Sumadores y Multiplicadores

Estructura de Computadores

Doble Grado de Biotecnología e Ingeniería Informática

Grado de Ingeniería Informática

Aleix Mariné i Tena

Marc Compitrompi ///

**Introducción**

La práctica consiste en la realización y el análisis de diversos circuitos digitales centrados en las operaciones de suma y de multiplicación. Para ello se utilizará como herramienta de soporte la aplicación TkGate.

Un sumador es un circuito digital que realiza la adición de números. En muchas computadoras y procesadores se utilizan sumadores en las unidades aritméticas lógicas, por lo que son esenciales para su funcionamiento. Sus funciones incluyen el cálculo de direcciones, el desplazamientos en los índices de tablas, las operadores de incremento y decremento y otras operaciones similares.

Los multiplicadores son circuitos digitales usados en computación para multiplicar dos números binarios. Suelen ser implementados como una multiplicación de productos parciales que luego son sumados para obtener el producto final. Para implementarlos se utilizan sumadores binarios.

En esta práctica, se propone implementar diferentes circuitos de manera incremental, de tal forma que las tareas de las fases posteriores necesitan la correcta implementación de los circuitos de las fases anteriores.

**FASE 0: Half Adder (HA) i Full Adder (FA)**

*TAREA 1: Realizad el circuito digital Half Adder (HA) de 1 bit que se muestra en la siguiente figura. Suponed que los retardos de las puertas lógicas utilizadas son de AND=1T y XOR=2T. (0.5P)*

A description...

El HA suma dos dígitos binarios simples A y B, denominados sumandos, y sus salidas son Suma (S) y Acarreo (C). La señal de acarreo representa un desbordamiento en el siguiente dígito en una adición de varios dígitos.

El HA tiene la limitación de que no admite un C de entrada, por lo que no admiten una conexión en serie para poder propagar el C y permitir la implementación de un sumador de varios bits. Para solventar este inconveniente, se utiliza el Full Adder (FA), explicado más adelante.

Nuestra implementación es la siguiente:

A description...

/// formatar taula

Entradas Salidas

A B C S

0 0 0 0

1 0 0 1

0 1 0 1

1 1 1 0

**Retardos**

Debido que tanto S como C solamente dependen del resultado de una puerta, los tiempos de retardo son directamente los indicados por el enunciado.

T(S) = R(XOR) = 2T

T(C) = R(AND) = 1T

***TAREA 2.*** *Realizad el circuito digital Full Adder (FA) de 1 bit con acarreo de entrada que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Suponed que los retardos de las puertas lógicas utilizadas son de AND=1T, OR=1T y XOR=2T (0.75P)*

El FA es un circuito que sirve para sumar dos números binarios de una cifra A y B. El FA admite un C de entrada, por lo que es adecuado en una conexión en serie para crear un sumador de más de un bit. Debido a la nueva entrada, la lógica del circuito es más compleja que en el HA.A description...

Podemos destacar que la puerta OR del C puede ser sustituida por una puerta XOR sin alterar la lógica del circuito por si hay algun tipo de limitación a la hora de usar diferentes tipos de puerta.

Además, el FA también puede ser implementado a partir de la unión de dos HA siguiendo el diagrama de la siguiente figura:

Nuestra implementación es la siguiente:A description...

A description...

/// formatar taula

Entradas Salidas

A B Cin Cout S

0 0 0 0 0

0 0 1 0 1

0 1 0 0 1

0 1 1 1 0

1 0 0 0 1

1 0 1 1 0

1 1 0 1 0

1 1 1 1 1

**Retardos y Area**

Para calcular el retardo usaremos el camino crítico para cada salida. Los retardos se calcularan de la misma manera a lo largo de la práctica.

T(S) = 2\*R(XOR) = 2\*2T = 4T

T(C) = R(XOR) + R(AND) + R(OR) = 2T + 1T + 1T = 4T

/// recalcular area a partir del tkgate (?)

