

Arquitectura de Computadores

I1

28 septiembre 2021

Inicio: 8.30 am

Entrega: hasta las 9.00 pm

0) Responde esta pregunta de manera manuscrita.

a) Nombre completo y número de alumno: _____

b) Me comprometo a no preguntar ni responder dudas de esta prueba, ya sea directa o indirectamente, a nadie que no sea parte del equipo docente del curso. **Firma:** _____

Responde sólo TRES de las siguientes cuatro preguntas

1)

Se realizan las primeras elecciones de alcalde del pequeño pueblo de ALUpolis y para estas se utiliza un moderno sistema computacional. El ayuntamiento de ALUpolis tiene una sala de votación con dos computadores, uno por cada candidato, Carlos y Valeria. Para emitir el voto, cada habitante de ALUpolis debe ingresar su rut y huella digital en el computador del candidato preferido, el cual lleva una cuenta de los votos emitidos en ese computador. Al emitir el voto en uno de los computadores, el rut del votante se bloquea en el otro computador. Estos computadores manejan números de 16 bits en complemento de 2. ALUpolis tiene una población de votantes de 1830 personas, pero no todas votan.

Llega el momento del conteo y los resultados son inquietantes:

- computador Carlos ProgramCounter: 3051_{10}
- computador Valeria Multiplexor: $1000\ 0011\ 0010\ 1010_2$

Después de un análisis exhaustivo, tú descubres que cada computador tiene un (y solo un) bit cambiado (de 0 a 1 o de 1 a 0) que arruina su conteo.

- ¿Cuál bit es erróneo en cada computador? ¿Cuáles son los resultados reales de la votación y quién, Carlos o Valeria, ganó la elección de alcalde de ALUpolis?
- Diseña un solo circuito aplicable a ambos computadores a partir de los bits del número de votos (erróneo) que solucione estos errores.

El estándar IEEE 754 utiliza números positivos para el exponente y para la fracción al representar un número de punto flotante.

- ¿Qué ventajas y desventajas nos daría utilizar complemento de 2 en vez de números estrictamente positivos para el exponente y la fracción?

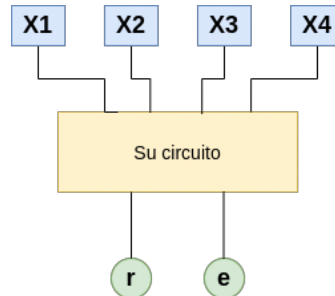
Arquitectura de Computadores

I1

28 septiembre 2021

2)

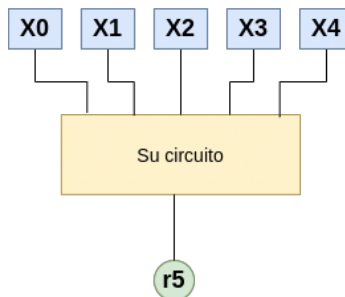
- a) Diseña un circuito digital combinacional que dado un input de 4 bits, $X1, \dots, X4$, produce un output, r , que corresponde al valor que más se repite (0 o 1) entre los inputs. El output e sólo se pone en 1 si hay empate entre los 4 inputs, en cuyo caso el valor de r es el que tú prefieras.



Por ejemplo,

inputs	r	e
1 0 1 1	1	0
0 0 0 1	0	0
1 0 1 0	x	1

- b) Ahora agregamos un quinto input, $X0$, como se muestra en la figura. Modifica tu circuito de a) tal que el nuevo único output $r5$ sea el valor (0 o 1) que más se repite entre los 5 bits del input. Puedes suponer que tu circuito de a) es correcto y que cuentas con los valores correctos de r y e .



