



Escuela de Ingeniería en Computadores
CE 3201 — Taller de Diseño Digital

Documento de diseño

Autores:

Fabricio González Cerdas
Jian Zheng Wu

Profesor:

Luis Barboza Artavia

1. Propuestas de diseño: FSM del juego *Memoria*

A continuación se presenta una propuesta de FSM basada en el modelo **Moore**, donde las salidas dependen únicamente del estado actual, y no de las entradas. Los estados en esta FSM están claramente definidos y reflejan la secuencia del juego.

1.1. Propuesta A: FSM Moore por fases

Descripción de los estados La FSM está diseñada para controlar el flujo del juego, específicamente en el manejo de las selecciones de cartas de los jugadores. Los estados son los siguientes:

- **S_IDLE**: Estado inicial del juego. Aquí se espera la acción de un jugador para comenzar una ronda.
- **S_ONE**: Este estado ocurre cuando el jugador ha seleccionado su primera carta. La FSM espera que el jugador seleccione una carta, y cuando lo haga, esta carta se marca como abierta. El temporizador de 15 segundos también comienza aquí.
- **S_TWO**: Este estado es alcanzado cuando el jugador ha seleccionado su segunda carta. En este momento, la FSM espera que el jugador seleccione una carta, y se marca también como abierta. Luego, las cartas se comparan en el siguiente estado, **S_CHECK**.
- **S_CHECK**: En este estado, la FSM verifica si las dos cartas seleccionadas son iguales. Si las cartas coinciden, pasa a **S_UPDATE_SCORE** para actualizar el puntaje y luego a **S_KEEP_TURN** para mantener el turno y volver a seleccionar cartas. Si no coinciden, se pasa a **S_HOLD** para mantener las cartas abiertas por un breve período antes de cambiar de turno.
- **S_UPDATE_SCORE**: Si las cartas coinciden, este estado actualiza el puntaje del jugador y marca las cartas como emparejadas.
- **S_HOLD**: Si las cartas no coinciden, en este estado se mantienen abiertas por un breve intervalo antes de pasar al siguiente turno.
- **S_KEEP_TURN**: Mantiene el turno del jugador actual.
- **S_SWITCH_TURN**: Cambia al siguiente jugador, moviéndose a **S_IDLE** para que el otro jugador comience su turno.
- **S_GAME_END**: Este estado indica el final del juego cuando se han completado las 8 parejas.

Diagrama de la Máquina de Estados (FSM) Aquí se muestra un diagrama que ilustra cómo los estados interactúan entre sí:

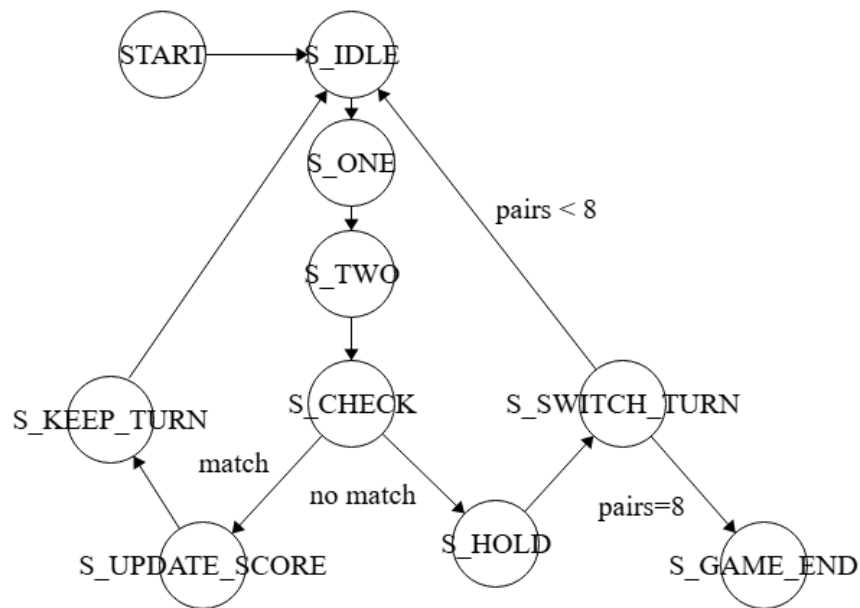


Figura 1: Diagrama de la máquina de estados para el juego Memoria.

1.2. Propuesta B: FSM Mealy orientada a eventos (compacta)

Idea. Reducir número de estados; ciertas salidas (abrir, cerrar, anotar) dependen tanto del estado como de la entrada (*evento*: selección válida, timeout, coincidencia inmediata).

Estados

- RESET, IDLE_SHOW, TURN_ARMED (espera eventos).
- ONE_OPEN (primera abierta), TWO_OPEN (segunda abierta).
- RESOLVE (compara y decide), GAME_END.

