

Alexander Paquette

EE316

Last Edited 02/05/2026

The I2C design that I have created has a few important components that help the system work as intended:

1. Clock Divider:

- Takes a 100MHz system clock, divides it down to 100kHz.
- Toggles SCL every time the divider hits the limit.
- Produces scl_en = '1' once per half-period so the FSM only advances on SCL edges.

2. SDA Logic:

- If sda_oe = '1' and sda_int = '0' => drive SDA low, otherwise => display SDA = 'Z'.
- External pull-up.

3. Internal Registers:

- Addr_w stores the 7-bit address and the R/W bit.
- data_tx stores the byte we want to write.
- Data_rx stores the byte we want to read.
- Bit_cnt counts down from 7 to 0 for each bit of a byte.

4. FSM:

- Shown in figure 1:

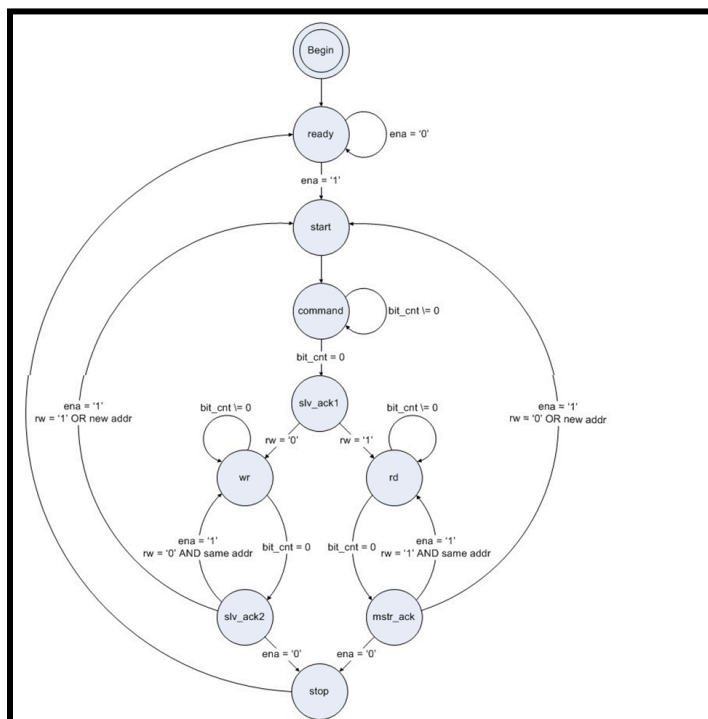


Figure 1: I2C FSM.

5. SDA / SCL Ports:

- Scl is driven push-pull
- Sda is open-drain (which is correct for I2C).

The waveform produced by the testbench for my system is shown below in figure 2:

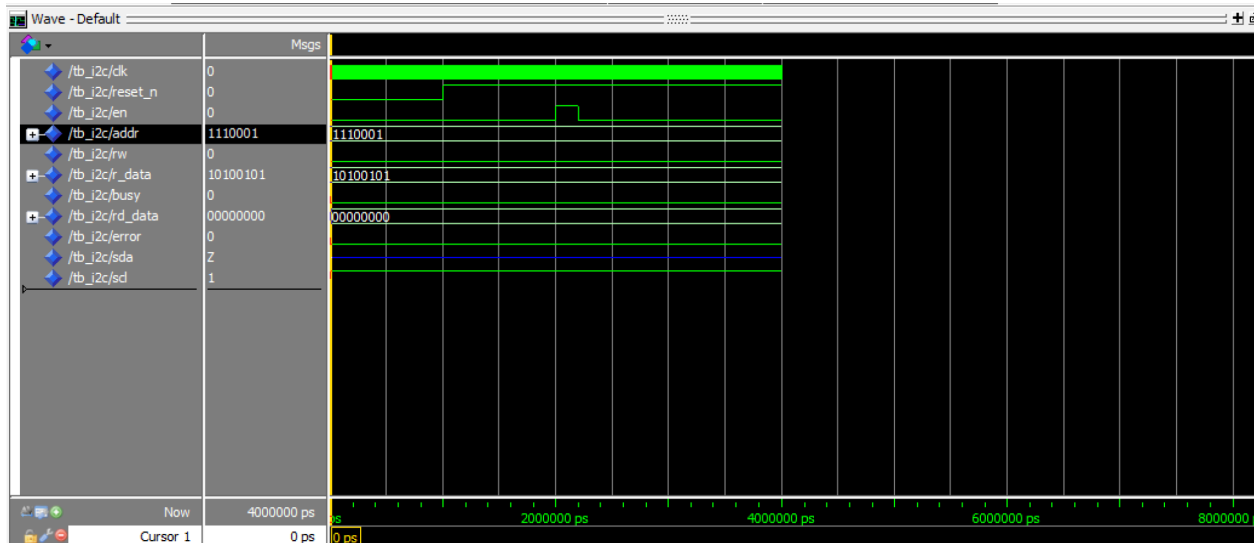


Figure 2: I2C Testbench Waveform

NOTE:

Inspiration for this code was taken from the “I2C Controller” Slides provided in the EE316 Moodle page.

This READ ME document and the I2C code will be uploaded to the Project 2 GitHub repository.