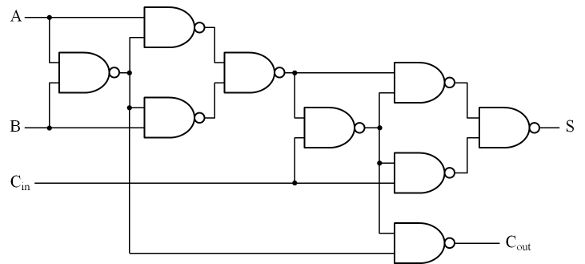
Área utilizada=2XOR+2AND+1OR=2\*8+2\*6+6=34

***TAREA 3.*** *Realizad una implementación alternativa al mismo circuito Full Adder (FA) de 1 bit con acarreo de entrada y comparad los tiempos de retardo y área con la solución anterior. (0.75P)*

Como implementación alternativa del FA hemos utilizado la implementación descrita en las transparencias.

A description...

Sin embargo existen otras alternativas, como por ejemplo el FA usando exclusivamente puertas lógicas NAND, como se indica en la siguiente figura:



**Retardo y area del FAv2**

T(S) = 2\*R(XOR) = 2\*2T = 4T

T(C) = R(AND) + 2\*R(OR) = 1T + 2\*1T = 3T

Área utilizada=2\*XOR+3\*AND+2\*OR=2\*8+3\*6+2\*6 = 46

**Comparación con la versión 1 del FA**

Podemos observar que la salida Cout en la versión 1 del FA depende de la salida de una puerta XOR, que es compartida tanto para el cálculo de S como de Cout. La puerta XOR es la que más retardo tiene, lo cual nos añade un tiempo extra de retardo respecto a la versión 2 del FA (4T).

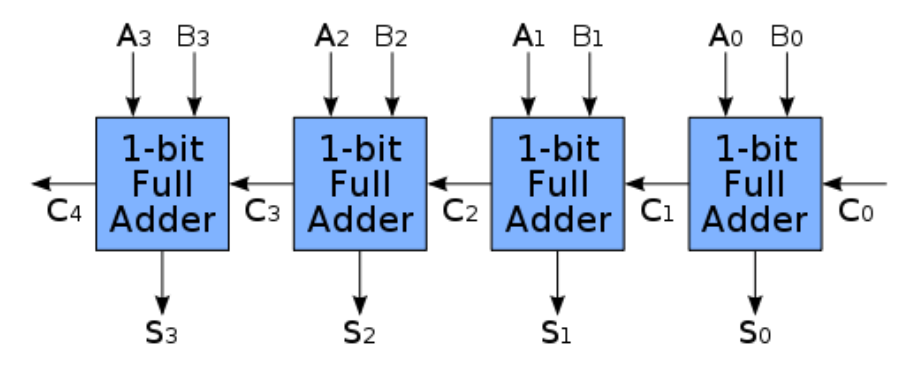
En la versión 2, el cálculo de Cout y S es independiente,es decir, el cálculo de cada salida es independiente del cálculo de la otra. De esta manera, el camino crítico de Cout es de 3T, ya que no hay ninguna salida compartida, reduciendo en 1T la latencia.

Sin embargo esta disminución de la latencia se ve penalizado por un mayor número de puertas lógicas en la versión 2 del FA, lo cual se traduce en una mayor area, concretamente, de 46 – 34 = 12 unidades más.

Por tanto, el uso de una u otra implementación dependerá de si queremos priorizar una disminución del retardo en el cálculo de Cout o bien un uso de menos puertas lógicas y menor area.

*FASE 1: Sumadores CPA*

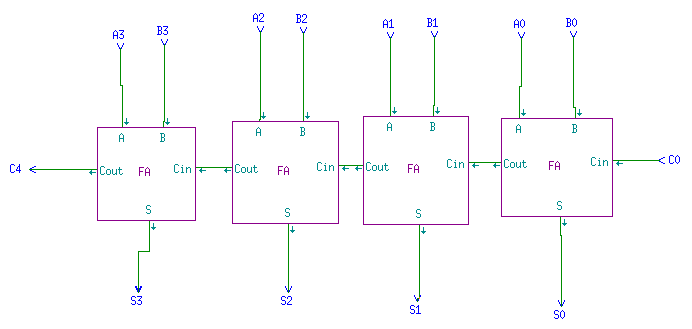
***TAREA 4.*** *Realizad el circuito digital Carry Propagate Adder (CPA) de 4 bits que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Realizad dos versiones del mismo CPA asumiendo las dos posibles implementaciones de Full Adder (FA) de 1 bit consideradas en tareas anteriores. (1P)*



