

**Code Choice Logic:**

```

if(strobe) begin
  case(keypad_sync)
    2'b01: begin
      code_choice = 1'b0;
      use_code = 1'b1;
    end
    2'b10: begin
      code_choice = 1'b1;
      use_code = 1'b1;
    end
    default: begin
      code_choice = 1'b0;
      use_code = 1'b0;
    end
  endcase
end
else begin
  code_choice = 1'b0;
  use_code = 1'b0;
end

```

**Keycode Logic:**

```

partial_code = keycode;
if(use_code && strobe && (state != s10 ||
state != s11 || state != s12)) begin
  partial_code = {keycode[7:0],
  code_choice};
end

```

**State Logic:**

```

next_state = state;
case(state)
  write: begin
    if (w_en)
      next_state = s0;
    else
      next_state = write;
  end
  s0: begin
    if(strobe)
      next_state = s1;
    else
      next_state = s0;
  end
  s1: begin
    if(strobe)
      next_state = s2;
    else
      next_state = s1;
  end
  s2: begin
    if(strobe)
      next_state = s3;
    else
      next_state = s2;
  end
  s3: begin
    if(strobe)
      next_state = s4;
    else
      next_state = s3;
  end
  s4: begin
    if(strobe)
      next_state = s5;
    else
      next_state = s4;
  end
  s5: begin
    if(strobe)
      next_state = s6;
    else
      next_state = s5;
  end
  s6: begin
    if(strobe)
      next_state = s7;
    else
      next_state = s6;
  end
  s7: begin
    if(strobe)
      next_state = s8;
    else
      next_state = s7;
  end
  s8: begin
    if(strobe)
      next_state = s9;
    else
      next_state = s8;
  end
end
s9: next_state = s10;

```

**State Logic Cont:**

```

s10: begin
  if(is_enter) // if true, this means that a register button has been
    enabled
    next_state = s11;
  else if(is_op && is_result)
    next_state = s12;
  else
    next_state = s10;
  end
  s11: next_state = s13; // num_enter state
  s12: next_state = s13; // result_ready state
  s13: begin // checking if we are done registering numbers
    if (r_en)
      next_state = write; // moves back to the protective write state
    else if (strobe) // continues to register more values
      next_state = s1;
    else // waiting for decision
      next_state = s13;
    end
  end

```

**Instruction Logic:**

```

if(state == s9) begin
  store_dig = 1;
  enter = 0;
  write_en = 0;

  // result_ready = 0;
end
else if (state == s11) begin
  store_dig = 0;
  enter = 1;
  write_en = 1;

  // result_ready = 0;
end
else if (state == s12) begin
  store_dig = 0;
  enter = 0;
  write_en = 0;
  // result_ready = 1;
end
else if (state == s10) begin
  store_dig = 0;
  enter = 0;
  write_en = 0;
end
else begin
  store_dig = 0;
  enter = 0;
  write_en = 0;
end

```

