```
1  `timescale 1ns / 1ps
2
3  module lcd_initialization(
4      input clk,
5      input nrst,
6      output reg en,
7      output reg rs,
8      output reg db7,
9      output reg db6,
10     output reg db5,
11     output reg db4
12 );
13
14     reg [5:0] state;
15     reg [32:0] delay_counter; // Counter for delays
16     reg [3:0] outData;      // 4-bit data/cmd bus
17     reg task_done;           // Completion flag
18     reg [32:0] max_counter;  // Maximum count for the current state
```

The code snippet above shows the inputs, outputs, and internal signals needed for the initialization part of the solution. Each of them are defined below and the usage for each.

Inputs and Outputs:

- **Inputs:**
  - clk: System clock signal.
  - nrst: Active low reset signal for initialization.
- **Outputs:**
  - en: Enable signal to trigger LCD commands.
  - rs: Register Select, differentiating between data (1) and commands (0).
  - db7-db4: 4-bit data bus for communication with the LCD.

Internal Signals:

- state: State machine variable to track the initialization sequence.
- delay_counter: Handles timing delays between commands.
- outData: Holds the 4-bit command or data to be sent to the LCD.
- task_done: Indicates when a specific sequence or task is completed.
- max_counter: Determines the maximum delay for the current state.

Line 21 to Line 25:

This block assigns the individual bits of the outData register to the corresponding data bus lines (db7-db4) of the LCD. It ensures that the correct 4-bit data or command is sent to the LCD during each step of communication. By using a combinational always block, it ensures that all changes in outData are immediately seen on db4 up to db7. This block is crucial because the LCD operates in 4-bit mode, requiring precise data transmission for proper initialization and operation. Without this, the LCD would receive incorrect or incomplete data, which will lead to code and the LCD not functioning as designed.

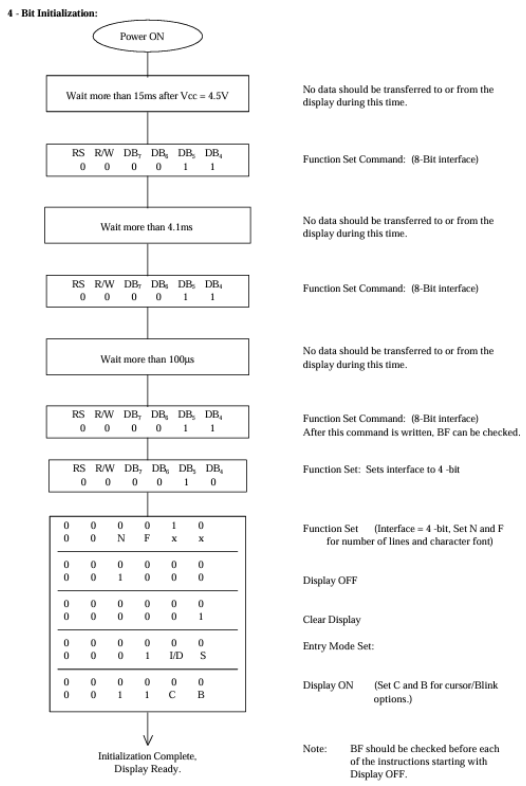
Line 29 to Line 51:

The `execute_sequence` task encapsulates the repetitive sequence for sending a 4-bit command or data to the LCD with precise timing. It manages the enable signal (`en`) and uses

`delay_counter` to ensure proper delays between actions, such as setting the data (`outData`) and toggling the enable signal off after the LCD latches the input. Additionally, it tracks when the operation is complete using the `finished` flag. This task is important because it centralizes and simplifies the process of handling timing-critical operations, reducing redundancy in the code and ensuring that the LCD receives commands with the correct timing required for stable operation.

Line 53 to Line 224:

From this point onward, the code from the 0th state up to the 16th state, as written, follows the 4-bit initialization process outlined in the user manual. Each state performs the actions specified in the flowchart shown below, which is why the code includes 16 states. Some of these states will be explained in more detail in the following sections. The timing and delays were also based on the user manual, with some additional values added to the required delays. However, this did not affect the functionality as long as the delays were not shorter than the required values or excessively long, which could cause major delays. Proper timing and delays are crucial because the system will not function correctly or display values on the LCD if the delays in the code are incorrect.



