Organización y Arquitectura de Computadoras

Facultad de Ciencias, UNAM

José Ethan Ortega González: 316088327
Etzael Iván Sosa Hedding: 316259305

1. Procedimiento

1. Construye las 3 compuertas básicas: Not, Or y And. Para corroborar la funcionalidad de sus compuertas, deberán construir circuitos para simular las siguientes funciones. Sólo puedes hacer uso de fuentes de alimentación power y ground, transistores, resistencias y pines de entrada y salida.

1. $A \leftrightarrow B$
2. $A \oplus B$

Solución:

- **NOT:** Para este circuito pensamos en dejar pasar la corriente de la fuente en un caso y bloquearla en el otro, auxiliándonos de un elemento tierra para indicar el 0. Utilizamos un transistor de tipo P que deja pasar la corriente cuando recibe un 0 y uno de tipo N que está abierto cuando recibe un 1. Es claro que necesitamos uno de cada tipo para que siempre haya uno abierto y deje pasar la información deseada.
- **AND:** Como necesitamos que ambas entradas sean 1, ponemos 2 transistores de tipo N para que dejen pasar la corriente de la fuente de poder al ser 1 ambos pues si no fueran iguales no pasaría la corriente ni tampoco pueden ser ambos de tipo P porque esas se cierran al ser 1. Sin embargo, también es necesario poner una resistencia para que por defecto tengamos un 0, ya que si ambas entradas son 0 no pasa nada y estaría indeterminada la salida.
- **OR:** Aquí necesitamos 2 transistores de tipo N para que dejen pasar la corriente cuando sean 1 pero necesitamos un transistor con 0 para que por defecto (cuando ambas entradas sean 0) se muestre un 0. Aquí con que una entrada sea 1 se permite pasar la corriente, en AND se tenía que dejar pasar la corriente por un transistor y luego por otro, aquí no ocurre eso, son independientes.
- **XOR:** Se evalúa a 1 cuando las entradas son distintas, por lo que utilizamos la definición que es la (negación de A y B) y (A o B) por lo que utilizamos los anteriores circuitos.
- **Equivalencia:** Como la tabla de verdad de XOR y la equivalencia son opuestas, basta con negar XOR.

2. Construye un circuito que resuelva las situaciones que se piden. Debes hacer uso de las compuertas que construiste en el ejercicio anterior.

1. Indicar si un número n es primo, con $n \in 0, \ldots, 15$.
2. Simular la siguiente función y reduce mediante mintérminos y maxtérminos. Agrega ambos circuitos.

$E3$	$E2$	$E1$	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Solución:

1. Primero identificamos los números primos del 0 al 15 e hicimos una tabla de verdad de 4 variables para poder representar los números. Para cada número primo el resultado de la función F será evaluada a 1, en otro caso se evalúa a 0. La tabla se muestra a continuación

x_0	x_1	x_2	x_3	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Apoyándonos en la tabla de verdad, obtuvimos la siguiente función:

$$F(x_1, x_2, x_3, x_4) = \overline{x_1}\;\overline{x_2}x_3\overline{x_4} + \overline{x_1}\;\overline{x_2}x_3x_4 + \overline{x_1}x_2\overline{x_3}x_4 + \overline{x_1}x_2x_3x_4 + x_1\overline{x_2}x_3x_4 + x_1x_2\;\overline{x_3}x_4$$

Podemos utilizar un mapa de Karnaugh para simplificar la función:

x_2x_3 \ x_0x_1	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

Del mapa de Karnaugh, obtenemos la siguiente función:

$$F(x_1, x_2, x_3, x_4) = x_1\overline{x_2}x_3 + \overline{x_0}x_1x_3 + \overline{x_0}\;\overline{x_1}x_2 + \overline{x_1}x_2x_3$$

Basta con representar la función F en un circuito para resolver el ejercicio.

2. Como debemos de resolver el ejercicio con mintérminos y maxtérminos, explicamos el procedimiento para ambos:

- Para los **mintérminos** tenemos que fijarnos en las filas de la tabla que se evalúan a 1. Cuando una entrada es 0 la denotamos con la negación y cuando es 1 la dejamos como está. Cambiaremos los nombres de las entradas por conveniencia. Sea $E3 = A, E2 = B$ y $E1 = C$. Entonces tenemos lo siguiente:

$$\begin{aligned}
\overline{A}\ \overline{B}\ \overline{C} + \overline{A}\ \overline{B}C + \overline{A}BC + A\overline{B}C &= \overline{A}(\overline{B}\,\overline{C} + \overline{B}C + BC) + A\overline{B}C && \text{(por distributividad)}\\
&= \overline{A}(\overline{B}(\overline{C} + C) + BC) + A\overline{B}C && \text{(por distributividad)}\\
&= \overline{A}(\overline{B}(1) + BC) + A\overline{B}C && \text{(por complemento)}\\
&= \overline{A}(\overline{B} + BC) + A\overline{B}C && \text{(por identidad)}\\
&= \overline{A}\,\overline{B} + \overline{A}BC + A\overline{B}C && \text{(por distributividad)}
\end{aligned}$$

- Para los maxtérminos tenemos que fijarnos en las filas de la tabla que se evalúan a 0. Cuando una entrada es 1 la denotamos con la negación y cuando es 0 la dejamos como está. Entonces tenemos lo siguiente:

$$(A + \overline{B} + C)(\overline{A} + B + C)(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C})$$

