I2c.v – Whitley Forman

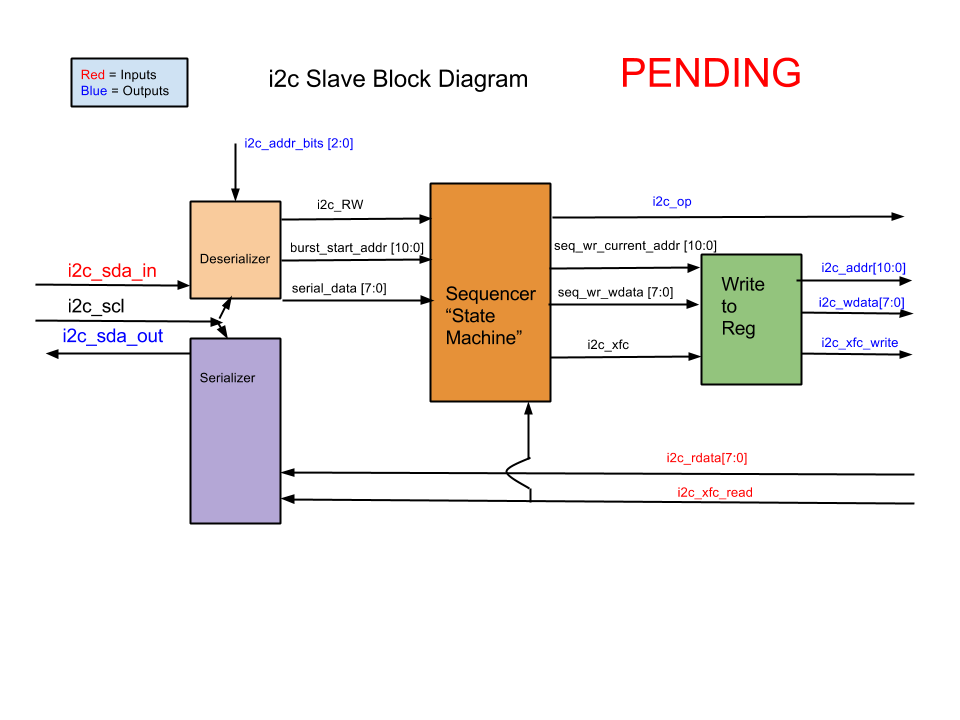
**Interfaces**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i2c interface |  |  |  |  |
|  | clk | in | 1 | clock |
|  | rst | in | 1 | reset |
|  | i2c\_scl | in | I2C Serial Clock | serial clock |
|  | i2c\_sda\_in | in | I2C Serial Data | external pin combines in and out using open drain |
|  | i2c\_sda\_out | out | I2C Serial Data | external pin combines in and out using open drain |
|  | i2c\_op | out | 1 | 1: write 0: read |
|  | i2c\_addr | out | 11 | register address |
|  | i2c\_wdata | out | 8 | data to be written for a write op |
|  | i2c\_xfc\_write | out | 1 | transfer complete |
|  | i2c\_xfc\_read | in | 1 | transfer complete |
|  | i2c\_addr\_bits | in | 3 | 3 LSB i2c address select |
|  | i2c\_rdata | in | 8 | data read from register block |

**Functional Requirements**

* I2c Address Selection
  + Provide a user definable address
  + 3 pin DIP Switch for 3 LSB of device address
* Receive and Send Read Requests from i2c bus Master
  + De-serialize incoming data from i2c bus
  + Respond to i2c address specified by external DIP switches
    - Read
      * Transfer Read Opcode and address to register block to be read
      * Serialize the returned data to the i2c bus
    - Write
      * Gather de-serialized data in bursts and increment corresponding register address accordingly
      * Transfer Write Opcode and register along with correlating data to register block
* Electrical and Data Rate
  + Operate at a minimum of 100kb/s to 400 kb/s for SCL Clock
  + Operate with pull-up resistors at 3.3Vdc and 5Vdc

**Micro Architecture**



|  |  |  |  |
| --- | --- | --- | --- |
| i2c internal |  |  |  |
|  | i2c\_RW | 1 | read/write command from deserializer |
|  | burst\_start\_addr | 11 | address from master, address start of burst data/write addr |
|  | serial\_data | 11 | data bursts |
|  | seq\_wr\_current\_addr | 11 | updated register address to write to |
|  | seq\_wr\_wdata | 8 | data corresponding to current address |
|  | i2c\_xfc | 1 | transfer complete, sequencer to write to reg |

**Verification**

* + Test Bench
    - Different addresses
    - Red and write
    - Burst write

FPGA

**Possibilities**

* + Possible error detection for invalid register address
  + Burst Overrun detection or overwrite from beginning