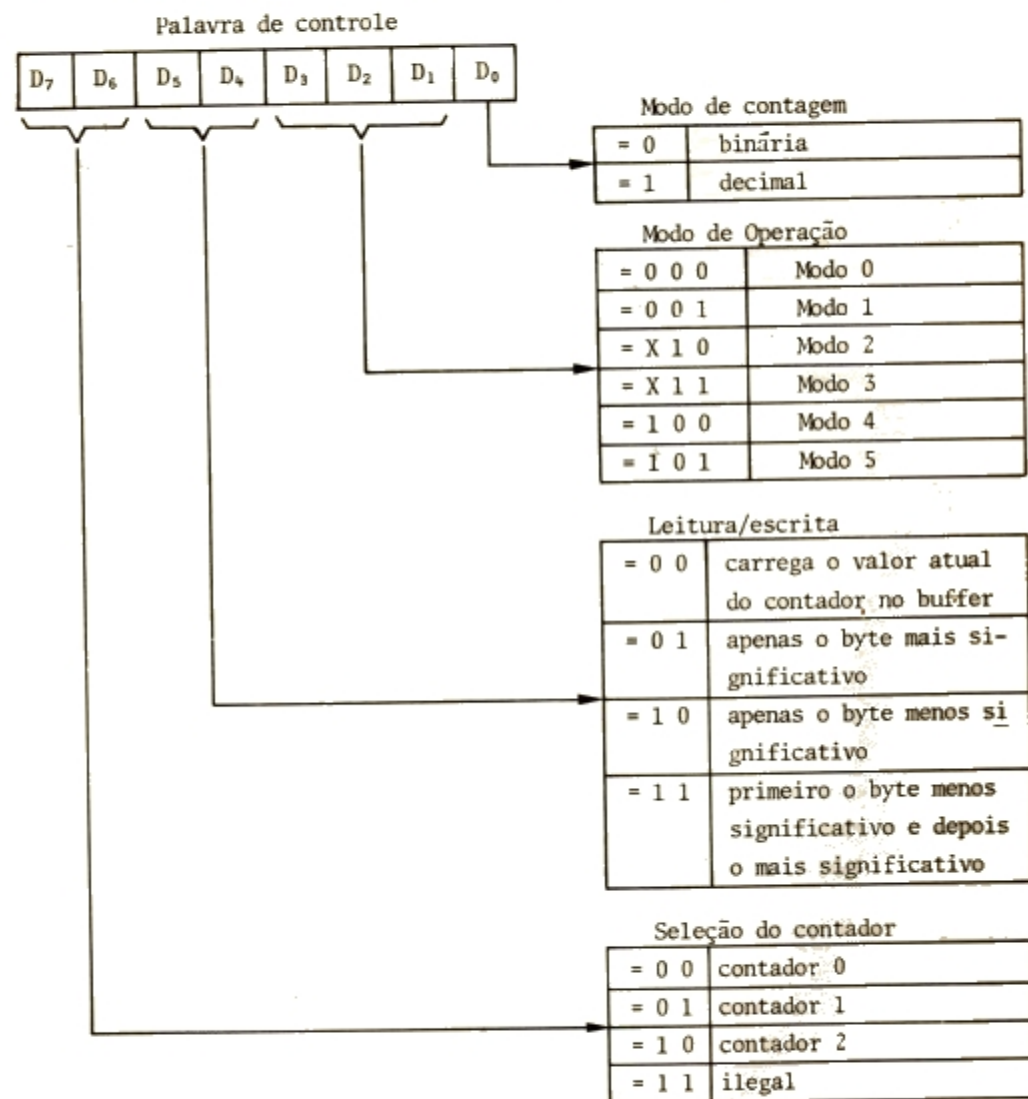


Fig. XIII.10 Formato da palavra de controle.

Com as possibilidades existentes no contador programável 8253 temos disponíveis contadores bem versáteis, capazes de executar quase todas as tarefas de contagem ou determinação de intervalos de tempo necessárias para sistemas de microprocessador e, pela sua facilidade de manipulação, reduz sensivelmente o software necessários para estas tarefas, facilitando assim a programação do microprocessador.

Para trabalharmos com um sistema de microprocessadores, devemos projetar o hardware e o software necessário para cada aplicação especial. Com o que já foi exposto nos capítulos anteriores, podemos partir para a delimitação do hardware.

