-- Arquitetura e Organização de Computadores

-- Arquivo: instruções.txt

-- Anderson Cottica

-- Erik Ryuichi Yamamoto

-- Data entrega: 08/11/17

Para o funcionamento de nosso microprocessador foi necessário implementar algumas instruções e sinais que são apresentados a seguir:

1 - clk – clock -- sinal de clock do sistema

2 - rst - reset -- reset do sistema

3 - estado\_s0 -- para máquina de estados, estado 0

4 - estado\_s1 -- para máquina de estados, estado 1

5 - estado\_s2 -- para máquina de estados, estado 2

6 - mov\_s -- habilita a instrução MOV

7 - add\_s -- habilita a instrução ADD

8 - sub\_s --habilita a instrução SUB

9 - cmp\_s --habilita a instrução CMP

10 - ld\_s -- habilita a inst. LD, manipulação de valores na memória

11 - st\_s -- habilita a instrução ST

12 - reg1 ~ reg6 -- registradores

13 - pc\_s -- saída do PC

14 - reg\_dest\_s -- registrador destino

15 - reg\_fonte\_s -- registrador fonte

Os opcodes implementados em nosso processador tem o seguinte formato para 15 bits da ROM:

Para o TIPO R:

MOV const, Rd -- 0100 CCCCCCC F DDD

MOV Rs, Rd -- 0100 SSS 0000 F DDD

ADD const, Rd -- 0101 CCCCCCC F DDD

ADD Rs, Rd -- 0101 SSS 0000 F DDD

SUB const, Rd -- 1000 CCCCCCC F DDD

SUB Rs, Rd -- 1000 SSS 0000 F DDD

CMP const, Rd -- 1001 CCCCCCC F DDD

CMP Rs, Rd -- 1001 SSS 00000 DDD

ST Rs, @Rd -- 1100 SSS 00000 DDD

LD Rs, @Rd -- 1101 SSS 00000 DDD

A operação MOV trata da movimentação entre valores de registradores ou entre registrador e um valor constante. Da mesma forma, ADD faz uma operação aritmética entre registradores ou entre registrador e uma constante. SUB trata-se da subtração entre valores de registradores ou entre registrador e uma constante. CMP é a comparação entre dois valores de registradores para definição se os valores são iguais ou diferentes a fim de realizar o salto para determinada parte do programa. ST e LD são instruções para leitura e escrita de valores na memória.

FLAGS:

O quarto bit (bit 3) menos significativo fica como indicador da *flag*: se a *flag* for 1, o registrador fonte é uma constante dada pelos próximos, assim podemos carregar constantes de 7 bits; caso a *flag* for 0, a fonte é um registrador. Neste caso, os bits seguintes são zerados.

Para o caso de saltos, se a *flag* indica a presença de um salto. Para o caso da flag ser 0, é feito um salto de acordo com a distância especificada na instrução BRL.

Para o TIPO J:

JMP endereço -- 001111 EEEEEEEEE

BRL distancia -- 001101 LLLLLLLLLLL

NOP -- 000000000000000

O tipo de instrução J representa os saltos do sistema.

OBS.:

C - valor da constante em binário

S - Valor do registrador fonte em binário

D - Valor do registrador de destino em binário

L - Valor que indica a distância (largura) do salto

F - Flag usada para os casos de salto

CMP - compara registrador de destino com uma constante ou com um registrador fonte e seta flag para o caso em que o Registrador de destino for maior ou igual, ou zera quando menor.

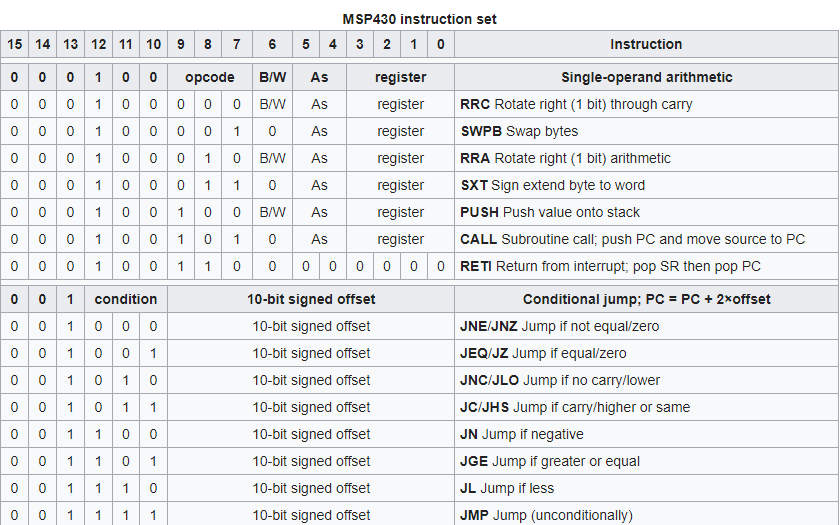
JMP - pula incondicionalmente para o endereço solicitado.

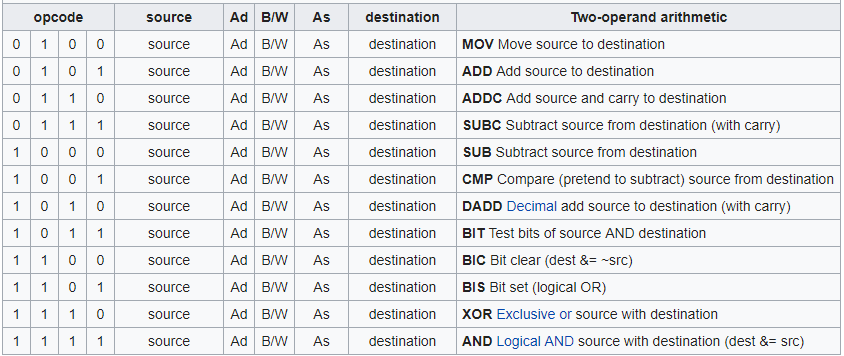
BRL - pula distancia dada, indicada por L quando flag for 0.

ST - carrega o valor de registrador fonte no endereço dado pelo registrador de destino.

LD - carrega o valor no endereço dado do registrador de destino, no registrador fonte.

A tabela a seguir apresenta as instruções destacadas a serem implementadas e seus respectivos opcodes:





Referência utilizada: <https://en.wikipedia.org/wiki/TI\_MSP430>