



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

INFORME 3, ARQUITECTURA Y ORGANIZACIÓN DE
COMPUTADORES.

IMPLEMENTACIÓN DE FLIP-FLOPS D DE DOS CAPAS PARA REALIZAR OPERACIONES SOBRE ENTRADAS DE 8 BITS

Gabriela Acuña, rol 201973504-7.
Profesor Mauricio Solar.

25 de Junio, 2021

Índice

1	Resumen	2
2	Introducción	2
3	Desarrollo	2
3.1	Circuito de Suma	3
4	Resultados	3
5	Análisis	3
6	Conclusión	3
7	Anexo	4

1 Resumen

Para realizar esta tarea se implementaron 8 flip-flops D de 2 capas, uno para cada bit del número X de entrada. Cada uno de estos flip-flops se generó desde la configuración conocida de compuertas lógicas. Luego de solucionar esto, se desarrolló individualmente cada uno de los casos posibles según el valor de M . Para la suma se utilizó el Full Adder conocido por la tarea anterior, y para la resta se utilizó la implementación propuesta en el enunciado. Para los otros dos casos se utilizaron compuertas de 8 bits. Se consideró una implementación exitosa ya que los resultados obtenidos son iguales a los dados como ejemplo en el enunciado de la tarea.

2 Introducción

El objetivo de esta tarea fue realizar una implementación de circuito con memoria interna, compuesta por flip-flops D, para así usar el valor anterior y actual de la variable de entrada X y realizar ciertas operaciones con estos, según el valor entregado en M .

Para el desarrollo de esta tarea se utilizaron Flip-Flops D¹ estos se pueden ver en la *figura 2* como *FF1* y *FF2*. La unión de estos dos Flip-Flops D genera un Flip-Flop D de 2 niveles. Un beneficio de utilizar 2 niveles es que el circuito recuerda su entrada previa y actual en todo momento, sin depender del cambio de ciclo de reloj. Para expandir este circuito y utilizarlo en 8 bits, se separó cada uno de los 8 bits de entrada, para recordarlos individualmente, al actual y al anterior. Para no reescribir el circuito muchas veces se utilizó el circuito presentado en la *figura 2* como subcircuito, el cual fue nombrado *logicFlipFlop*.

Por otro lado, al igual que para el desarrollo de la tarea pasada, se implementó la suma sin carry de salida a través de un full adder por bit de salida. Para la resta, se utilizó la implementación de la suma para realizar el atajo presentado en el enunciado que se ve en la *ecuación 1*.

$$A - B = A + \overline{B} + 1 \quad (1)$$

Por último, para realizar el *AND* y *OR*, se utilizaron compuertas de 2 entradas y 8 bits. Ya que los pines de entrada eran de 8 bits, facilitó bastante el proceso.

3 Desarrollo

La tarea se organizó a través de subcircuitos, como se puede ver en la *figura 1* del circuito general de la implementación completa. Por esta razón se procedió a explicar detalladamente el funcionamiento de cada uno de ellos.

3.1 Circuito de Suma

Titulado **Suma** y perteneciente a **SLogic**, como ya fue mencionado, la base de esta implementación es el circuito conocido *full adder*, que se utiliza para los 8 bits a sumar.

¹P114 libro

3.2 Circuito de Resta

Para el desarrollo de este circuito se utilizo el subcircuito ya explicado de Suma, y con el atajo entregado en el enunciado quedo como una operacion bastante simple, con la negación de $P(X)$ y la suma de la constante 1.

4 Resultados

5 Análisis

6 Conclusión

7 Anexo

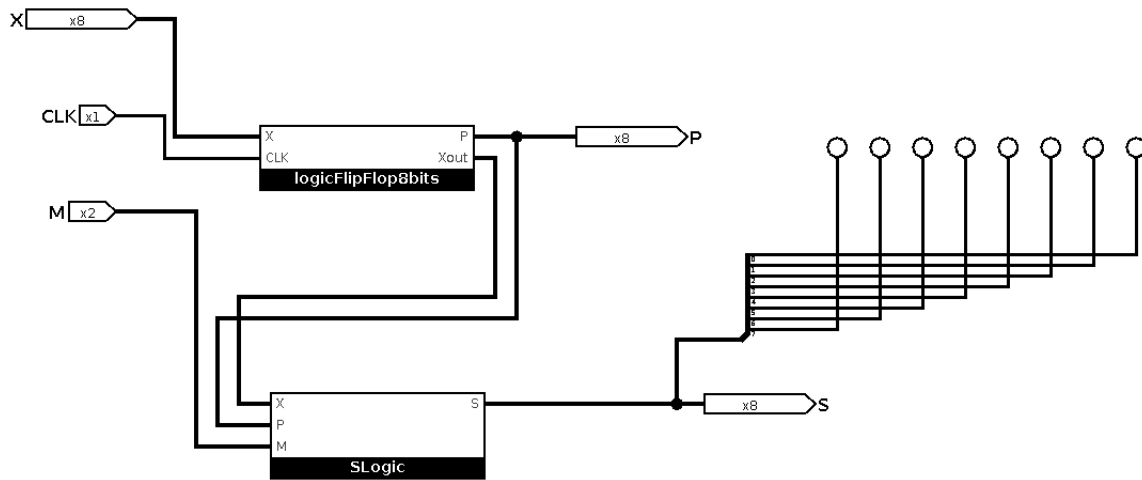


Figure 1: Circuito general, con salida en leds y salidas en binario para referencia

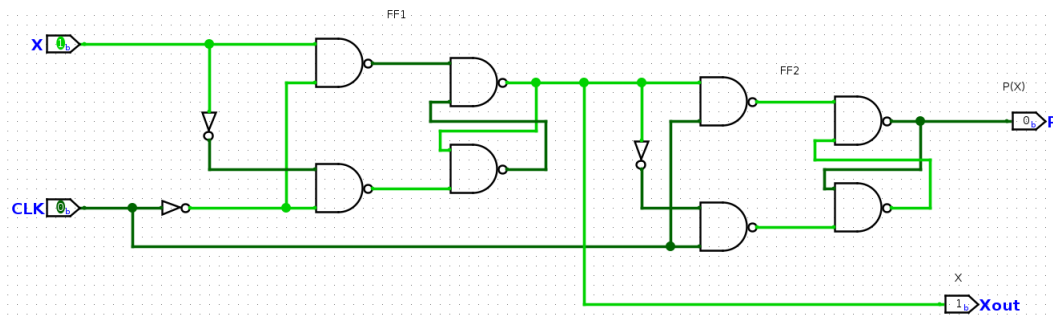


Figure 2: Flip-Flop D de 2 capas para 1 bit.