

Intel 8051 - Estudo Preliminar

Felipe Augusto Lee, Victor Hugo Garrett

Estrutura e nome dos registradores:

1. Tem 8 registradores de uso geral R0-R7, mapeados na IRAM. Existem quatro mapeamentos possíveis, equivalentes a quatro bancos desses registradores que podem ser selecionados: de 0x00 até 0x07, de 0x08 até 0x0F, de 0x10 até 0x17 e de 0x18 até 0x1F.
2. Tem um acumulador (A), mapeado na posição 0xE0 da IRAM. Este registrador especial é usado na maioria das instruções.
3. Tem um registrador chamado B, mapeado na posição 0xF0 da IRAM. Usado como uma extensão do acumulador.
4. Tem um Data Pointer (DP), um registrador de 16 bits utilizado para endereçar a memória de programa ou uma memória externa. Esse registrador está mapeado nas posições 0x82 e 0x83 da IRAM.
5. Tem um Stack Pointer (SP), mapeado na posição 0x81 da IRAM. Ele é incrementado e decrementado pelas instruções push e pop respectivamente.
6. Tem um Program Status Word (PSW), mapeado na posição 0xD0 da IRAM. Contém flags de status em cada bit:
 - a. 0: P: Paridade do acumulador (XOR dos bits).
 - b. 1: UD: (User Defined) Lido e escrito por software, não interferido por hardware.
 - c. 2: OV: Setado quando a adição produz overflow.
 - d. 3: RS0: Se refere ao bit menos significativo para acessar as partes do banco de registradores. (Lembrando que na IRAM temos de 0x00 até 0x1F, e temos 8 registradores de uso geral)
 - e. 4: RS1: Se refere ao bit mais significativo para acessar as partes do banco de registradores.
 - f. 5: F0: (Flag Zero) Lido e escrito por software, não interferido por hardware.
 - g. 6: AC: (Aux. Carry) Setado quando adição produz carry do bit 3 pro bit 4.
 - h. 7: C: (Carry Bit) Utilizado comumente como registrador geral para computações, ou como “acumulador booleano”.

Instruções originais do processador para operações de:

1. Carga de constante:
 - a. MOV <dest_byte>, <src_byte>
 - b. O byte do src_byte é copiado para o dest_byte. src_byte não afetado. Nenhuma flag é setada.
2. Cópia de valor entre registradores;
 - a. A instrução MOV <operand1> <operand2> também é usada para mover valores entre registradores, uma vez que pode copiar o valor de um endereço qualquer da IRAM para outro, de <operand2> para <operand1>.
 - b. Os operandos devem ser o acumulador e um registrador de uso geral, ou, alternativamente, dois endereços fixos da IRAM

3. Soma de dois valores:
 - a. `ADD <A>, <src_byte>`
 - b. Adiciona o byte da variável indicada no acumulador, deixando o resultado neste.
 - c. As flags carry e auxiliary carry são sentadas, respectivamente, se há carry-out dos bits 7 e bit 3. Se não há, são limpas. Na soma de unsigned integers, o carry indica um overflow.
 - d. A flag OV é setada se há carry-out do bit 6 e não do bit 7, ou ao contrário; senão a flag é limpa.
 - e. Quatro maneiras de endereçar o `src_byte`: registrador, direta, indireta com registrador ou imediato.
4. Subtração de dois valores:
 - a. A instrução `SUBB A <operand2>` subtrai do acumulador o valor do segundo operando com a carry flag. `<operand2>` pode ser:
 - i. Um dos registradores de uso geral (R0 - R7)
 - ii. Uma constante imediata.
 - iii. Um valor de memória cujo endereço é passado na instrução.
 - iv. Um valor de memória endereçado por um registrador.
 - b. As flags que podem ser afetadas são: carry, auxiliary carry e overflow. As flags carry e auxiliary carry são setadas quando há um empréstimo nos bits 7 e 3 respectivamente. A flag de overflow é setada quando uma subtração entre dois valores signed de 8 bits sai do escopo de -128 até 127.
5. Desvio incondicional:
 - a. O desvio incondicional é feito pelas instruções `JMP`, `AJMP`, `SJMP` e `LJMP`.
 - b. A instrução `JMP @A+DP` pula para o endereço representado pela soma do acumulador com o DP.
 - c. A instrução `AJMP <11_bit_address>` pula absolutamente para um endereço, mas afeta apenas os 11 bits menos significativos do PC, isto é, pula absolutamente dentro de um certo bloco da memória de programa.
 - d. A instrução `SJMP <relative_address>` soma o valor de `<relative_address>` ao PC.
 - e. A instrução `LJMP <code_address>` pula para o endereço `<code_address>`, fornecido como imediato na instrução.
6. Desvio condicional
 - a. As instruções que realizam pulo condicional são: `CJNE`, `DJNZ`, `JC`, `JNC`, `JZ`, `JNZ`, `JB`, `JNB` e `JBC`.
 - b. A instrução `CJNE <operand1> <operand2> <relative_address>` compara os valores de `<operand1>` e `<operand2>` soma `<relative_address>` no PC caso eles sejam diferentes. Se `<operand2>` for um imediato, `<operand1>` pode ser qualquer registrador de uso geral, acumulador ou acesso indireto à memória. Se `<operand2>` for um endereço de memória, `<operand1>` deve ser o acumulador.
 - c. A instrução `DJNZ <operand> <relative_address>` decrementa o valor de `<operand>` e, se ele não atingir 0, soma `<relative_address>` ao PC.

<operand> pode ser qualquer registrador de uso geral ou um endereço fixo da IRAM.

- d. A instrução JC <relative_address> soma <relative_address> no PC se a flag de carry estiver setada.
- e. A instrução JNC <relative_address> soma <relative_address> no PC se a flag de carry não estiver setada.
- f. A instrução JZ <relative_address> soma <relative_address> no PC se o acumulador for 0.
- g. A instrução JNZ <relative_address> soma <relative_address> no PC se o acumulador não for 0.
- h. A instrução JB <bit_address> <relative_address> soma <relative_address> no PC se o bit indicado por <bit_address> estiver setado.
- i. A instrução JNB <bit_address> <relative_address> soma <relative_address> no PC se o bit indicado por <bit_address> não estiver setado.
- j. A instrução JBC <bit_address> <relative_address> soma <relative_address> no PC se o bit indicado por <bit_address> estiver setado e então o limpa.

7. Acesso à memória (modo de endereçamento indireto)

- a. O acesso indireto a IRAM pode ser feito com a instrução MOV já apresentada, usando os operandos @R0 ou @R1 que acessam a IRAM nos endereços contidos por R0 e R1 respectivamente.
- b. O acesso a uma memória externa é feito pela instrução MOVX <dest> <source>. As possíveis combinações de operandos são restritas: um deles deve obrigatoriamente ser o acumulador, enquanto o outro é @R0, @R1 ou @DPTR, indicando acesso aos endereços contidos por cada um desses registradores.