Un sumador con propagación de acarreo, también denominado sumador de acarreo serie, es aquel constituido por varios sumadores completos en los cuales el acarreo de salida se conecta a la entrada de acarreo del sumador siguiente. Podemos observar que el primero de los sumadores puede ser reemplazado por un semisumador suponiendo que el acarreo de entrada Cin es cero.

La principal limitación del sumador es que este es relativamente ya que cada etapa debe esperar que se calcule un acarreo de entrada de la etapa previa.

Nuestra implementación es la siguiente



Hemos utilizado dos versiones diferentes del FA para implementar el CPA.

**Retardos y Area**

**Versión 1 del FA**

Cuando usamos este FA, el cálculo de Cout y S tiene una puerta XOR en común, por lo que para calcular Cout necesitamos el resultado de esa puerta. Esto implica que el primer FA necesita 4T para estabilizar la señal tanto de Cout como de S. Sin embargo, el resto de FA pueden llevar a cabo la operación de esta XOR de manera paralela sin utilizar el Cout del anterior FA, es decir, que esta operación es paralelizable, disminuyendo así el camino crítico tanto para el cálculo de Si para i > 0 como para Ci para i > 0. De esto se deduce que:

Retardo C = TC(bit 0) + (n-1 bits) \* (TC – PP) = (R(XOR) + R(AND) + R(OR)) + (4-1) \* ((R(XOR) + R(AND) + R(OR)) – (R(XOR))) = (2T + 1T + 1T) + 3 \* (2T + 1T + 1T - 2T) = 10T

Para el retardo de S, podemos utilizar la misma fórmula, pero teniendo en cuenta que en el último FA debemos llevar a cabo la operación con la XOR para sacar el último Si, que es la última parte del camino crítico. Por tanto se deduce que:

Retardo S = TC(bit 0) + (n-2 bits) \* (TC – PP) + (TS – PP) = (R(XOR) + R(AND) + R(OR)) + (4-2) \* ((R(XOR) + R(AND) + R(OR)) – (R(XOR))) + (2 \* R(XOR) – R(XOR)) = (2T + 1T + 1T) + 3 \* (2T + 1T + 1T - 2T) + (2 \* 2T – 2T) = 4T + 3 \* 2T + 4T – 2T = 12T

Área utilizada = 4 \* FAv1 = 4 \* 34 = 136

**Versión 2 del FA**

Análogamente al razonamiento anterior, se deduce que no se puede paralelizar el cálculo en ninguna puerta, ya que contrariamente a lo que pasaba en el anterior FA, no hay ninguna puerta que pueda calcularse usando únicamente A y B, por lo que el cálculo de Cout siempre debe esperar a que se estabilice la señal de Cin, teniendo 0 partes paralelizables.

Retardo C = TC(bit 0) + (n-1 bits) \* (TC – PP) = TC(bit 0) + 3 \* TC = 4 \* TC = 4 \* (R(AND) + 2 \* R (OR)) = 4 \* (1T + 2 \* 1T) = 12T

Para el retardo de S, vemos que la primera XOR es paralelizable ya que solamente depende de las entradas A y B, sin embargo la segunda XOR depende del Cout del anterior sumador; por lo que siempre deberemos esperar al carry para calcular la última XOR, es decir, que solamente tendremos en cuenta la paralelización en el último FA debido a que es que marca el camino crítico:

Retardo S = TC(bit 0) + (n-2 bits) \* (TC – PP) + (TS – PP) = TC + (4-2) \* (TC – 0) + (TS – PP) = 3 \* TC + (TS – PP) = 3 \* (R(AND) + 2 \* R (OR)) + (2 \* R(XOR) – R(XOR)) = 3 \* (1T + 2 \* 1T) + 2 \* 2T – 2T = 9 T + 2T = 11T

