

→ motive: to connect our low-level i2c - master to a higher level system bus using a "simple" register interface

module shu-i2c-core (

 input logic cs → goes high when
 All read/write operation master wants to
 are ignored if cs talk to this specific
 is low.

 — read, write → when CPU
 wants to read from
 this module.

 — [4:0] adds: Address bus to
 select which internal
 register to read / write from

 — wr-data

 — rd-data.

 — tri su, sda);

// signal declaration

logic [15:0] dwsr_reg ;
____ wr-i2c, wr-dwsr ; → one cycle pulses

wr_i2c : send a command
to i2c-master now → write a divisor
register now.

____ [7:0] dcnt

____ read, ack.

// i2c master instantiation

// registers

always @ (_____)

if (wr-dwsr) → high

dwsr_reg <= wr-data[15:0];

// decoding

WR-DVGR \leftarrow when chip is selected,
& in write operation,
& address is $\text{xx}61$

WR-I2C \leftarrow chip is selected, write
op & address is $\text{xx}10$

// read data.

rd-data = {22'bo, ack, ready, dont};

[7:0] \rightarrow dont

[8] \rightarrow ready

[9] \rightarrow ack

[31]:10] \rightarrow 22'bo

Flow

① CPU configures speed

(write operation)

\rightarrow CS = 1, write = 1, read = 0

→ $\text{addr} = 5' b \dots 01$ (Address 1).

→ $\text{wr_data}[15:0] = 125 \rightarrow 125$ to set
dver to 100kHz

→ This asserts wr_dver &
dver_ready becomes 125.

② CPU sends START.

(write operation)

→ $\text{CS} = 1, \text{write} = 1, \text{read} = 0$

→ $\text{addr} = 5' b \dots 10$ (Address 2)

→ $\text{wr_data}[10:8] = 3'b000$ (start_CMD)

→ This asserts wr_i2c & the i2c master
starts generating a START condition.

③ CPU Polls for Ready

(read operation)

→ $\text{CS} = 1, \text{write} = 0, \text{read} = 2$

→ addr (doesn't matter)

→ It checks rd_data[3] (ready)

in a loop until it sees ready == 1.

④ CPU sends Address

(write operation)

→ $\text{CS} = 1, \text{write} = 1, \text{read} = 0$

→ $\text{addr} = 5' b \dots 20$ (Address 2).

→ $\text{wr_data}[10:8] = 3'b001$ (WR-(MD))

→ wr-data[7:0] = 8'h78 (slave address + write bit).

→ This asserts wr-i2c, and I2C master sends the byte.

⑤

CPU Polls for Ready & check ACK:

- CPU polls rd-data[8] (ready) again until 1.
- When ready == 1, it also checks rd-data[9] (ack). If ack == 1, the slave didn't respond, the CPU should abort.