O sistema de microcomputador é constituído dos seguintes blocos básicos:

- unidade central de processamento
- memória
- dispositivos de entrada e saída.

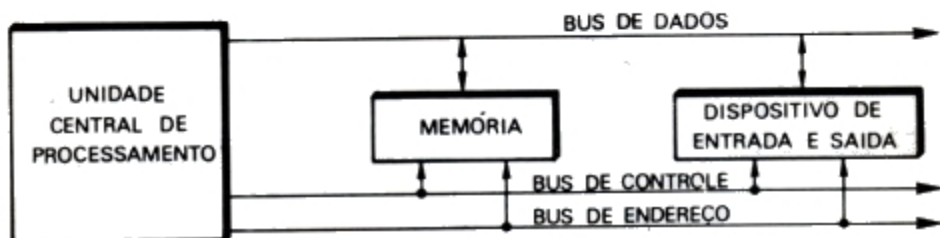


Fig. XIII.1 Configuração básica de microcomputador.

Para a unidade central de processamento, devemos escolher qual o microprocessador que iremos utilizar conforme a conveniência. Com o microprocessador 8080 para a configuração da unidade central de processamento, devemos utilizar também o gerador de clock 8224 e controlador do sistema 8228. As memórias podem ser divididas em memórias de dados, utilizando-se memórias RAM, e memórias programadas, utilizando-se memórias ROM. Para os dispositivos de entrada e saída, podemos ter interface de comunicação serial com o 8251 e interface de comunicação paralela com o 8255. Opcionalmente, podemos utilizar circuitos especiais como contador programável 8253. Agrupando-se, ordenadamente, todos estes circuitos, teremos um microcomputador como o mostrado em diagrama de blocos da figura XIII.2.

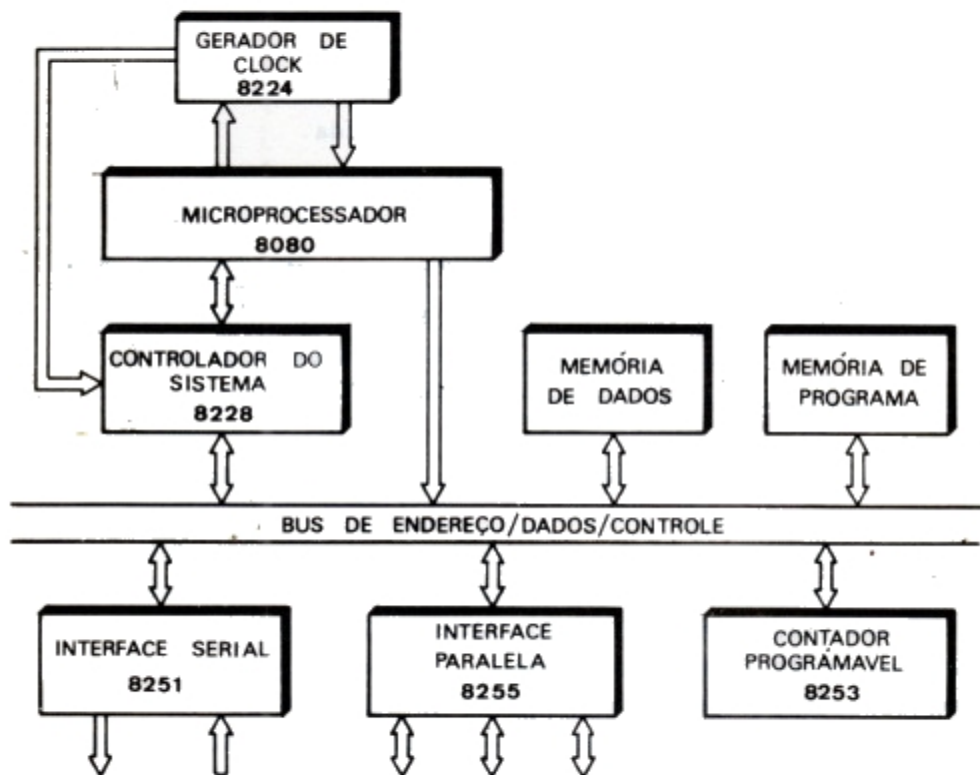


Fig. XIII.2

Caso seja selecionado um outro microprocessador, a estrutura do sistema não será alterada.

