

MARCOS ROGÉRIO ALONSO

TKCONTROL
SISTEMA DE PROCESSAMENTO BASEADO NO
MICROCONTROLADOR 8031

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da Disciplina Projeto Orientado, para a obtenção do título de Bacharel em Ciência da Computação.

Orientador
Prof. WILIAN SOARES LACERDA

LAVRAS
MINAS GERAIS – BRASIL
2003

MARCOS ROGÉRIO ALONSO

TKCONTROL
SISTEMA DE PROCESSAMENTO BASEADO NO
MICROCONTROLADOR 8031

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da Disciplina Projeto Orientado curso de Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 16 de junho de 2003.

Prof. Giovanni Francisco Rabelo

Prof. Luciano Mendes dos Santos

Prof. Wilian Soares Lacerda
UFLA
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2003

DEDICATÓRIA

Este trabalho é dedicado aos meus pais, Ângela Maria Romin e Joel Alonso pelo amor, carinho e atenção que sempre me deram.

Sabor da conquista

“Pensamos em todo dia que passa,
em realizar sonhos.
Damos tudo de nós,
sangue, suor e raça,
para mostrarmos a taça
que um dia conquistamos”.

Tuca

AGRADECIMENTOS

Agradeço a todos que me deram forças para continuar este trabalho com alegria e satisfação, em especial a Luciana Arbex e Fernando Alonso, pelo incentivo e apoio que sempre me deram.

A todos os amigos com quem morei, sendo sempre verdadeiros e fiéis, tornando a minha passagem pela cidade de Lavras mais agradável.

A todos os amigos, que mesmo distantes, sabem que podem contar com este outro amigo aqui.

Agradeço também aos amigos da UFLA, que batalharam juntos comigo nesta fase tão importante de nossas vidas.

A todos os professores que até aqui contribuíram com o meu aprendizado e ao Wilian, meu professor e orientador, que mesmo em outra cidade não deixou de me orientar e me cobrar responsabilidade por este trabalho.

RESUMO

Este projeto, desenvolvido para a disciplina de “Projeto Orientado” do Curso de Ciência da Computação, tem como finalidade a confecção de um protótipo em hardware controlado por software baseado no microcontrolador 8031. Este protótipo poderá se comunicar com um computador PC compatível através da porta serial de dados, pela qual este também poderá receber o programa de controle.

ABSTRACT

This project, developed to matter of “Guided Project” of the Course Computer Science, it has as purpose the making of a prototype in hardware controlled by software based on the microcontroler 8031. This prototype can communicate with a PC computer compatible through the serial port of data, for the which this will also be able to receive the control program.

SUMÁRIO

1.	Introdução	10
2.	Referencial Teórico	13
2.1.	Microprocessadores	13
2.1.1.	Sistema Seqüencial Síncrono	15
2.1.2.	Sistemas Microprocessados	15
2.1.3.	Principais Componentes de um Sistema Microprocessado. 17	
2.1.3.1.	Memórias Somente de Leitura - Não voláteis.....	18
2.1.3.2.	Memórias de Leitura e Escrita – Voláteis.....	18
2.1.3.3.	Memórias de Leitura e Escrita – Não voláteis	19
2.1.3.4.	Temporizadores	19
2.1.3.5.	Comunicação Serial	19
2.2.	Microcontroladores	19
2.2.1.	O Microcontrolador 8031	24
2.2.1.1.	Porquê o 8031 (8051).....	24
2.2.1.2.	Principais Características de Hardware	25
2.2.1.3.	Pinagem do 8031 (8051)	27
2.2.1.4.	Memória no 8031 e 8051.....	30
2.2.1.5.	Reset no 8031 (8051)	38
2.2.1.6.	Clock no 8031 (8051).....	40
2.2.1.7.	Interrupção para o 8031 (8051)	41
2.2.1.8.	Temporizadores e Contadores no 8031 (8051).....	43
2.2.1.9.	Canal Serial para o 8031 (8051)	49
2.3.	O software (Linguagem Assembly).....	53
2.4.	Aplicações Interessantes com o 8031	54
	a) Tranca Eletrônica - Camargo (2001)	54
	b) Cadeira de Rodas - Camargo (2001)	54
3.	Metodologia	55
4.	Desenvolvimento e Resultados.....	60
4.1.	Hardware	60
4.2.	Software.....	70
5.	Conclusão.....	73
6.	Trabalhos Futuros.....	74
7.	Bibliografia	75

Anexos.....77

- A – Databook TS80C31X2
- B – Databook GAL 16V8
- C – MAX 232
- D – Registrador 74373
- E – Buffer Bidirecional 74245
- F – Databook EPROM 27C512A
- G – Databook RAM HM62256B
- H – Esquemático
- I – Programa de Gravação da PAL
- J – Leiaute da Placa do Sistema
- K - Programa Teste e Instruções
- L – Resumo e Artigo

LISTA DE FIGURAS

Figura 2.1: Sistema Genérico com Microprocessador.....	16
Figura 2.2: Diagrama em Blocos Simplificado de um Microcontrolador.....	21
Figura 2.3: Organização da Estrutura Interna do 8051.....	23
Figura 2.4: Pinagem do 8031/51 em Encapsulamento DIL.....	27
Figura 2.5: Memória de Programa.....	30
Figura 2.6: Memória de Dados.....	31
Figura 2.7: Diagrama em blocos da parte baixa da RAM interna.....	33
Figura 2.8: Expandindo RAM e ROM.....	37
Figura 2.9: Reset no 8031/51.....	38
Figura 2.10: Clock para o 8031/51.....	40
Figura 2.11: Registrador TCON.....	43
Figura 2.12: Registrador TMOD.....	43
Figura 2.13: Funcionamento dos Timers nos modos 1 e 0.....	46
Figura 2.14: Funcionamento dos times no modo 2.....	47
Figura 2.15: Timer 0 no modo 2 e alterações no timer 1.....	48
Figura 2.16: Registrador SCON.....	49
Figura 2.17: Registrador PCON.....	49
Figura 4.1: Esquema do Sistema de Processamento.....	61
Figura 4.2: Circuito lógico dos sinais /IOR e /IOWR.....	65
Figura 4.3: Circuito de seleção das portas de I/O mapeada em memória.....	69
Figura 4.4a: Fluxograma do Programa Teste.....	70
Figura 4.4b: Continuação.....	71
Figura 4.5: Fluxograma simplificado de um programa monitor.....	72

LISTA DE TABELAS

Tabela 2.1 – Descrição das funções dos pinos do 8031.....	28
Tabela 2.2 – Funções especiais dos pinos da porta três.....	29
Tabela 2.3 – Seleção do banco de registradores.....	34
Tabela 2.4 – Como o reset afeta os registros de função especial.....	39
Tabela 2.5 - Modo de Funcionamento do Canal Serial.....	50
Tabela 2.7 – Taxa e recarga com clock de 11,059 MHz.....	52
Tabela 4.1 – Lógica de decodificação e controle do sistema	63
Tabela 4.2 – Endereços RAM 1, RAM2 e I/O mapeada	66
Tabela 4.3 – Lógica de leitura e escrita na RAM.....	66
Tabela 4.4 - Pinos de Entrada e Saída da PAL.....	67
Tabela 4.5 - Descrição dos pinos do barramento de expansão.....	68

1. Introdução

Com o avanço da tecnologia e a utilização da eletrônica digital por grande parte das empresas, o emprego de microcontroladores vem sendo muito requisitado para um melhor desenvolvimento da produção, diminuindo os custos e trazendo benefícios para as empresas que utilizam esse sistema. É importante salientar que, considerando a relação custo/benefício, os microcontroladores podem não só ser usados em empresas de médio/grande porte, como podem também ser utilizados em vários projetos de eletrônica, na substituição de vários componentes digitais, obtendo-se assim no final do projeto um melhor acabamento. Assim um microcontrolador ocuparia um menor espaço físico e teria uma maior eficiência e praticidade, uma vez que todos os comandos seriam executados via software.

Antes de falar sobre o uso dos microcontroladores é bom conhecer um pouco mais como surgiu esta importante máquina de processar dados que cabe na palma da mão.

Na década de 70 os microprocessadores começaram a ser utilizados em microcomputadores para que estes pudessem processar dados de modo mais eficiente. A Intel foi uma das precursoras a lançar estes poderosos processadores. Foram evoluindo as tecnologias e então começou a surgir a necessidade de melhorar cada vez mais o processamento de dados destes componentes. Com isto houve a preocupação em se fazer um componente que além da arquitetura do processador, também tivesse os seus periféricos encapsulados em um único chip. Assim surgiram os microcontroladores, e também o 8031 e 8051 da Intel.

O uso deste microcontrolador pode ser atribuído não apenas pela facilidade de desenvolvimento do projeto de um sistema de processamento, mas também o preço de um projeto baseado no microcontrolador 8031 é relativamente baixo. Então, pode ser usado para vários projetos de controle, principalmente quando há necessidade de uma arquitetura específica, onde o uso de controle dado por um computador ou mesmo de um projeto que usa um processador de alta tecnologia torna-se desnecessário e inviável em relação ao seu custo.

Este microcontrolador pode ser usado em indústrias que exigem uma automação e controles dos seus processos, por exemplo, pode ser projetado um CLP simples de baixíssimo custo.

Hoje em dia tem se pensado muito em relação com o que realmente é preciso para que se possa implantar determinada tecnologia em uma máquina ou em um sistema autômato, para se fazer uma indústria automatizar sem desperdiçar dinheiro. Pode-se perceber em alguns casos que existem máquinas que usam processadores super avançados e as mesmas não precisam de tais tecnologias para ser eficientes, sendo que com um processador menos sofisticado ou mesmo com o microcontrolador 8031 poderia solucionar o problema. É claro que a tecnologia em relação à quantidade de coisas que um “super processador” faz não se discute, mas tem -se que pensar o que realmente uma máquina ou um sistema microprocessado está precisando. Se o uso de um microcontrolador como o 8031 já satisfaz a necessidade, então se tem uma redução no custo do projeto.

Com isto tem-se que pensar também que uma máquina microprocessada pode se tornar ultrapassada de acordo com a sua velocidade. Então neste aspecto deve-se ter cuidado para que possa fazer melhorias assim que necessário.

Claro que o uso de um microcontrolador pode se expandir o quanto se deseja, ou seja, pode-se projetar uma infinidade de coisas com um microcontrolador. Assim sendo a quantidade destas “coisas” se dá pela capacidade que o indivíduo tem de inventar a cada dia que passa, um aplicativo novo, um controle específico, a automação de um processo qualquer, entre outros.

2. Referencial Teórico

2.1. Microprocessadores

No livro que é adotado como livro texto para o desenvolvimento deste projeto, a definição de Silva Junior (1990), microprocessador é um componente eletrônico, que devido as suas avançadas técnicas de fabricação, o mesmo consegue efetuar com eficiência e rapidez, várias funções e operações lógicas e aritméticas, sob controle de um “programa” externo, que dita para a máquina a “seqüência” em que as funções e os operandos serão utilizados (executados).

Na definição de Lima & Rosa (2001) microprocessador é um chip responsável pelo processamento em um microcomputador. Este é um elemento complexo, contendo, entre outras coisas, uma unidade lógica e aritmética (ULA) e diversos registros (registradores) especiais.

Segundo Marinho & Marinho (2001), o processador ou unidade central de processamento (CPU) é a parte do sistema que faz o processamento das informações para que as instruções sejam executadas; as instruções devem estar armazenadas na memória de programa em seqüência, formando assim o programa. A CPU possui um registrador chamado de contador de programa (PC) que contém o endereço da próxima instrução que deve ser executada.

Toda vez que uma instrução é retirada da memória pela unidade central de processamento, automaticamente o contador de programa é incrementado para que, após o processamento desta instrução, quando a CPU for buscar a próxima instrução, basta usar o endereço contido no contador de programa.

Toda vez que a CPU é ligada ou resetada, automaticamente o seu contador de programa é zerado, desta forma, a primeira tarefa que a CPU irá realizar é a execução da instrução contida na posição de memória de endereço “0000”. Cada instrução possui duas fases distintas: o ciclo de busca e o ciclo de

execução. Durante o ciclo de uma instrução a CPU faz com que o conteúdo do contador de programa seja colocado no barramento de endereços, endereçando, desta maneira, a posição de memória que contém a instrução que deve ser executada.

Segundo Garcia et al. (2002), processador é um dos componentes que mais determina a performance de um sistema. O primeiro processador usado em um PC (Personal Computer) foi o 8086 que tem 16 Bits e foi fabricado pela Intel no final dos anos 70. Os computadores baseados neste processador eram chamados de PC-XT. Antes do 8086 podia-se encontrar os seguintes processadores de 8 bits, tais como: 8080, 8085, Z80, 6502, 6800.

Um microprocessador é um circuito integrado construído numa fina peça de silício (wafer). Contém milhares, ou mesmo milhões, de transistores, que estão interligados por superfinos traços de alumínio. Os transistores trabalham em conjunto armazenando e manipulando dados de forma a que o microprocessador possa executar uma vasta variedade de funções úteis. A evolução tem sido um fator constante no mundo dos microprocessadores. Pode ser constatado este fato, verificando que o primeiro microprocessador da Intel, o 4004, introduzido em 1971 continha 2300 transistores, já o processador Pentium® II lançado em meados da década de 90 também da Intel, contém 7,5 milhões de transistores.

Uma das aplicações mais comuns dos microprocessadores é em computadores pessoais, embora sua aplicação não esteja restrita aos PC's. Eles também são utilizados para dar "inteligência" a aparelhos que são usados no dia a dia. Por exemplo, utilizando microprocessadores é possível adicionar funcionalidades aos aparelhos domésticos, como, realizar uma rediscagem automática em um telefone, desligar automaticamente um termostato de aquecimento, aumentar a segurança dos automóveis bem como a sua eficiência, etc.