Área utilizada = 4 \* FAv2 = 4 \* 46 = 184

***TAREA 5.*** *Indicad las fórmulas que describen los tiempos de retardo del circuito digital Carry Propagate Adder (CPA) de 4 bits implementado en la tarea anterior. Aplicando esa fórmula, mostrad los tiempos de retardo que introduciría un CPA de 8 bits, 16 bits, 32 bits y 64 bits.*

* CPA de 8 bits

Retardo C = 4T + (8-1) \* (4T – 2T) = **18T**

Retardo S = 4T + (8-2) \* (4T-2T) + (4T-2T) = **18T**

* CPA de 16 bits

Retardo C = 4T + (16-1) \* (4T – 2T) = **34T**

Retardo S = 4T + (16-2) \* (4T-2T) + (4T-2T) = **34T**

* CPA de 32 bits

Retardo C = 4T + (32-1) \* (4T – 2T) = **66T**

Retardo S = 4T + (32-2) \* (4T-2T) + (4T-2T) = **66T**

* CPA de 64 bits

Retardo C = 4T + (64-1) \* (4T – 2T) = **130T**

Retardo S = 4T + (64-2) \* (4T-2T) + (4T-2T) = **130T**

***TAREA 5.*** *Asumid un tiempo de retardo para la puerta lógica XOR de 1T. Realizad de nuevo el circuito digital Carry Propagate Adder (CPA) de 4 bits y comparadlo con los resultados obtenidos anteriormente.*

TC = 3T

TS = 2T

* CPA de 4 bits

Retardo C = 3T + (4-1) \* (3T – 1T) = **9T**

Retardo S = 3T + (4-2) \* (3T-1T) + (2T-1T) = **8T**

* CPA de 8 bits

Retardo C = 3T + (8-1) \* (3T – 1T) = **17T**

Retardo S = 3T + (8-2) \* (3T-1T) + (2T-1T) = **16T**

* CPA de 16 bits

Retardo C = 3T + (16-1) \* (3T – 1T) = **33T**

Retardo S = 3T + (16-2) \* (3T-1T) + (2T-1T) = **32T**

* CPA de 32 bits

Retardo C = 3T + (32-1) \* (3T – 1T) = **65T**

Retardo S = 3T + (32-2) \* (3T-1T) + (2T-1T) = **64T**

* CPA de 64 bits

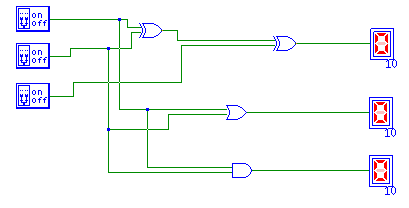
Retardo C = 3T + (64-1) \* (3T – 1T) = **129T**

Retardo S = 3T + (64-2) \* (3T-1T) + (2T-1T) = **128T**

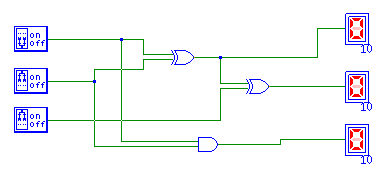
**FASE 2: Sumadores CLA y CSA**

***TAREA 6.*** *Realizad las dos versiones de circuito digital Partial Full Adder (PFA) de 1 bit con acarreo de entrada que se muestran en las siguientes figuras e indicad los tiempos de retardo y el área utilizada. Suponed que los retardos de las puertas lógicas utilizadas son de AND=1T, OR=1T y XOR=2T.*

