Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica

IEE2713 — Sistemas Digitales — 2' 2025

Interrogación N°2

Profesor: Carlos Milovic
23 de Octubre de 2025

Instrucciones

- **Tiempo Límite**: La duración total de esta evaluación es de 2:30 horas (150 minutos). No habrá extensión de tiempo.
- **Formato de Respuesta**: Responda cada pregunta en hojas (cuadernillos) separados. Por favor, inicie la respuesta a cada pregunta principal en una hoja distinta para facilitar la corrección.
- **Advertencia**: No se revisarán respuestas de preguntas distintas que se encuentren mezcladas en la misma hoja.

Pregunta 1

El departamento de Ingeniería Eléctrica se enfrenta a un desafío en la implementación de una nueva experiencia práctica de Sistemas Digitales. Debido a la escasez global de silicio, el inventario de componentes es crítico. Don Mario, encargado del pañol, reporta que se han agotado todos los flip-flops JK (FF-JK).

El profesor, confiando en sus sólidas competencias de diseño, le ha asignado la tarea de implementar (construir) la funcionalidad de un FF-JK utilizando únicamente el inventario restante.

Su tarea es diseñar un sistema que emule el comportamiento de un flip-flop JK síncrono (activado por flanco). Para ello, debe utilizar un flip-flop D (FF-D) como elemento base.

Deberá diseñar el circuito combinacional necesario que genera la señal de excitación para la entrada D del FF-D, basándose en las entradas de control J, K y el estado actual del flip-flop (salidas Q y/o $\bar{Q}$). El inventario es estricto y los componentes son de diferentes familias lógicas:

- 1 x Flip-Flop tipo D (Modelo: 74LS74 - Familia TTL)
- 4 x Compuertas NAND de 2 entradas (Modelo: 74LS00 - Familia TTL)
- 3 x Compuertas NOR de 2 entradas (Modelo: 74HC02 - Familia CMOS)

Entregables Requeridos:

1. La ecuación de excitación (o característica) óptima que define la entrada D en función de J, K y Q ($D = f(J, K, Q)$) para que el FF-D se comporte como un FF-JK.
2. El inventario disponible mezcla familias TTL y CMOS. Basándose en la tabla de voltajes adjunta, analice la compatibilidad de conexión:
   - ¿Puede la salida de una compuerta TTL (NAND 74LS00) conectarse directamente a la entrada de una compuerta CMOS (NOR 74HC02)? Justifique.
   - ¿Puede la salida de una compuerta CMOS (NOR 74HC02) conectarse directamente a la entrada de una TTL (NAND 74LS00 o FF-D 74LS74)? Justifique.

3. Dibuje el diagrama lógico final que implemente su diseño óptimo (Punto 1), utilizando únicamente los componentes del inventario y respetando las conclusiones de su análisis de interfaz (Punto 2). Su diseño debe ser óptimo, es decir, utilizar la mínima cantidad de compuertas lógicas del inventario para resolver el problema.

Table 1: Niveles de Voltaje Estándar (con $V_{CC} = 5V$)

Familia Lógica	$V_{IH(min)}$	$V_{IL(max)}$	$V_{OH(min)}$	$V_{OL(max)}$
74LS (TTL)	2.0 V	0.8 V	2.7 V	0.4 V
74HC (CMOS)	3.5 V	1.5 V	4.9 V	0.1 V

Pregunta 2

Usted ha sido contratado/a como ingeniero/a de hardware junior en "DigitalBet S.A.", una compañía líder en el desarrollo de sistemas de juego automatizados. Su primer proyecto es diseñar el módulo de decisión para un nuevo juego de mesa digital. El juego es simple: dos jugadores "lanzan" un dado digital de 6 caras cada uno (Dado 1 y Dado 0). El sistema debe encender una luz de "Victoria" ($z = 1$) si y sólo si el resultado del Dado 1 es estrictamente mayor que el del Dado 0 ($D1 > D0$).

El sistema recibe los valores de los dados y debe procesarlos. Usted deberá diseñar dos versiones del módulo para dos arquitecturas de sistema distintas.

