

Examen Virtual de Organización del computador Modalidad Manuscrito

Por favor firmar todas las ojas y Se les recuerda a los estudiantes que, según la resolución RD-2020-197-E-UNC-DEC#FAMAF, en el examen en la modalidad manuscrito, el/la estudiante, deberá firmar todas las hojas de su examen antes de digitalizarlo y enviarlo para su corrección. Al final del mismo deberá introducir la leyenda "Por la presente declaro que la resolución de este examen es obra de mi exclusiva autoría y respetando las pautas y criterios fijados en los enunciados. Asimismo declaro conocer el régimen de infracción de los estudiantes cuyo texto ordenado se encuentra en el apéndice de la Res. Rec. 1554/2018", con una foto de su Documento Nacional de Identidad, ocultando su número de trámite, en carácter de Declaración Jurada.

Enviar por email apenas finalizado el examen a pablo.ferreyra@unc.edu.ar

1)

a) Expresar el número 10111100100110101000110100110000 en decimal considerando que se encuentra en formato IEEE754. b) Mostrar el contenido de los distintos campos del formato en binario.

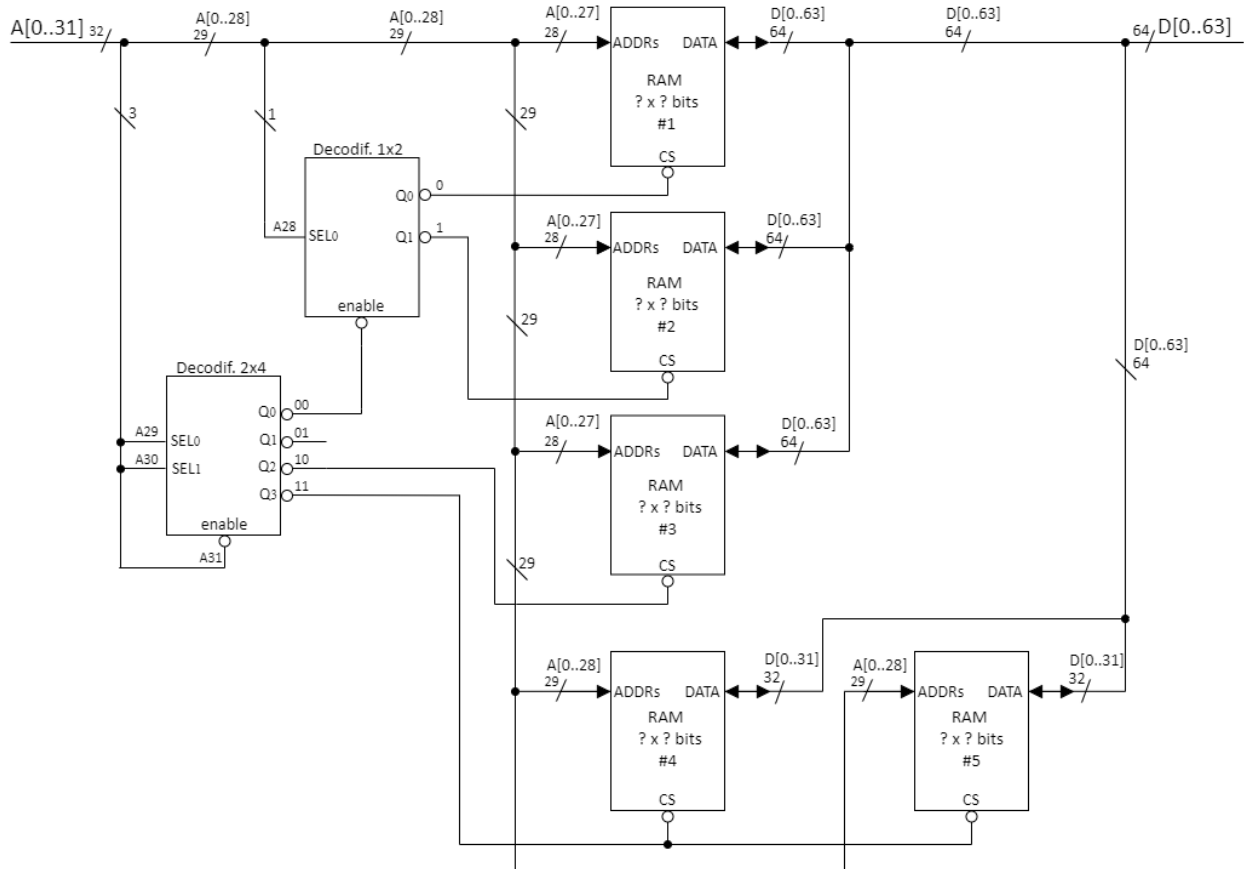
2)

