

Microcontrolador PIC18F452

Organização e Arquitetura

Eduardo José dos Reis Nobre Junior¹

¹Universidade Estadual de Santa Cruz (UESC)
Caixa Postal 45662-900 Campus Soane Nazaré de Andrade,
Rodovia Jorge Amado, Km 16, Bairro Salobrinho, Ilhéus-Bahia

Abstract. *The PIC18F452 Microcontroller used in monitoring activities of equipment, instrumentation and monitoring, data acquisition. Using the RISC architecture, equipped with internal memory divided into instruction memory and data memory. Capable of parallel communication and simultaneous access to data and instructions by using Harvard architecture, thus having separate buses for memory and instructions. This article aims to present the characteristics from the point of view of its organization and architecture and how certain characteristics can be used by the user.*

Resumo. *O Microcontrolador PIC18F452 utilizado em atividades de monitoramento de equipamento, instrumentação e monitoramento, aquisição de dados. Utilizando a arquitetura RISC, dotado de memória interna dividida em em memória de instrução e memória de dados. Capaz de comunicação paralela e acesso simultâneo a dados e instruções por utilizar a arquitetura Harvard, tendo portanto barramentos separados para memória e instruções. Esse artigo visa apresentar as características a partir do ponto de vista da sua organização e arquitetura e como certas características podem ser utilizadas pelo usuário.*

Visão Geral e Organização

O Microcontrolador PIC18F452 utilizado em atividades de monitoramento de equipamento, instrumentação e monitoramento, aquisição de dados. Utilizando a arquitetura RISC, dotado de memória interna dividida em em memória de instrução e memória de dados. Capaz de comunicação paralela e acesso simultâneo a dados e instruções por utilizar a arquitetura Harvard.

Organização de memória

A memória é dividida em três grupos descritos abaixo

- Memória de Programa
- RAM de Dados
- EEPROM de Dados

Memória de Programa

Um Program Counter de 21 bits é capaz de endereçar o espaço de memória de 2 MB. Acessar um espaço entre a memória física e a memória de 2 MB causará a leitura de todos zeros, uma instrução NOP.

O tendo 32 KBytes de memória Flash, podendo armazenar palavras de instruções de 16K. O endereço do vetor de RESET é encontrado em 00000H e os vetores de interrupções se encontram nos endereços 00008h e 00018h.

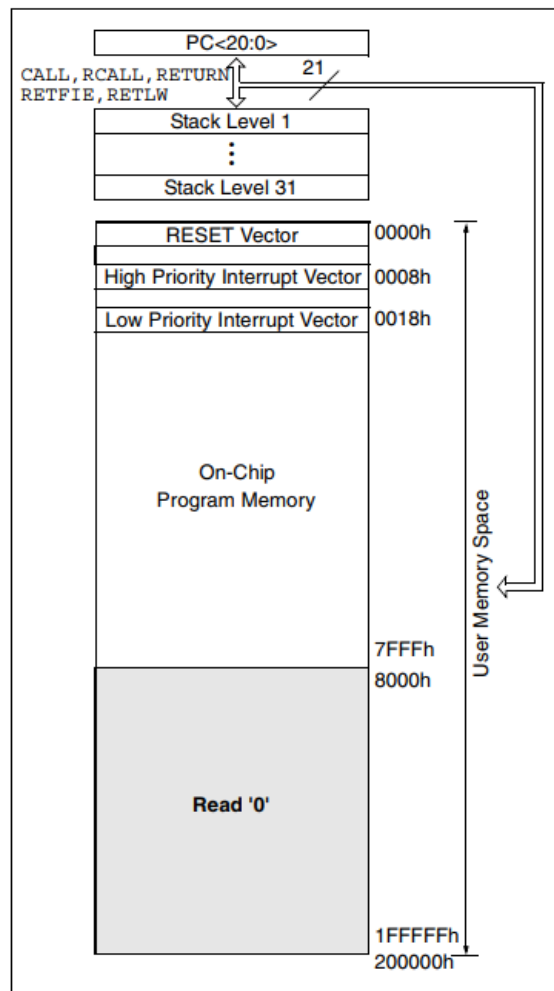


Figure 1. Figura 1. Mapa de Memória e Pilha, Fonte: [Technology Inc 2006]

Instruções na Memória Programa

A memória de instruções é acessada em bytes. Instruções são armazenadas como 2 ou 4 bytes na memória. O Least Significant Byte (LSB) de uma palavra de instrução sempre é armazenado num local de memória de endereço par (LBS='0').