*Figura 1*



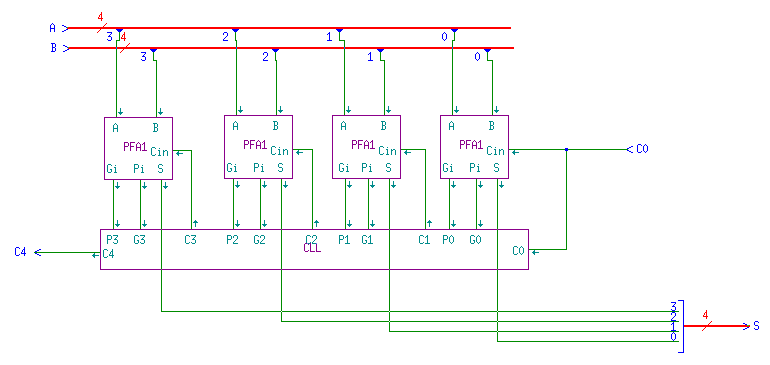
*Figura 2*



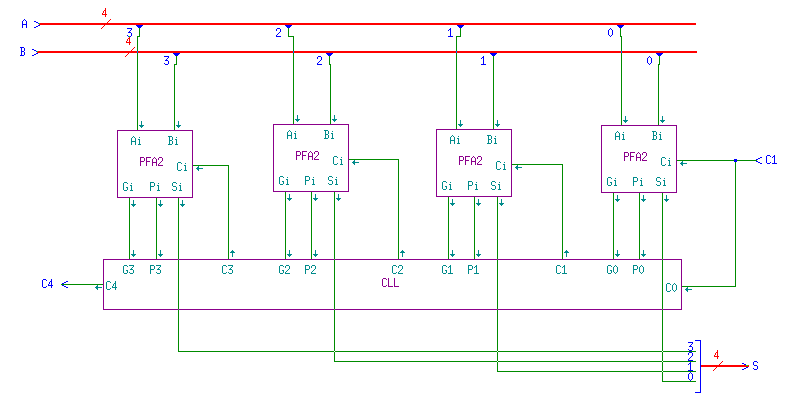
|  |  |
| --- | --- |
| Figura 1  TC = 4T  TPi = 1T  TGi = 1T  Área utilizada = 28 | Figura 2  TPi = 2T  TSi = 4T  TGi = 1T  Área utilizada = 22 |

***TAREA 7.*** *Realizad dos versiones del circuito digital Carry Look-Ahead Adder (CLA) de 4 bits que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Asumid para cada una de las versiones los diseños de Partial Full Adder (FA) de 1 bit implementados en la tarea anterior. Suponed que los retardos de las puertas lógicas utilizadas son de AND=1T, OR=1T y XOR=2T. .*

*V1.*



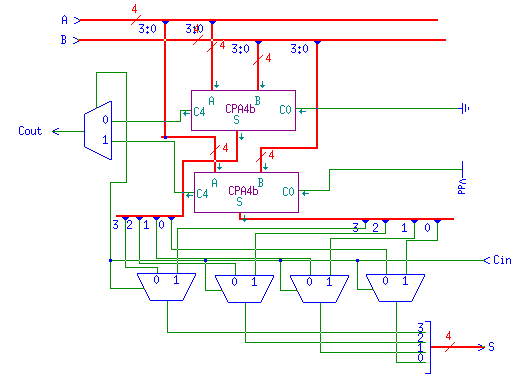
*V2.*



|  |  |  |  |
| --- | --- | --- | --- |
| Versión 1 | | Área utilizada = 226 | |
| Operación | Datos entrada | | Retardo |
| Generar y propagar | A, B | | 1T |
| C1 a Cn | P, G, C0 | | 2T |
| Suma S | C, A, B | | 2T |
| **Retardo total** | | | **5T** |

|  |  |  |  |
| --- | --- | --- | --- |
| Versión 2 | | Área utilizada = 202 | |
| Operación | Datos entrada | | Retardo |
| Generar y propagar | A, B | | 2T |
| C1 a Cn | P, G, C0 | | 3T |
| Suma S | C, A, B | | 2T |
| **Retardo total** | | | **7T** |

***TAREA 8.*** *Realizad el circuito digital multiplicador Carry Select Adder (CSA) de 4 bits que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Asumid el diseño de Carry Propagate Adder (CPA) de 4 bits implementado en una tarea anterior y un retardo para el multiplexor de 1T.*



RCSA = RCPA + RMux

RCPA = 10T RMUX = 1T

RCSA = 10T + 1T = **11T**

Área utilizada = 312

***TAREA 9.*** *Indicad las fórmulas que describen los tiempos de retardo del circuito digital Carry Select Adder (CSA) de 4 bits que se ha implementado en la tarea anterior. Aplicando esa fórmula, mostrad los tiempos de retardo que introduciría un CSA de 8 bits, 16 bits, 32 bits y 64 bits.*