Diagrama de estados Mealy

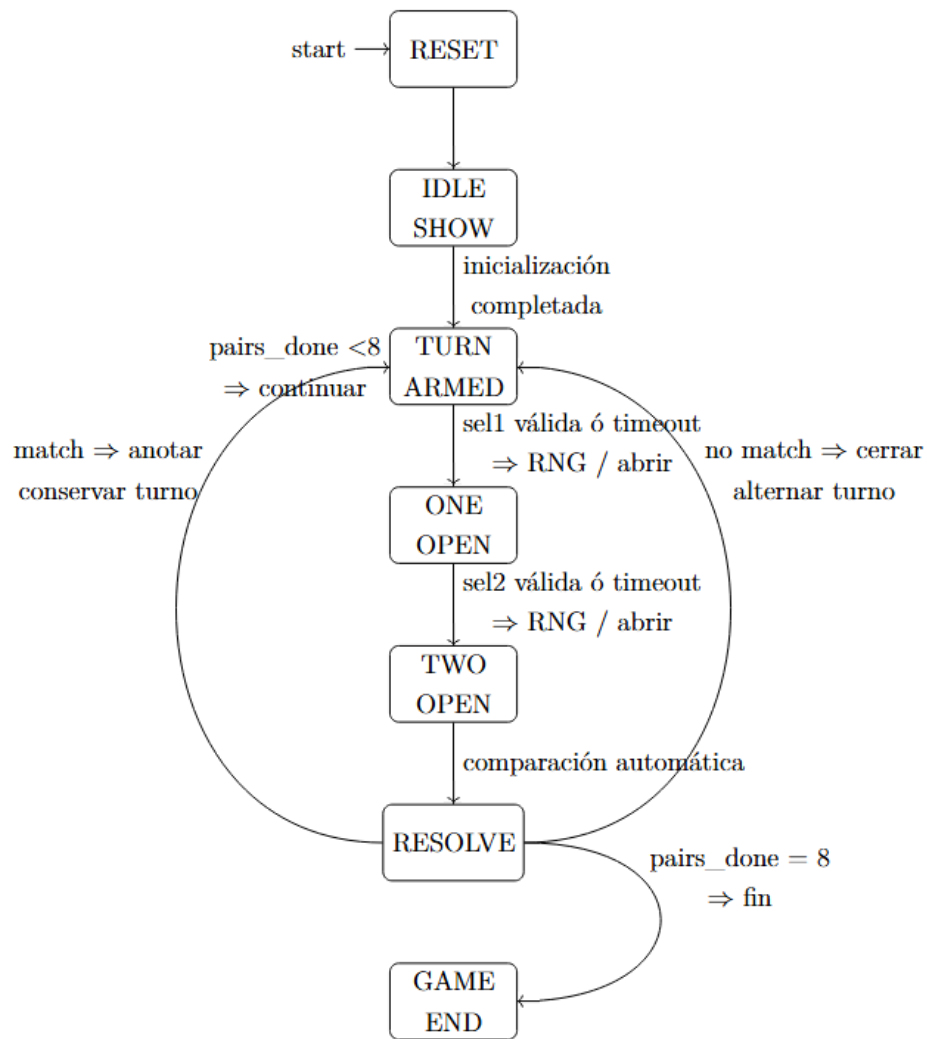


Figura 2: Diagrama Mealy para el juego Memoria.

2. Módulos auxiliares y señales

Temporizador de 15 s

Contador descendente (`load_timer`, `tick_1Hz` derivado de reloj del sistema). Señal `timeout=1` cuando llega a 0. Para 7 segmentos, exponer `bcd_tens`, `bcd_units`.

Debounce y One-shot

Para cada botón/switch: *debouncer* sincronizado al reloj y *one-shot* para un pulso limpio de un ciclo (`btn_pulse`).

RNG/LFSR

Registro LFSR (p. ej., 8 bits) para índice pseudoaleatorio `rand_idx`; en timeout se enmascaran cartas inválidas y se repite hasta obtener una válida.

Banco de cartas

Memoria de 16 entradas con estado por carta: *face-down* / *face-up* / *matched* y *id de símbolo*. Señales: `open_idx`, `close_pair`, `lock_matched`.

Puntaje y turno

Registro de turno (`p_turn`) y contadores de parejas `score_P1`, `score_P2`. Señal `pairs_done` (`==8`).

7 segmentos

Multiplexado por dígitos: tiempo restante y marcadores de parejas por jugador (mapeo configurable).

Controlador VGA 640×480 (modelo de estructura)

Contadores H/V, generación de `hsync/vsync`, ventana `visible`, mapeo (x,y) a celda (4×4), y *mux* de color por estado de carta (reverso, abierta con símbolo, par encontrado), coordenadas de resaltado según selección/turno.

3. Conclusión

Se documentaron dos propuestas de FSM para el juego *Memoria* y se seleccionó la variante Moore por fases por su claridad, facilidad de pruebas y control estable de salidas. La arquitectura modular permite cumplir los requerimientos del laboratorio (VGA, 7 segmentos, timeout, RNG) y facilita la integración en FPGA.