APB I2C SLAVE

The I2C slave responds to the transaction made by the master.

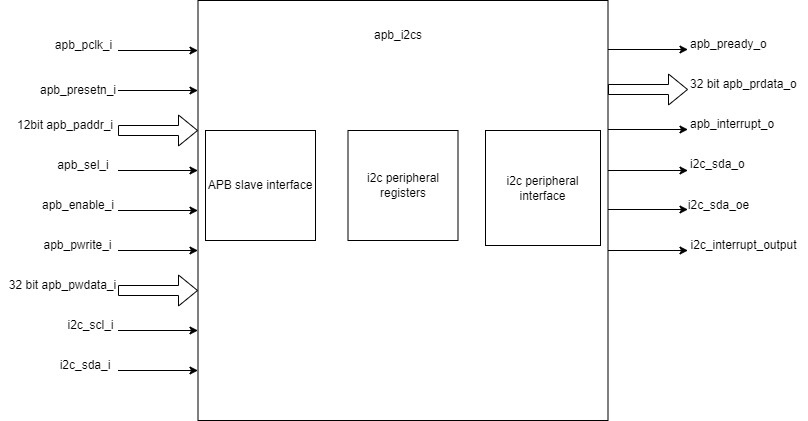
FEATURES:

* Supports 32 bit read and write data.
* Interrupts may be generated in any direction.
* Support 7 bit I2C addressing.
* Data is transferred in the sequence of 8 bits.

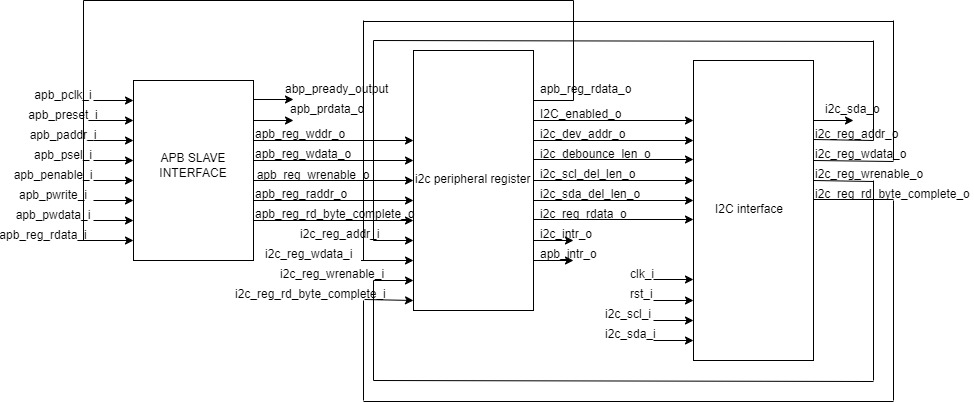
THEORY OF OPERATION:

I2C slave contains i2c peripheral interface and APB slave interface. There are FIFO and registers for handling communication with external I2C controllers.

Block diagram of APB I2C Peripheral:



Block diagram of internal modules:



APB SLAVE INTERFACE:

* APB slave interface is driving the APB output ports and input to register module.
* A successful reset will clear apb\_reg\_waddr\_o, apb\_reg\_wdata\_o, apb\_reg\_wrenable\_o, apb\_reg\_rd\_byte\_complete\_o.
* apb\_pready\_o is driven high if apb\_psel\_i and apb\_penable\_i are high.
* apb\_prdata\_o is driven by apb\_reg\_rdata\_i which we get from the i2c peripheral register.
* apb\_reg\_raddr\_o and apb\_reg\_waddr\_o are driven by apb\_paddr\_i.
* apb\_reg\_wdata\_o is driven by apb\_pwdata\_i.
* apb\_reg\_wrenable\_o is driven high if apb\_psel\_i, apb\_penable\_i and apb\_pwrite\_i are high else it is driven low.
* apb\_reg\_rd\_byte\_complete\_o is driven high if apb\_psel\_i and apb\_penable\_i are high and apb\_pwrite\_i is low else it is driven low.

I2C PERIPHERAL REGISTER:

I2C peripheral register is assigning value to CSRs and driving the interrupt port for APB and I2C. It is taking input from APB slave interface and i2c interface.

