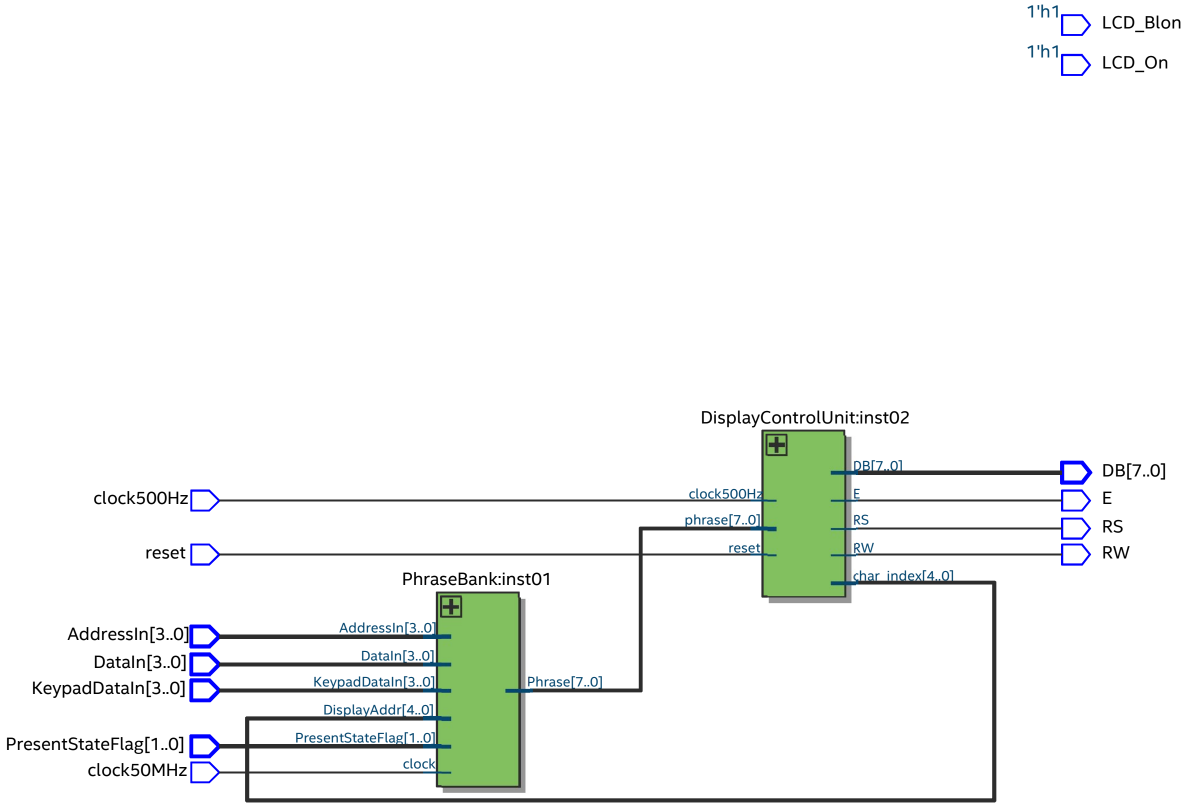
```verilog
module DisplayControlUnit (
    input wire clock500Hz, reset,

    input wire[7:0] phrase,

    output reg[4:0] char_index,

    output reg RS,RW,
    output wire E,
    output reg[7:0] DB
);

    assign E = clock500Hz;

    // Estados
    parameter
        FS1 = 4'd0,
      FS2 = 4'd1,
        FS3 = 4'd2,
        FS4 = 4'd3,
        ClearDisplay = 4'd4,
        DisplayControl = 4'd5,
        EntryMode = 4'd6,
        ReturnHome = 4'd7,
        SetAddress = 4'd8,
        WriteChar = 4'd9;

    reg[3:0] PresentState,NextState;

    // Bloco sequencial - Estados
     always @(posedge reset or posedge clock500Hz) begin
        if(reset) begin
            PresentState <= FS1;
            char_index <= 5'd0;
        end
        else begin
            PresentState <= NextState;
            if(PresentState == WriteChar) begin
                char_index <= char_index + 5'd1;
            end
            if(NextState == ReturnHome) begin
                char_index <= 5'd0;
            end

        end
    end

    // Bloco Combinacional - Estados
    always @(*) begin
        case(PresentState)

            default: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00111000;
                NextState = FS1;
            end

            FS1: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00111000;
                NextState = FS2;
            end

            FS2: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00111000;
                NextState = FS3;
            end

            FS3: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00111000;
                NextState = FS4;
            end

            FS4: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00111000;
                NextState = ClearDisplay;
            end

            ClearDisplay: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00000001;
                NextState = DisplayControl;
            end

            DisplayControl: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00001100;
                NextState = EntryMode;
            end

            EntryMode: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b00000110;
                NextState = WriteChar;
            end

            ReturnHome: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b10000000;
                NextState = WriteChar;
            end

            SetAddress: begin
                RS = 1'b0;
                RW = 1'b0;
                DB = 8'b11000000;
                NextState = WriteChar;
            end

            WriteChar: begin
                RS = 1'b1;
                RW = 1'b0;
                DB = phrase;


                if (char_index == 5'd15) begin
                    NextState = SetAddress;
                end
                else if (char_index == 5'd31) begin
                    NextState = ReturnHome;
                end
                else begin
                    NextState = WriteChar;
                end
            end
        endcase
    end

endmodule
```

