// # Load operations

mem[0] = 32'b00000000000000000010111000110111; // LUI x28, 2

mem[1] = 32'b00000000000000000010111000010111; // AUIPC x28, 2

mem[2] = 32'b00000000000000000010000010000011; // LW x1, 0(x0)

mem[3] = 32'b00000000000000000001000100000011; // LH x2, 4(x0)

mem[4] = 32'b00000000000000000000000110000011; // LB x3, 0(x0)

mem[5] = 32'b00000000000000000100001000000011; // LBU x4, 0(x0)

mem[6] = 32'b00000000000000000101001010000011; // LHU x5, 0(x0)

//Arithmetic operations

mem[7] = 32'b00000000001000001000001100110011; // ADD x6, x1, x2

mem[8] = 32'b01000000001000001000001110110011; // SUB x7, x1, x2

mem[9] = 32'b00000000001000001100010000110011; // XOR x8, x1, x2

mem[10] = 32'b00000000001000001110010010110011; // OR x9, x1, x2

mem[11] = 32'b00000000001000001111010100110011; // AND x10, x1, x2

// # Immediate operations

mem[12] = 32'b00000000010000001000010110010011; // ADDI x11, x1, 4

mem[13] = 32'b00000000010000001100011000010011; // XORI x12, x1, 4

mem[14] = 32'b00000000010000001110011010010011; // ORI x13, x1, 4

mem[15] = 32'b00000000010000001111011100010011; // ANDI x14, x1, 4

mem[16] = 32'b00000000010000001010011110010011; // SLTI x15, x1, 4

mem[17] = 32'b00000000010000001011100000010011; // SLTIU x16, x1, 4

// Shift operations

mem[18] = 32'b00000000001000001101100010110011; // SRL x17, x1, x2

mem[19] = 32'b01000000001000001101100100110011; // SRA x18, x1, x2

mem[20] = 32'b01000000000100001101100110010011; // SRAI x19, x1, 1

mem[21] = 32'b00000000001000001001101000110011; // SLL x20, x1, x2

mem[22] = 32'b00000000000100001001101010010011; // SLLI x21, x1, 1

mem[23] = 32'b00000000000100001101101100010011; // SRLI x22, x1, 1

// # Set less than operations

mem[24] = 32'b00000000001000001010101110110011; // SLT x23, x1, x2

mem[25] = 32'b00000000001000001011110000110011; // SLTU x24, x1, x2

// # Store operations

mem[26] = 32'b00000000000100000010000000100011; // SW x1, 0(x0)

mem[27] = 32'b00000000000100000000000000100011; // SB x1, 0(x0)

mem[28] = 32'b00000000000100000001000000100011; // SH x1, 0(x0)

// # Branch operations

mem[29] = 32'b00000000000000000000001001100011; // BEQ x0, x0, 4

mem[30] = 32'b00000000000111111000111110010011; //increment x31

mem[31] = 32'b00000000001000001001001001100011; // BNE x1, x2, 4

mem[32] = 32'b00000000000111111000111110010011; //increment x31

mem[33] = 32'b00000000010000011100001001100011; // BLT x3, x4, 4

mem[34] = 32'b00000000000111111000111110010011; //increment x31

mem[35] = 32'b00000000011000101101001001100011; // BGE x5, x6, 4

mem[36] = 32'b00000000000111111000111110010011; //increment x31

mem[37] = 32'b00000000100000111110001001100011; // BLTU x7, x8, 4

mem[38] = 32'b00000000000111111000111110010011; //increment x31

mem[39] = 32'b00000000101001001111001001100011; // BGEU x9, x10, 4

mem[40] = 32'b00000000000111111000111110010011; //increment x31

// # Jump operations

mem[41] = 32'b00000000010000000000111011101111; // JAL x29, 4

mem[42] = 32'b00000000000000000000111101100111; // JALR x30, 0(x0)

// # Fence operation,Environment call and break

mem[43] = 32'b00001111111100000000000000001111; // FENCE

mem[44] = 32'b00000000000100000000000001110011; // EBREAK

mem[45] = 32'b00000000000000000000000001110011; // ECALL

**DATAMEM:**

mem[0]=8'd5;

mem[1]=8'd0;

mem[2]=8'd0;

mem[3]=8'd0;

mem[4]=8'd2;

mem[5]=8'd0;

mem[6]=8'd0;

mem[7]=8'd0;

mem[8]=8'd3;

mem[9]=8'd0;

mem[10]=8'd0;

mem[11]=8'd0;

mem[12]=8'd12;

mem[13]=8'd13;

mem[14]=8'd14;

mem[15]=8'd15;

mem[16]=8'd16;

mem[17]=8'd17;

mem[18]=8'd18;

mem[19]=8'd19;

mem[20]=8'd20;

mem[21]=8'd21;

mem[22]=8'd22;

mem[23]=8'd23;

mem[24]=8'd24;

mem[25]=8'd25;

mem[26]=8'd26;

mem[27]=8'd27;

mem[28]=8'd28;

mem[29]=8'd29;

mem[30]=8'd30;

mem[31]=8'd31;

mem[32]=8'd32;

mem[33]=8'd33;

mem[34]=8'd34;

mem[35]=8'd35;

mem[36]=8'd36;

mem[37]=8'd37;

mem[38]=8'd38;

mem[39]=8'd39;

mem[40]=8'd40;

mem[41]=8'd41;

mem[42]=8'd42;

mem[43]=8'd43;

mem[44]=8'd44;

mem[45]=8'd45;

mem[46]=8'd46;

mem[47]=8'd47;

mem[48]=8'd48;

mem[49]=8'd49;

mem[50]=8'd50;

mem[51]=8'd51;

mem[52]=8'd52;

mem[53]=8'd53;

mem[54]=8'd54;

mem[55]=8'd55;

mem[56]=8'd56;

mem[57]=8'd57;

mem[58]=8'd58;

mem[59]=8'd59;

mem[60]=8'd60;

mem[61]=8'd61;

mem[62]=8'd62;

mem[63]=8'd63;