PRUEBA ELIMINATORIA. MODELO A

ESTRUCTURA DE COMPUTADORES

Duración: 90 minutos.

1. (0.1p) Añade al final de la siguiente secuencia de bits el bit de paridad par que corresponda: 1100011.
   1. Pasa la respuesta de binario a hexadecimal.

**110001102 → C616**

* 1. ¿Cuál es su correspondiente valor decimal?

**12\*16+6=19810**

1. (0.3p) Un determinado receptor ha recibido la siguiente palabra: 1010000
   1. ¿Se ha dado algún error entre los procesos de envío y recepción?
   2. ¿Cuál es el dato enviado?

**C1= 1 xor 1 xor 0 xor 0 = 0 Error en la posición 2**

**C2= 0 xor 1 xor 0 xor 0 = 1 (bit redundante de paridad)**

**C4= 0 xor 0 xor 0 xor 0 = 0**

**Dato enviado → 1000**

1. (0.2p) ¿Cuál es la limitación del criterio de bit de paridad comparado con la codificación *Hamming*? ¿Y cuál es su ventaja?

**La limitación del criterio del bit de paridad pasa por aun detectándose errores simples en la transmisión de datos no pueden corregirse porque no se sabe en qué posición ha sucedido. Sin embargo, el criterio del bit de paridad es bastante más sencillo de implementar.**

1. En la siguiente operación: if R2<0, salto + 1Dh, ¿la situación de qué *flag* se analiza (0.1p)? Si se cumple la condición, ¿cómo se actualiza el Contador de Programa (PC) (0.1p)? ¿Y si no se cumple la condición (0.1p)?

**Se analiza la situación del flag de signo S. Si la condición se cumple, el PC deberá actualizarse con la nueva dirección (PC=PC+Offset). Si no se cumple la condición, el PC seguirá actualizándose sabiendo que las instrucciones se hallan seguidas (PC=PC+1).**

1. La Ruta de Datos es gobernada por los valores de ciertas señales de selección, ¿qué forma la combinación de dichas señales? (0.1p)

**La palabra de control.**

1. ¿Cuál es el cometido de la Unidad de Control? (0.2p)

**El cometido de la unidad de control es el de a partir de una instrucción de memoria, codificada en lenguaje máquina, obtener la palabra de control.**

1. Un computador tiene un formato de instrucción con un “campo de operación” de 6 bits y una unidad de control microprogramada de ciclo múltiple. Si cada instrucción se ejecuta en cuatro ciclos de reloj, ¿cuántos bits necesita el bus de direcciones de la memoria de control? (0.3p)

**Campo de operación de 6 bits → 26 = 64 operaciones.**

**Si cada instrucción se ejecuta en 4 ciclos de reloj, hacen falta en total 64 x 4 = 256 operaciones. 28 = 256, con lo que el bus de direcciones de la memoria de control necesita 8 bits.**

**26 x 22 = 28 → 8 bits.**

1. (0.4p) ¿Se ejecutará correctamente la siguiente secuencia de instrucciones en una canalización dividida en las cuatro fases IF / DOF / EX / WB? ¿Por qué? Realiza un dibujo para apoyar la explicación.