As instruções CALL e GOTO possuem um endereço absoluto de memória na instrução. Instruções são sempre armazenadas no espaço de uma palavra, os dados contidos em uma instrução são um endereço.

Organização da Memória de Dados

A memória de dados é implementada com o uso de SRAM. Cada registro na memória de dados tem um endereço de 12 bits, permitindo o armazenamento de até 4096 bytes. O mapa de memória é dividido em 16 bancos contendo 256 bytes cada. O últimos 4 bits do Bank Select Register (BSR) selecionam qual banco será acessado. Os primeiros 4 bits do BSR não são implementados. A memória de dados contém Special Function Registers (SFR) e General Purpose Registers (GPR). Os SFR são utilizados para controle, status do controlador e funções auxiliares, enquanto GPR são utilizados para armazenamento de dados.

Toda memória pode ser acessada diretamente ou indiretamente. Acesso direto talvez necessite do uso de um registrador BSR. Acesso indireto necessita do uso de um File Select Register (FSRn) e um correspondente Indirect (INDFn). Cada FSR armazena endereços de 12 bits que podem ser utilizados para acessar qualquer valor do mapa de da memória de dados.

O conjunto de instruções e arquitetura permite operações entre todos os bancos. Isso pode se realizado por endereçamento indireto com o uso da instrução MPVFF.

Memória de Programa Flash

A memória de instruções Flash suporta operações de leitura e escrita. Uma leitura na memória de instrução é executada um byte por vez. Uma escrita realizada na memória de instrução é executada em blocos de 8 bytes por vez.

Escrever ou apagar a memória de instruções irá interromper operações de buscas até que a operação seja realizada. A memória não pode ser acessada durante a escrita ou remoção de instruções, portanto, código não pode ser executado.

Memória de dados EEPROM

A memória de dados EEPROM suporta operações de leitura e escrita durante a operação normal. A memória de dados não é diretamente mapeada no espaço de do registrador, é registrado portanto por Special Function Registers.

Existem quatro SFR utilizados para leitura e escrita de programas e dados na memória EEPROM. Esses registradores são EECON1 EECON2 EEDATA EEADR.

Pilha de retorno de endereço

A pilha de retorno de endereço permite até 31 chamadas de instruções e interrupções em qualquer combinação. O PC é colocado no topo da pilha quando as instruções CALL ou RCALL são executadas ou uma interrupção é reconhecida, armazenando o endereço anterior para posterior utilização usando as instruções RETURN, RETLW ou RETFIE para pegar o valor de PC da pilha.

A pilha opera com uma estrutura de 31 palavras de 21 bits na RAM e uma pilha de ponteiros de 5 bits, com a pilha de ponteiros inicializada com 00000b após todos os RESETS. Não há RAM associada com o pilha de ponteiros 00000b.

O espaço de pilha não faz parte nem da memória de instruções nem de dados. A pilha de ponteiro aceita operações de leitura e escrita, o topo da pilha pode ser acessado através de registradores SFR. Dados podem ser colocados e removidos do topo da pilha utilizando operações SFR. Um bit de sinal indica se o ponteiro da pilha se encontra acima ou no limite de 31 fornecidos.

Fluxo de dados e Pipeline

Um ciclo de instruções consiste em quatro Q ciclos (Q1, Q2, Q3, Q4). A instrução de busca e execução passam pelo pipeline de forma que a operação de busca dure um ciclo de instrução enquanto a decodificação e execução da instrução duram outro ciclo. Se uma instrução fizer com que o PC se altere, dois ciclos serão necessários para a conclusão da instrução.

Um ciclo de busca começando com o PC sendo incrementado para Q1. No ciclo de instrução, a instrução buscada é levada ao IR no ciclo Q1. Essa instrução então é decodificada e executada nos ciclos seguintes. Dados são lidos durante Q2 e escritos em Q4, na leitura do operando e escrita do destino.

