

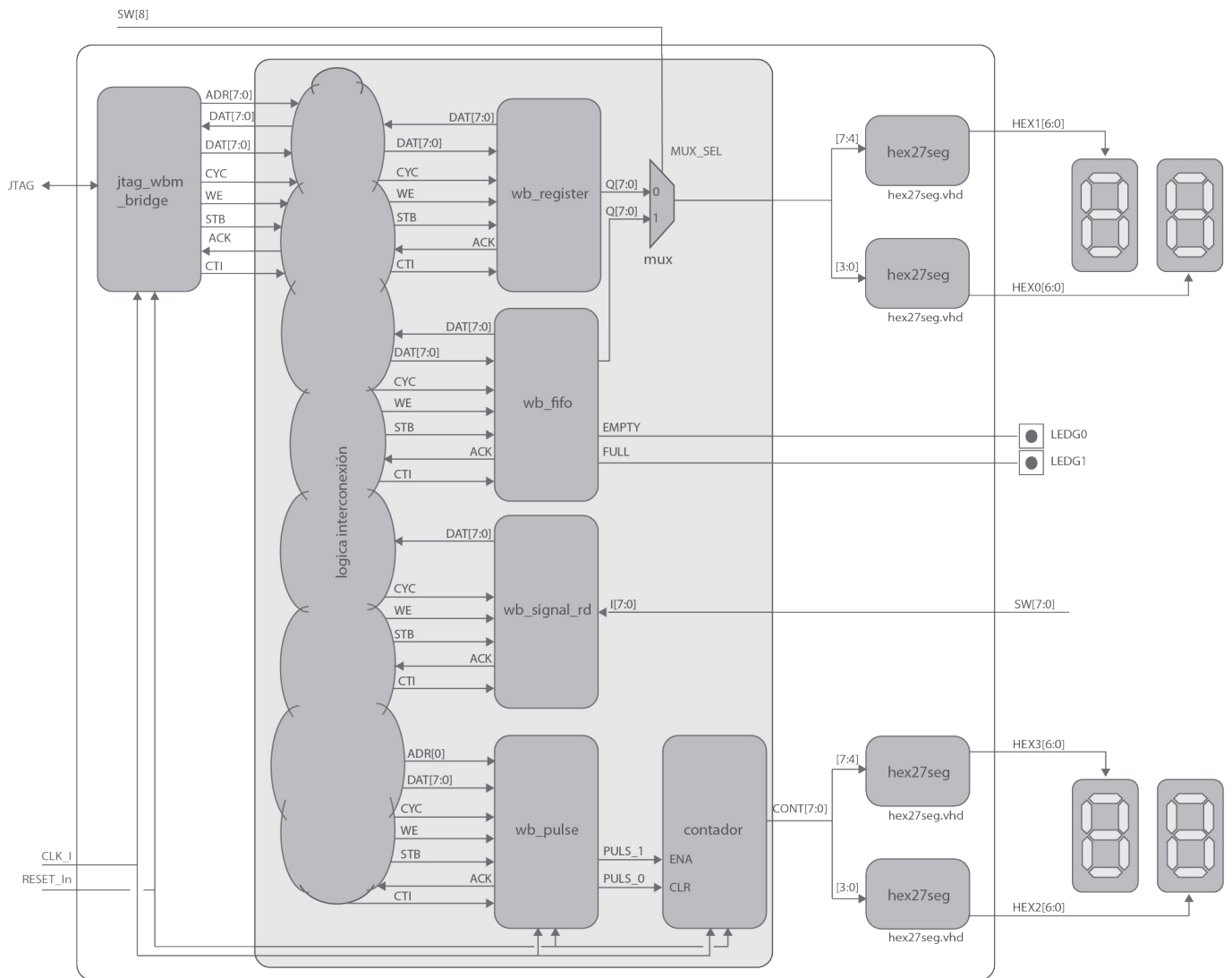
# Práctica 4

## Objetivos

- Tomar contacto con el estándar Wishbone para interfaces entre módulos (IP cores) dentro de un sistema integrado en un chip (System on Chip o SoC). Se usarán transferencias simples de lectura y escritura.

## Descripción

El sistema a diseñar permitirá, desde un PC, escribir y leer varios dispositivos esclavos conectados a un bus Wishbone. Para esto se utilizará un IP Core que recibe comandos a través de la conexión JTAG y los traduce en ciclos Wishbone.



El sistema deberá contar con los siguientes elementos:

- **jtag\_wbm\_bridge**: interfaz jtag a wishbone master
- **wb\_register**: registro de 8 bits con interfaz wishbone esclavo para su lectura/escritura y salida Q[7:0] donde se muestra el contenido del registro.
- **wb\_fifo**: fifo de ancho 8 bits y profundidad 8 bytes con interfaz wishbone esclavo para su lectura/escritura y salidas: Q[7:0] donde se muestra el último dato, EMPTY y FULL que indican estado del FIFO. Para el fifo debe utilizarse una instancia de LPM\_FIFO.
- **wb\_signal\_rd**: bloque con interfaz wishbone esclavo que permite leer el estado de 8 señales conectadas en su entrada I[7:0]
- **wb\_pulse**: bloque con interfaz wishbone esclavo que genera pulsos de ancho un período de reloj CLK\_I cada vez que se realice un ciclo de escritura. Los pulsos se generan en las salidas PULS\_1 o PULS\_0 dependiendo de si su entrada ADR\_I[0]=1 o ADR\_I[0]=0 respectivamente.
- **Contador**: contador de ancho 8 que se incrementa en cada flanco de CLK\_I en el que ENA=1 y se borra en cada flanco en que CLR=1.
- Multiplexor 2 a 1 de ancho 8bits y conversores HEX27SEG. Dependiendo del valor de la señal MUX\_SEL permiten visualizar en los display 7 segmentos:
  - MUX\_SEL=0 : Q de **wb\_register**
  - MUX\_SEL=1 : Q de **wb\_fifo**