Entregables Requeridos:

1. **Especificación de E/S (Entrada/Salida)**: Determine sus variables de entrada y el número de bits para cada una.
2. **Módulo Combinacional**: Los valores binarios de $D1$ y $D0$ están disponibles simultáneamente en un bus de datos paralelo.
   - Diseñe un circuito combinacional con salida $z = 1$ si y sólo si $D1 > D0$.
   - Entregue la expresión booleana minimizada para z.
   - Dibuje el diagrama lógico final.
3. **Módulo Secuencial**:En este modo, el módulo recibe los bits de $D1$ y $D0$ de forma serial. El sistema envía un par de bits (a_k, b_k) en cada ciclo de reloj, comenzando por el par de bits más significativos (MSB) y terminando con el par de bits menos significativos (LSB).
   - Rediseñe el comparador como un circuito secuencial (Máquina de Moore). Dibuje el diagrama de estados (Moore) que describa esta lógica.
   - Entregue las tablas de transición de estados y la tabla de salida.
   - Dibuje el circuito lógico final (implementado con Flip-Flops D y compuertas).
4. **Análisis Comparativo**: Compare brevemente ambas implementaciones.
   - ¿Cuál es más rápido en entregar un resultado y por qué?
   - ¿Cuál es más "simple" en términos de hardware?

Pregunta 3

Usted ha sido ascendido a ingeniero/a *mid-level* en "DigitalBet S.A." y se ha asignado el diseño de la lógica central para una nueva máquina de apuestas digital.

El juego se centra en un pozo de dinero digital que cambia según el lanzamiento de una moneda (Cara o Sello). Su tarea es diseñar el módulo de puntuación usando una Máquina de Estados Finitos (FSM). Este módulo debe rastrear el dinero en el pozo y controlar un marcador digital de un dígito.

Especificaciones de Funcionamiento: El módulo es un sistema síncrono. Las reglas del juego se evalúan en cada ciclo de reloj basándose en el resultado del lanzamiento de una moneda.

1. El sistema debe permanecer en un estado inactivo hasta que reciba una señal de START (o RESET).
2. Al recibir la señal de START, el juego se (re)inicia y el pozo se carga con un valor inicial de $3.
3. Mientras el juego está activo, se procesa un nuevo lanzamiento de moneda en cada ciclo:
   - Cara: Se añaden $2 al pozo (Pozo = Pozo + 2).
   - Sello: Se retiran $2 del pozo (Pozo = Pozo - 2).
4. El juego se detiene y entra en estado inactivo si ocurre uno de estos dos eventos:
   - "Bancarrota": Si el pozo contiene $1 y el resultado del lanzamiento es Sello.
   - "Jackpot": Si el pozo contiene $9 y el resultado es Cara.

Entregables Requeridos:

1. Dibuje el diagrama de estados que define al sistema utilizando la maquina de Mealy.
2. Su tarea es diseñar la lógica de salida para el segmento **e**.
   - Entregue las tablas de transición de estados y la tabla de salida.
   - Dibuje el circuito lógico final (óptimo) implementado con Flip-Flops D y compuertas.

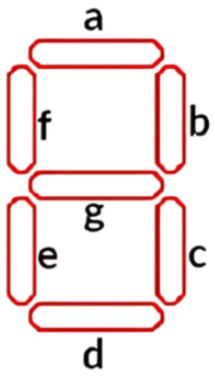


Figure 1: Diagrama display 7-segmentos.

Pregunta 4

Usted ha sido contratado como ingeniero/a de hardware *senior* en la empresa "Overflow Systems". Su primera tarea es auditar un módulo de lógica crucial (pay_code), diseñado por un ingeniero anterior que dejó la compañía sin la documentación adecuada.

Este módulo es el núcleo de las máquinas de pago de la compañía. Antes de que pueda ser reutilizado en la nueva generación de productos, la gerencia necesita un informe técnico completo sobre su funcionamiento exacto.

Analice el siguiente código (escrito en SystemVerilog) exactamente como está escrito. El ingeniero anterior era conocido por su estilo "peculiar", y cualquier inconsistencia o priorización lógica no documentada debe ser identificada y replicada.