Apesar do microprocessador 8080 ser ainda hoje um dos mais utilizados, escolhemos para estudo do circuito detalhado o microprocessador 8085 pelas suas vantagens: maior velocidade, menor número de circuitos integrados no sistema, necessita de fonte mais simples e possui algumas instruções a mais. A estrutura do sistema de microcomputador com o microprocessador 8085 é mostrada na figura XIII.3.

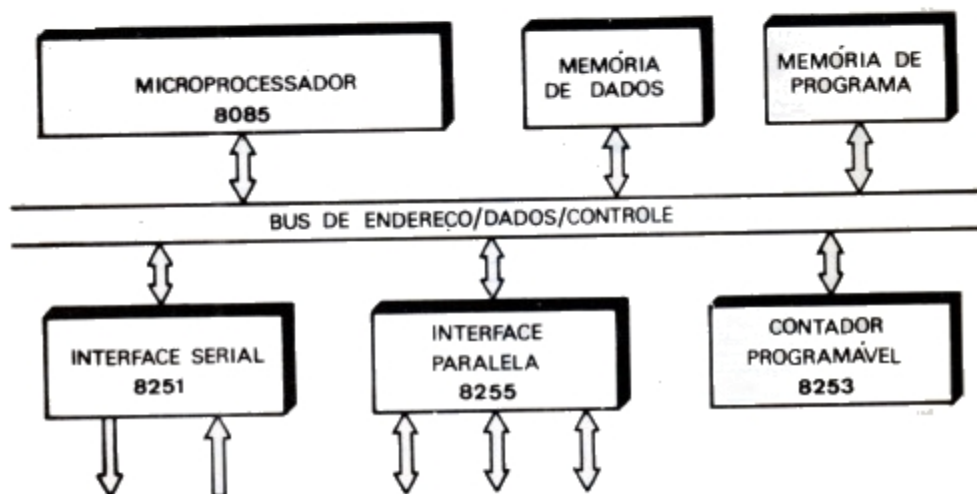


Fig. XIII.3

Definido o microprocessador para a unidade central de processamento, o próximo passo é a ligação para cada um de seus pinos.

Como o microprocessador escolhido é o 8085 e este possui pinos que são multiplexados para endereço e dados, torna-se necessária a utilização de portas para a separação das informações neste pino. Como foi visto no capítulo IV, as portas são buffers com latch, isto é, a partir de um comando externo, transferem as informações de suas entradas para as saídas e retêm estas informações até um novo comando externo. Entre os circuitos integrados disponíveis para esta aplicação, vamos selecionar o circuito integrado 8212.

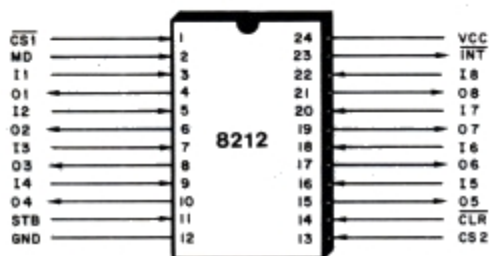


Fig. XIII.4

Este circuito é composto por oito portas paralelas e sua lógica de controle é a seguinte: o pino de modo de operação (MD) é usado para selecionar por onde é recebido o sinal de comando para transferir as informações das entradas das portas para as saídas. Quando o nível lógico neste pino é "0", o comando de controle é efetuado pelo pino de strobe (STB); quando o nível lógico é "1", o comando é efetuado pelo produto dos dois pinos de seleção do circuito (\overline{CS}_1 , \overline{CS}_2).

Quando o microprocessador 8085 coloca informações de endereço em seus pinos multiplexados de dados e endereços, coloca automaticamente para sincronização em seu pino ALE um nível "1" e, quando as informações são dados, coloca um nível "0".

Utilizando este sinal do pino ALE do 8085 conectado ao pino CS₂ do 8212, os outros sinais de controle são conectados da seguinte maneira:

MD = "1"

STB = "1"

\overline{CS}_1 = "0"

Desta forma, quando em CS₂ tivermos "1", informações das entradas das portas são transferidas para a saída; quando em CS₂ tivermos "0", as informações da saída permanecerão inalteradas mesmo com a variação das entradas das portas.

O pino de entrada de clear (CLR) não será utilizado nesta aplicação e é conectado a "1" para que não interfira no funcionamento das portas e o pino de saída de interrupção para o sistema (INT), que também não é utilizado nesta aplicação, ficará não conectado.

O circuito de ligação da 8212, bem como sua conexão com o microprocessador 8085, é mostrado na figura XIII.5.