Sea D $= \overline{B} + C$. Entonces tenemos:

$$(A + D)(\overline{A} + B + C)(\overline{A} + D)(\overline{A} + \overline{B} + \overline{C})$$

Por conmutatividad tenemos:

$$(A + D)(\overline{A} + D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por distributividad:

$$(\overline{A}(A + D) + D(A + D))(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por distributividad:

$$(\overline{A}A + \overline{A}D + DA + DD)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por complemento y por idempotencia:

$$(0 + \overline{A}D + DA + D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por identidad y por conmutatividad:

$$(D\overline{A} + DA + D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por distributividad y por complemento:

$$(D(\overline{A} + A) + D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C}) = (D(1) + D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

Por identidad, por idempotencia y sustituyendo D:

$$(D)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C}) = (\overline{B} + C)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

3. En una planta de manejo de residuos tóxicos cuentan con la mejor tecnología para tratar desechos peligrosos y mantener a sus trabajadores seguros. Una parte fundamental de su sistema de protección consta de tres filtros que mantienen la toxicidad del área prácticamente nula. Sin embargo estos filtros pueden fallar, lo que volvería nocivo permanecer en la planta. Con la finalidad de monitorear el estado de los filtros, estos tienen un sensor que indica si es que el filtro esta fallando o no. Si uno de los filtros falla es posible trabajar con normalidad pero es necesario notificar al servicio técnico para que lo reparen a la brevedad. En caso de que fallen dos, la planta puede seguir trabajando pero los empleados se deben retirar mas temprano. Finalmente, si es que fallan los 3, sera necesario activar el protocolo de alerta y evacuar inmediatamente la planta. Se necesita un mecanismo que indique la cantidad de filtros que fallan para que puedan tomar las acciones pertinentes según sea el caso.

Solución:

Para poder representar el problema de la planta, lo que hicimos fue crear un circuito con tres entrada y tres salidas; si ninguna de las salidas está encendida, entonces ningún filtro está averiado y todo puede transcurrir con normalidad; si la primer salida está encendida, entonces exactamente un filtro de los tres que hay está averiado, por lo que se puede trabajar con normalidad, pero se debe de notificar al servicio técnico; si la segunda salida está encendida, entonces exactamente dos filtros están averiados, por lo que se puede trabajar con normalidad pero los trabajadores se deben de retirar más temprano; si la tercer salida está encendida, entonces los tres filtros están averiados, por lo que se activa el protocolo de alerta se activa y todos deben de evacuar.

Para implementar esto, hicimos una tabla de verdad de tres variables y tres funciones de conmutación F_1, F_2, F_3 donde cada función corresponde a cada salida y el número de radares averiados; es decir, F_1 verifica si un radar está averiado y se conecta a la primer salida, F_2 verifica si dos radares estén averiados y se conecta a la segunda salida y F_3 verifica si tres radares están averiados y se conecta a la tercer salida. La tabla de verdad y las funciones de conmutación se muestran a continuación:

Tabla de verdad:

x	y	z	F_1	F_2	F_3
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	1

Funciones de conmutación:

$$F_1(x, y, z) = \overline{x}\,\overline{y}z + \overline{x}b\overline{c} + a\overline{b}\,\overline{c}$$
$$F_2(x, y, z) = \overline{x}yz + x\overline{b}c + ab\overline{c}$$
$$F_3(x, y, z) = abc$$

2. Preguntas

1. ¿Cuál es el procedimiento a seguir para desarrollar un circuito que resuelva un problema que involucre lógica combinacional?

Solución:

- **Paso 1:** Entender y plantear el problema en términos de lógica combinacional, es decir, poner el problema en términos de 0's y 1's; saber cuándo cierta entrada nos debe dar una salida en particular.
- **Paso 2:** Aunque es opcional, es buena idea hacer una tabla de verdad para visualizar todas las posibles combinaciones de la entrada y las posibles salidas.
- **Paso 3:** Convertir los resultados de la tabla a una función de conmutación.
- **Paso 4:** Aunque podría ser parte del paso 3, es importante minimizar con algún método, ya sea mediante minitérminos, maxitérminos o mapas de Karnaugh, según convenga.
- **Paso 5:** Implementar los circuitos necesarios para el problema.
- **Paso 6:** Asegurarse de que funciona para cada combinación de la entrada.

2. Si una función de conmutación se evalúa a más ceros que unos ¿es conveniente usar mintérminos o maxtérminos? ¿En el caso que se evalúe a más unos que ceros?

Solución:

Si una función se evalúa a más ceros que unos conviene utilizar minitérminos porque así nos tenemos que fijar únicamente en los resultados que se evalúan a unos (que son menos). En el caso contrario podemos utilizar maxitérminos que se enfocan en los resultados evaluados a cero.

3. Analizando el trabajo realizado, ¿cuáles son los inconvenientes de desarrollar los circuitos de forma manual?

Solución:

- Si el problema no se entiende bien es muy fácil diseñar algo erróneo o hecho para alguna otra situación diferente de la que nos solicitan.
- Mientras más entradas o combinaciones tengamos más complejo se vuelve el diseño y por lo tanto más difícil se vuelve encontrar alguna falla o incluso se puede hacer confusa su implementación.