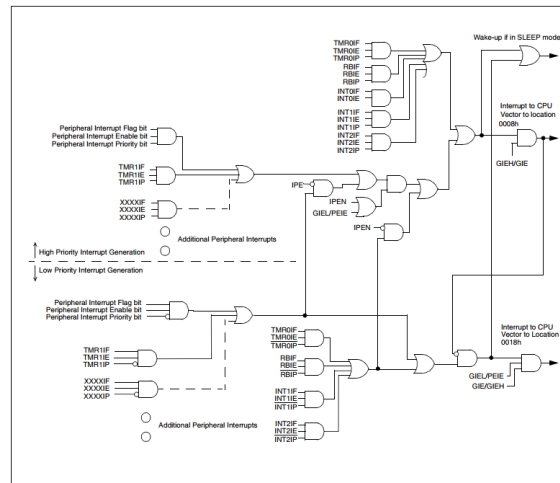


Figure 2. Figura 2. Estrutura do sistema de interrupções, Fonte: [Technology Inc 2006]

Interrupções

O microcontrolador possui múltiplas fontes de interrupção e uma característica que permite cada fonte de interrupção ser associada a um nível de prioridade. O vetor de interrupções de alta prioridade se encontra no endereço 000008h e o vetor de interrupções de baixa prioridade se encontra no endereço 000018h. Interrupções de alta prioridade sobrescreveram interrupções que esteja em andamento. O dispositivo possui 10 registradores responsáveis por controlar as operações de interrupção.

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

Cada fonte de interrupção, exceto INT0, possui três bits para controlar suas operações. Esses bits são:

- Bit de flag para indicar que um evento de interrupção ocorreu.
- Bit de enable que permite que o programa em execução se desvie quando esse bit está acionado.
- Bit de prioridade configurado para high ou low.

Quando uma interrupção é respondida, o Global Interrupt Enable (GIE) é limpo para desabilitar futuras interrupções. Se o bit de prioridade estiver limpo esse se torna o bit de GIE. Se níveis de prioridade forem utilizados, estes serão os bits GIEH ou GIEL.

O endereço de retorno é colocado na pilha de endereços citadas na sessão X o PC é carregado com o endereço do vetor de instruções (000008h ou 000018h). Uma vez que no Interrupt Service Routine, a fonte da interrupção pode ser determinada ao se ler os bits flags da interrupção. Os bits de interrupção devem ser limpos antes do religamento do sistema de interrupções para evitar interrupções recursivas.

Entrada e Saída

Dependendo do modelo do dispositivo, existem cinco ou três portas disponíveis. Alguns dos pinos de I/O são multiplexados com uma função alternativa.

Cada porta tem três registradores para essa operação, sendo eles:

- TRIS
- PORT
- LAT

Os registradores acima são separados em 5 grupos A, B, C, D e E criando $TRIS_A$, $TRIS_B$ e assim por diante.

Table 1. Tabelas de Características do Registrador PORT

PORT	A	B	C	D	E
Largura	7 bits	8 bits	8 bits	8 bits	3 bits
Bi-directional	Sim	Sim	Sim	Sim	Sim

Cada registrador PORT possui um conjunto de pinos R_A , R_B ... Esses pinos podem ser utilizados para leitura ou gravação, analógica ou digital.

O registrador $TRIS_A$ controla a direção dos pinos R_A , mesmo quando eles estão sendo utilizados de forma analógica, trocando os pinos de entrada para saída.

O usuário deve garantir que os bits no registrador $TRIS_A$ estejam configurados quando utilizando R_A como entrada analógica.

O registrador LAT_A está mapeado na memória. Operações de leitura e escrita em LAT_A escreve os valores como entrada ou saída em $PORT_A$. Essa situação se mantém verdadeira para LAT_E e $PORT_E$, LAT_C e $PORT_C$ e por seguinte.

Conjunto de Instruções

O PIC18 possui um conjunto de instruções que melhora a versão anterior PICmicro, mantendo ainda a fácil migração do conjunto de instruções do PICmicro.

A maioria das instruções são palavras únicas de 16 bits, mas existem três instruções que requerem dois espaços de instrução.