2.1.1. Sistema Seqüencial Síncrono

Antes de descrever sobre sistemas microprocessados é necessário que se entenda o significado de um sistema seqüencial síncrono. Por Silva Junior (1994), sistema seqüencial síncrono é o sistema que tem um compasso, ou seja, clock (marcador de tempo). A cada pulso deste clock ou grupo de pulsos (pode ser de 3 em 3 pulsos, por exemplo) executam uma função específica.

Um exemplo mais conhecido é o relógio digital, que possui um oscilador interno que gera uma seqüência conhecida e constante. E esta freqüência é dividida até gerar um sinal de um segundo de ciclo, e este sinal incrementa o contador dos segundos. Então um circuito lógico verifica se o contador de segundos já se encontra em 60, e se isso acontecer ele irá para o novo valor 0, e assim prossegue verificando os minutos e as horas.

2.1.2. Sistemas Microprocessados

Segundo Lima & Rosa (2001), sistemas microprocessados são aqueles que têm por elemento central um microprocessador. O microprocessador funciona como um sistema seqüencial síncrono. Entre os microprocessadores mais conhecidos pode-se citar o 80386, 80486, Pentium (I, II, III e IV) e Athlon.

Embora já existam microprocessadores que trabalhem a centenas de MHz, o 8031 (microcontrolador utilizado neste projeto) possui tipicamente um clock de 12 MHz, com tempos de execução de cada instrução variando entre 1 μ s e 4 μ s.

Um sistema básico como o apresentado na Figura 2.1 possui não apenas o microprocessador, mas também alguns outros dispositivos como cita Lima & Rosa (2001):

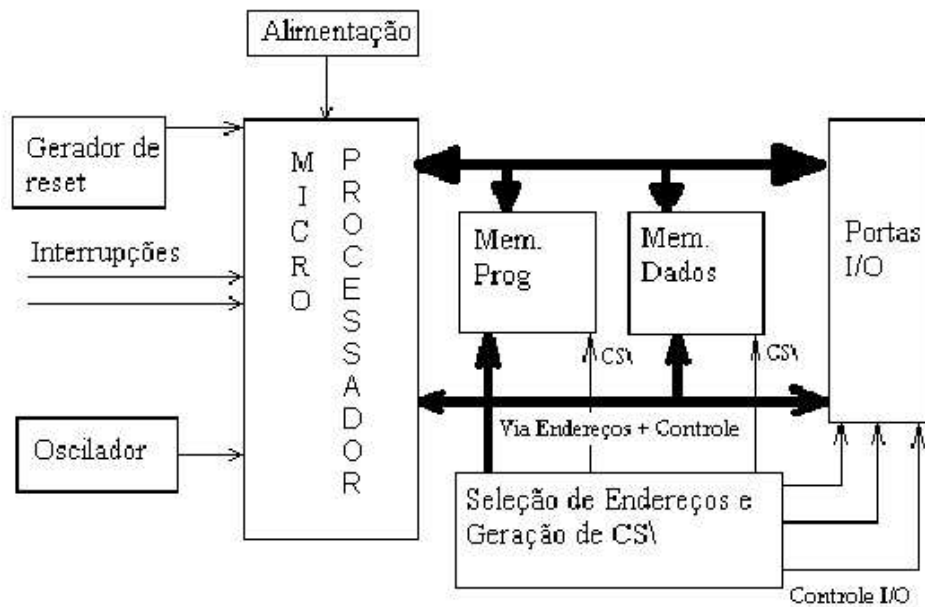


Figura 2.1- Sistema Genérico com Microprocessador

- *Interrupções*: são entradas que a partir de um sinal externo promovem a interrupção do processamento e seja iniciada uma sub-rotina específica.
(Obs.: o 8031 tem interrupções com estrutura *nesting*, onde uma interrupção pode interromper outra que está sendo atendida, desde que tenha maior prioridade).

- *Gerador de Reset*: responsável por iniciar o sistema ao ligar ou quando acionado forçadamente.
- *Gerador de Clock*: gera os pulsos necessários ao sincronismo do sistema.
- *Memória de Programa*: memória onde o microprocessador procura as instruções a executar. Em sistemas dedicados costuma-se utilizar memórias ROM, embora em alguns casos memórias RAM também sejam utilizadas.
- *Memória de Dados*: memória onde o microprocessador lê e escreve dados durante a operação normal. Geralmente é do tipo volátil, embora memórias não-voláteis possam ser utilizadas.
- *Seleção de Endereços*: lógica para escolher qual memória ou periférico o microprocessador vai utilizar.
- *Portas de I/O (Input/Output) ou simplesmente portas E/A (Entrada/Saída)*: sua função é a comunicação com o mundo externo. Através delas, os dispositivos como teclados, impressoras, displays, entre outros, comunicam-se com o sistema.

2.1.3. Principais Componentes de um Sistema Microprocessado

As memórias, os temporizadores e a Comunicação serial são componentes de grande importância para um sistema de processamento, portanto, os itens a seguir possuem suas descrições.

2.1.3.1. Memórias Somente de Leitura - Não voláteis

Estas memórias são do tipo ROM, e são divididas nos seguintes tipos segundo Silva Junior. (1994):

- ROM (Read Only Memory) – São gravadas no processo de fabricação. Tem custo final baixo, mas isto quando se precisa de um grande volume.
- PROM (Programmable Read Only Memory) – Estas são as ROM's programáveis, ou seja, vem sem nada gravado, e pode ser gravada apenas uma vez pelo usuário. Tem custo superior as ROM's, mas é ideal para médios lotes.
- EPROM (Eraseble Programmable Read Only Memory)– Estas são as PROM's apagáveis, ou seja, as mesmas depois de gravadas podem ser apagadas quando exposta à luz ultravioleta, são de fácil aquisição e ideais para protótipos.

2.1.3.2. Memórias de Leitura e Escrita - Voláteis

Segundo Silva Junior (1994) esta é a memória RAM, que também pode ser estática ou dinâmica. As *memórias estáticas* mantêm os dados enquanto forem alimentadas, é de rápido acesso e também são mais caras que as dinâmicas. As *memórias dinâmicas* têm custo bem inferior, sendo que estas são as mais utilizadas nos PC's, mas também perdem seus dados. O problema é que esta precisa de um sinal de *refresh*, o qual repassa a memória periodicamente, senão a mesma perde seus dados. Para que isso ocorra corretamente existe um circuito que é dedicado para este fim.

2.1.3.3. Memórias de Leitura e Escrita – Não voláteis

Este é um tipo especial de memória chamada de EEPROM ou E2PROM. Esta é uma memória que pode ser programada eletricamente (assim como as outras), mas também pode ser apagada eletricamente não necessitando assim da exposição à luz ultravioleta.

2.1.3.4. Temporizadores

Os temporizadores são dispositivos capazes de contar eventos, gerar pulsos periodicamente, medir a largura de pulsos, entre outros. Estes são facilmente programados e interfaceados pela CPU.

2.1.3.5. Comunicação Serial

Para Idoeta & Capuano (1984), comunicação serial é um canal onde a comunicação com o processador pode ser feita de modo seqüencial, um bit é enviado um após o outro. Assim sendo codificado por algum dispositivo ou periférico de saída, ou pelo próprio processador.

2.2. Microcontroladores

Neste item será abordado o cérebro deste projeto que é o microcontrolador, em especial o 8031. Serão referenciados principalmente os microcontroladores 8051 e 8031, porque a única diferença entre eles é que o 8051 possui ROM interna.

O microcontrolador para Lima & Rosa (2001), hoje em dia, é um elemento indispensável para o engenheiro eletricista, eletrônico ou ainda para o técnico de nível médio da área, em função de sua versatilidade e da enorme

aplicação. Entre algumas das aplicações de um microcontrolador pode-se citar a automação industrial, telefones celulares, auto-rádios, fornos de microondas e videocassetes. Além disso, a tendência da eletrônica digital é de se resumir a microcontroladores e a chips que concentram grandes circuitos lógicos, como os PLDs (Programmable Logic Devices). Para sistemas dedicados, o microcontrolador apresenta-se como a solução mais acessível, em função do baixo custo e facilidade de uso.

Para Silva Junior (1990) e (1994), um microcontrolador é um componente eletrônico que já tem incorporado em seu invólucro vários blocos como o microprocessador, as RAM's e ROM's, os temporizadores, os contadores, o canal de comunicação serial e as portas de I/O. E pode ainda conter elementos para uso industrial, tais como conversores A/D e D/A, PLL, PWM, etc.

Através do uso de microcontroladores consegue-se atingir uma compactação da construção dos sistemas de controle.

Além do tamanho reduzido deve-se destacar a facilidade de programação, pois seus periféricos são vistos pela CPU interna como memória, já que esses periféricos trazem um buffer de entrada.

Por Silva & Braile (2000), basicamente o microcontrolador (anteriormente chamado de microcomputador de um só chip) é um componente que possui todos os componentes dos microprocessadores comuns embutidos em uma só pastilha, facilitando assim o desenvolvimento de sistemas, complexos e sofisticados.

A Figura 2.2 apresenta um simples diagrama esquemático em blocos de um microcontrolador.

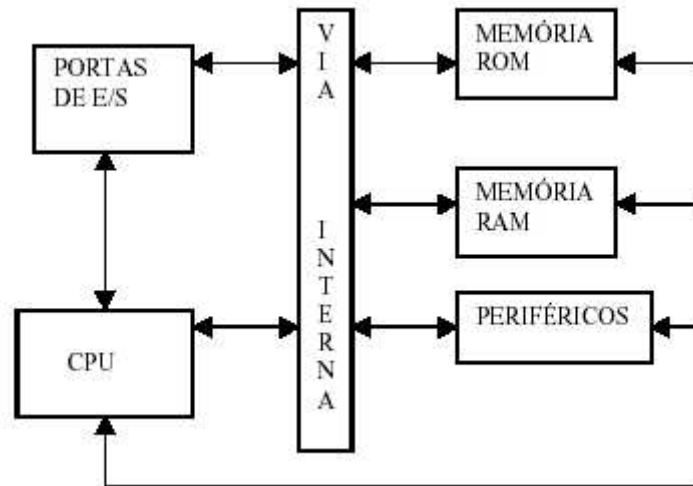


Figura 2.2 – Diagrama em Blocos Simplificado de um Microcontrolador

A maioria dos microprocessadores comuns, e assim como vários microcontroladores existentes no mercado têm sua estrutura interna de memória de dados e programa baseados na arquitetura de Von Neumann, que prevê uma única via de comunicação entre memórias e CPU. Portanto, com uma única via a CPU pode se comunicar com seus periféricos, porém um de cada vez.

Alguns Microcontroladores utilizam uma arquitetura conhecida como Harvard, que prevê várias vias de comunicação entre CPU e periféricos, permitindo-se assim a realização de várias operações ao mesmo tempo, implicando em um aumento considerável na velocidade de execução do software.

Esta arquitetura trouxe algumas vantagens, entre elas a terminologia chamada RISC (Reduced Instruction Set Computer – Computador com Conjunto de Instruções Reduzido) onde é possível programar microcontroladores com apenas 35 instruções (em alguns casos), contra mais de 100 instruções da arquitetura anterior.

De acordo com Zelenovsky & Mendonça (2002), o termo controlador é usado para designar o dispositivo que controla um processo ou algum parâmetro do ambiente. O controlador de temperatura do condicionador de ar, por exemplo, liga ou desliga o compressor em função da temperatura ambiente. Antigamente, os controladores usavam lógica discreta e, por isso, tinham um tamanho que dificultava seu emprego em sistemas pequenos. Hoje em dia, usam-se os circuitos integrados microprocessados e todo o controlador cabe em uma pequena placa de circuito impresso.

O microcontrolador, com o avanço da microeletrônica, recebe dentro do seu CI uma quantidade de recursos cada vez maior. Isso leva à primeira definição: microcontrolador é um CI com alta densidade de integração que inclui, dentro do chip, a maioria dos componentes necessários para o controlador. É por isso que usa o apelido: "solução com único chip". Existe uma quantidade expressiva de microcontroladores, porém os mais conhecidos são: 8051, 8096, 68HC705, 68HC11 e os Pics.

A Figura 2.3 apresenta a organização detalhada da estrutura interna do 8051.

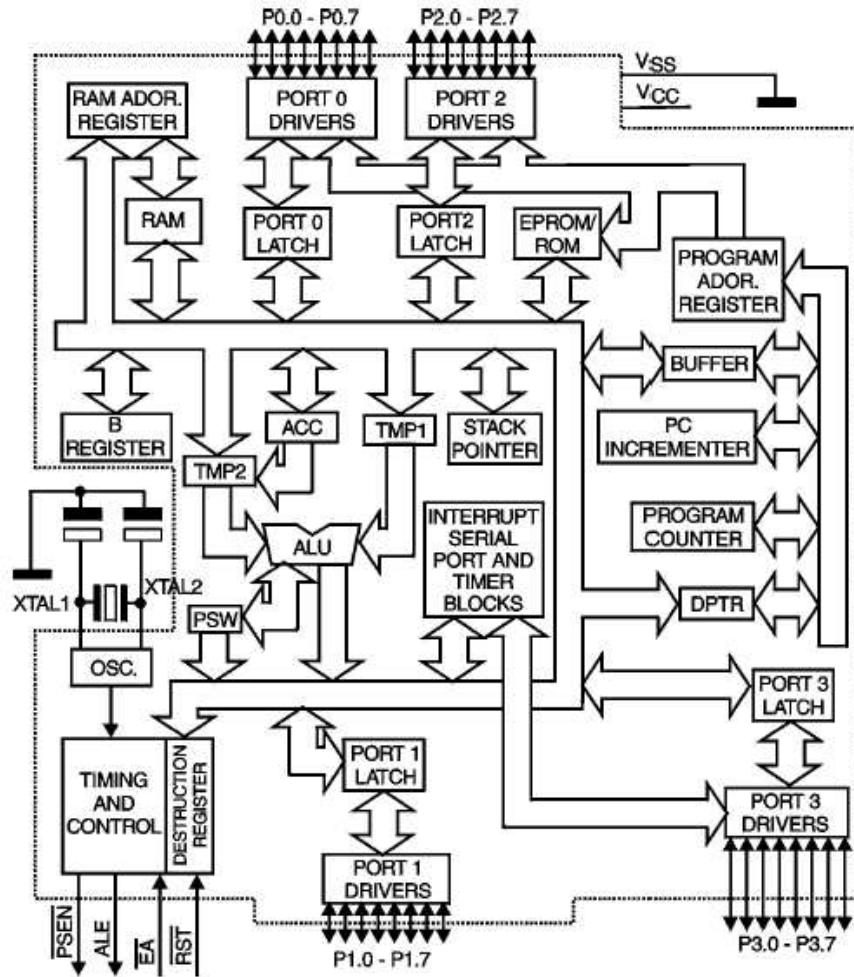


Figura 2.3 - Organização da Estrutura Interna do 8051

2.2.1. O Microcontrolador 8031

Este é o microcontrolador alvo de estudo, no qual este projeto é baseado. Lembrando novamente que o 8031 é igual ao 8051 só que não possui ROM interna (chamado também por *ROM less* na literatura inglesa). A disposição dos pinos também são iguais e quando qualquer um dos dois microcontroladores forem trabalhar com a memória de programa externa, o pino /EA tem que ser aterrado assumindo nível lógico 0, e para memória de programa interna (somente para o 8051) o pino /EA assume nível lógico 1. Quando isto é feito o 8051 lê os primeiros 4 KBytes internos e o que exceder esta memória será procurado externamente. Portanto daqui a diante se estiver sendo citado o 8051 em vez do 8031, no contexto não fará diferença uma vez que se trabalha apenas com memória de programa externa.