Para os demais pinos do microprocessador 8085 teremos:

- para clock, pode-se ter um cristal de, por exemplo, 6,144 MHz conectado diretamente aos pinos 1 e 2, que resultará numa frequência de clock de operação de 3,072 MHz.

- ao pino de reset, pode-se conectar uma chave externa para a geração manual do sinal com um circuito para a temporização adequada, como visto na figura XIII.5.

- os pinos de interrupção estão disponíveis para a conexão de qualquer circuito integrado que solicite uma interrupção, como por exemplo, as saídas do circuito contador programável 8253, interrompendo, assim, o processamento da unidade central de processamento, também, conectar resistores ligados à terra para que, quando nenhum sinal lhes for aplicado, eles recebem através dos resistores, nível lógico "0", não alterando assim o perfeito funcionamento da unidade central de processamento.

- Os pinos de RDY e HOLD são utilizados apenas para aplicações especiais, tais como sincronização com memórias, ou periféricos lentos e operações de acesso direto à memória, respectivamente. Como não iremos utilizar nenhuma aplicação espe

Tendo a unidade central de processamento definida, vamos passar à definição das memórias.

O microprocessador 8085 possui 16 bits para endereçamento de memória ($A_{15} - A_0$); desta maneira, ele pode endereçar 64k (2^{16}) endereços diferentes de memória, porém a quantidade necessária dependerá de cada aplicação. Vamos estabelecer 4k de memória ROM e 1k de memória RAM o que acreditamos ser suficiente para a memória das aplicações gerais.

Desta maneira, podemos ter programas com até 4096 bytes e 1024 bytes para armazenamento de dados que podem ser utilizados para programa.

Para as memórias ROM, vamos utilizar o circuito integrado 2716 e, para as memórias RAM, o circuito integrado 2114; desta maneira, são necessários dois 2716 que resultarão em um total de 4kx8 e dois 2114, para um total de 1kx8.

É sempre aconselhável que as memórias ROM sejam colocadas a partir do endereço inicial (A_{15} à $A_0 = 0$) devido ao fato de toda vez que o circuito é ligado ou ressetado o microprocessador inicia a execução da instrução do endereço inicial.

Desta maneira, como foi estabelecido 4096 bytes de memória ROM e o endereçamento é feito de modo binário temos:

| | A_{15} | A_{14} | A_{13} | A_{12} | A_{11} | A_{10} | A_9 | A_8 | A_7 | A_6 | A_5 | A_4 | A_3 | A_2 | A_1 | A_0 |
|---|----------|----------|----------|----------|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| endereço do 1º byte da memória ROM. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| endereço do último byte da memória ROM. | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

o que em hexadecimal corresponde a dizer que o endereço da memória ROM é de 0000 à 0FFF que correspondem a 4096 endereços diferentes.

Para possibilidade de futuras expansões de memória RAM, vamos projetar o circuito reservando os 12k endereços seguintes.

As 1024 posições de memória RAM podem, por exemplo, serem colocadas entre o endereço 4000 à 43FF e os 7k seguintes, reservados para futuras expansões. Desta maneira, o nosso mapa de memória é como se vê na figura XIII.6.