Diseñar un circuito de registros de 4 bits que cumpla con el siguiente comportamiento:

00	Salida en paralelo
01	Retención del dato
10	Reset
11	Entrada en paralelo

Cuando el circuito debe retener los datos, cada flip-flop debe mantener en su salida el mismo dato aunque la entrada se modifique. En la salida en paralelo, los datos almacenados salen por los pines `parallel_out[0-3]` y el dato se retiene sin modificarse. En la entrada en paralelo, ingresan datos nuevos por el pin `parallel_in[0-3]` y las salidas (`parallel_out[0-3]`) debe permanecer en cero. Finalmente, en *reset* todos los datos almacenados son reemplazados por unos en el mismo ciclo de reloj.

3) Basados en el sistema de memoria de la figura:



Se pide:

- Indicar el tamaño de cada bloque de memoria (RAM #1, #1, #2, #3, #4, #5) expresado en cantidad de palabras x ancho de palabra en bits.
- Dibujar el mapa de memoria implementado, indicando la dirección de inicio y final de cada bloque.
- Indicar si esta implementación presenta posiciones imagen o espejo. De ser así, indicar su ubicación en el mapa y a que bloque que real corresponden.
- Calcular la capacidad total (expresada en bytes) de memoria **implementado** (no se consideran las posiciones imagen)

4)

Diseñar un sistema secuencial de dos bits de entrada (B, A) y una salida (LOCK), el cual se utiliza para controlar la apertura de un candado. Las entradas se codifican de la siguiente manera:

Codigo	B	A
No permitido	0	0

'A'	0	1
'B'	1	0
No permitido	1	1

El sistema debe iniciar con LOCK = 1. La secuencia de apertura es: 'A'-'A'-'B'-'A'. Si esta secuencia se ingresa correctamente de manera consecutiva, la señal de salida se desbloquea LOCK =0. Para cualquier otra secuencia de 'A' y 'B' el sistema debe permanecer bloqueado.

Una vez desbloqueado el sistema, el mismo se volverá a bloquear solo ante la recepción de 'B'-'B' en forma consecutiva.

La combinación de entrada de los bit A y B simultáneamente en 0 o 1 no esta permitida, y su aparición representa un error en el sistema. De registrarse alguno de estos dos casos, el sistema debe ir inmediatamente a un estado de error (con LOCK = 1) del cual se sale solo ante la ocurrencia de la secuencia 'A'-'B' de manera consecutiva o no.

Se pide:

- Diagrama de estados
- Tablas de transición de estados y de salida
- Simplificar mediante diagramas de Karnaugh todas las funciones que lo permitan
- Implementar los circuitos combinacionales de transición de estados y de salida utilizando compuertas lógicas de cualquier tipo y número de entradas.

5) Dado el siguiente código:

```
.data
    info: .dword 0x4142434445464748,0x494a4b4c4d4e4f50,
           0x4142434445464748,0x494a4b4c4d4e4f50
.text
    LDR X2, =info           // Get info base address
    ADDI X12, X2, 0x1F
    ADD X8, XZR, XZR
a:  ADD X7, X2, X8
    LDURB W20, [X7]
    LDURB W21, [X12]
    STURB W20, [X12]
    STURB W21, [X7]
    SUBI X12, X12, #1
    ADDI X8, X8, #1
    SUBI X9, X8, #16
    CBNZ X9, a
end: b end
```

- Indicar el estado de la memoria de datos cuando el programa inicia y el estado de la memoria cuando el programa termina.

(Escribir los datos en formato arreglo de bytes)

- ¿Cuántas instrucciones ejecuta el programa antes del label "end"?
- ¿Qué hace el programa? (Traducir de LEGv8 a C)