2.2.1.1. Porquê o 8031 (8051)

Segundo Casare (2001), existem no mercado muitos tipos de microcontroladores, sendo o 8051 o mais popular. O microcontrolador reúne num único componente vários elementos de um sistema, antes baseado em microprocessador e que eram desempenhados por vários componentes independentes tais como RAM, ROM, comunicação serial, etc. A memória de programa pode ser ROM, FLASH ou outro tipo. A ATMEL possui uma enorme família de componentes com as mesmas características do 8051, alguns até com as mesmas pinagens dos registradores; outros com pinagens diferentes, mas com o mesmo set de instruções, com clock de 4 à 10 MHz aproximadamente.

A DALLAS Semiconductor tem um microcontrolador de alta performance, de até 90 MHz, compatível com 8051.

Já por Lima & Rosa (2001), o 8051 da Intel, é sem dúvida, o microcontrolador mais popular atualmente. O dispositivo em si é um microcontrolador de 8 bits relativamente simples, mas com ampla aplicação. Porém, o mais importante é que não existe somente o CI 8051, mas sim uma família de microcontroladores baseada no mesmo. Entende-se *família* como sendo um conjunto de dispositivos que compartilha os mesmos elementos básicos, tendo também um mesmo conjunto básico de instruções.

2.2.1.2. Principais Características de Hardware

O microcontrolador 8031 (8051) apresenta dois modos básicos de funcionamento:

- a) *Modo mínimo* - modo onde somente recursos internos são utilizados pela CPU. Neste modo, estão disponíveis 4 KBytes de ROM (somente para o 8051) para memória de programa e 128 bytes de RAM para memória de dados. Além da economia de componentes e espaço físico, o modo mínimo apresenta a vantagem de poder utilizar as quatro portas de oito bits cada para controle (I/O);
- b) *Modo expandido* – modo onde a memória de programa (ROM), a memória de dados (RAM) ou ambas podem ser expandidas para 64 KBytes, através do uso de CI's externos. No entanto, apresenta a desvantagem de "perder" duas das quatro portas para comunicação com as memórias externas.

Este projeto é desenvolvido no modo expandido.

As características deste microcontrolador:

- Clock de 12 MHz;
- 64 KBytes de memória de dados e mais 64 KBytes de memória de programa, isoladamente;
- Capacidade de expansão da memória de programa, (quatro KBytes interna para o 8051) e 60 KBytes externa ou 64 KBytes externa para o 8031.
- Capacidade de expansão de memória RAM;
- Possui 4 portas de I/O de 8 bits sendo endereçáveis individualmente;
- Interrupção com estrutura ‘nesting’*;
- Dois temporizadores/contadores de 16 bits;
- Oscilador de clock interno;
- Canal de comunicação serial tipo UART full-duplex.

* ‘nesting’ é o nome que se dá ao processo pelo qual uma interrupção pode interromper outra que já estiver sendo atendida, desde que tenha maior prioridade.

2.2.1.3. Pinagem do 8031 (8051)

A Figura 2.4 apresenta o diagrama de pinagem, a Tabela 2.1 possui a descrição das funções dos pinos e a Tabela 2.2 a descrição das funções especiais dos pinos da porta três.

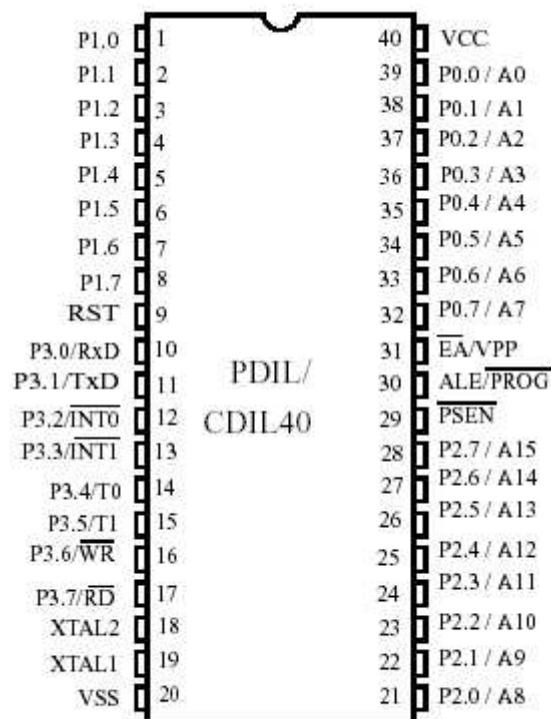


Figura 2.4 – Pinagem do 8031/51 em Encapsulamento DIL

Tabela 2.1 - Descrição das funções dos pinos do 8031

Pino	Nome	Função
1 a 8	P1.0 a P1.7	Porta bidirecional de 8 bits, que podem ter seus pinos individualmente endereçáveis;
9	RST/VPD	Reseta o sistema pela aplicação de nível lógico 1 TTL, durante 2 ou mais ciclos de máquina;
10 a 17	P3.0 a P3.7	Porta de I/O bidirecional de 8 bits;
18	XTAL2	Saída do amplificador inversor do oscilador e entrada para o gerador de clock interno. Se for usado clock externo, serve como entrada para o mesmo.
19	XTAL1	Entrada do amplificador inversor do oscilador deve ser conectado a terra se usarmos clock externo;
20	Vss	Conexão de terra;
21 a 28	P2.0 a P2.7	Porta bidirecional de 8 bits.
29	/PSEN	Saída para a habilitação do programa externo. É o strobe da memória de programa externa.
30	ALE	Saída habilitadora do latch de endereços;
31	/EA	Entrada de seleção de memória;
32 a 39	P0.7 a P0.0	Porta de 8 bits bidirecional com dreno aberto.
40	Vcc	+5V durante a operação normal.

Tabela 2.2 - Funções especiais dos pinos da porta três

Porta	Função Especial	Descrição
P3.0	RXD/data	Receptor da porta serial assíncrona ou entrada e saída de dados síncronos (expansão de I/O pela porta serial).
P3.1	TXD/clock	Saída de transmissão da porta serial assíncrona, ou saída de clock para os registradores de deslocamento externos (expansão de I/O pela porta serial).
P3.2	/INT0	Interrupção externa numero 0, ou bit de controle para o timer/counter 0.
P3.3	/INT1	Interrupção externa numero 1, ou bit de controle para o timer/counter 1.
P3.4	T0	Entrada externa para o timer/contador de eventos 0.
P3.5	T1	Entrada externa para o timer/contador de eventos 1.
P3.6	/WR	Strobe (sinalizador) de escrita na memória de dados externa.
P3.7	/RD	Strobe (sinalizador) de leitura na memória de dados externa.

2.2.1.4. Memória no 8031 e 8051

Organização da memória no 8051 - Silva Junior (1994).

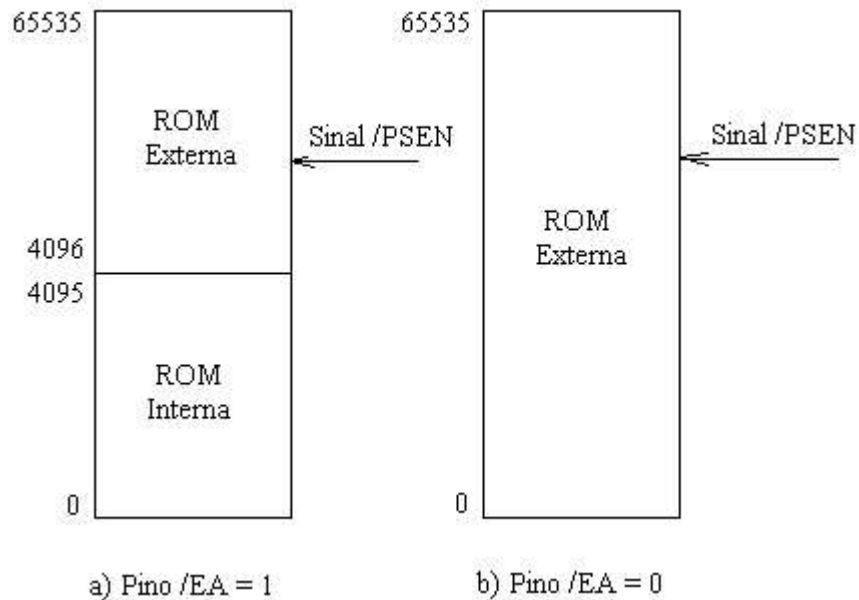


Figura 2.5 – Memória de Programa

Na Figura 2.5-a percebe-se que o 8051 no seu modo mínimo trabalha com programas gravados nos 4 KBytes de memória de programa interna (iniciando em 0000H).

No 8051 tem-se então duas possibilidades de utilização de memória de programa externa, uma possibilidade é utilizar 64 KBytes de memória de programa externa, e a outra possibilidade é de utilizar os 4 KBytes que são gravados internamente no 8051 com os 60 KBytes restantes, sendo que os 4

KBytes internos são invioláveis. Para a versão 8031 deve-se utilizar somente as memórias de programa externas, e aterrar o pino /EA.

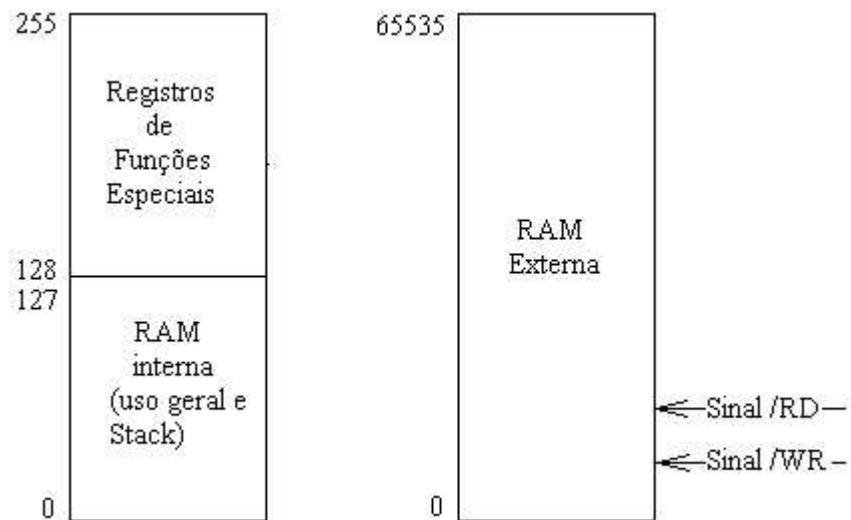


Figura 2.6 – Memória de Dados

Já para a expansão externa da memória de dados, como apresentado na Figura 2.6, pode-se colocar mais 64 KBytes de RAM, independentemente da RAM interna. Como salienta Junior, apud Lima & Rosa (2001), diferentemente de outros sistemas baseados em microprocessador, onde cada endereço de memória identifica uma única posição física, no 8051 o mesmo endereço hexadecimal pode identificar 3 posições físicas diferentes (e até 4, no caso do 8052, que tem 256 bytes de RAM interna). Por exemplo, o endereço 23H na RAM interna, o endereço 23H na RAM externa e o endereço 23H na ROM externa. Mesmo com esses endereços "iguais", não há risco de conflito, pois as instruções e o modo de endereçamento são diferentes. O conteúdo para estas três

memórias com o mesmo endereço é carregado no acumulador de acordo com os exemplos:

a) *da RAM interna*

```
MOV A,23H ;end.direto
```

b) *da RAM externa*

```
MOVX R0,#23H ;imediato
```

```
;R0 como ponteiro
```

```
MOVX A,@R0 ;indireto
```

c) *da ROM externa*

```
CLR A
```

```
MOV DPTR,#0023H
```

```
MOVC A,@A+DPTR
```

Os registros de funções especiais incluem locações referentes às portas de I/O, ponteiros do sistema, registros de interrupção, registros da porta serial, dos temporizadores e registros aritméticos.

Os registros de funções especiais são tratados pela CPU da mesma forma que a RAM é tratada, podendo efetuar alterações, e o mais importante, endereçar algumas dessas posições como bit.

Na RAM interna, sua parte baixa está dividida em 4 bancos de registradores de 8 Bytes cada, 16 Bytes cujos bits são individualmente endereçáveis pela CPU e o resto da memória que é endereçado somente Byte-a-Byte, como apresentado na Figura 2.7.