RCSA = RCPA + (Mmodulos - 1) \* RMux

* CSA de 8 bits

RCSA = 10T + 1 \* 1T = **11T**

* CSA de 16 bits

RCSA = 10T + 3 \* 1T = **13T**

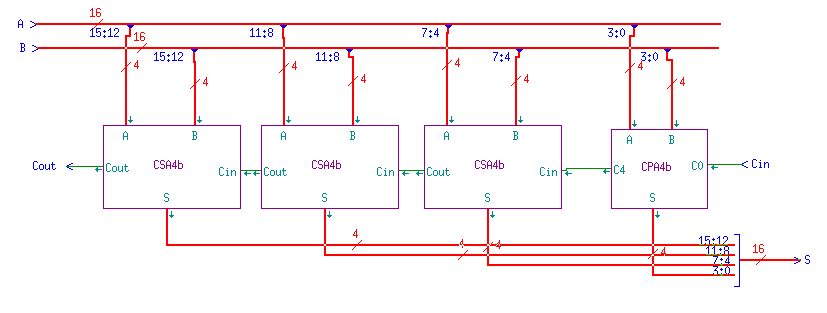
* CSA de 32 bits

RCSA = 10T + 8 \* 1T = **17T**

* CSA de 64 bits

RCSA = 10T + 16 \* 1T = **25T**

***TAREA 10.*** *Realizad el circuito digital Carry Select Adder (CSA) de 16 bits e indicad los tiempos de retardo y el área utilizada. Utilizad para esta implementación circuitos Carry Select Adder (CSA) de 4 bits implementados en una tarea anterior. Sin necesidad de implementarlo, comparad los tiempos de retardo con un Carry Select Adder (CSA) de 16 bits que usa Carry Propagate Adders (CPA) de 16 bits.*