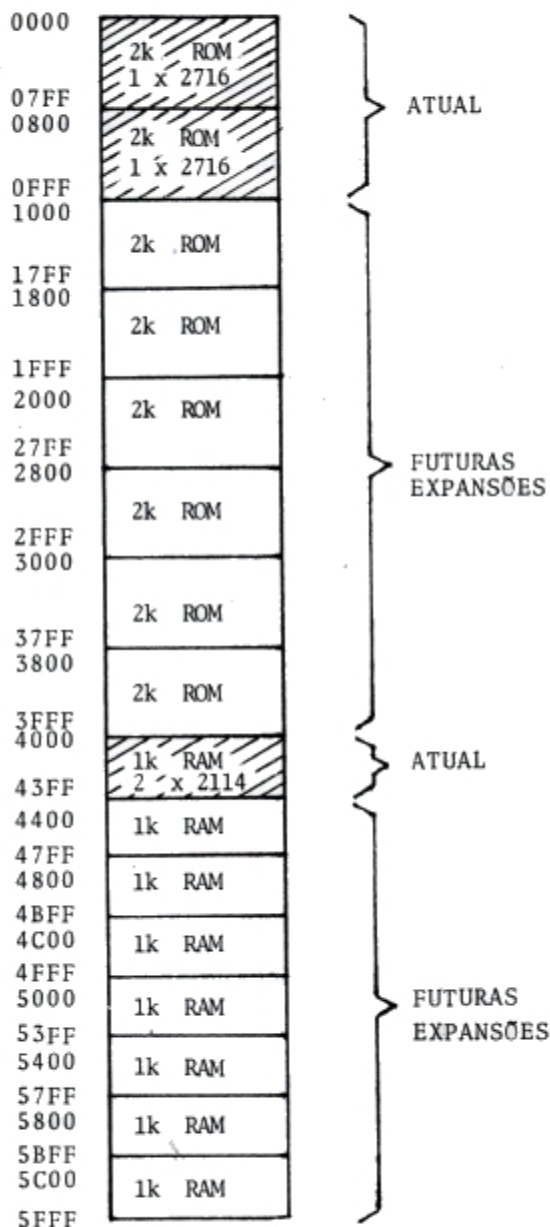


Figura XIII.6.

Para a formação desta configuração de memórias, os bits de endereço A_{10} a A_0 são conectados diretamente aos circuitos 2716, e os bits A_9 a A_0 , aos circuitos 2114. Os outros bits são utilizados para a seleção de cada circuito integrado e, para esta seleção, devemos fazer uma decodificação adequada dos bits. Esta decodificação pode ser feita pelo decodificador 8205 cuja configuração de pinos pode ser vista na figura XIII.7.

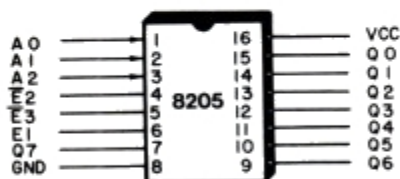


Fig. XIII.7

Quando os sinais nos pinos de habilitação "E" do decodificador 8205 tiverem a seguinte configuração: $E_1 = "1"$; $\bar{E}_2 = "0"$; $\bar{E}_3 = "0"$ uma de suas saídas "Q" terá nível "0" e as demais terão nível "1", dependendo da configuração em seus pinos de endereço "A".

Na tabela da figura XIII.8, temos as configurações possíveis do decodificador 8205.

| E_1 | E_2 | E_3 | A_2 | A_1 | A_0 | Q_0 | Q_1 | Q_2 | Q_3 | Q_4 | Q_5 | Q_6 | Q_7 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Fig. XIII.8

Utilizando-se o decodificador 8205, as conexões, como se vê na figura XIII.9, podem ser feitas para a lógica de endereçamento descrita na figura XIII.6:

- o sinal I/\overline{M} do microprocessador 8085 habilitará os decodificadores das memórias.

- o bit A_{14} de endereço, quando tiver nível lógico "0", habilitará o decodificador das memórias ROM e desabilitará o de codificador das memórias RAM. Quando o nível neste bit de endereço for "1", a habilitação será exatamente inversa.

- os bits A_{13} , A_{12} , A_{11} selecionarão um dos circuitos integrados 2716.

- os bits A_{12} , A_{11} , A_{10} selecionarão um dos conjuntos compostos por dois circuitos integrados 2114.