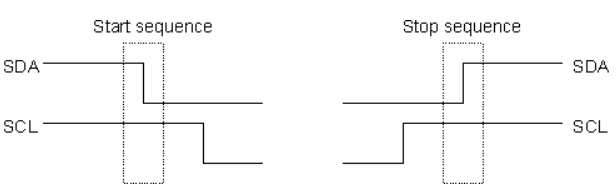
There are two FIFO instantiated in this module:

* FIFO\_sync\_256x8\_i2c\_to\_apb: Transfer data from i2c to apb.
* FIFO\_sync\_256x8\_apb\_to\_i2c: Transfer data from apb to i2c.

I2C PERIPHERAL INTERFACE:

START AND STOP CONDITION:

* The start and stop sequence mark the start and end of the transaction.



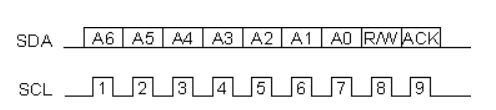
* To generate the start condition, the data line should change from high to low while the clock is high.
* To generate the stop condition, the data line should change from low to high while the clock is high.

READ/WRITE BIT:

* When sending out the 7 bit address we still send 8 bits. The last bit is used to inform if the master wants to write to the slave or read from the slave. If the bit is 0, master is writing to the slave else it is reading from the slave.

ACKNOWLEDGEMENT BIT:

* For every 8 bit transfer the device receiving the data sends an acknowledgement bit.



* Low acknowledgement bit sent by the receiving device indicate that it has received the data and it is ready to accept another byte.
* High acknowledgement bit sent by the receiving device indicate that it cannot accept new data and the master should terminate the transfer.

I2C STATES:

* I2C slave has 10 states:
  + ST\_IDLE:
    - Initially slave is in this state.
    - Slave may also come to this state if stop is detected.
  + ST\_DEVADDR:
    - Slave comes to this state after the start sequence is detected and i2c\_enabled\_i is high.
    - Slave receives the device address and transfer type (read/write).
    - Stop the transfer if the device address is not received.
  + ST\_DEVADDRACK:
    - Slave comes to this state after receiving the i2c device address and sends the acknowledgement.
    - i2c\_sda\_o is driven low to indicate successful acknowledgement.
    - Acknowledgement is released by driving i2c\_sda\_o to high before new transfer.
    - Read operation sets i2c state to ST\_REGRDATA. I2c\_reg\_rddata\_i is driven to i2c\_sda\_o.
    - Write operation sets i2c state to ST\_REGADDR.
  + ST\_REGADDR:
    - If the master wants to write then the slave comes to this state.
    - Slave receives the register address inside the device where the master wants to write. This register address is driven to i2c\_reg\_addr\_o.
  + ST\_REGADDRACK:
    - After receiving the register address successfully the slave comes to this state and sends acknowledgement.
    - i2c\_sda\_o is driven low to indicate successful acknowledgement.
    - Acknowledgement is released by driving i2c\_sda\_o to high before new transfer.
  + ST\_REGWDATA:
    - After sending acknowledgement, the slave comes to this state and writes data to the register.
    - I2c\_reg\_wrenable is driven high.
  + ST\_REGWDATAACK:
    - After successfully writing the data, an acknowledgement bit is sent.
    - I2c\_reg\_wrenable is driven low.
    - i2c\_sda\_o is driven low to indicate successful acknowledgement.
    - Acknowledgement is released by driving i2c\_sda\_o to high before new transfer.
  + ST\_REGRDATA:
    - Slave comes to this state if the master wants to read the data.
    - After successful read i2c\_rd\_byte\_complete is driven high.
  + ST\_REGRDATAACK:
    - After successful reading, acknowledgement is received.
    - I2c\_rd\_byte\_complete is cleared.
    - If negative acknowledgement is received, transfer is stopped.
    - If successful acknowledgement is received then i2c state is set to ST\_REGRDATA and more data is read.
  + ST\_WTSTOP:
    - Slave comes to this state if there is no more transaction or we want to stop the transfer.

