**1 –** Na Arquitetura Harvard a memória do programa e a memória de dados são inteiramente separados.

**2 -** Os pinos são usados para a programação do dispositivo. O PIC usa o GP0, GP1 e GP3 para fazer as programações. Tabela

Descrição gerada automaticamente

**- GP0** é um pino de E/S bidirecional.

**- ICSPDAT** é um pino de dados, que possui uma programação serial no circuito.

**- GP1** é um pino de E/S bidirecional.

**- ICSPCLK** é um pino de clock, que possui uma programação serial no circuito.

**- GP2** é um pino de E/S bidirecional.

**- T0CLK** é o clock de entrada para TMR0.

**- GP3** é um pino de entrada.

**- MCLR** é o master clear (reset).

**- VPP** que é o pino ultizado quando queremos gravar o PIC. Colocamos uma tensão alta nele, para gravar os dados que vamos gravar via serial.

**- GP4** é um pino de E/S bidirecional.

**- OSC2** é usado para se conectar a um cristal externo.

**- GP5** é um pino de E/S bidirecional.

**- OSC1** é usado para se conectar a um cristal externo.

**- CLKIN** pode ser usado para entrada de clock externo, para comandar todo o microcontrolador.

**3 -** **\_CONFIG**: ele seta os bits conforme o necessário.

**\_MCLRE\_OFF**: Ele desabilita o reset externo. Quando habilitado processador será resetado, já desabilitado o pino é usado como entrada GP3.

**\_CP\_OFF**: Ele desabilita a proteção de código. Que faz permitir ou bloqueia o código de ser lido o chip depois de gravado.

**\_WDT\_OFF**: Ele desabilita o watchdog Timer. Ele serve para resetar o pic quando o programa falha. O watchdog é ruim na fase de prototipagem, então normalmente deixamos desligado.

**\_IntRC\_OSC**: Ele configura os bits FOSC1 e FOSC0 pois assim ele trabalha junto com o oscilador RC interno.

**4 -** A função CALL tem 8 bytes e a GOTO tem 9 bytes, então instrução GOTO pode pular para qualquer posição da memória, Já o CALL só pode chamar sub-rotinas que estejam nos primeiros 256 bytes de memória.

**5 -** O bit 6 (GPPU) é usado para habilitar ou desabilitar os pull-ups internos. Para poder desabilitar o pull-up interno deve setar o valor 1.

**7 -** A diretiva banksel seleciona os bancos corretos para acesso corretamente as variáveis do programa, O banksel funciona para posições de memória fixas, também para posições de memória definidas pelo usuário, se as variáveis estiverem todas em uma única seção de dados, só é necessária uma diretiva banksel.

**8-** A fim de que estes registradores funcionem de forma correta, além do conteúdo que deve ser neles armazenado, deve-se configurar o microcontrolador para que estas portas funcionem a contento. Diferente da porta que utilizamos para entrar numa sala, nos microcontroladores, as portas só funcionam num sentido, ou são entradas ou são saídas. Não existe a possibilidade de uma porta ser no mesmo instante entrada e saída e a razão disso é fácil de se entender, circuitos eletrônicos trabalham com diodos, transistores, MOSFET e outros dispositivos que conduzem corrente apenas num sentido. Portanto o usuário deve escolher e especificar qual a direção que se espera para a circulação de corrente naquele momento no microcontrolador, Para realizar esta configuração é preciso utilizar os registradores TRISx. TRISA e TRISB, São registradores de funções especiais que tem a missão de configurar as portas. De outra forma dependendo do valor colocado num bit do registrador TRISA, o bit correspondente no registrador PORTA será uma entrada ou uma saída. Se os registradores TRISA e TRISB estiverem mapeados em memória, não é preciso usar a instrução TRIS.