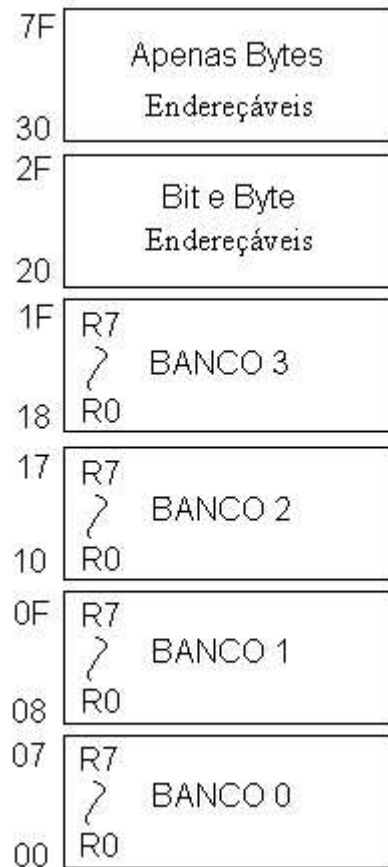


Figura 2.7 – Diagrama em blocos da parte baixa da RAM interna

Os bancos de registradores são simples posições de memória RAM que permitem seus endereçamentos pelos nomes dados aos registros, além de seus endereçamentos pela posição de memória.

A seleção do banco de registradores é dada pelo valor dos bits RS0 e RS1, conforme mostra a tabela 2.3 onde R0 a R7 é o nome do registrador.

Tabela 2.3 – Seleção do banco de registradores

RS1	RS0	Banco	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	00	01	02	03	04	05	06	07
0	1	1	08	09	0A	0B	0C	0D	0E	0F
1	0	2	10	11	12	13	14	15	16	17
1	1	3	18	19	1A	1B	1C	1D	1E	1F

Alguns registros de funções especiais também têm seus bits endereçáveis, sendo que alguns deles possuem um nome mnemônico, para maior facilidade de desenvolvimento de software em compiladores.

Os Registros de funções especiais e seus endereços em hexadecimal na RAM por Silva Junior (1990) e (1994):

P0(80), P1(90), P2(A0) e P3(B0) – São posições da RAM que contém os dados das quatro portas de I/O do microcontrolador, caso as mesmas sejam usadas como tal.

Uma escrita em um desses registros altera automaticamente o conteúdo presente na saída do chip, e uma leitura dos mesmos coloca o estado presente nos pinos dentro da posição desejada. Essa operação é possível devido aos buffers e latches de cada porta, que só são ativos na leitura ou escrita das mesmas.

SP(81) – É o stack pointer, que indica o ultimo endereço do armazenagem na pilha.

TH1(8D), TLI(8B), TH0(8C) e TL0(8A) – São registros de dados dos temporizadores/contadores (1 e 0 respectivamente). Contém os valores contados, no caso de ser usado as funções desse importante periférico.

TCON(88) e TMOD(89) – Registros de controle e de modo de operação dos temporizadores/contadores. É nestes registros que é efetuado a programação de como se quer que estes periféricos trabalhem.

PCON(87) – *Power control* permite adaptar o chip para situações em que não há processamento, mas não é desejado que se perca o conteúdo das memórias internas do chip, como pode ocorrer em uma falha de alimentação.

SCON(98) e SBUF(99) – No SCON, coloca-se os bits que se deseja trabalhar com a porta de comunicação serial conforme desejado, e o SBUF é o registro no qual a porta armazenará o dado recebido, ou escrito o dado que se deseja transmitir.

IE(A8) e IP(B8) – Estes registros permitem que seja escolhido qual ou quais interrupções serão habilitadas (ou desabilitadas) e qual a prioridade (maior ou menor) de cada uma.

PSW(D0) – O *program status word* contém bits que indicam quais as ocorrências da ALU na última operação lógica e aritmética, além de indicar qual banco de registradores foi acessado pela última vez.

ACC(E0) e B(F0) – Equivalem ao acumulador e ao registro B dos microprocessadores de 8 bits. O interessante aqui é que inclusive os registros da CPU são acessíveis como simples posições de memória.

DPH(83) e DPL(82) – Estes dois registros de 8 bits podem ser referenciados como um registro de 16 bits, chamado DPTR. Este registro é usado quando se deseja fazer endereçamento indireto para a leitura de constantes armazenadas na memória de programa, para ler ou escrever variáveis na memória externa de dados e para o desvio para memória de programa de até 64 KBytes.

A Figura 2.8 mostra um sistema simples baseado no microcontrolador 8051, com RAM e ROM externas. Quando é retirada a RAM neste caso, o sistema é chamado de sistema mínimo. Pode-se reparar ainda que se tem um latch que faz uma multiplexação dos dados e da metade menos significativa dos endereços. No primeiro ciclo de máquina o 8051 coloca nos pinos da porta 0 o Byte menos significativo de endereço (A0-A7) e leva o pino ALE (Habilitador do Latch de Endereço) a nível lógico 1, de modo que o latch coloque em suas saídas esses bits e logo em seguida leva o pino ALE ao nível 0, para que estes bits fiquem travados na saída do latch. Assim ficarão fixos porque a saída está habilitada pelo pino /OE (Habilita a saída quando em 0 e HiZ quando em 1). Neste instante, o 8051 coloca nos pinos da porta 0 o Byte de dados.

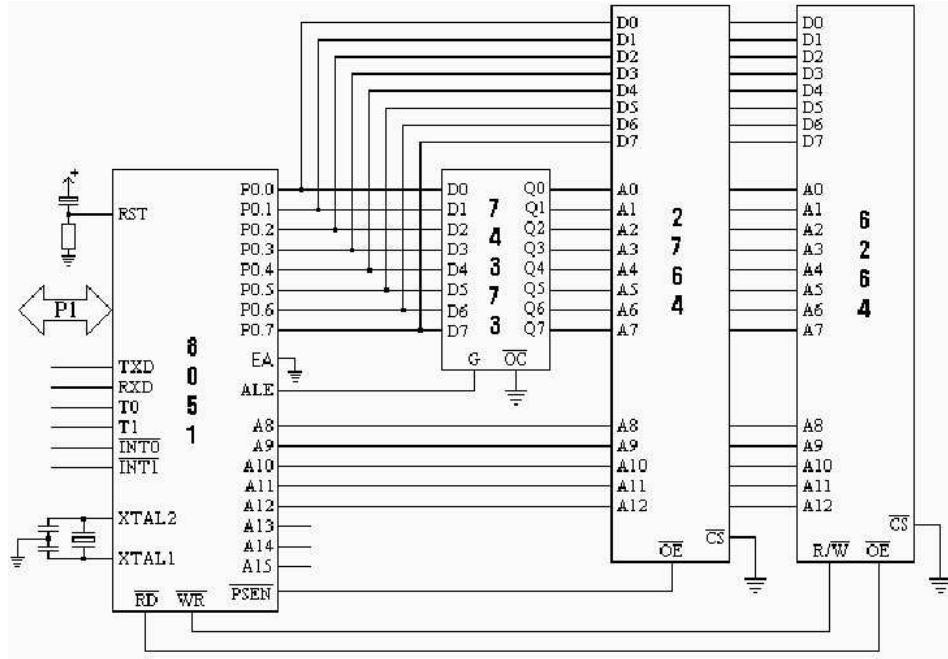


Figura 2.8 – Expandindo RAM e ROM

2.2.1.5. Reset no 8031 (8051)

Na Figura 2.9 tem-se a implementação do reset onde se pode também provocar um reset forçado dado que o esquema possui uma chave de contato momentâneo NA (push-botton NA) em paralelo com o capacitor, e ao pressioná-la e soltá-la se tem uma nova operação de reset.

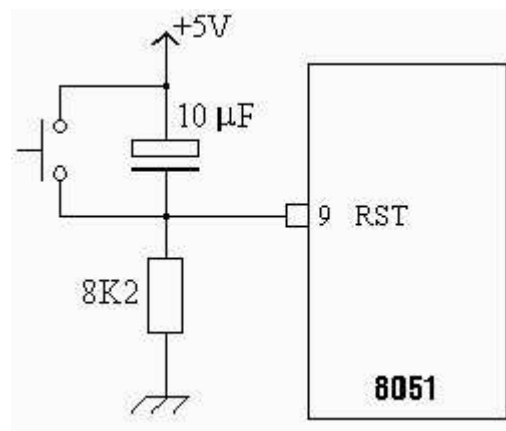


Figura 2.9 – Reset no 8031/51

Segundo Lima & Rosa (2001), o reset é conseguido mantendo-se nível alto no pino 9 (RST) por pelo menos dois ciclos de máquina (24 pulsos de clock).

Para a versão CMOS (80C51) o resistor se torna desnecessário. Se existir não interfere em nada.

A Tabela 2.4 mostra como o RESET afeta os registros de função especial.

Tabela 2.4 – Como o reset afeta os registros de função especial

REGISTRO	VALOR	REGISTRO	VALOR
PC	0000 H	TMOD	00 H
ACC	00 H	TCON	00 H
B	00 H	TH0	00 H
PSW	00 H	TL0	00 H
SP	07 H	TH1	00 H
DPTR	0000 H	TL1	00 H
P0 a P3	FF H ou	SCON	00 H
	11111111 B	SBUF	Indeterminado
IP	0XX00000 B	PCON(NMOS)	0XXXXXXXX B
IE	00 H	PCON	0XXX0000 B

A RAM não é afetada por RESET forçado. Em caso de RESET por inicialização o seu valor será aleatório.

Seqüência de funções de inicialização:

Inicialização do sistema básico.

- Sempre ocorre no Reset

Inicialização do sistema montado.

- Prepara o equipamento para a operação para a o qual foi projetado, por exemplo, ajustando os contadores, as portas, entre outros.

2.2.1.6. Clock no 8031 (8051)

Por Lima & Rosa (2001)

Frequência de clock mínima = 3,5 MHz

Frequência de clock máxima = 33 MHz (algumas versões) e 12 MHz (mais comum)

Na Figura 2.10 é mostrada a implementação do clock tanto externo quanto interno do microcontrolador 8031.

Clock externo – utiliza um sinal externo que possa ser aproveitado de outro sistema. Para a implementação deste, basta aterrar o pino 19 (XTAL1) e injetar o sinal no pino 18 (XTAL2), que desta forma irá diretamente para o sistema de controle e temporização do chip.

Clock interno – Se utilizar o clock gerado pelo oscilador interno, deve-se intercalar aos pinos XTAL1 e XTAL2 um cristal ou um filtro cerâmico (o cristal é mais indicado) na frequência desejada, e dois capacitores de realimentação.

Capacitores:

- 40 pF + - 10 pF para filtro ressonante de cerâmica
- 30 pF + - 10 pF para cristal oscilador

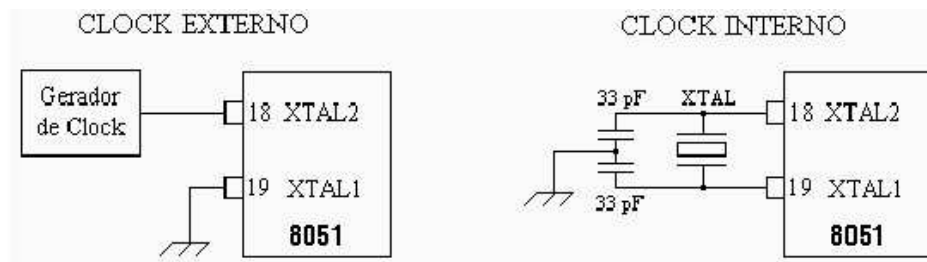


Figura 2.10 – Clock para o 8031/51

2.2.1.7. Interrupção para o 8031 (8051)

Por Silva Junior a interrupção é o processo pelo qual a execução de um programa é interrompida (ou não), de acordo com as necessidades de eventos externos ou internos.

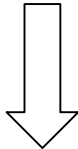
O 8051 pode ser interrompido de cinco maneiras:

- pela interrupção externa 0 (INT0 \ - pino P3.2);
- pela interrupção externa 1 (INT1 \ - pino P3.3);
- pelo contador/temporizador 0;
- pelo contador/temporizador 1;
- pelo canal de comunicação serial.

Observação: O canal serial e as interrupções externas usam alguns pinos da porta três, portanto se for usado algumas dessas interrupções, deve-se tratar a porta três como porta de I/O apenas para bit endereçável.

O 8051 possui um processo de prioridade interno, para o caso de duas interrupções de mesma prioridade estarem aguardando atendimento.

Ordem de atendimento das interrupções:

- | | |
|-------------------------|------------------|
| • Interrupção externa 0 | Maior prioridade |
| • Timer 0 | |
| • Interrupção externa 1 | |
| • Timer 1 | |
| • Canal serial | Menor Prioridade |
- 

Neste microcontrolador, cada interrupção é desviada para um certo endereço fixo, no qual começa a rotina de interrupção.

Os endereços de desvio são:

- INT 0\ 03H
- TIMER/COUNTER 0 0BH
- INT 1\ 13H
- TIMER/COUNTER 1 1BH
- SERIAL 23H

Sabe-se que o Reset inicia no endereço 00H, e desta forma, por exemplo, se for usada todas as interrupções, tem-se que usar os três bytes disponíveis para o reset, num desvio incondicional para outra parte da memória, onde realmente teria início o programa.

O 8051 possui dois registros de controle de 8 bits cada um, na sua memória RAM mais alta (Registros de funções especiais- SFR) que servem para a habilitação ou não das interrupções, e indica para a CPU a prioridade de cada uma delas. Para entender o funcionamento dos registros de controle das interrupções pode-se utilizar o Anexo A (página 18).