La decodificación del espacio de direcciones entrada/salida wishbone deberá ser:

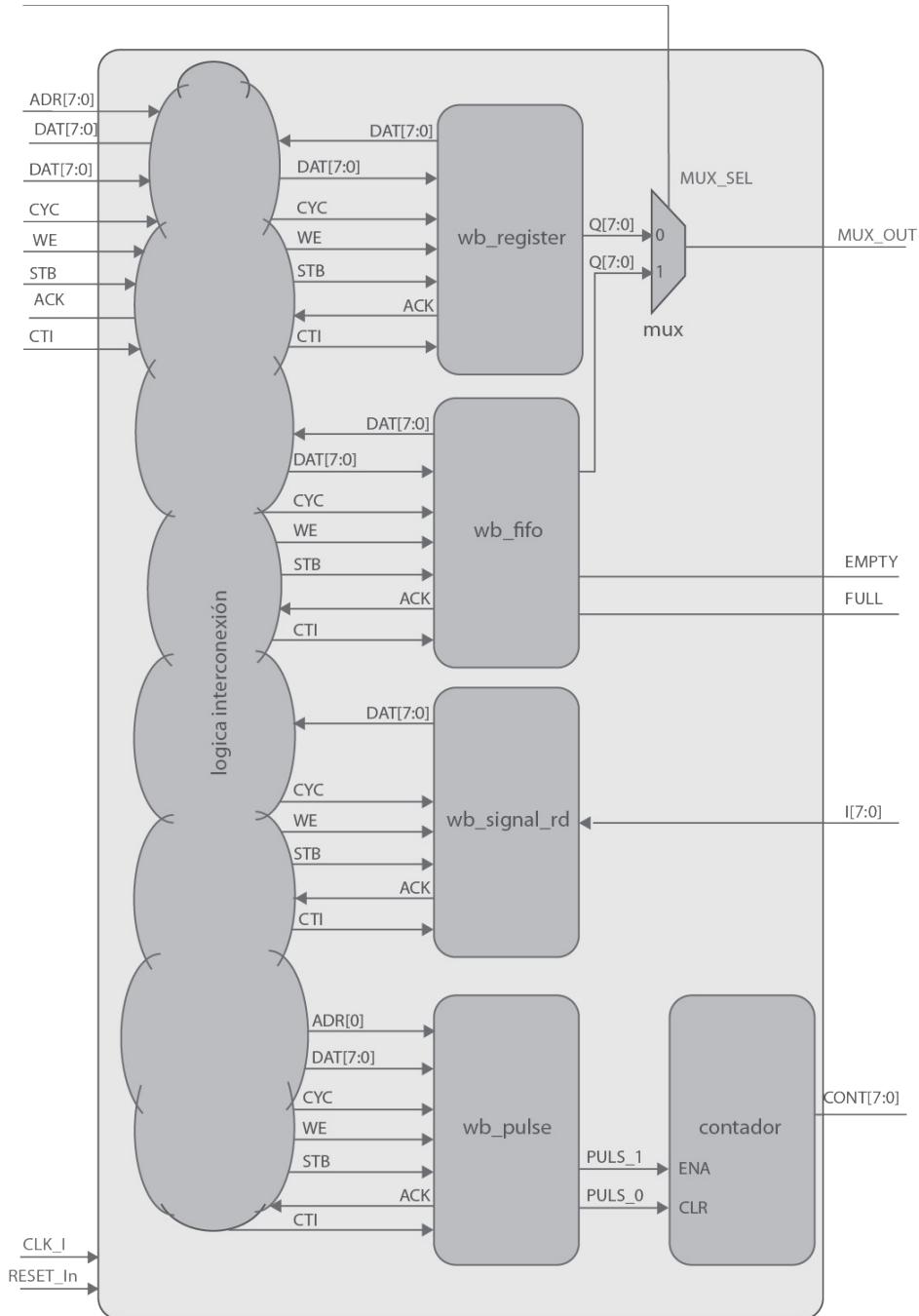
ADR[7:0]	WR	RD
0000 0000 (00h)	wb_register	wb_register
0000 0001 (01h)	wb_fifo	wb_fifo
0000 0010 (02h)	---	wb_signal_rd
0000 0100 (04h)	wb_pulse (PULS_0)	---
0000 0101 (05h)	wb_pulse (PULS_1)	---

## Parte 1

1.a) Diseñar y simular por separado cada uno de los bloques: **wb\_fifo**, **wb\_signal\_rd**, **wb\_pulse** y **contador**

Parte 2

2.a) Crear una una entidad que contenga instancias de todos los bloques wishbone esclavo y la lógica de interconexión.



2.b) Verificar, mediante ciclos de lectura y escritura wishbone a distintas direcciones, que el sistema contenido en la entidad se comporta en forma apropiada.

### Parte 3

3.a) Modificar el diseño de referencia que hace uso del `jtag_wbm_bridge`. Sustituir el registro Wishbone por la entidad validada en el punto 2.a.

3.b) Asignar pines para todas las entradas y salidas y probar el sistema completo en la placa.