Arquitectura de Computadores

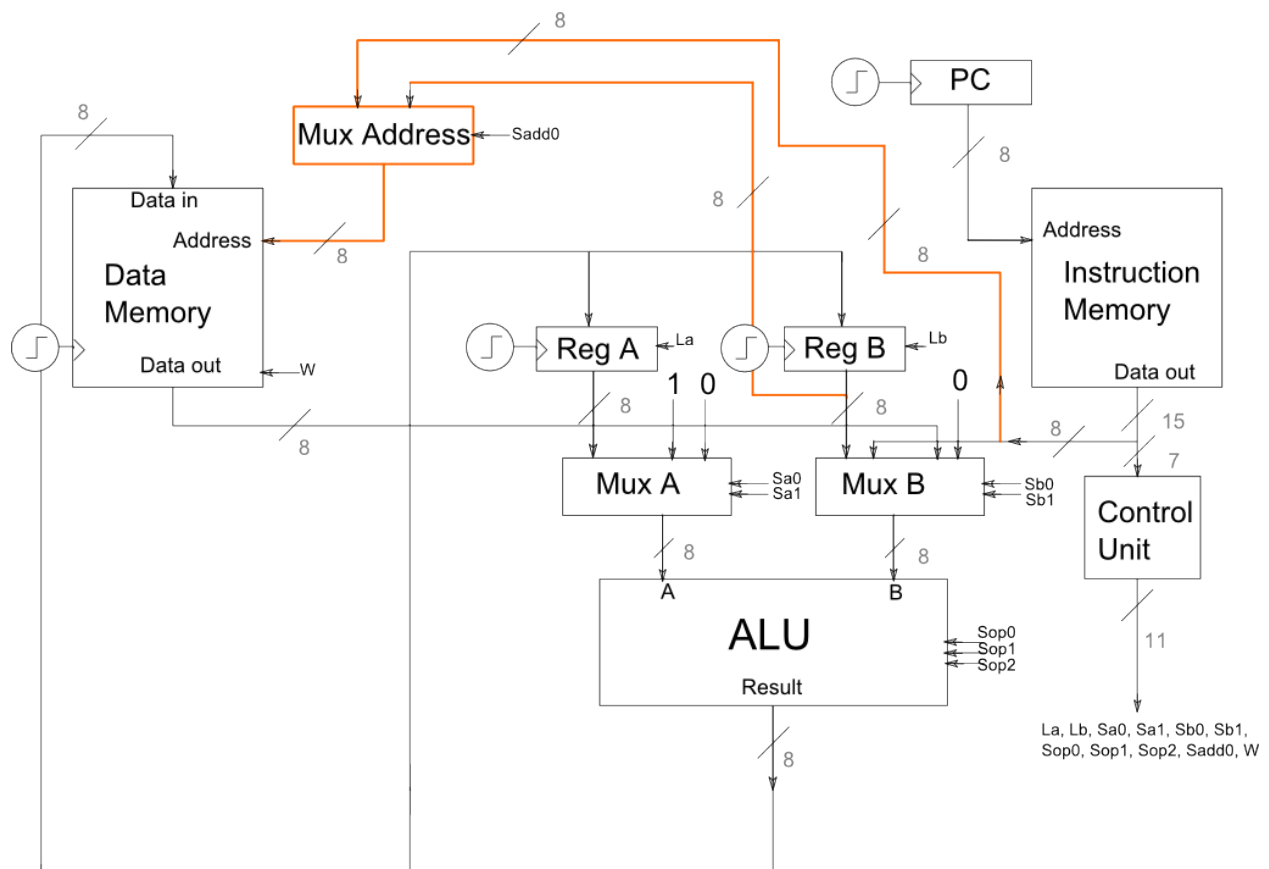
I1

28 septiembre 2021

- 3) Considera la siguiente configuración del computador básico. Como vimos en clases, esta configuración permite tener acceso (lectura y escritura) a datos almacenados en la *Data Memory* usando direccionamiento directo y direccionamiento indirecto.

Queremos agregar a esta configuración la capacidad de tener acceso a la *Data Memory* a través de una *dirección base* más un *desplazamiento*. Considera que la dirección base se almacena en el registro *B* y el desplazamiento se almacena en un nuevo registro especializado, *D*.

- a) Dibuja y explica de manera clara y precisa los cambios necesarios en el circuito del computador básico: destaca claramente componentes, cables, buses de datos nuevos/modificados y cualquier otro cambio que hagas. La ALU no cambia en cuanto a que sigue teniendo dos inputs, *A* y *B*, y un output, *Result*, y que todos son de 8 bits.
- b) Especifica las nuevas instrucciones en cuanto a opcodes, señales de control, y operación, similarmente a las otras instrucciones en el set y de manera coherente con tu dibujo en a).



Arquitectura de Computadores

I1

28 septiembre 2021

4) Usando exclusivamente el lenguaje *assembly* del computador básico (no en RISC-V):

- a) Escribe un programa que multiplique dos números enteros (el multiplicando p y el multiplicador q) de la siguiente manera: multiplica p por la potencia de 2 más grande, digamos 2^r , que sea menor o igual que q ; llamemos m a este resultado parcial. Si $q = 2^r$, entonces m es el resultado final; de lo contrario, tu programa tiene que sumar p a m tantas veces ($q - 2^r$) como haga falta. Tu programa debe determinar r .

Por ejemplo,

p	q	r	2^r	m	$q - 2^r$
3	5	2	4	12	1 → todavía hay que sumar 3 a m una vez más
3	8	3	8	24	0
5	15	3	8	40	7 → todavía hay que sumar 5 a m siete veces más

- b) Considera que tu programa de a) está correcto y corresponde a una función llamada **Mult**. Escribe un programa en el *assembly* del computador básico (no en RISC-V) que calcule el factorial de un número s , utilizando la función **Mult**.