mov R1, R2

add R2, R5, 3

mov R1, R2

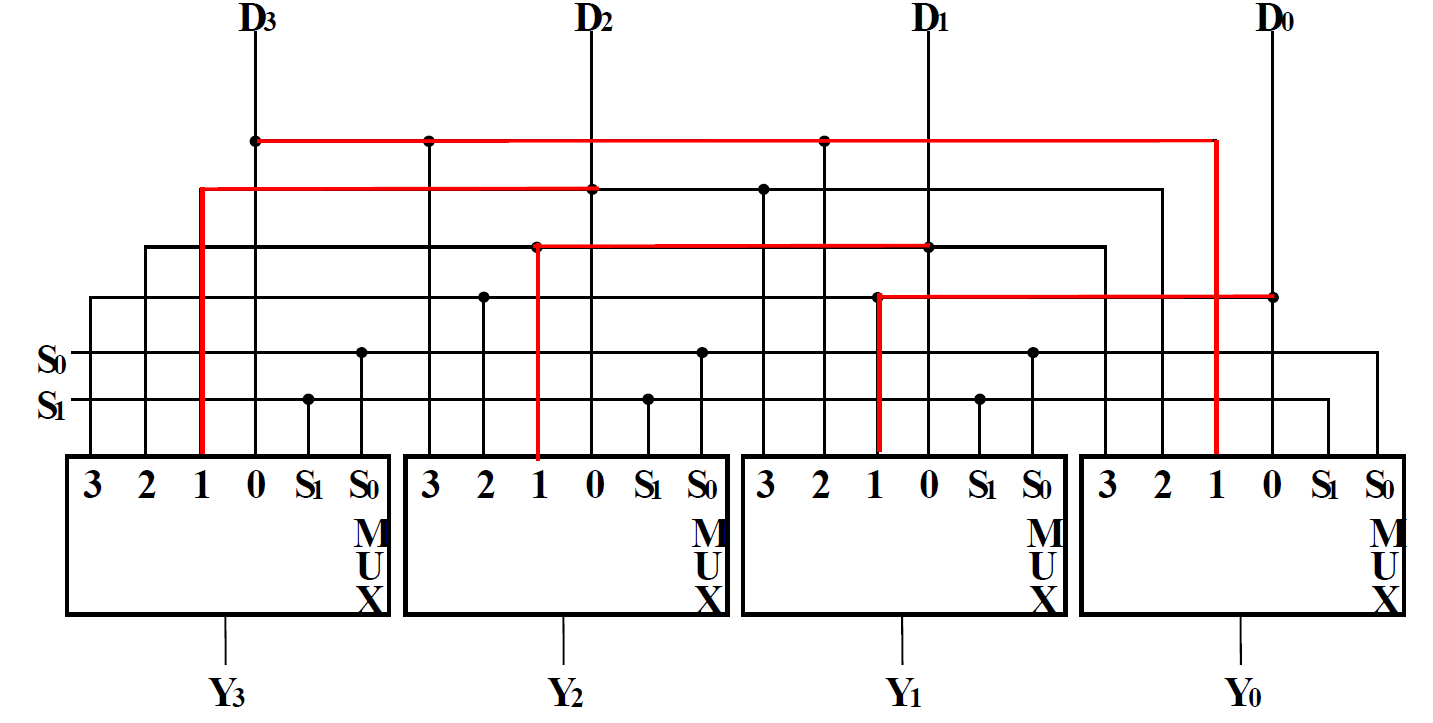
**Entre las dos primeras instrucciones no hay riesgo de datos porque en la primera instrucción no se modifica el valor del registro R2. Sin embargo, la instrucción 3, no se ejecutaría correctamente porque el registro R2 se actualizaría en su ciclo de WB, un ciclo después de que por canalización la tercera instrucción haya leído su valor para pasarlo al registro R1.**

1. ¿Por qué son necesarias las señales de selección TA, TB, TD y AX, BX, DX en una Unidad de Control de ciclo múltiple? (0.2p) ¿Por qué no aparecen en una Unidad de Control cableada de ciclo único? (0.1p) En una Unidad de Control microprogramada, ¿qué cambios son necesarios para poder cambiar una instrucción del juego de instrucciones? (0.1p)

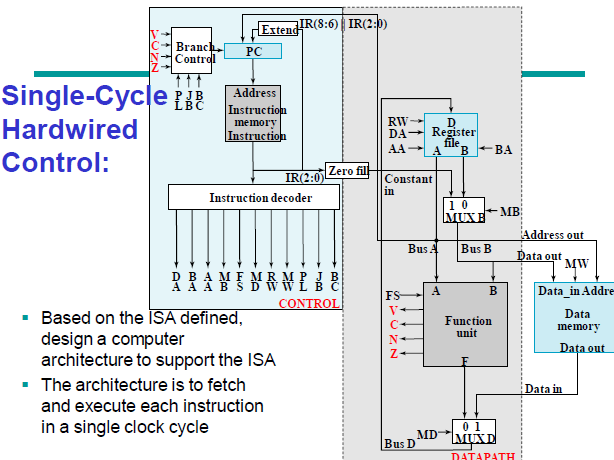
**Las señales TA, TB, TD, AX, BX, DX son necesarias para poder gestionar la utilización de los registros ocultos R8...R11 en los campos de destino, bus A y bus B. Estos registros ocultos, accesibles solo por el firmware, son necesarios porque una unidad de control microprogramada pasa muchas veces por la UAL y tiene mucha más carga computacional que una unidad de control cableada.**