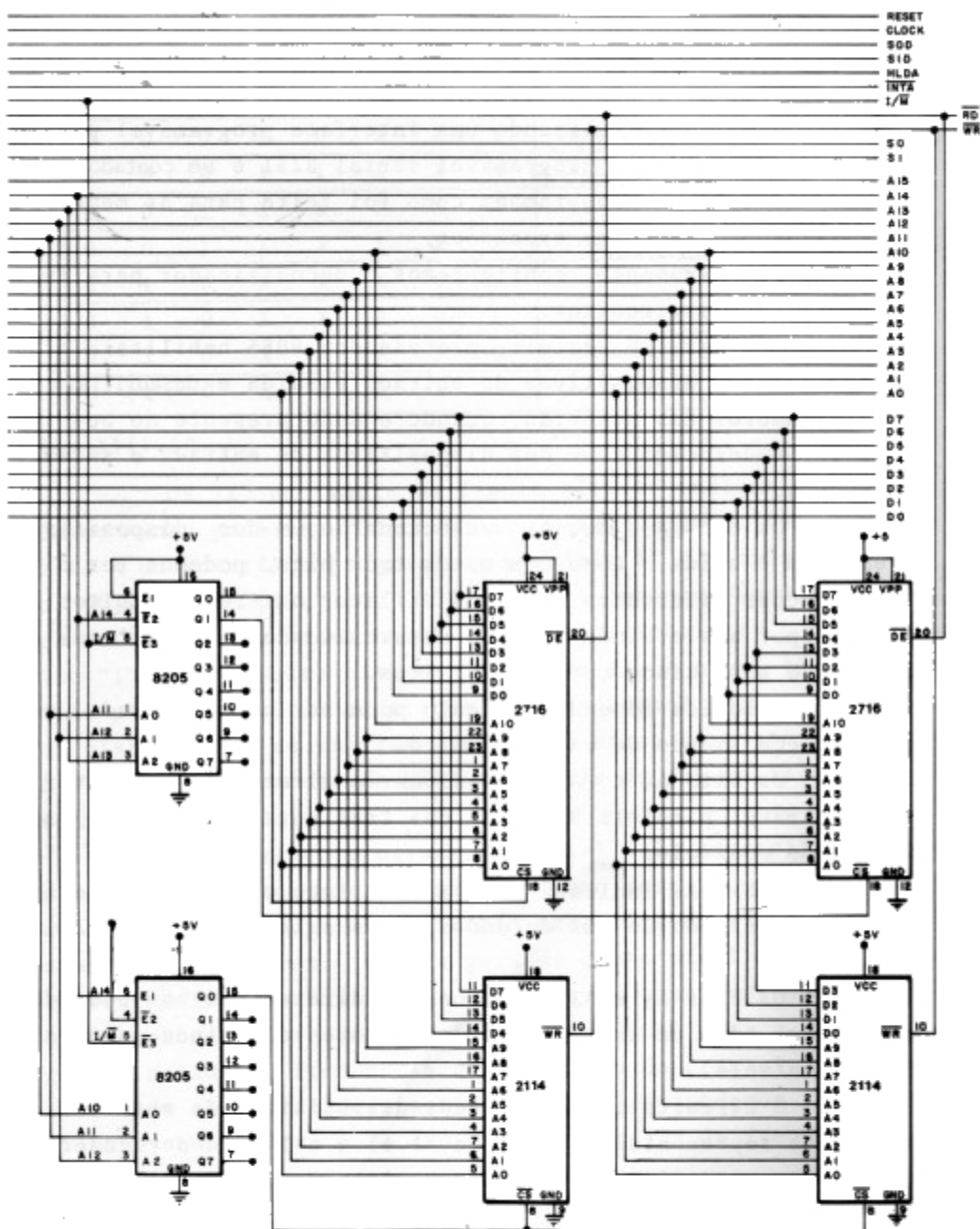


Fig. XIII.9

O microprocessador 8085 pode ainda utilizar até 512 dispositivos de entrada e saída, selecionados por seus 8 bits menos significativos de endereço ($A_7 - A_0$). Vamos então completar nosso circuito, utilizando uma interface programável paralela 8255, uma interface programável serial 8251 e um contador programável 8253, deixando, também como foi feito para as memórias, possibilidade de futuras expansões.

Primeiramente, configuremos o decodificador para endereçamento destes circuitos:

O sinal I/M do microprocessador 8085 habilitará o decodificador dos dispositivos de entrada e saída e desabilitará os decodificadores das memórias, quando o dado presente no bus de dados for endereçado a um dos dispositivos de entrada e saída e vice-versa.

Os bits A_7 , A_6 , A_5 selecionarão um dos dispositivos de entrada e saída. Como com estes três bits, podemos ter oito configurações diferentes, podemos utilizar os três circuitos escolhidos por nós e ainda teremos possibilidades para futuras expansões de mais cinco outros circuitos.

A lógica de endereçamento pode ser complementada, conforme a necessidade de cada circuito.

Os bits A_4 e A_3 selecionarão diretamente na interface programável paralela 8255 uma das três portas ou, ainda, sua palavra de controle.

O bit A_2 indicará para a interface programável serial 8251 se o conteúdo do bus de dados é uma palavra de controle ou um dado.

Os bits A_1 e A_0 selecionarão diretamente no contador programável 8253 um de seus três contadores ou, ainda, sua palavra de controle.