Entregables Requeridos:

1. Identifique el tipo de Máquina de Estados Finitos (FSM) implementada (Moore o Mealy).
2. Describa en lenguaje natural de forma clara, precisa y concisa qué hace la máquina.
3. Dibuje el diagrama de estados que describe la máquina implementada en el código.
4. Entregue las tablas de transición de estados y la tabla de salida.
5. Comente de forma breve (1-2 líneas) qué sucede si las entradas m50 y m100 se activan exactamente en el mismo ciclo de reloj. ¿El diseño maneja esta condición? ¿Cómo?

```
module pay_code (
  input  logic clk,
  input  logic rst,
  input  logic m50,
  input  logic m100,
  output logic open
);

  typedef enum logic [1:0] { S0, S1, S2 } state_t;
  state_t state, state_nxt;
  logic   open_nxt;

  always_comb
    open_nxt = (state==S1  && m100) ||
               (state==S2 && m50);

  always_comb
    unique case (state)
      S0:    state_nxt = m100 ? S2 : (m50 ? S1  : S0);
      S1:   state_nxt = m100 ? S0   : (m50 ? S2 : S1);
      S2:  state_nxt = m50  ? S0   : (m100? S0   : S2);
      default: state_nxt = S0;
    endcase

  always_ff @(posedge clk)
    if (rst)       {state, open} <= {S0, 1'b0};
    else           {state, open} <= {state_nxt, open_nxt};

endmodule
```

Pauta de Corrección (60 pts por pregunta)

P1 — Adaptar FF-D → JK (60 pts)

1. **Derivación de la ecuación de excitación** $D = J\overline{Q} + \overline{K}\,Q$ (tabla/razonamiento JK→D y verificación de casos) ..... **20 pts**

2. **Compatibilidad de niveles TTL/CMOS (VIL/VIH, VOL/VOH) y conclusiones**
   Conclusión esperada: TTL→CMOS *no garantizado*; CMOS→TTL *sí* ..... **15 pts**

3. **Implementación con 4×NAND y 3×NOR** (Los 3 NOR se deben usar para implementar las capa superior de los ANDs, conectando directamente Q a uno de los ANDs. Para el OR e inversores de la capa inferior, se usan 3 NANDs (el 4to es un distractor)) ..... **25 pts**

P2 — Comparador de dados (combinacional + FSM) (60 pts)

(a) **Especificación E/S** (D1=$a_2a_1a_0$, D0=$b_2b_1b_0$, rango 1..6, z=1 si $D1>D0$) ..... **10 pts**

(b) **Solución combinacional jerárquica (MSB→LSB)**
Ejemplo aceptable (o equivalente):

$$g_2{=}a_2\overline{b_2},\ e_2{=}\overline{a_2 \oplus b_2},\ g_1{=}a_1\overline{b_1},\ e_1{=}\overline{a_1 \oplus b_1},\ g_0{=}a_0\overline{b_0},\ z{=}g_2 \vee (e_2g_1) \vee (e_2e_1g_0)$$

+ diagrama ..... **20 pts**

(c) **FSM Moore secuencial (lectura MSB→LSB)**
Estados $S_{\mathrm{EQ}}, S_{\mathrm{GT}}, S_{\mathrm{LT}}$; transición desde S_{EQ} según par (a_k, b_k); tablas y circuito ..... **20 pts**

(d) **Comentario comparativo** (simplicidad, lógica vs. FF, latencia/área) ..... **10 pts**

P3 — Juego con moneda (Mealy) + segmento ee (60 pts)

(a) **FSM Mealy conforme a reglas** (estados de monto alcanzable, +2/–2, condiciones de término y opcional reinicio – No bajar puntaje si no se especifican estados o condiciones si se apreta START o RESET en medio del juego. Tampoco considerar el mostrar -1.) ..... **30 pts**

(b) **Tablas/ecuaciones** de próximo estado y salidas para el segmento E: $y_e = 0$. ..... **15 pts**

(b) **Lógica del segmento e (dígito LSD)**
Basta con conectar a tierra ..... **15 pts**

P4 — Ingeniería inversa de `pay_exact_150_simple` (60 pts)

(a) **Tipo de máquina** (esperado: Mealy) ..... **10 pts**

(b) **Descripción funcional clara**
Describir en palabras que es una maquina de pagos que recibe solo el monto exacto; sobrepago vuelve a $S0$ sin abrir ..... **15 pts**

(c) **Diagrama de estados** (condiciones en aristas; marcar dónde *open*=1) ..... **15 pts**

(d) **Tablas de transición y Ecuación/es (Mealy)** $open = (state{=}S1 \wedge m100) \vee (state{=}S2 \wedge m50)$ **15 pts**

(e) **Manejo de bugs** Como las entradas se manejan con la logica combinacional, solo se detectará la última activa antes del flanco de subida del clock.
5 pts

Observaciones generales de corrección

- Se aceptan variaciones de las máquinas de estado y los circuitos, siempre que tiendan a ser óptimos. Penalizar estados innecesarios o reducciones simples no hechas.