Line 53 to Line 60:

This part initializes and manages the state of the LCD during a reset or clock edge. When the reset signal (`nrst`) is low, it sets critical control signals (`en` and `rs`) to 0, resets the `state` and `delay_counter`, clears `outData`, and sets `max_counter` to a default value for a

15ms delay. This ensures that the LCD initialization process starts cleanly and predictably after a reset. This is important because it guarantees that the LCD begins operation from a known state, avoiding undefined behavior, and provides the proper timing for the LCD's power-on initialization sequence, which is crucial for stable operation. The 15ms delay is also important because it is stated

Line 62 to Line 134:

This portion of the code demonstrates the initial steps for 4-bit initialization as per the user manual. It begins by waiting for power stabilization (state 0) with a delay of over 15 ms. Subsequently, it issues the "Function Set" command three times (states 1, 3, and 5) with the data `4'b0011`, each followed by specific delays (4.1 ms, 100 µs, etc.) to ensure proper LCD timing as per the manual's requirements for transitioning from 8-bit to 4-bit mode. Finally, in state 6, the "Function Set" command is issued with `4'b0010`, signaling the switch to 4-bit operation. Delays are carefully implemented between states to meet timing constraints outlined in the manual, ensuring stable initialization and preparation for subsequent commands.

Line 136 to Line 224:

This section of the code continues the LCD initialization process in 4-bit mode and configures the display settings. In state 7, the upper nibble of the "Function Set" command (`4'b0010`) is executed, followed by the lower nibble (`4'b1000`) in state 8 to set a 2-line display with 5x7 dots. States 9 and 10 turn the display off by executing the "Display OFF" command in two nibbles (`4'b0000` and `4'b1000`). States 11 and 12 clear the display using the "Clear Display" command (`4'b0000` and `4'b0001`), with a 16 ms delay to ensure the operation completes. States 13 and 14 configure the "Entry Mode Set" command (`4'b0000` and `4'b0110`), enabling cursor increment and shift. Finally, states 15 and 16 turn the display on with the "Display ON" command (`4'b0000` and `4'b1111`). Proper delays are maintained between these steps to ensure the LCD operates reliably according to the user manual specifications.

Line 225 to Line 253:

This part of the code is responsible for positioning the cursor on the LCD at the first line and first column (address 00H), and ensuring the LCD's register settings are updated correctly. In state 17, the upper nibble of the "Set DDRAM Address" command (`4'b0000`) is executed, followed by the lower nibble in state 18. This action sets the cursor to the beginning of the first line. After the command is executed, a small delay (15 ms) is provided in state 19 to allow the LCD to register the command before switching to the data register (RS set to 1). Finally, in state 20, a brief delay is applied before data can be written to the display. These steps ensure that the cursor is correctly positioned, and the necessary time is given for the LCD to process each command, as specified in the manual.

Line 255 to Line 343:

This sequence outlines the process of sending characters to an LCD display in 4-bit mode by splitting each character's ASCII code into its upper and lower nibbles. Each state

represents the transmission of either the upper or lower nibble of a character, such as 'J', 'E', 'H', 'A', and 'D'. The `execute_sequence` function is used to transmit a nibble, and the system waits for the `task_done` signal to ensure the operation is completed before progressing. After transmitting each nibble, a delay (`max_counter`) is introduced to comply with the LCD's timing requirements, ensuring proper operation. Each character's code is sent sequentially, transitioning through states to build the full display content.

Line 345 to Line 385:

This segment handles setting the cursor to the second line of the LCD display in 4-bit mode. State 30 provides a buffer time, ensuring the register is correctly set for command input by selecting the command register (`rs <= 0`) and incrementing a delay counter until `max_counter` is reached. States 31 and 32 send the command to move the cursor to the second line (address 40H) by transmitting the upper (`4H`) and lower (`0H`) nibbles of the address. State 33 then sets a buffer time again, switching to the data register (`rs <= 1`) and resetting the delay counter to prepare for the next operation.

Line 387 to Line 502:

This part is identical to Lines 255 through 343, but it will display the characters ‘A’, ‘L’, ‘I’, ‘B’, ‘A’, ‘S’, ‘E’, and ‘R’. It also uses the same function, ‘execute_sequence’, to transmit a nibble and waits for the ‘task_done’ signal to ensure that the operation is complete. The only difference between this part and Lines 255 through 343 is that it will display the characters on the second line of the LCD display in 4-bit mode, as the cursor was set to it on Lines 345 through 385.

Line 504 to Line 566:

This section manages displaying a name, clearing the screen, and reactivating the display. State 48 sends the upper nibble (`4H`), and state 49 sends the lower nibble (`FH`) to complete the final character display with a delay of 2 seconds for visibility (`max_counter <= 201_000_000`). State 50 ensures a buffer time by selecting the command register (`rs <= 0`) and waiting for the delay counter to complete before proceeding. States 51 and 52 execute the "Clear Display" command by sending its upper (`0H`) and lower (`1H`) nibbles with appropriate delays. Finally, states 53 and 54 turn the display back on by transmitting the "Display ON" command (`0H` upper nibble and `FH` lower nibble), ensuring the LCD is ready for further operations with a final delay for stability.