2.2.1.8. Temporizadores e Contadores no 8031 (8051)

O 8051 é dotado de dois temporizadores/contadores internos programáveis, deste ponto em diante chamados simplesmente de *timers*. São controlados por software e também por sinais externos aplicados ao microcontrolador. Os timers são geralmente utilizados para a geração de eventos depois de decorrido um determinado intervalo de tempo (*timeouts*), na geração de pedidos de interrupção periódicos, ou ainda na contagem e na medição da largura de pulsos externos. Os registradores associados à programação dos timers, TCON e TMOD, aparecem nas Figuras 2.11 e 2.12.

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Figura 2.11– Registrador TCON

7	6	5	4	3	2	1	0
Gate1	C/T1	M1.1	M0.1	Gate0	C/T0	M1.0	M0.0

Figura 2.12- Registrador TMOD

Os bits TF0 e TF1 (5 e 7) de TCON são flags associados aos Timers 0 e 1, respectivamente. Cada vez que ocorre um overflow (passagem de FFH para 0) na contagem de um timer, o bit correspondente é setado, o que gera um pedido de interrupção (que será atendido se a interrupção correspondente estiver habilitada). TR0 e TR1 (bits 4 e 6) permitem habilitar ou desabilitar (parar) a

contagem, respectivamente para Timer 0 e Timer 1. Os bits de 0 a 3 deste registrador dizem respeito às interrupções.

O registrador TMOD (Figura 2.12) define o modo de funcionamento dos timers, discutido logo adiante.

Habilitação da contagem

Diz-se que um timer está *habilitado* quando estão satisfeitas todas as condições para que sua contagem progrida com o passar do tempo. A habilitação dos timers depende dos valores atribuídos aos bits C/T.x e GATE.x, bem como dos sinais externos aplicados aos pinos T.x e INT.x (entenda-se, deste ponto em diante, 'x' como um nome genérico para 0 ou 1).

As condições de habilitação de um timer são as seguintes:

- O bit TR.x deve estar em 1;
- Se o bit GATE.x estiver em 0, o timer estará habilitado;
- Caso contrário (bit GATE.x = 1), a habilitação pode ser controlada pelo usuário através do pino externo INT.x, da seguinte forma:
 - INT.x = 0: contador desabilitado;
 - INT.x = 1: contador habilitado.

Satisfeitas estas condições, a contagem avançará de acordo com a configuração do timer:

- Temporizador (bit C/T.x = 0): a contagem é incrementada a cada 12 ciclos do oscilador interno, o que significa que o contador avança a cada ciclo de instrução do microcontrolador;
- Contador (bit C/T.x = 1): neste modo, a contagem não avança de acordo com o oscilador interno, mas sim a cada transição descendente de um

sinal externo, colocado no pino T.x (não confundir com o pino INT.x, utilizado na habilitação).

Aplicações

A possibilidade de se habilitar um timer externamente através do pino INT.x permite medir a largura de pulsos externos. Para tanto, conecta-se o sinal cuja largura se deseja medir ao pino INT.x, com o timer correspondente configurado como temporizador ($C/T.x = 0$) e com o bit $GATE.x = 1$. Desta forma, a contagem avançará enquanto o sinal externo estiver em nível alto, obtendo-se uma contagem proporcional à duração do pulso. A largura do pulso pode então ser calculada a partir do período do sinal de clock.

É possível também contar pulsos externos. Para tanto, conecta-se o sinal a medir ao pino T.x, com o contador configurado como contador ($C/T.x = 1$) e com o bit $GATE.x = 0$. Desta forma, a contagem avançará a cada pulso do sinal externo.

Os dois mecanismos acima podem ainda ser combinados para contar pulsos de um sinal externo somente durante um intervalo de tempo determinado por outro sinal externo. Para tanto, configura-se o timer como contador (bit $C/T.x = 1$) com o bit $GATE.x = 1$. Desta forma, a contagem avançará de acordo com os pulsos recebidos no pino T.x, mas somente enquanto o nível no pino INT.x estiver em 1.

Modos de operação

Cada timer pode operar em um de quatro modos de operação diferentes. O modo de operação é definido pelos valores dos bits $M0.x$ e $M1.x$ do registrador TMOD.

Modo 0 ($M1.x = 0$ e $M0.x = 0$)

Neste modo, os registradores TLx e THx são vistos como um contador de 13 bits, formado por THx e pelos 5 bits menos significativos de TLx. Os 3 bits mais significativos de TLx são indeterminados e devem ser ignorados. O registrador THx pode ser visto como um contador de 8 bits, cuja frequência de contagem é igual a 1/32 da frequência de clock. Este modo existe por razões de compatibilidade com a família MCS48, mais antiga.

Modo 1 ($M1.x = 0$ e $M0.x = 1$)

O funcionamento deste modo é semelhante ao do modo 0. A única diferença é que o par de registradores TLx/THx é visto como um contador de 16 bits.

A Figura 2.13 representa o funcionamento dos timers nos modos 0 e 1. Note a representação do bit C/T.x, em forma de uma chave que seleciona a fonte de pulsos de clock para o timer e também a saída “overflow”, que gera o pedido de interrupção associado. Essa figura mostra ainda o mecanismo de habilitação da contagem, descrito no início desta seção.

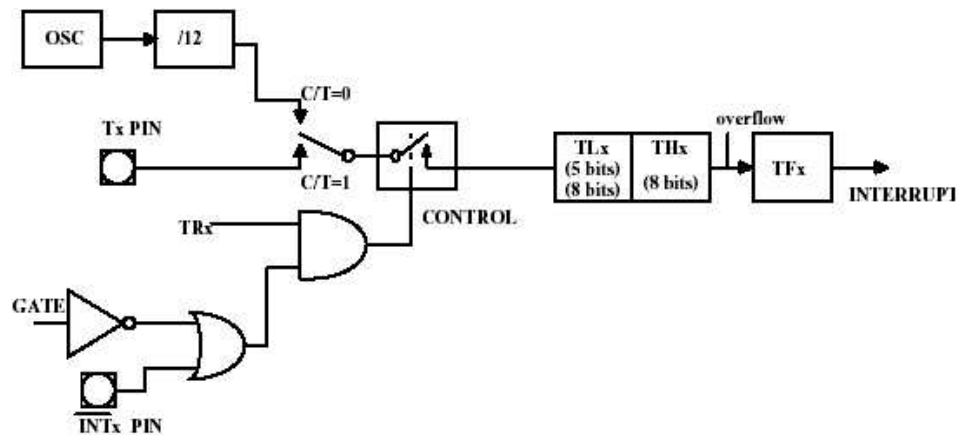


Figura 2.13 – Funcionamento dos Timers nos modos 1 e 0.

Modo 2 ($M1.x = 1$ e $M0.x = 0$)

Neste modo, o registrador TLx funciona como um contador recarregável de 8 bits, conforme a Figura 2.14. O registrador THx contém o valor que é recarregado automaticamente em TLx sempre que a contagem deste sofre um overflow (passagem de FFH para 00). O overflow faz também com que o flag TFx seja setado e portanto gera um pedido de interrupção, que será atendido de acordo com as regras de habilitação das interrupções. O valor de THx não sofre qualquer alteração e TLx retoma a contagem a partir do valor carregado.

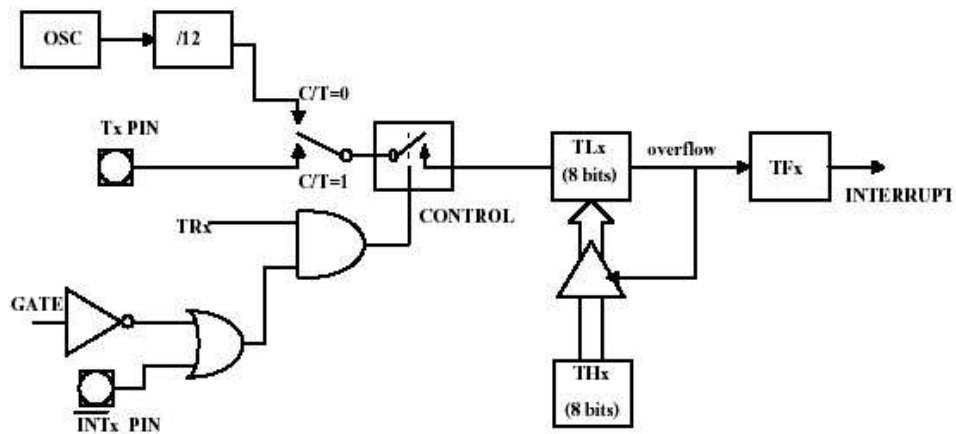


Figura 2.14 – Funcionamento dos times no modo 2

Modo 3 ($M1.x = 1$ e $M0.x = 1$)

TL0 e TH0 funcionam como dois contadores independentes de 8 bits, conforme a Figura 2.15. Note que TR1 e TF1 são desviados do Timer 1 e utilizados para controlar TH0. O par de registradores TL1TH1 continua funcionando como um contador de 16 bits. Contudo, não se pode desabilitar sua contagem através de TR1, e TF1 não reage quando ocorre overflow da contagem em TL1TH1. Para desabilitar a contagem de TL1TH1 quando TL0TH0 estiver no modo 3, deve-se colocar TL1TH1 também no modo 3. Então, se o Timer 0

estiver no modo 3 e o Timer 1 for colocado no modo 3, o par TL1TH1 pára de contar e preserva seu valor até que outro modo de funcionamento seja escolhido para o Timer 1.

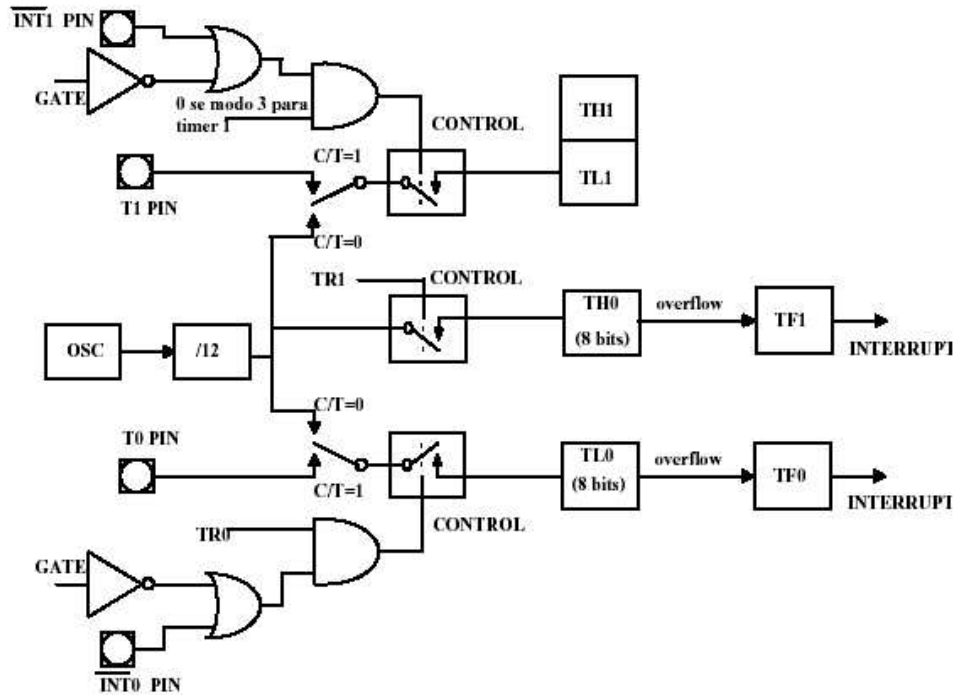


Figura 2.15 – Timer 0 no modo 2 e alterações no timer 1

2.2.1.9. Canal Serial para o 8031 (8051)

Segundo Silva Junior (1994), a interface serial no 8051 conta com dois registradores de dados, um deles utilizado na transmissão e outro na recepção. O conjunto de instruções, contudo, referencia ambos pelo nome SBUF. A distinção entre eles é feita de acordo com a natureza da operação, escrita ou leitura. Desta forma, escrever em SBUF implica no envio do byte escrito através da interface serial; analogamente, a leitura desse registrador retorna o último byte recebido.

O controle do canal serial é feito pelo registrador SCON, apresentado na Figura 2.16, e pelo bit SMOD do registrador PCON, que aparece na Figura 2.17.

Os bits SM0 e SM1 do registrador SCON selecionam o modo de funcionamento, de acordo com a Tabela 2.5.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	T1	R1

Figura 2.16 – Registrador SCON

7	6	5	4	3	2	1	0
SMOD	GF1	GF0	PD	IDL

Figura 2.17 – Registrador PCON

Tabela 2.5 – Modo de Funcionamento do Canal Serial

SM0	SM1	Modo	Tipo	Bits	Taxa de TX
0	0	0	Síncrona, HD	8	Fclock/12
0	1	1	Assíncrona, FD	10	Variável
1	0	2	Assíncrona, FD		Fclock/32 ou /64
1	1	3	Assíncrona, FD		Variável

Para o melhor entendimento dos registros SCON e PCON, pode-se ler o anexo A (da página 12 à 17).

A geração das taxas de transmissão por Silva Junior (1990) e (1994):

- a) Para o modo 0 a taxa é fixa e vale 1/12 da frequência de clock.

- b) Para o modo 2, a taxa de transmissão depende apenas do valor do bit SMOD no PCON:
 Se SMOD = 0, taxa = 1/64 da frequência de clock (SMOD=0 no reset);
 Se SMOD = 1, taxa = 1/32 da frequência de clock.

- c) Para os modos 1 e 3, a taxa de transmissão é fornecida pelo T/C1, onde cada ordem de transmissão é gerada ao decorrer o overflow deste contador. Neste caso, deve-se desabilitar a interrupção do T/C1.

Tem-se, nesse caso, uma fórmula para o cálculo desta taxa:

$$\text{Taxa} = \frac{2^{\text{SMOD}}}{32} \times (\text{taxa de overflow do T/C1})$$

O T/C1 pode ser configurado para temporizador ou contador, em qualquer um dos três modos de operação.

Caso especial: T/C1 no modo 2.