Cada palavra é dividida em um OPCODE, que especifica o tipo da instrução e um ou mais operandos. O conjunto de instruções é extremamente ortogonal e é agrupado nas categorias abaixo:

- Instruções orientadas a Byte
- Instruções orientadas a Bit
- Instruções Literais
- Instruções de Controle

A maioria das instruções **orientadas a Byte** contêm três operandos:

1. O registrador de arquivo (**f**)
2. O destino do resultado (**d**)
3. A memória acessada (**a**)

O registrador de arquivo f especifica qual registrador de arquivo deverá ser utilizado pela instrução.

O destino do resultado *d* especifica onde o resultado da operação será colocado. Se o *d* tem o valor de 0, o resultado deverá ser colocado no registrador WREG. Se o valor de *d* for 1, o resultado deverá ser colocado no registrador especificado na instrução.

A maioria das instruções **orientadas a Bit** contem três operandos:

1. O registrador de arquivo (**f**)
2. O bit do registrador de arquivo (**b**)
3. A memória acessada (**a**)

O bit do registrador de arquivo *b* seleciona o número do bito afetado pela operação, enquanto o designador do registrador de arquivo *f* representa o número do arquivo no qual o bit está localizado.

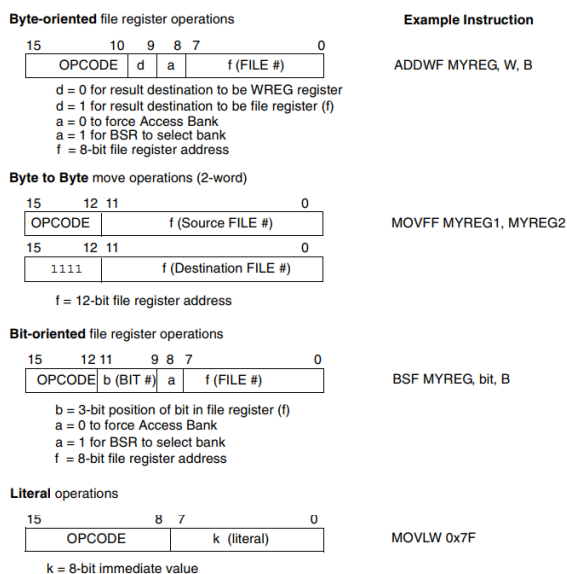


Figure 3. Figura 3. Estrutura geral das Instruções, Fonte: [Technology Inc 2006]

Instruções **literais** podem usar os seguintes operandos:

1. Um valor literal a ser carregado em um registrador de arquivo (**k**)
2. O registrador FSR a ser carregado o valor do literal (**f**)
3. Operando não requerido (**–**)

Instruções de **controle** podem usar os seguintes operandos:

1. Um endereço de memória de programa (**n**)
2. O modo das instruções de Call e Return (**s**)
3. O modo das instruções Table Read e Table Write (**s**)
4. Operando não requerido (**–**)

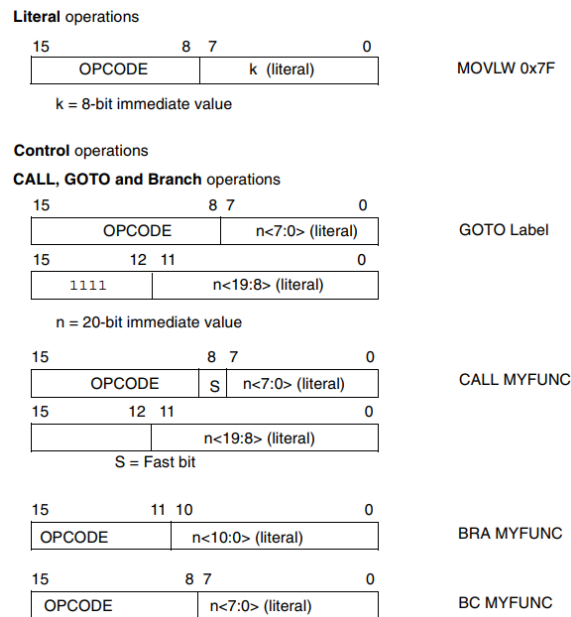


Figure 4. Figura 3. Estrutura geral das Instruções, Fonte: [Technology Inc 2006]

References

Technology Inc, M. (2006). *PIC18FXX2 Data Sheet*
<http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>. Microchip Technology, 1th edition.