**TAREA No. 1**

**Nombre 1: Aylin Rojas Orrego RUT: 20.834.561-3**

**Nombre 2: Valentina Muñoz Rebanal RUT: 21.074.294-8**

**Problema**

Un determinado microprocesador usa códigos de operación (se identifican como Op-Code), que es una única combinación de cuatro bit, para especificar cada una de las posibles instrucciones de máquina de esta CPU.

Los códigos de operación son recibidos por un decodificador (decoder), el cual debe decidir qué es lo que mandata la instrucción de máquina y generar las señales de control pertinentes.

Esta decodificación es realizada generalmente en varios pasos con los Op-Code, siendo primero clasificados en tipos básicos. Así algunas operaciones requerirán que alguna palabra sea leída desde la memoria en cuyo caso el decoder debe generar la señal RO (Read Operando).

Otras operaciones requerirán que una palabra de computador sea almacenada en memoria, en cuyo caso el decoder debe generar la señal WO (Write Operando).

Otras categorías de operaciones son JUMP (J), Iterativa (IT), y Entrada/Salida (IO). Estas categorías no son mutuamente exclusivas, es decir una operación puede caer en más de una categoría.

Las operaciones que caen en cada una de las categorías son listadas en la tabla mostrada a continuación:

|  |  |  |
| --- | --- | --- |
| Categoría | Señal | Operación |
| Lee operando | RO | 2-10, 12, 13,15 |
| Escribe operando | WO | 4-6, 9,11 |
| JUMP (saltar) | J | 1-3,5 |
| Iterativo | IT | 7,8,14 |
| Entrada/Salida | IO | 13, 15 |

1.- Obtenga una función mínima SoP (mintérminos) para generar las señales de control RO, WO, J, IT, IO, en respuesta al Op-Code de cuatro bit. Identifique los bits del Op-Code como C3, C2, C1, C0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C3** | **C2** | **C1** | **C0** | **BIN** | **RO** | **WO** | **J** | **IT** | **IO** |
| 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | **1** | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | **2** | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | **3** | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | **4** | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | **5** | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | **6** | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | **7** | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | **8** | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | **9** | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | **10** | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | **11** | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | **12** | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | **13** | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | **14** | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | **15** | 1 | 0 | 0 | 0 | 1 |

**Read Operando:**

0 0 1 1 1 1 0 0

0 0 1 0 1 1 0 1

0 1 1 1 1 0 0 0

0 1 1 0 1 0 0 1

C3 X C1 X C3 X C1 X

0 1 0 1 1 0 0 0

0 1 1 1 1 0 1 0

1 1 0 1 C3 C2 X C0

1 1 1 1

X C2 X C0

0 1 0 0

0 1 0 1

1 1 0 0

1 1 0 1

X C2 C1 X

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C3**  **C2**  **C1**  **C0** | **00** | **01** | **11** | **10** |
| **00** |  | 1 | 1 | 1 |
| **01** |  | 1 | 1 | 1 |
| **11** | 1 | 1 | 1 |  |
| **10** | 1 | 1 |  | 1 |

**Write Operando:**

0 1 0 0

0 1 0 1

C3 C2 C1 X

1 0 0 1

1 0 1 1

C3 C2 X C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C3**  **C2**  **C1**  **C0** | **00** | **01** | **11** | **10** |
| **00** |  | 1 |  |  |
| **01** |  | 1 |  |  |
| **11** |  |  |  | 1 |
| **10** |  |  |  | 1 |

**Jump:**

0 0 0 1

0 1 0 1

C3 X C1 C0

0 0 1 0

0 0 1 1

C3 C2 C1 X

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2**  **C1**  **C0** | **00** | **01** | **11** | **10** |
| **00** |  |  |  |  |
| **01** | 1 | 1 |  |  |
| 11 | 1 |  |  |  |
| 10 | 1 |  |  |  |

**Iterative:** *(se dejó tal cual ya que no había forma de minimizar con el M-K)*

**Input / Output:**

1 1 0 1

1 1 1 1

C3 C2 X C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C3**  **C2**  **C1**  **C0** | **00** | **01** | **11** | **10** |
| **00** |  |  |  |  |
| **01** |  |  | 1 |  |
| **11** |  |  | 1 |  |
| **10** |  |  |  |  |

2.- Implemente el circuito resultante para la señal J en TinkerCad.

Un reloj en la noche

Descripción generada automáticamente con confianza baja

En este Diagrama podemos apreciar el cómo debe funcionar nuestro circuito con cada compuerta lógica. Primero he de explicar el porqué usé un M-K de 4 bits en vez de tres ya que todos los bits correspondientes a C3 son 0’s, esto no sería lo correcto ya que los mintérminos que buscamos serían distintos ya que la obviar C3 tanto 0001 como 1001 contarían como mintermino, siendo que en la descripción del problema 1001 no es parte de estos.

Con esto aclarado pasaré a explicar cómo se realizó el circuito y cómo funciona:

Imagen que contiene Diagrama

Descripción generada automáticamente

Nuestras entradas C3, C2, C1 y C0 se verán reflejadas en estos 4 botones o palancas pasando cada uno a un color específico; C3=blanco, C2=celeste, C1=naranjo y C0=verde.

Lo primero que haremos será negar C3, C2 y C1 como nos indica la ecuación de J, por lo que nuestros nuevos valores serán; =gris, =azul y =café.

Luego cómo en cada paréntesis de la ecuación hay tres valores que se multiplican *(o sea una compuerta AND de tres entradas)* uniremos estos tres cables, donde el paréntesis lo representa el cable morado y en paralelo el paréntesis se representa con el cable rosado.

Finalmente, estos dos cables son unidos en una compuerta OR *(debido a la suma en la ecuación)*, resultando en el cable amarillo el cual está unido a un led con su debida resistencia para probar que la salida sea correcta.

Solo en caso de que desee probar el TinkerCard adjunto el link:

<https://www.tinkercad.com/things/0orrwQSjFSL-sizzling-snaget-albar/editel?sharecode=NcoHwl5Mmenl6PkbT4MEor4fyRqzBcOVmGn8YycnNmc>