O normal é usa-lo no modo de recarga automática (timer de 8 bits) e nesse caso, a seguinte fórmula é usada:

$$\text{Taxa} = \frac{2^{\text{SMOD}}}{32} \times \frac{(\text{Frequência de clock})}{12 \times (256 - \text{TH1})}$$

Outro modo de representar a mesma fórmula:

$$\text{Recarga} = 256 - \frac{\text{clock}(\text{MHz} \times 10^6)}{384 \times \text{taxa}(\text{BPS})}$$

Onde Recarga em decimal e com SMOD=0. Se SMOD=1 tem-se o dobro da Taxa.

Na Tabela 2.7 se tem valores para uso prático na transmissão, considerando um valor de clock de 11,059 MHz, Timer 1 no modo 2 e SMOD=0.

Tabela 2.7 - Taxa e recarga com clock de 11,059 MHz

TAXA (BPS)	RECARGA (Decimal)
300	160
1200	232
2400	244
4800	250
9600	253

2.3. O software (Linguagem Assembly)

A programação do microcontrolador 8031 é feita em linguagem assembly. A primeira razão para se trabalhar com o assembly é a oportunidade de conhecer melhor o funcionamento do microcontrolador, que permite o desenvolvimento de programas de forma mais consistente. Uma outra razão é que programas assembly são mais rápidos e mais compactos do que os criados com outras linguagens.

A linguagem assembly foi a primeira linguagem que surgiu nesta área de computação, sendo formada por mnemônicos. Cada mnemônico possui o um código correspondente em hexadecimal, e cada ‘família’ de microcontrolador possui um conjunto de mnemônicos diferente.

A linguagem assembly tem como finalidade facilitar o manuseio das instruções para desenvolver programas, desde uma simples rotina até um sofisticado programa para controle industrial. Todo microcontrolador precisa de uma lista de instruções para ser seguida passo-a-passo, lista essa chamada de programa, que diz exatamente o que o microcontrolador deve fazer. O microcontrolador só faz duas coisas: lê e processa números binários.

Cada fabricante de microcontrolador define os mnemônicos para os seus produtos.

Algumas linguagens assembly utilizam o mesmo código (instrução de máquina) para diversas CPU's, porém antes de se iniciar a tradução deve-se informar ao tradutor para qual CPU deve ser feita a tradução. A linguagem assembly é classificada de baixo nível, porque na elaboração do programa há a necessidade de elaborar rotinas para o completo controle da CPU.

2.4. Aplicações Interessantes com o 8031

a) Tranca Eletrônica - Camargo (2001)

Projeto de controle para abertura de trancas eletrônicas de acesso. É utilizado um teclado de dezesseis teclas padrão telefônico, onde a tranca abre ao se digitar a senha correta, de seis dígitos. Esta senha pode ser alterada facilmente através do próprio teclado, mas isso só pode ser feito se o usuário conhecer também uma outra senha, chamada de senha “master”, além do procedimento para realizar a operação. Isso quer dizer que se pode fornecer a senha de abertura para qualquer pessoa desejada, mas se em algum momento se deseja restringir o acesso de alguns, é só trocar a senha, o que só uma pessoa autorizada (que conhece a senha master) pode fazer. Todas as operações realizadas podem ser visualizadas em um display LCD (Display de cristal líquido), facilitando as tarefas.

A principal vantagem desse projeto é a possibilidade de se trocar a senha sem a necessidade de se desmontar o circuito (quando há essa possibilidade), o que é necessário a outros equipamentos não microcontrolados. Além de difícil, essa tarefa pode danificar o sistema.

b) Cadeira de Rodas - Camargo (2001)

Controle de uma cadeira de rodas elétrica, com 2 motores de corrente contínua acoplados nas rodas traseiras.

A parte de controle lógico da cadeira foi feita inteiramente com um microcontrolador. O programa lê de um teclado de 16 teclas os comandos de ir para frente, para trás, para um lado ou outro, e também a velocidade desejada pelo usuário. Então, ele manda os comandos para um circuito de potência que liga os motores da forma desejada. Para controlar a velocidade, o microcontrolador manda um sinal de PWM (Pulse Width Modulation) com precisão de 8 bits.

3. Metodologia

Para o desenvolvimento deste projeto, foi utilizada a metodologia descrita neste capítulo.

A pesquisa teórica foi realizada em livros e sites da internet sobre o assunto. O projeto do hardware e do software foi baseado na pesquisa teórica feita.

Com os estudos feitos em cima de cada componente utilizado, o esquemático foi a primeira coisa a ser feita. No seu desenvolvimento foi utilizada uma PAL para decodificação dos sinais de controle de barramento. O mapa de gravação da PAL é gerado pelo programa PALASM, segundo as equações lógicas editadas em arquivo ASCII. Logo depois de concluída a idéia sobre o sistema de processamento a ser desenvolvido, o leiaute foi projetado. No desenvolvimento do esquemático e do leiaute foram utilizados os softwares do IVEX, sendo o WinDraft (Figura 3.1) para o desenvolvimento do esquemático e o WinBoard (Figura 3.2) para o leiaute.

A placa foi confeccionada por uma empresa especializada por causa da sua complexidade, e a montagem feita manualmente.

O desenvolvimento do software de controle do 8031 foi feito em assembly e compilado no LSI51 e linkado no LSILK para criar o arquivo (.HEX), e depois é simulado no SIM31. Além do simulador SIM31 foi usado o *TS Controls Emulator 8051* para Windows (Figura 3.3). Depois do desenvolvimento do software, das correções necessárias e testes feitos, procedeu-se a gravação do software na EPROM.

Após a gravação da EPROM, foram feitos os testes de hardware usando um programa de teste para verificação do funcionamento deste sistema de processamento.

