Máquinas de estados tipo DOW

Creador: David Rubio G.

Entrada: https://soceame.wordpress.com/2024/11/21/maquinas-de-estados-tipo-dow/

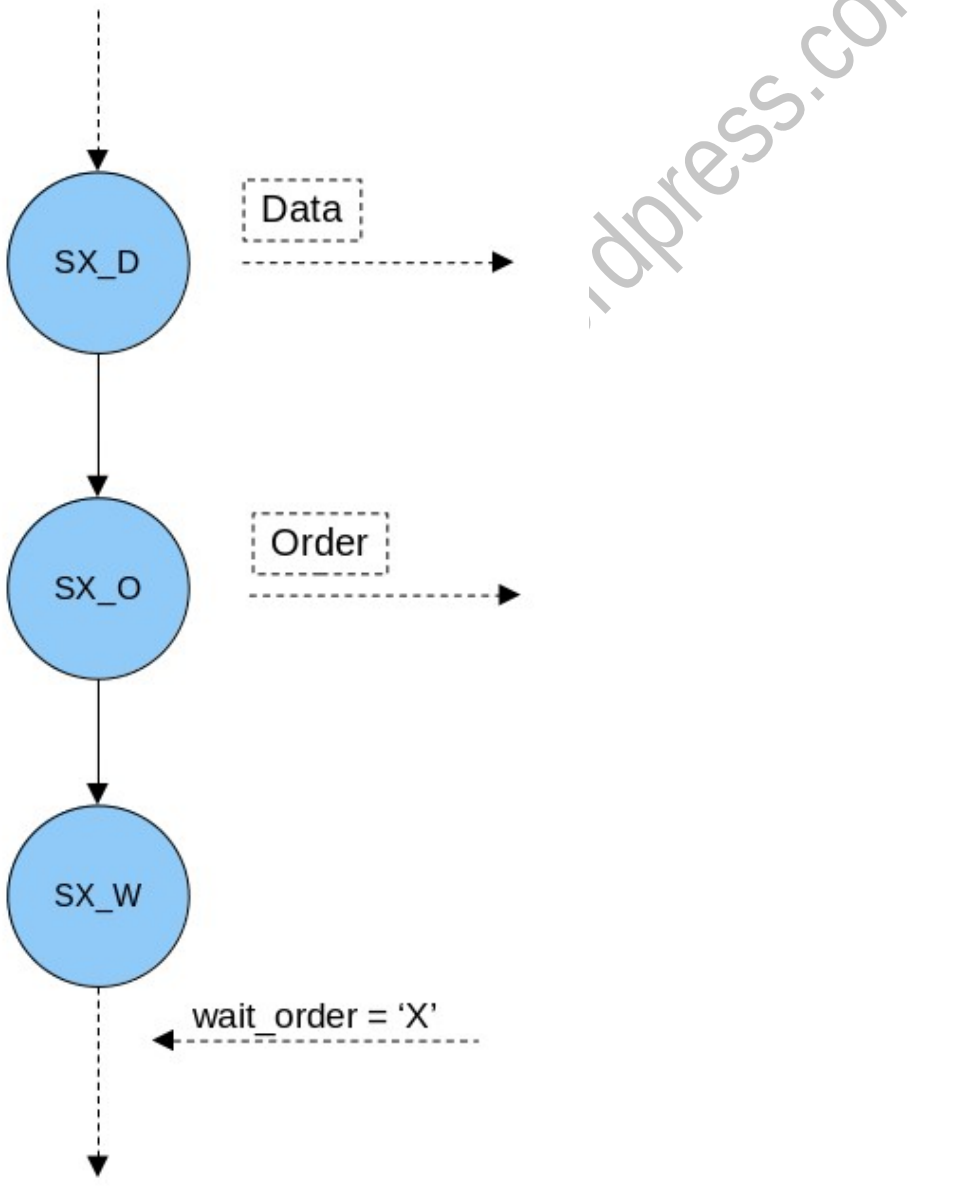
Blog: https://soceame.wordpress.com/

GitHub: https://github.com/DRubioG

Fecha última modificación: 23/02/2025

Las máquinas de estados tipo **DOW** vienen del nombre *data-order-wait (en castellano: DOE, dato-orden-esperar)*. Este tipo de máquinas de estado se basan en garantizar las comunicaciones entre máquinas de estado, para que cuando se le manda la orden a siguiente máquina de estado se pueda garantizar que el dato esté disponible en la línea que esta orden dure un ciclo de reloj.
Para ello lo que se hace es romper las máquinas de estados normales para realizar una acción por estado.

NOTA: cuando se entra en sistemas de alta complejidad, dificultad o tamaño en FPGAs, este tipo de sistemas se acaban imponiendo.



Para conseguir esto se tiene que tener al menos 3 estados.

- **Primer estado**, es para poner el dato en la línea que se va a comunicar a la siguiente máquina de estado.
- **Segundo estado**, es para mandar la orden a la siguiente máquina de estados
- **Tercer estado**, es para tirar la línea de comunicaciones que manda la orden (para que dure un ciclo de reloj) y para esperar que se produzca algún cambio en la señal para continuar la máquina de estados.
  Este estado *wait* es bloqueante, por lo que se recomienda añadir un sistema que elimine el posible bloqueo, ya sea un *watchdog* interno, un sistema de recepción con certidumbre de que devuelve el dato o un sistema que esté por encima que pueda resetear la máquina de estados..

Este tipo de máquinas suelen ser de grandes dimensiones porque multiplican por 3 el número de estados, pero son las más garantistas con las comunicaciones, a sabiendas de que no se va a producir una carrera de datos por un skew, o por otra causa.

Ejemplo

Un ejemplo de este tipo de máquinas es el siguiente.

```
when S2_D =>
    data <= data_in;

    state <= S2_O;

when S2_O =>
    data_ok <= '1';

    state <= S2_W;

when S2_W =>
    data_ok <= '0';

    if data_ok_in = '1' then
        state <= S3_D;
    else
        state <= S2_W;
    end if;
```

Con un sistema de protección se para evitar bloqueos interno sería como el siguiente.

```
when S2_D =>
    data <= data_in;

    state <= S2_O;

when S2_O =>
    data_ok <= '1';

    state <= S2_W;

when S2_W =>
    data_ok <= '0';

    if data_ok_in = '1' then
        state <= S3_D;
    elsif watchdog = '1' then
        state <= S0;
    else
        state <= S2_W;
    end if;
```