O circuito de ligação dos dispositivos de entrada e saída está representado na figura XI.11 e a lógica de endereçamento para estes circuitos pode ser vista na figura XIII.10.

| A ₇ | A ₆ | A ₅ | A ₄ | A ₃ | A ₂ | A ₁ | A ₀ | | HEXA |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------|------|
| 0 | 0 | 0 | 0 | 0 | X | X | X | Porta A da 8255 | 07 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | Porta B da 8255 | 0F |
| 0 | 0 | 0 | 1 | 0 | X | X | X | Porta C da 8255 | 17 |
| 0 | 0 | 0 | 1 | 1 | X | X | X | Controle da 8255 | 1F |
| 0 | 0 | 1 | X | X | 0 | X | X | Dado da 8251 | 3B |
| 0 | 0 | 1 | X | X | 1 | X | X | Controle da 8251 | 3F |
| 0 | 1 | 0 | X | X | X | 0 | 0 | Contador nº 0 8253 | 5C |
| 0 | 1 | 0 | X | X | X | 0 | 1 | Contador nº 1 8253 | 5D |
| 0 | 1 | 0 | X | X | X | 1 | 0 | Contador nº 2 8253 | 5E |
| 0 | 1 | 0 | X | X | X | 1 | 1 | Controle da 8253 | 5F |

Fig. XIII.10

Os "X" na tabela da figura XIII.10 indicam que o nível lógico tanto pode ser "0" como "1". Assim, para a seleção da porta A da interface programável paralela 8255, temos o endereço "00₁₆" ou "01₁₆" ou "02₁₆" etc, qualquer um desses, sempre selecionará a mesma porta. É usual estabelecer o nível lógico "1" para aqueles bits em que tanto faz o nível lógico. Se assim procedermos, os endereços em hexadecimal para a seleção dos dispositivos de entrada e saída do nosso circuito é como está representado na coluna HEXA na tabela da figura XIII.10, na forma hexadecimal.

A interface programável serial 8251 necessita de sinal de clock tanto para transmissão como para recepção, assim como o contador programável também necessita de sinal de clock para seus contadores. Para a geração deste sinal de clock em diferentes frequências, vamos utilizar o sinal da saída clock do microprocessador 8085 que fornece um sinal com frequência igual à metade da frequência de oscilação do cristal. Para a geração de sinais de diferentes frequências, podemos utilizar circuitos contadores para a divisão sucessiva da frequência do sinal originada do microprocessador.

O circuito integrado 7490, adequadamente interligado, fornece em sua saída Q_A o sinal de frequência 10 vezes menor do que a frequência do sinal de sua entrada I_B. Colocando-se este sinal à entrada I_A de um circuito integrado 7493, também ade

quadamente interligado, este circuito fornece em sua saída Q_A um sinal de frequência 2 vezes menor; em Q_B , quatro vezes menor; em Q_C , oito vezes menor e, em Q_D , dezesseis vezes menor. Assim sendo, da maneira em que os contadores estão ligados no circuito da figura XIII.11, a frequência dos sinais das saídas de A à H é a seguinte:

$$A = 307.200 \text{ Hz}$$

$$B = 153.600 \text{ Hz}$$

$$C = 76.800 \text{ Hz}$$

$$D = 38.400 \text{ Hz}$$

$$E = 19.200 \text{ Hz}$$

$$F = 9.600 \text{ Hz}$$

$$G = 4.800 \text{ Hz}$$

$$H = 2.400 \text{ Hz}$$

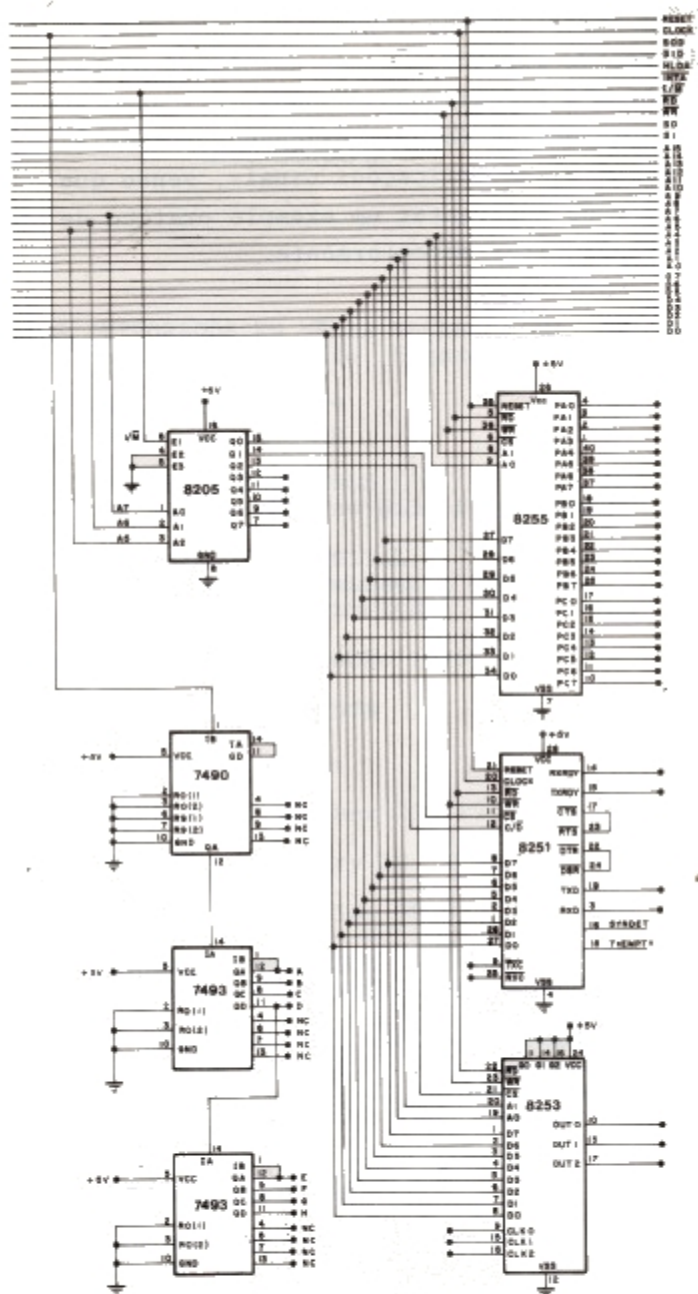


Fig. XIII.11

Os circuitos das figuras XIII.5, XIII.9, XIII.11 formam um sistema de microcomputador completo, podendo, entretanto, variar conforme a necessidade de cada aplicação geral. Este circuito aqui apresentado foi dimensionado de maneira que possa atender à maioria das aplicações usuais, sendo que o principal objetivo foi o de apresentar um exemplo prático de hardware, com os circuitos descritos anteriormente.

BIBLIOGRAFIA

- CIRCUITOS INTEGRADOS EM MÉDIA ESCALA E EM LARGA ESCALA
João Antonio Zuffo, Edgard Blücher Ltda
- COMPONENT DATA CATALOG 1979
Intel Corporation
- DIGITAL COMPUTER FUNDAMENTALS
Yaohan Chu, McGraw-Hill Book Company
- ELEMENTOS DE ELETRÔNICA DIGITAL
Ivan V. Idoeta e Francisco G. Capuano,
Distr. de Livros Érica Ltda
- ELETRICIDADE MODERNA, setembro 1975
- ELETRONICS, February 16, 1978
- ELETRONICS, March 15, 1979.
- ELETRONICS DESIGN 21, October 11, 1977
- ELETRONICS DESIGN 2, January 18, 1980
- ELETRONICS DESIGN 10, May 10, 1976
- ELETRONICS ENGINEERS HANDBOOK
Donald G. Fink, McGraw-Hill Book Company
- INTRODUCTORY EXPERIMENTS IN DIGITAL ELETRONICS AND
8080A MICROCOMPUTER PROGRAMING AND INTERFACING
Peter Rony, David G. Jarsen, Jonathan A.
Titus, McGraw-Hill Book Company
- MICROCOMPUTER COMPONENTS - DATA BOOK 1979
Mostek Corporation
- MICROPROCESSOR DEVICES - DATA BOOK 1976/77
Siemens A. G.
- MICROCOMPUTER INTERFACING WITH THE 8255 PPI CHIP
P. F. Goldsbrough, P. R. Rony,
McGraw-Hill Book Company



OUTRAS PUBLICAÇÕES

- ☐ INFORMÁTICA
- ☐ ELETRÔNICA
- ☐ TELECOMUNICAÇÕES
- ☐ PROCESSAMENTO DE DADOS
- ☐ MICROPROCESSADORES
- ☐ ELETROTÉCNICA
- ☐ MECÂNICA
- ☐ CIÊNCIAS EXATAS



ÉRICA

LIVROS DE CONTEÚDO, COM QUALIDADE
RUA JARINÚ, 594 - TATUAPÉ - CEP 03306
CX. P. 15617 - TEL.: (011) 294-8686 - S.P.