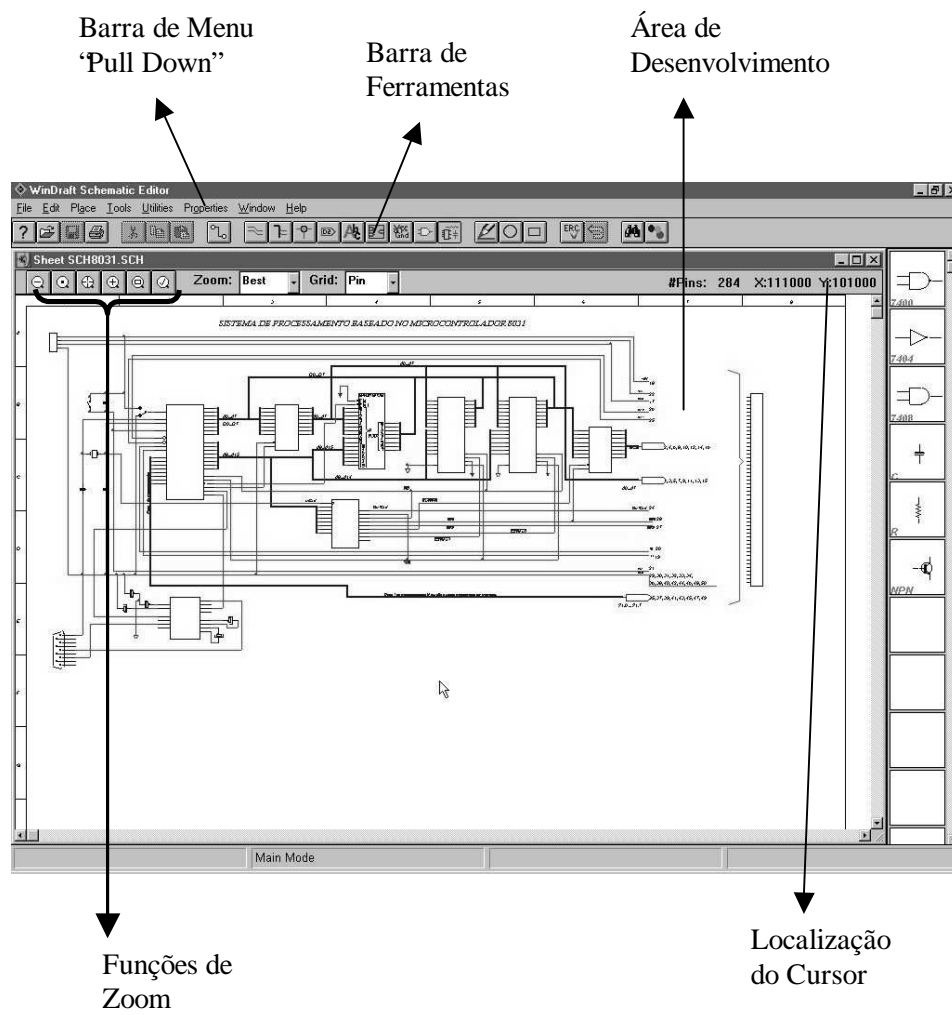


Figura 3.1 – Tela do WinDraft

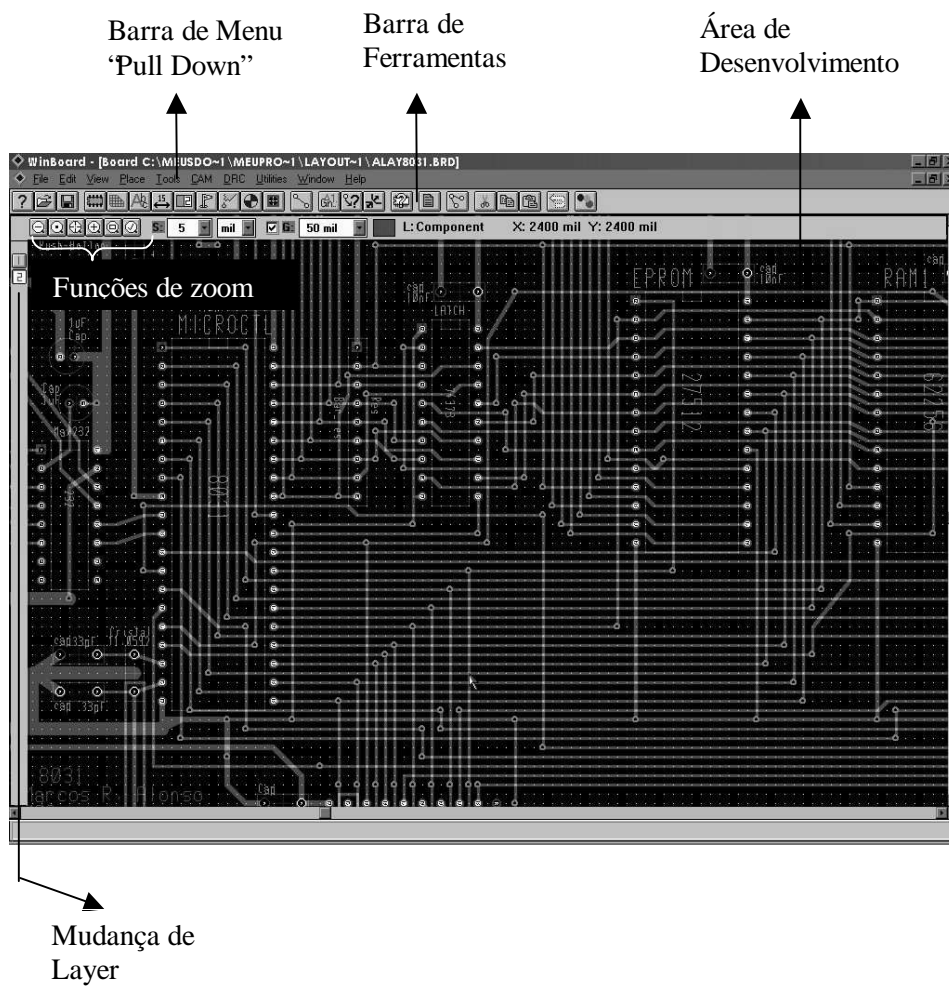


Figura 3.2 – Tela do WinBoard

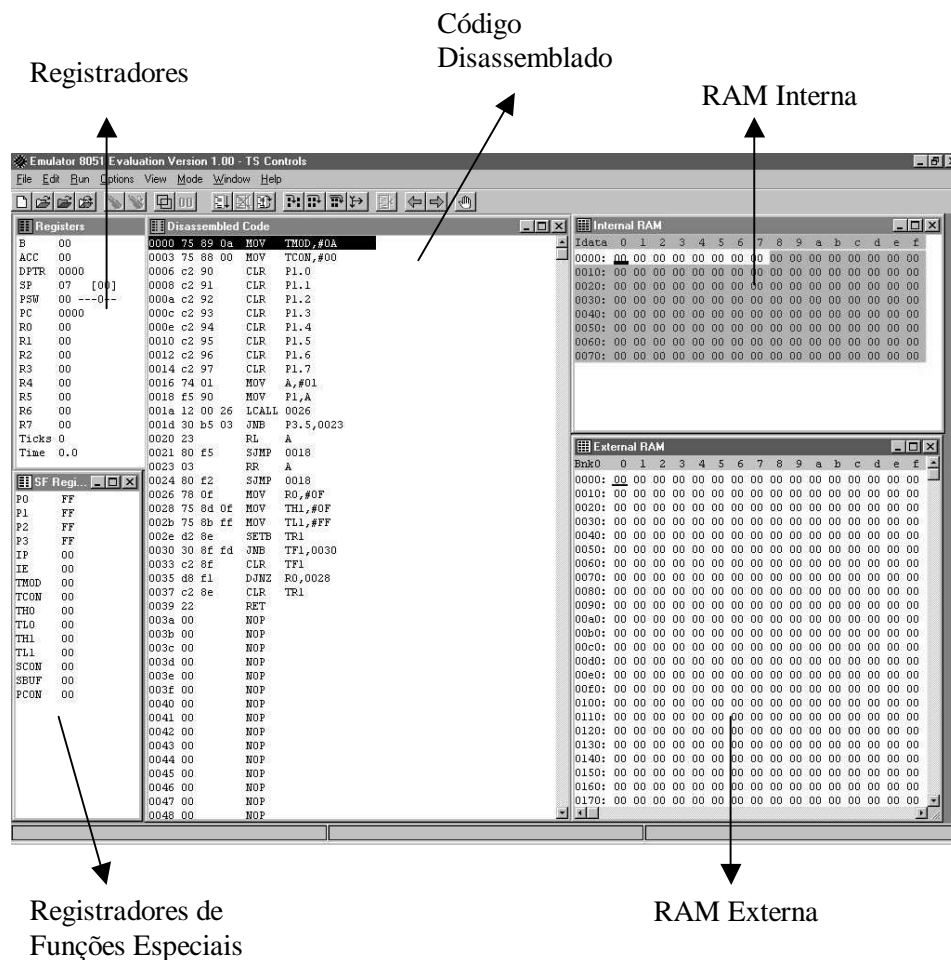
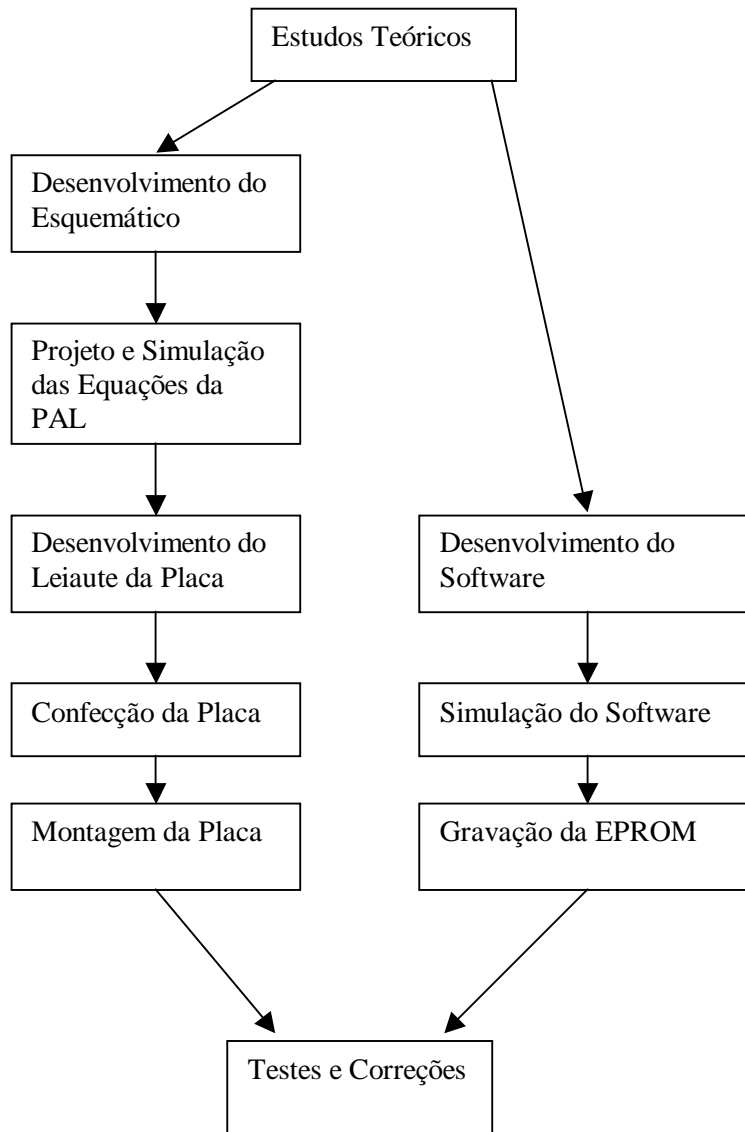


Figura 3.3 – Tela do TS Controls Emulators 8051

Fluxograma Sequencial



4. Desenvolvimento e Resultados

4.1. Hardware

Depois dos estudos feitos em cima de todo referencial teórico e dos “Databooks”¹ que seguem anexos, pode-se chegar a idéia final de uma arquitetura satisfatória para o objetivo do desenvolvimento deste sistema de processamento.

Assim sendo é descrito aqui toda parte de hardware desenvolvido e mostrado no esquemático da Figura 4.1(ou mais detalhado no anexo H).

O circuito para a execução do projeto terá como componentes principais:

- Microcontrolador TSC80C31-12CA
- PAL 16V8
- MAX 232
- Latch 74373
- 74245 – Transceptor de barramento octal
- 27512 – EPROM - 64KBytes
- 62256 – RAM – 32KBytes
- Cristal 11.059MHz

¹ “Databook” – É um manual técnico que contém as especificações e características técnicas dos componentes.

Neste ponto é explicada de modo sucinto a especificação de cada componente e seu uso no projeto. Maiores detalhes são encontrados nos databooks anexados.

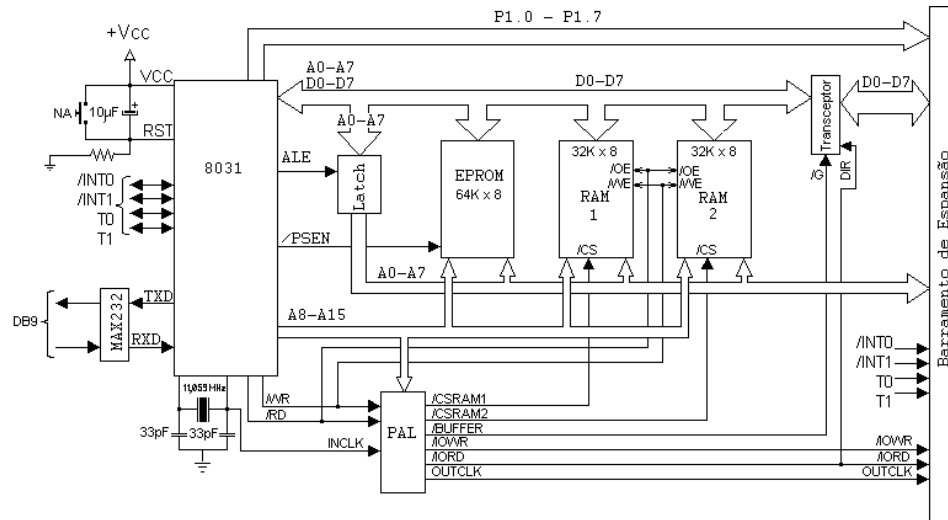


Figura 4.1 – Esquema do Sistema de Processamento

O CI MAX 232 (Anexo C) é um conversor RS232-TTL e TTL-RS232 reunidos em um único chip. Ele possui alimentação de +5 Volts e a conversão TTL-RS232 dá-se por um multiplicador de tensão construído por capacitores.

Este CI então substituirá o 1488 e o 1489 que são respectivamente conversores TTL-RS232 e RS232-TTL. Deste modo economiza-se espaço na placa e assim não é necessária a alimentação de +12 e -12 Volts para suprir o 1488.

Usando este CI pode-se ter uma conexão serial do 8031 (Anexo A) com um PC utilizando o conector DB-9 mostrado no esquemático. Então o mesmo é ligado em um canal serial do PC em questão, bastando apenas desenvolver o software adequado.

Na memória EPROM (Anexo F) será armazenado o programa monitor. A EPROM é acessada através de seu pino /OE pelo pino /PSEN do 8031. O pino /CS é aterrado já que é a única memória EPROM e ainda na capacidade máxima do sistema.

O microcontrolador utiliza a porta 0 e a porta 2 para endereçar memória externa, assim é expandida a capacidade de dados e de programas no 8031. Num primeiro instante todo o barramento de dados do microcontrolador servirá para endereçar a memória, porém os bits menos significativos (A7-A0) serão armazenados em um latch pela porta 0, onde este latch receberá um pulso de nível lógico 1 pelo pino ALE do 8031. Após isto os dados (D7-D0) trafegam pelo barramento.

Como no uso de memórias externas se perdem duas portas para fazer o endereçamento, então é usado de um artifício para que o problema de restrição de portas seja resolvido. É usado, portanto, o que se chama de I/O mapeada como memória. Assim utiliza-se uma lógica de habilitação adequada para endereçar não só a memória como também as portas de I/O. Quando se usa I/O mapeada como memória, é como se tivesse trabalhando com escrita e leitura da memória, mas na verdade estaria escrevendo ou lendo os dados de uma porta. Isto é possível, porque são reservados os últimos 256 bytes da área de memória RAM externa para uso como 512 portas de I/O de 8 bits cada.

Os sinais de leitura e escrita são exclusivos, por isso há no mesmo endereço uma porta de saída e uma porta de entrada. Portanto, 256 bytes reservados da memória equivalem a 512 portas, sendo 256 de entrada e 256 de saída.

Para o entendimento da lógica de mapeamento é usada a Tabela 4.1 por meio da qual se obtém as equações de controle desejável para o sistema.

Tabela 4.1 – Lógica de decodificação e controle do sistema

Entradas da PAL										Saídas da PAL						Função
A8	A9	A10	A11	A12	A13	A14	A15	/RD	/WR	/CSRAM1	/CSRAM2	/IORD	/IOWR	/BUFFER	DIR	
X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	X	Lê ROM
X	X	X	X	X	X	X	0	0	1	0	1	1	1	1	X	Lê RAM 1
X	X	X	X	X	X	X	0	1	0	0	1	1	1	1	X	Escreve na RAM 1
0	X	X	X	X	X	X	1	0	1	1	0	1	1	1	X	Lê RAM 2
X	0	X	X	X	X	X	1	0	1	1	0	1	1	1	X	
X	X	0	X	X	X	X	1	0	1	1	0	1	1	1	X	
X	X	X	0	X	X	X	1	0	1	1	0	1	1	1	X	
X	X	X	X	0	X	X	1	0	1	1	0	1	1	1	X	
X	X	X	X	X	0	X	1	0	1	1	0	1	1	1	X	
X	X	X	X	X	X	0	1	0	1	1	0	1	1	1	X	
0	X	X	X	X	X	X	1	1	0	1	0	1	1	1	X	Escreve na RAM 2
X	0	X	X	X	X	X	1	1	0	1	0	1	1	1	X	
X	X	0	X	X	X	X	1	1	0	1	0	1	1	1	X	
X	X	X	0	X	X	X	1	1	0	1	0	1	1	1	X	
X	X	X	X	0	X	X	1	1	0	1	0	1	1	1	X	
X	X	X	X	X	0	X	1	1	0	1	0	1	1	1	X	
X	X	X	X	X	X	0	1	1	0	1	0	1	1	1	X	
1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	Lê I/O mapeada
1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	Escreve na I/O mapeada

Note que na Tabela 4.1 se têm basicamente sete “modos” de endereçamento, sendo:

- Acesso à memória EPROM
- Acesso para escrita na memória RAM 1
- Acesso para leitura da memória RAM 1
- Acesso para escrita na memória RAM 2
- Acesso para leitura da memória RAM 2
- Acesso para escrita em I/O mapeada como memória
- Acesso para leitura em I/O mapeada como memória

Pode-se perceber que quando se utilizam as portas mapeadas neste projeto, as últimas 256 posições da memória é que são usadas como I/O (FF00H até FFFFH). Por isso todos os pinos da porta 2 (A15 à A8) terão que estar em nível lógico 1. Neste caso as memórias devem ser desabilitadas, ou seja /CS das RAM's têm que estar em nível lógico 1. Deve-se gerar dois sinais de controle (leitura e escrita) para a correta seleção das portas mapeadas, as quais são chamadas de /IORD e /IOWR. Estes sinais serão controlados pelos pinos /RD e /WR do microcontrolador. Na Figura 4.2 é apresentado o circuito lógico dos sinais /IORD e /IOWR.

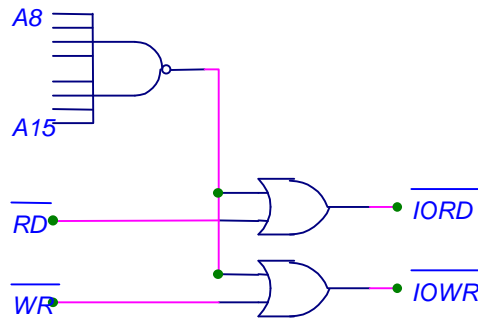


Figura 4.2 – Circuito lógico dos sinais \overline{IORD} e \overline{IOWR}

Para a escrita e leitura da Porta 1 não é preciso nenhum tipo de controle externo ao microcontrolador, já que a mesma não é utilizada para o endereçamento das memórias e nem para funções especiais.

Quando \overline{RD} e \overline{WR} estiverem ambos desabilitados (em nível lógico 1), o microcontrolador poderá acessar a EPROM (\overline{PSEN} em nível 1), e as memórias de dados estarão desabilitadas assim como as I/O mapeadas.

Para o controle de leitura e escrita nas memórias RAM's usa-se a Tabela 4.3. De acordo com a tabela é ligado o \overline{WR} e o \overline{RD} do 8031 no \overline{WE} e \overline{OE} das memórias RAM's respectivamente.

Pode-se perceber pela Tabela 4.1 que a memória RAM 1 é acessada somente quando o bit de endereçamento A15 estiver em nível 0 e for operação de escrita ou leitura. Isto acontece por que a memória RAM 1 vai até o endereço 7FFFH (como mostra a Tabela 4.2).

A RAM 2 é acessada somente quando o bit de endereçamento A15 for nível 1, pelo menos um bit de A8 à A14 for nível 0 e for operação de leitura ou escrita.

A saída /BUFFER mostrada na Tabela 4.1 habilita o “transceiver” 74245 (Anexo E) e é usado o pino /IORD para indicar o sentido do mesmo. Ou seja, se for leitura B->A e se for escrita A->B.

Tabela 4.2 – Endereços RAM 1, RAM2 e I/O mapeada

	Início	Fim
RAM 1	0000H	7FFFH
RAM 2	8000H	FEFFH
I/O Mapeada	FF00H	FFFFH

Tabela 4.3 – Lógica de leitura e escrita na RAM

/RD	/WR	/OE	/WE	
1	1	1	1	Espera
0	1	0	1	Lê
1	0	1	0	Escreve
0	0	0	0	Não Ocorre

A lógica do sistema é implementada em uma PAL 16V8 (Anexo B). Esta PAL contém a lógica do circuito.