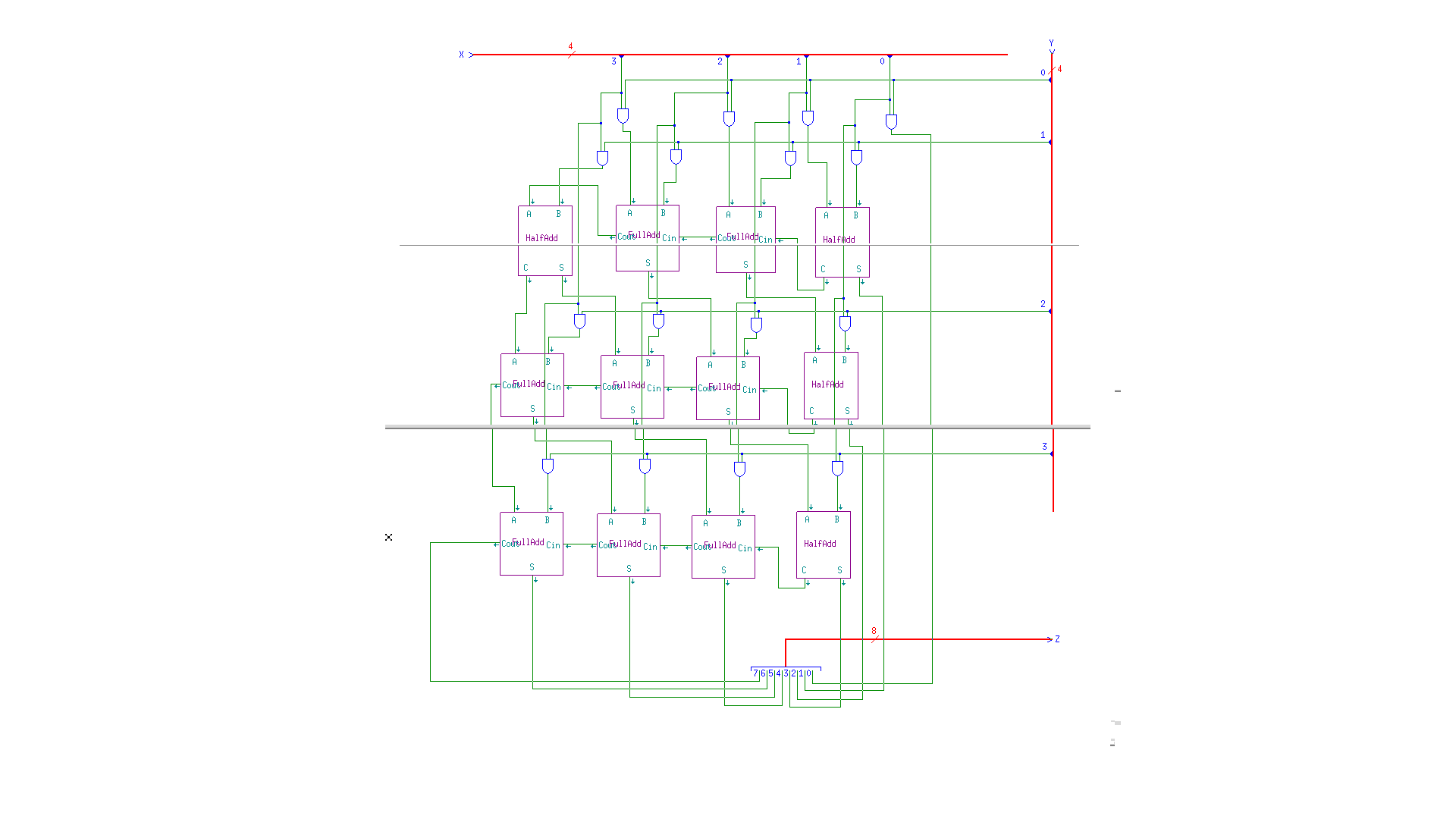
RCPA = 10T RMUX = 1T

RCSA = 10T + 3 \* 1T = **13T**

Área utilizada = 1072

**FASE 3: Multiplicador RCA**

***TAREA 11.*** *Realizad el circuito digital multiplicador Riple Carry Array de 4 bits que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Asumid los diseños de Half Adder (HA) y Full Adder (FA) de 1 bit implementados en las tareas anteriores. Suponed también que los retardos de las puertas lógicas utilizadas son de AND=1T, OR=1T y XOR=2T.*



En nuestro caso se trata de la siguiente fórmula:

FA y HA con PP = TAND + (N-1)\*(TS-PP) + (N-1)\*TC + (N-3)\*(TC-PP)

FA : TS = 4T, TC = 4T, PP = 2T

Retardo = 1 + 1 + 3\*2 + 3\*4 + 1\*2 = 21T

Área utilizada = 424

***TAREA 12.*** *Indicad la fórmula que describe el retardo del circuito digital multiplicador Riple Carry Array de 4 bits que se ha implementado en la tarea anterior. Aplicando esa fórmula, mostrad los tiempos de retardo que introduciría un multiplicador RCA de 8 bits, 16 bits, 32 bits y 64 bits.*

FA y HA con PP = TAND + (N-1)\*(TS-PP) + (N-1)\*TC + (N-3)\*(TC-PP)

* CSA de 8 bits

FA y HA con PP = 1+ 7\*(4-2) + 7\*4 + 4\*(4-2) = 1 +14 + 28 + 8 = 51T

* CSA de 16 bits

FA y HA con PP = 1 + (15\*(4-2)) +15\*4 + 13\*(4-2) = 1 + 30 + 60 + 26 = 117T

* CSA de 32 bits

FA y HA con PP = 1 + 31\*(4-2) + 31\*4 + 29\*(4-2) = 1 + 31\*2 + 31\*4 + 58 = 245T

* CSA de 64 bits

FA y HA con PP = 1 + 63\*(4-2) + 31 + 61\*(4-2) = 1 + 63\*2 + 63\*4 + 61\*2 = 501T

Bibliografia

<https://en.wikibooks.org/wiki/Practical_Electronics/Adders>

[https://en.wikipedia.org/wiki/Adder\_%28electronics%29](https://en.wikipedia.org/wiki/Adder_(electronics))