**Para cambiar una instrucción del juego de instrucciones haría falta acceder al microcódigo y cambiar el microprograma (las microinstrucciones) relacionado con dicha instrucción.**

1. ¿Qué valor toma Y[3..0] si D (1001) y S (10)? (0.2p) **0110**



1. Basándote en la siguiente imagen, explica el funcionamiento de la UCP; contesta, además, a las siguientes preguntas:
   1. ¿Por qué hace falta la cajetilla ‘Zero fill’? (0.1p) **Porque a partir de un campo de 3 bits, se debe extender el signo (rellenar con 0 porque el operando es siempre positivo) para igualar en tamaño al tamaño de datos que maneja la UAL.**
   2. ¿Por qué hace falta la cajetilla ‘Extend’? (0.1p) **Porque a partir de un campo de 6 bits, se debe extender el signo, con 0s si el MSB es 0 y con 1s si el MSB es 1, para sin modificar el valor del offset, se pueda igualar en tamaño a la longitud del PC, para actualizarlo en caso de que haya saltos o bifurcaciones.**
   3. ¿Cuál es la diferencia entre ambas? (0.1p) **La diferencia entre ambos es que, si bien extienden el signo para no cambiar el número, Zero Fill cuenta solo con valores positivos y Extend cuenta con valores positivos (rellenar con 0) y valores negativos (rellenar con 1). Siendo además Zero Fill una cajetilla que se dirige hacia la ruta de datos y Extend hacia la actualización del registro PC.**



1. ¿Cuántas señales de selección necesita la unidad que realiza las siguientes operaciones? (0.1p). Diseña la lógica de dicha unidad (0.5p). ¿De qué tipo de unidad se trata? (0.1p)

A+1

A-1

A+B

A-B

**4 operaciones, 2 señales de selección. Se trata de una unidad aritmética.**

1. Una Ruta de Datos está formada por 5 elementos (A, B, C, D y E) en anillo. Cada elemento introduce el siguiente retardo: A (2 ns), B (2 ns), C (2 ns), D (2 ns) y E (2 ns):
   1. (0.1p) ¿Cuál es la frecuencia máxima del sistema?

**fmax= 1/10ns = 100 MHz**

* 1. (0.1p) ¿Cuánto tiempo necesita el sistema para ejecutar 6 instrucciones?

**6 x 10 ns = 60 ns**

* 1. Disponemos de dos registros para implementar la canalización; cada uno de ellos añade un nanosegundo de retardo. ¿Dónde pondrías dichos registros para sacar el mayor rendimiento al sistema? (0.1p) ¿Por qué? (0.1p)

**En cualquiera de los lugares en los que los 2 bloques queden lo más equitativos posible en tiempo; así, un bloque deberá abarcar 2 elementos y el otro bloque 3 elementos. Como los elementos son de misma duración, no es relevante siempre que se cumpla 2/3.**

* 1. ¿Cuál es ahora la frecuencia máxima del sistema? (0.1p)

**fmax = 1/7 ns = 1/7 GHz≈ 140 MHz**

* 1. ¿Cuántos nanosegundos hacen falta para ejecutar 6 instrucciones mediante la canalización (*pipe-line*)? (0.1p)