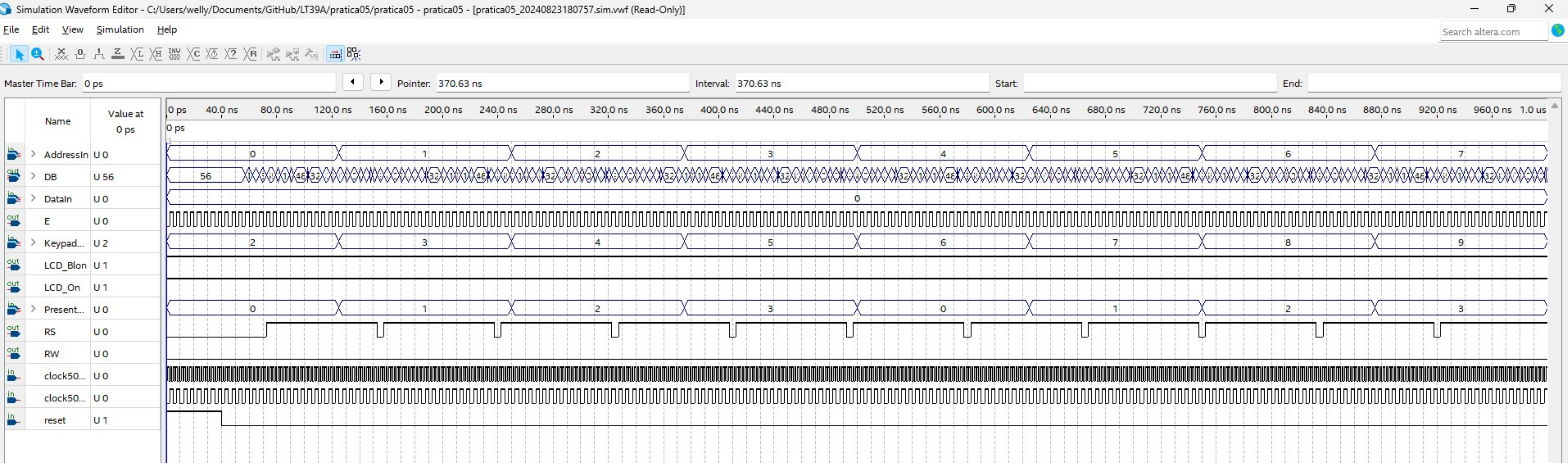
```
module PhraseBank(
    input wire clock,
    input wire[4:0] DisplayAddr,

    input wire[3:0] AddressIn, DataIn, KeypadDataIn,
    input wire[1:0] PresentStateFlag,

    output reg[7:0] Phrase
);

    wire[7:0] OutPadrao;
    wire[3:0] AddressInUnit, AddressInTens, DataInUnit, DataInTens, KeypadDataInUnit, KeypadDataInTens;

    Deconcatener inst00(AddressIn, DataIn, KeypadDataIn, AddressInUnit, AddressInTens, DataInUnit, DataInTens, KeypadDataInUnit, KeypadDataInTens);
    RomPadrao inst01 (clock,DisplayAddr, OutPadrao);

    reg[7:0] Numbers [0:9];
    initial begin
        Numbers[0] = "0";
        Numbers[1] = "1";
        Numbers[2] = "2";
        Numbers[3] = "3";
        Numbers[4] = "4";
        Numbers[5] = "5";
        Numbers[6] = "6";
        Numbers[7] = "7";
        Numbers[8] = "8";
        Numbers[9] = "9";
    end

    reg[7:0] RomState [0:2];
    initial begin
        RomState[0] = "I";
        RomState[1] = "W";
        RomState[2] = "R";
    end

    always @(*) begin

        if(DisplayAddr == 5'd4) Phrase = Numbers[AddressInTens];
        else if(DisplayAddr == 5'd5) Phrase = Numbers[AddressInUnit];

        else if(DisplayAddr == 5'd14) Phrase = RomState[PresentStateFlag];

        else if(DisplayAddr == 5'd20) Phrase = Numbers[KeypadDataInTens];
        else if(DisplayAddr == 5'd21) Phrase = Numbers[KeypadDataInUnit];

        else if(DisplayAddr == 5'd29) Phrase = Numbers[DataInTens];
        else if(DisplayAddr == 5'd30) Phrase = Numbers[DataInUnit];

        else Phrase = OutPadrao;
    end

endmodule
```

```
module RomPadrao (
    input wire clock,
    input wire[4:0] addr,

    output reg[7:0] dataOut
);

    reg[7:0] phrase [0:31];
    initial begin
        phrase[0]  = "A";
        phrase[1]  = "i";
        phrase[2]  = "n";
        phrase[3]  = ":";
        phrase[4]  = "X";     // Dezena Address
        phrase[5]  = "X";     // Unidade Address
        phrase[6]  = " ";
        phrase[7]  = " ";
        phrase[8]  = "S";
        phrase[9]  = "t";
        phrase[10] = "a";
        phrase[11] = "t";
        phrase[12] = "e";
        phrase[13] = ":";
        phrase[14] = "Y";     // R ou W
        phrase[15] = " ";



        phrase[16] = "D";
        phrase[17] = "i";
        phrase[18] = "n";
