

Máquina de Estados Finitos para Sistema de Ascensor

1. Definición de Estados y Registro de Estado

Estados definidos:

```
vhdl

type estado_tipo is (
    IDLE,           -- Ascensor inactivo
    DOOR_OPENING,   -- Puerta abriéndose
    DOOR_OPEN,      -- Puerta abierta
    DOOR_CLOSING,   -- Puerta cerrándose
    MOVING_UP,      -- Ascensor subiendo
    MOVING_DOWN,    -- Ascensor bajando
    ERROR          -- Estado de error
);
```

Registro de Estado:

```
vhdl

-- Registro de estado
process(clk, rst)
begin
    if rst = '1' then
        estado_actual <= IDLE;
        piso_reg <= "001"; -- Iniciar en piso 1
    elsif rising_edge(clk) then
        estado_actual <= estado_siguiete;

        -- Actualizar piso actual solo cuando haya Llegado a un piso
        for i in 0 to 4 loop
            if estado_actual = MOVING_UP or estado_actual = MOVING_DOWN then
                if sensor_piso(i) = '1' then
                    piso_reg <= std_logic_vector(to_unsigned(i+1, 3));
                end if;
            end if;
        end loop;
    end if;
end process;
```

El estado inicial es **IDLE** y el piso inicial es el piso 1 (representado como "001"). El registro de estado se actualiza en cada flanco de subida del reloj, y el piso actual se actualiza cuando el ascensor está en movimiento y detecta un sensor de piso.

2. Lógica de Transición de Estados

La lógica de transición determina el próximo estado basado en el estado actual y las entradas. En este diseño, las transiciones dependen de:

- Estado actual
- Solicitudes de pisos (botones internos y externos)
- Sensores de piso
- Botones de control (abrir, cerrar, emergencia)
- Temporizadores (puerta, inactividad, entre pisos)
- Señales de error (sobrecarga, anomalía de energía)

vhd1

```
-- Lógica de próximo estado
process(estado_actual, solicitud_pendiente, btn_subir, btn_bajar, btn_piso, sensor_piso,
        btn_abrir, btn_cerrar, btn_emergencia, tiempo_puerta, tiempo_inactivo, tiempo_entre_pis
        sobrecarga, anomalia_energia, hay_solicitud_arriba, hay_solicitud_abajo, piso_reg)
begin
    -- Valor predeterminado para evitar latches
    estado_siguiete <= estado_actual;

    case estado_actual is
        when IDLE =>
            -- ...

        when DOOR_OPENING =>
            -- ...

        when DOOR_OPEN =>
            -- ...

        when DOOR_CLOSING =>
            -- ...

        when MOVING_UP =>
            -- ...

        when MOVING_DOWN =>
            -- ...

        when ERROR =>
            -- ...

    end case;
end process;
```



Transiciones principales:

- **IDLE → DOOR_OPENING:** Cuando hay una solicitud pendiente
- **DOOR_OPENING → DOOR_OPEN:** Cuando el tiempo de apertura de puerta se cumple
- **DOOR_OPEN → DOOR_CLOSING:** Cuando el tiempo de espera se cumple o se presiona el botón de cerrar
- **DOOR_CLOSING → MOVING_UP/DOWN:** Según las solicitudes pendientes

- **MOVING_UP/DOWN** → **DOOR_OPENING**: Cuando llega a un piso solicitado
- **Cualquier estado** → **ERROR**: En caso de anomalía de energía, botón de emergencia o sobrecarga
- **ERROR** → **IDLE**: Cuando se resuelven las condiciones de error

3. Lógica de Salida

Las salidas dependen del estado actual (máquina de Moore). Las principales salidas son controles para:

- Motor (subir/bajar)
- Puerta (abrir/cerrar)
- Luces y alarmas
- Reset de temporizadores
- Indicador de piso actual

vhd1

```

-- Lógica de salidas
process(estado_actual, piso_reg)
begin
    -- Valores predeterminados para evitar latches
    motor_subir <= '0';
    motor_bajar <= '0';
    abrir_puerta <= '0';
    cerrar_puerta <= '0';
    luz_cabina <= '0';
    alarma_sonora <= '0';
    alarma_visual <= '0';
    estado_error <= '0';
    reset_timer_puerta <= '0';
    reset_timer_inactividad <= '0';
    reset_timer_entre_pisos <= '0';

    -- Siempre mostrar el piso actual
    piso_actual <= piso_reg;

    case estado_actual is
        when IDLE =>
            -- Luz apagada en inactividad
            luz_cabina <= '0';
            reset_timer_inactividad <= '1';

        when DOOR_OPENING =>
            abrir_puerta <= '1';
            luz_cabina <= '1';
            reset_timer_puerta <= '1';
            alarma_sonora <= '1'; -- Señal sonora de apertura
            alarma_visual <= '1'; -- Señal visual de apertura

        when DOOR_OPEN =>
            luz_cabina <= '1';
            reset_timer_puerta <= '1';

        when DOOR_CLOSING =>
            cerrar_puerta <= '1';
            luz_cabina <= '1';
            reset_timer_puerta <= '1';
            alarma_sonora <= '1'; -- Señal sonora de cierre
            alarma_visual <= '1'; -- Señal visual de cierre
    end case;
end process;

```

```

when MOVING_UP =>
    motor_subir <= '1';
    luz_cabina <= '1';
    reset_timer_entre_pisos <= '1';

when MOVING_DOWN =>
    motor_bajar <= '1';
    luz_cabina <= '1';
    reset_timer_entre_pisos <= '1';

when ERROR =>
    alarma_sonora <= '1';
    alarma_visual <= '1';
    estado_error <= '1';
    -- Puerta cerrada en caso de anomalía de energía
    cerrar_puerta <= '1';

end case;
end process;

```

Tabla de Salidas por Estado

Estado	Motor Subir	Motor Bajar	Abrir Puerta	Cerrar Puerta	Luz Cabina	Alarma Sonora	Alarma Visual	Estado Error
IDLE	0	0	0	0	0	0	0	0
DOOR_OPENING	0	0	1	0	1	1	1	0
DOOR_OPEN	0	0	0	0	1	0	0	0
DOOR_CLOSING	0	0	0	1	1	1	1	0
MOVING_UP	1	0	0	0	1	0	0	0
MOVING_DOWN	0	1	0	0	1	0	0	0
ERROR	0	0	0	1	0	1	1	1

Diagrama de Tiempo Simplificado