**7 x 7 ns = 49 ns**

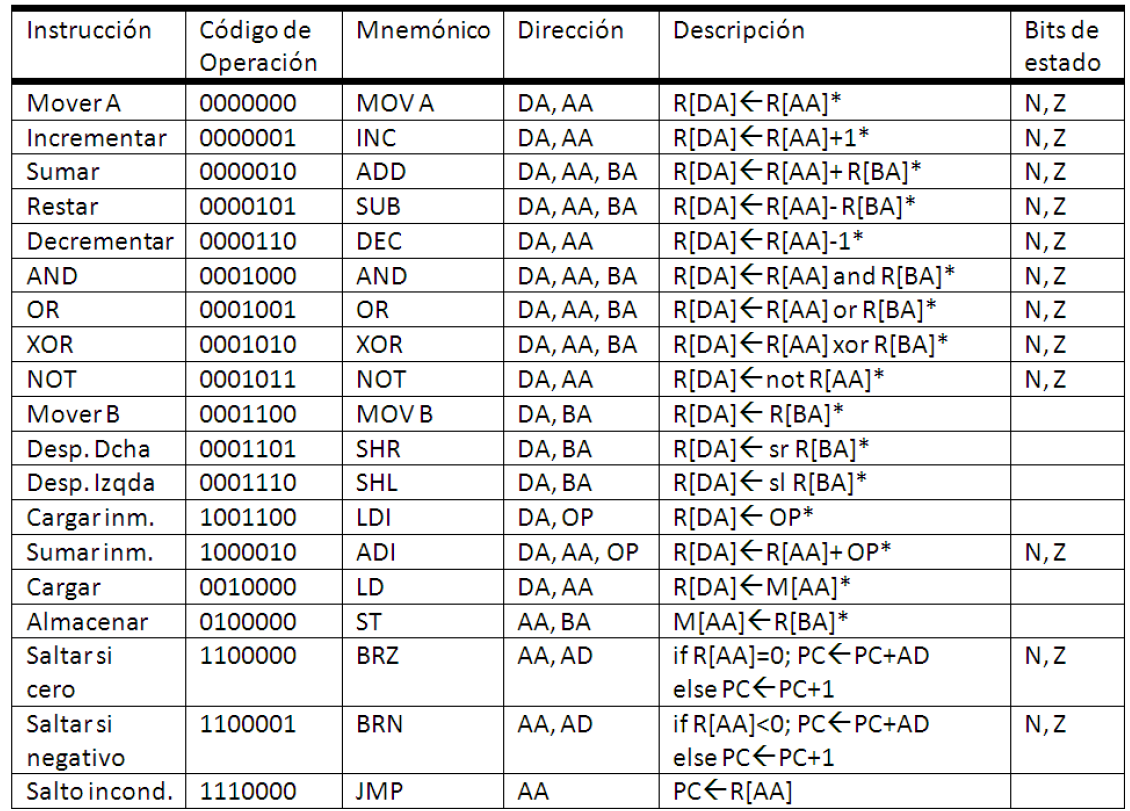
1. Basándote en la información de la tabla

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Opcode | | | | | | | DA | | | AA | | | BA | | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |

1. ¿Qué instrucción se va a ejecutar? (0.2p)

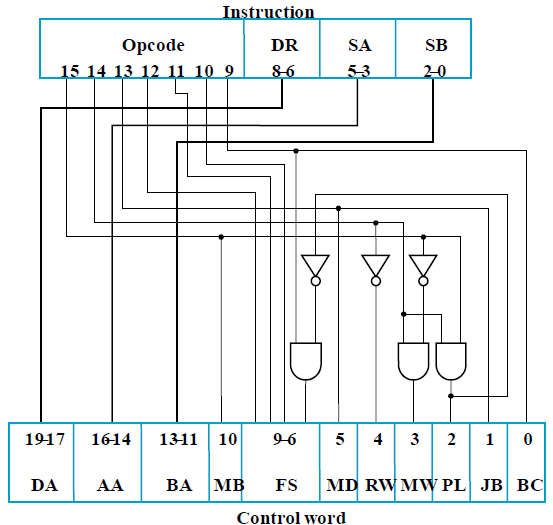
**Suma inmediata, R3 ← R3 + 3**

**R3 = 74h y los demás registros, quedan igual.**

Información útil:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R1 | 02h |  | R2 | 00h |
|  |  |  |  |  |
| R3 | 71h |  | R4 | D4h |
|  |  |  |  |  |
| R5 | 00h |  | R6 | 53h |
|  |  |  |  |  |
| R7 | 01h |  | R8 | A8h |
|  |  |  |  |  |

1. ¿Qué palabra de control le corresponde a esta instrucción en código máquina? (0.2p)



Palabra de control en hexadecimal:

**6DC90h**