Especificações trabalho:

Tamanho das Instruções: 16 bits

Definição das Instruções:

|  |  |  |
| --- | --- | --- |
| Instruções | Descrição | Código de Máquina |
| JMP | Jump incondicional | 0 000 0001 00 MMMMMM |
| JZ | Jump condicional, quando o valor comparado for zero | 0 000 0010 00 MMMMMM |
| Load | Carrega valor da memória a um registrador | 1 000 0001 RR MMMMMM |
| iLoad | Carregar um valor indiretamente a um registrador | 1000 0100 RR RR VVVV |
| Move | Move o valor de um registrador para outro | 1010 1000 RR XXXX RR |
| ShiftAdd | Dobra um registrador e soma com outro | 1010 0101 RR XX RR RR |
| Store | Carrega valor a memória | 1 000 0010 RR MMMMMM |
| Add | Operação de soma na ULA | 1 010 0001 RR XX RR RR |
| Sub | Operação de subtração na ULA | 1 010 0010 RR XX RR RR |
| And | Porta AND entre registradores feito na ULA | 1 010 0011 RR XX RR RR |
| Or | Porta OR entre registradores feito na ULA | 1 010 0100 RR XX RR RR |
| Halt | Interrompe o programa | 1111 1111 1111 1111 |
| Nop | Não faz nada | 0000 0000 0000 0000 |

Exemplos de funcionamentos:

Branch MEM= Branch 10

Bzero MEM= Branch 10

Load MEM REG= LOAD 10 B

iLoad REG Offset(REG)(só pode 4 bits de valor do offset por limitação no hardware)

Store REG MEM= LOAD B 10

Add REG1 REG2 REG3= Add A B C (A=B+C)

ShiftAdd REG1 REG2 REG3= Shift REG2 ADD o valor do REG3 e guarda no REG1

Sub REG1 REG2 REG3= Sub A B C (A=B-C)

And REG1 REG2 REG3= And A B C (A=B && C), com A armazenando 1 ou 0

Or REG1 REG2 REG3= Or A B C (A=B || C), com A armazenando 1 ou 0

Halt

Nop4

Tamanho da memória: 6 bits de endereço

Banco de Registradores: quatro registradores=R0,R1,R2,R3

Se o RegWrite estiver ativado, a data escrita será definida por um Mux entre os registradores MDR e a saída a ULA.