        phrase[19] = ":";
        phrase[20] = "X";     // Dezena DataTeclado
        phrase[21] = "X";     // Unidade DataTeclado
        phrase[22] = " ";
        phrase[23] = " ";
        phrase[24] = "D";
        phrase[25] = "o";
        phrase[26] = "u";
        phrase[27] = "t";
        phrase[28] = ":";
        phrase[29] = "Y";     // Dezena DataMemoria
        phrase[30] = "Y";     // Unidade DataMemoria
        phrase[31] = " ";
    end


    always @(posedge clock ) begin
        dataOut <= phrase[addr];
    end

endmodule
```

```
module Deconcatener (
    input wire[3:0] AddressIn , DataIn , KeypadDataIn ,

    output wire[3:0] AddressInUnit , AddressInTens , DataInUnit , DataInTens ,
KeypadDataInUnit , KeypadDataInTens
);

assign AddressInUnit = AddressIn % 10;
assign AddressInTens = AddressIn / 10;

assign DataInUnit = DataIn % 10;
assign DataInTens = DataIn / 10;

assign KeypadDataInUnit = KeypadDataIn % 10;
assign KeypadDataInTens = KeypadDataIn / 10;

endmodule
```

```
module Saida (
    input wire clock50MHz, clock500Hz, reset,

   input wire[3:0] AddressIn, DataIn, KeypadDataIn,
   input [1:0] PresentStateFlag,

    output wire RS,RW,
    output wire E,
    output wire[7:0] DB,
    output wire LCD_Blon, LCD_On
);

assign LCD_Blon = 1'b1;
assign LCD_On = 1'b1;

wire[4:0] DisplayAddr;
wire[7:0] Phrase;
PhraseBank inst01 (clock50MHz, DisplayAddr, AddressIn, DataIn, KeypadDataIn,
PresentStateFlag, Phrase);

DisplayControlUnit inst02 (clock500Hz, reset, Phrase, DisplayAddr, RS, RW, E, DB);

endmodule
```


1'h1
LCD_Blon
1'h1
LCD_On
DisplayControlUnit:inst02
DB[7..0]
clock500Hz
E
phrase[7..0]
RS
reset
RW
char_index[4..0]
DB[7..0]
E
RS
RW
clock500Hz
reset
PhraseBank:inst01
AddressIn[3..0]
DataIn[3..0]
KeypadDataIn[3..0]
DisplayAddr[4..0]
PresentStateFlag[1..0]
clock
Phrase[7..0]
AddressIn[3..0]
DataIn[3..0]
KeypadDataIn[3..0]
PresentStateFlag[1..0]
clock50MHz



Simulation Waveform Editor - C:/Users/welly/Documents/GitHub/LT39A/pratica05/pratica05 - pratica05 - [pratica05_20240823180757.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 370.63 ns

Interval: 370.63 ns

Start:

End:

Name	Value at 0 ps
AddressIn	U 0
DB	U 56
DataIn	U 0
E	U 0
Keypad...	U 2
LCD_Blon	U 1
LCD_On	U 1
Present...	U 0
RS	U 0
RW	U 0
clock50...	U 0
clock50...	U 0
reset	U 1