Tabela 4.4 – Pinos de Entrada e Saída da PAL

PINOS	Descrição	PINOS	Descrição
1	INCLK	11	NC
2	A8	12	/CSRAM1
3	A9	13	/CSRAM2
4	A10	14	/BUFFER
5	A11	15	/IOWR
6	A12	16	/IORD
7	A13	17	/RD
8	A14	18	/WR
9	A15	18	OUTCLK
10	GND	20	VCC

Obs: De A15 à A8, /WR e /RD são entradas e os outros são os pinos de saída da PAL que fornecerão o controle do sistema. O clock está apenas sendo “bufeizado”².

O programa que gera as equações da PAL está no Anexo I. Como comentado na metodologia este programa é testado e simulado no software PALASM.

² Buferizado – Derivado de buffer, um circuito que tem por objetivo o armazenamento temporário de dados. Aqui está sendo usado como sinônimo de driver (amplificador de corrente).

O reset do sistema é implementado do mesmo modo que é mostrado na Figura 2.9. O clock é interno, ou seja, o sistema não aproveita nenhum sinal de periféricos externos, do mesmo modo que mostra a Figura 2.10. Porém o mesmo é enviado para a expansão passando pela PAL como buffer.

A Tabela 4.5 mostra a descrição de como são dispostas as ligações no barramento de expansão da placa (veja Anexo H).

Tabela 4.5 – Descrição dos pinos do barramento de expansão

PIN	Descrição	PIN	Descrição	PIN	Descrição	PIN	Descrição
1	+5V	11	A0	21	P1.0	31	/INT0
2	GND	12	A1	22	P1.1	32	/INT1
3	D7	13	A2	23	P1.2	33	TO
4	D6	14	A3	24	P1.3	34	T1
5	D5	15	A4	25	P1.4	35	RST
6	D4	16	A5	26	P1.5	36	GND
7	D3	17	A6	27	P1.6	37	GND
8	D2	18	A7	28	P1.7	38	-5V
9	D1	19	GND	29	/IOWR	39	OUTCLK
10	D0	20	GND	30	/IORD	40	GND

O barramento de expansão serve para enviar e receber dados assim como seus controles para um funcionamento adequado. Este barramento pode servir para comunicar-se com diversos periféricos e controlá-los. Para que seja usado o barramento de dados mapeados tem-se que construir um periférico que irá usar os controles de leitura e escrita em I/O mapeada além dos endereços de A7 à A0

que selecionará através de DMUX o chip select de uma latch a qual servirá como uma porta. Na Figura 4.3 é mostrada a implementação desta arquitetura.

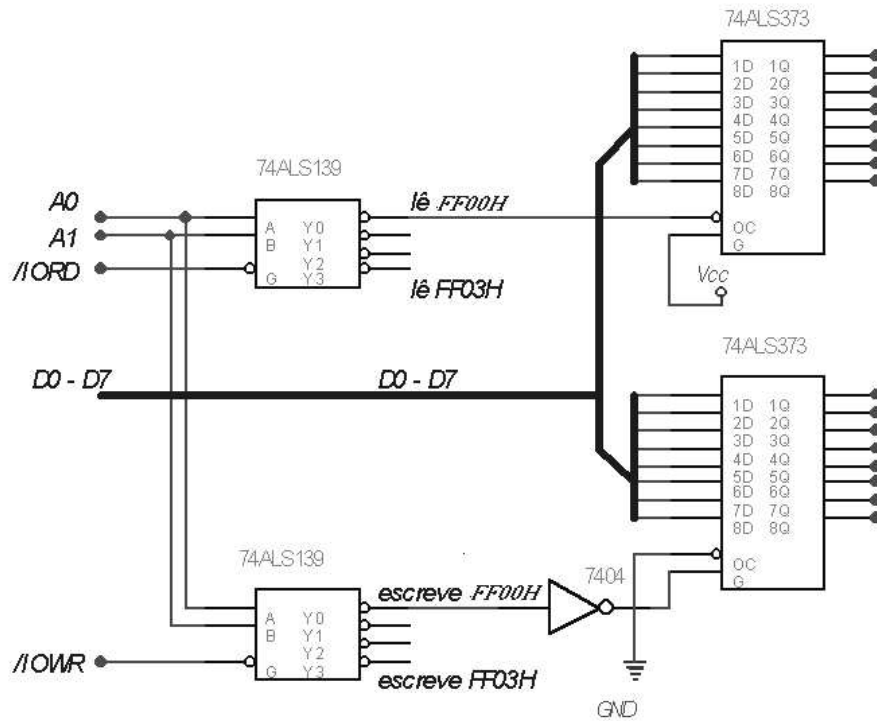


Figura 4.3 – Circuito de seleção das portas de I/O mapeada em memória

4.2. Software

Com relação ao software, foi elaborado um programa teste (Anexo K) para a verificação do funcionamento do sistema de processamento projetado. O programa faz gravações e leituras das memórias (RAM1 e RAM2), da porta P1 e das I/O's mapeadas. A Figura 4.4 abaixo mostra seu Fluxograma.

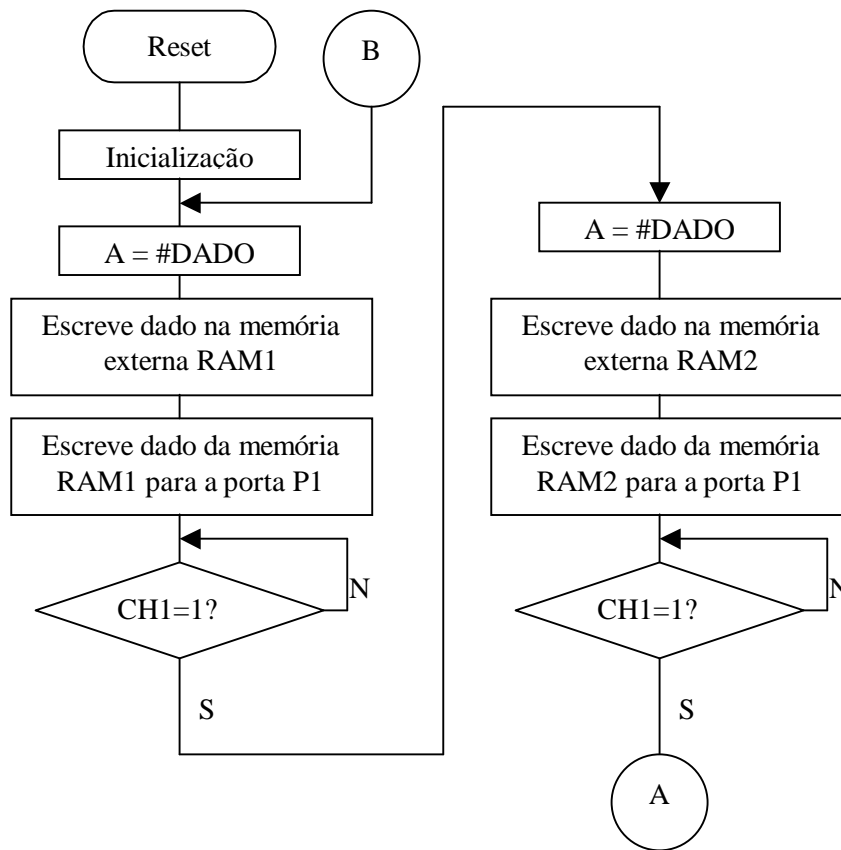


Figura 4.4a – Fluxograma do Programa Teste

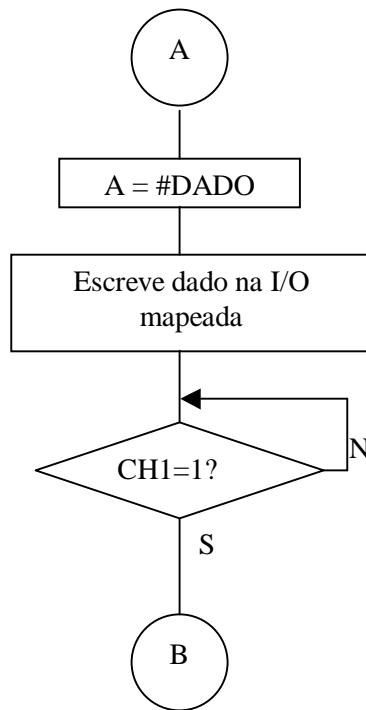


Figura 4.4b – Continuação

Este programa é um programa simples que serve apenas para a verificação do funcionamento do sistema. Como proposta de um programa monitor definitivo tem-se o seguinte:

O programa monitor ficará na memória de programa e não precisará mais fazer gravações em EPROM, isto porquê quando quiser inserir um programa (o programa de usuário), poderá enviá-lo através da interface serial (PC->Sistema de processamento) e o programa monitor se encarregará de armazená-lo e executá-lo na RAM.

O programa é mostrado no fluxograma simplificado da Figura 4.5.

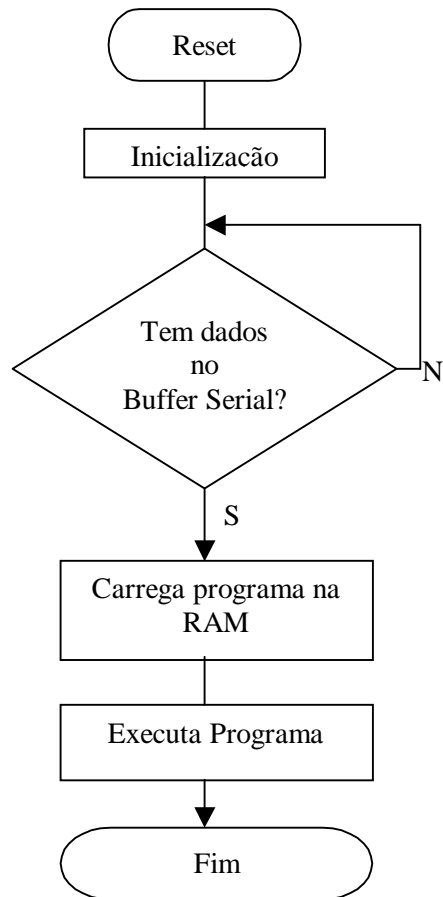


Figura 4.5 – Fluxograma simplificado de um programa monitor

Esta proposta é muito interessante quando não se deseja um módulo independente para um objetivo específico, caso seja necessário deve-se elaborar o software de acordo com o que se precisa.

5. Conclusão

Este trabalho é muito interessante para o aprendizado de um tipo de arquitetura muito utilizado para soluções de problemas dos mais variados em cima de controle de processos. Assim foi desenvolvido um sistema interessante sobre um microcontrolador conhecido, largamente utilizado, e muito barato em comparação com outros que existem no mercado.

Na elaboração deste sistema o objetivo da confecção do protótipo foi tido como a meta mais importante de todo o trabalho. Isto porque, ao elaborar-se um projeto deste tipo, todas as partes da confecção deste protótipo trazem um substancial aprendizado com relação à prática em sistemas digitais microprocessados.

As quais são:

- O envolvimento com os estudos de cada componente utilizado;
- O desenvolvimento de um esquemático satisfatório;
- O desenvolvimento do leiaute;
- A montagem;
- A gravação da EPROM;
- E a gravação da PAL.

Portanto se faz deste trabalho um interessante sistema que além da sua vasta utilidade em sistemas de controle de processos, também é muito útil para quem o desenvolve.

6. Trabalhos Futuros

Com este trabalho desenvolvido pode-se pensar em melhorias ou ainda em aplicações muito interessantes para este projeto. Uma melhoria interessante é a transmissão do programa do PC ao sistema de processamento via rádio, precisando desenvolver dois transceptores seriais de dados, onde um ficaria no PC e outro no sistema de processamento para que os mesmos se comuniquem entre si para fazerem as trocas de dados necessárias.

Aplicações interessantes que podem ser citadas são: controlador de velocidade de motores, controle de display, teclado, motores de passo para controlar um robô, etc.

Temos uma infinidade de aplicações que podem ser feitas dependendo apenas da idéia e do desenvolvimento do mesmo para este ser implementado utilizando o sistema de processamento desenvolvido.

7. Bibliografia

Silva Junior - Microcontrolador 8051 (Hardware e Software). São Paulo: Editora Érica – 2ª edição, 1990.

Silva Junior - Microcontrolador 8051 (Hardware e Software). São Paulo: Editora Érica – 3ª edição, 1994.

Idoeta & Capuano - Elementos de Eletrônica Digital. São Paulo: Editora Érica -16ª edição, 1984.

Marinho & Marinho - Mini-Curso de microcontrolador – Saber Eletrônica - jan, 2001.

Ribeiro & Melo – WinBoard e WinDraft for Windows. São Paulo: Editora Érica Ltda – Edição revisada e Ampliada, 1996.

Tscontrol – Emulador do 8051 – março 2003

<http://www.tscontrol.com>

Lima & Rosa - Microcontrolador 8051 – julho 2002

<http://www.dmat.furg.br/~vagner/8051emu/apostila/>

Moraes & Silva – Aplicações com o microcontrolador 8051 – julho 2002

<http://www.ufpa.br/dee/lacos/andreya-reinaldo.htm>

Vieira, julho 2002.

<http://www.eps.ufsc.br/disserta99/vieira/cap1.html>

Microeletrônica - Projetos com microcontroladores, julho 2002

http://geocities.yahoo.com.br/metronic8051/M_eletronica.html

Santos, D. – Projeto de controladores de posição angular baseado em algoritmos adaptativos – julho 2002.

<http://www.ufpa.br/dee/lacos/davi.htm>

Casare, julho 2002.

http://www.facens.br/site/ensino/projetos/microcont_leandro/INTRODUO.htm

Cefet – PR, julho 2002.

<http://www.lit.cpdtt.cefetpr.br/~claudio/Microcontroladores.htm>

News Letter – Braile Biomédica, julho 2002.

<http://www.braile.com.br>

ANEXOS