

P3 Sessió 3-4: Unitat de Control del mestre I²C

Introducció

Un cop dissenyats els mòduls bàsics del nostre mestre I²C, només ens falta dissenyar la unitat de control encarregada de gestionar la transmissió dels bits d'informació i control així com la generació del senyal rellotge. Per simplificar el seu desenvolupament, la unitat de control es divideix en dos mòduls: el controlador de comandes a nivell de byte i el controlador de comandes a nivell de bit, que anomenarem controlador de byte i controlador de bit.

El controlador byte, gestiona el trànsit I²C a nivell de byte. Prenen les dades del registre de comandament i les tradueix en seqüències basades en la transmissió d'un sol byte. Això ho aconsegueix dividint cada operació de byte en operacions de bit separades, que s'envien al controlador de bit encarregat de gestionar la transmissió real de les dades. Per exemple, per a una lectura d'un sol byte, el controlador de bit rep vuit instruccions de lectura del controlador de byte.

Per tant, el disseny de la unitat de control del mestres I²C es dividirà en dues fases. La primera verificar un controlador de bit donat, i la segona el disseny des de zero del controlador de byte basat en una màquina d'estats.

P3 Sessió 3: Controlador de Bit

Com hem dit el controlador de bit gestiona la transmissió real de cada bit, això implica la generació dels nivells específics per a les senyals *Start*, *Stop*, *repetició d'Start*, així com la temporització controlant les línies SCL i SDA mitjançant les senyals dels buffers de tres estats (veure secció Connexions al Bus de l'Introducció). Per fer-ho, cada operació de bit es divideix en quatre etapes (A, B, C i D), a excepció de *Start* i repetició de *Start* que consten de sis etapes tal i com mostra la **Figura 8**. La transició entre cada estat es controlada per un temporitzador extern.

Objectius

- Verificació a nivell RTL d'una màquina d'estats donada.
 - Planificació de testbench.
-

Material

El material de suport el podeu descarregar del campus virtual:

- `i2c_master_bit_ctrl.v` : RTL dels controlador de bit del I²C.
- `i2c_slave_model.v` : model d'esclau I²C.
- `sys_model.v` : model del sistema, genera el rellotge base i el reset actiu per nivell baix.
- `tb_i2c_master_bit_ctrl.v` : testbench amb bàsic del controlador de bit del I²C.
- `timescale.v` : defineix l'escala temporal pel fitxer on està inclòs.
- Plantilla informe de la pràctica.

Tasques a realitzar

1. Examineu el contingut del fitxer `i2c_master_bit_ctrl.v`, identifiqueu les diferents estructures i comenteu el codi.
2. Examineu el contingut del fitxer `tb_i2c_master_bit_ctrl.v` i comproveu si verifiquen totes les funcionalitats del controlador de bit.
3. Simuleu i verifiquen el controlador de bit (`i2c_master_bit_ctrl.v`).
4. Enumereu i descriviu els errors que heu trobat i com els heu corregit.
5. Contesteu les següents preguntes a l'informe de la pràctica.
 - Perquè la generació del senyal *Start* és de 6 etapes en lloc de 4?
 - Quina és la fórmula que ens dona la freqüència de rellotge o temps de bit?
 - Com s'implementa la detecció de col·lisions? I el *clock stretching*?Dibuixeu l'esquema RTL de la lògica i mostreu una captura de simulació a nivell RTL que ho mostri com funcionen.
6. Verifiquen que el codi es sintetitzable (sintetitzeu-lo per la FPGA Cyclone IV-E EP4CE22F17C6) per una freqüència de rellotge del sistema de 100 MHz (genereu el fitxer de restriccions temporals bàsic!).

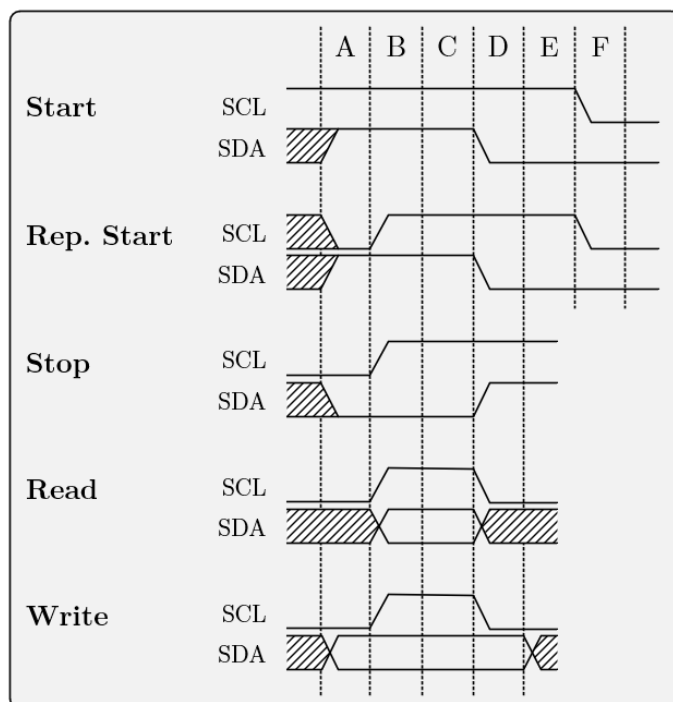


Figura 8 Les cinc operacions possibles de bit i la seva divisió en etapes.

Entrega

Un fitxer ZIP amb el directori de treball:

1. A la carpeta **rtl**: el codi RTL.
2. A la carpeta **tb**: el codi del testbench complet.
3. Dins la carpeta **doc** hi heu de posar l'informe complimentat que trobareu al campus virtual, que constarà de:
 - Verificació funcional: Enumereu i indiqueu les línies de codi que heu modificat perquè funcioni correctament el Controlador de Comandes de Bit.
 - Tipus i explicació de la màquina d'estats i diagrama d'estats (correcte).
 - Responen les qüestions.