APB I2C CSR’s:

I2CS\_DEV\_ADDRESS:

Offset = 0x000

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:7 | RW |  | Reserved |
| SLAVE\_ADDR | 6:0 | RW | 0X6F | I2C device address |

I2CS\_ENABLE:

Offset = 0X004

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:1 | RW |  | Reserved |
| IP\_ENABLE | 0:0 | RW |  | IP enabling bit |

I2CS\_DEBOUNCE\_LENGTH:

Offset = 0x008

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| DEB\_LEN | 7:0 | RW | 0X14 | Represents the number of system clocks over which each I2C line (SL and SDA) should be debounced. |

I2CS\_SCL\_DELAY\_LENGTH:

Offset = 0x00C

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| SCL\_DLY\_LEN | 7:0 | RW | 0X14 | Represents the number of system clocks over which the SCL line will be delayed relative to SDA line. |

I2CS\_SDA\_DELAY\_LENGTH:

Offset = 0x010

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| SDA\_DLY\_LEN | 7:0 | RW | 0X14 | Represents the number of system clocks over which the SDA line will be delayed relative to the SCL line. |

I2CS\_MSG\_I2C\_APB:

Offset = 0x040

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| I2C\_TO\_APB | 7:0 | RW | 0X00 | This register provide a method for passing a single byte message from the I2C interface to the APB interface. |

I2CS\_MSG\_I2C\_APB\_STATUS:

Offset = 0x044

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:1 | RW |  |  |
| I2C\_TO\_APB\_STATUS | 0:0 | RW | 0x00 | This register provides a method for passing a single byte message from the I2C interface to the APB interface. |

I2CS\_MSG\_APB\_I2C:

Offset = 0x048

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| APB\_TO\_I2C | 7:0 | RW | 0X00 | This register provides a method for passing a single byte message from the APB interface to the I2C interface. |

I2CS\_MSG\_APB\_I2C\_STATUS:

Offset = 0x4C

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:1 | RW |  |  |
| APB\_TO\_I2C\_STATUS | 0:0 | RW | 0X00 | This register provides a method for passing a single byte message from the APB interface to the I2C interface. |

I2CS\_FIFO\_I2C\_APB\_WRITE\_DATA\_PORT:

Offset = 0x080

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| I2C\_APB\_WRITE\_DATA\_PORT | 31:0 | RW |  | This is the write data port for the I2C to APB fifo. |

I2CS\_FIFO\_I2C\_APB\_READ\_DATA\_PORT:

Offset = 0x084

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| I2C\_APB\_READ\_DATA\_PORT | 31:0 | RW |  | This is the read data port for the I2C to APB FIFO. |

I2CS\_FIFO\_I2C\_APB\_FLUSH:

Offset = 0x088

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:1 | RW |  | RESERVED |
| ENABLE | 0:0 | RW |  | Writing a 1 to this register bit will flush the I2CtoAPB FIFO clearing all the contents and rendering the FIFO to be empty. |

I2CS\_FIFO\_I2C\_APB\_WRITE\_FLAGS:

Offset = 0x08C

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | RW |  | Reserved |
| FLAGS | 2:0 | RW |  | Represent the number of spaces left in FIFO. |

I2CS\_FIFO\_I2C\_APB\_READ\_FLAGS:

Offset = 0x90

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | RW |  | Reserved |
| FLAGS | 2:0 | RW |  | Represent the items present in FIFO to read. |

I2CS\_FIFO\_APB\_I2C\_WRITE\_DATA\_PORT:

Offset = 0X0C0

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| I2C\_APB\_WRITE\_DATA\_PORT | 31:0 | RW |  | This is the write data port for the APBtoI2C FIFO |

I2CS\_FIFO\_APB\_I2C\_READ\_DATA\_PORT:

Offset = 0X0C4

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| I2C\_APB\_READ\_DATA\_PORT | 31:0 | RW |  | This is the read data port for the APBtoI2C FIFO. |

I2CS\_FIFO\_APB\_I2C\_FLUSH:

Offset = 0X0C8

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:1 | RW |  | RESERVED |
| ENABLE | 0:0 | RW |  | Writing a 1 to this register bit will flush the APBtoI2C FIFO, clearing all contents and rendering the FIFO to be empty. |

I2CS\_FIFO\_APB\_I2C\_WRITE\_FLAGS:

Offset = 0X0CC

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | R |  |  |
| FLAGS | 2:0 | R |  | Represent number of spaces left in FIFO |

I2CS\_FIFO\_APB\_I2C\_READ\_FLAGS:

Offset = 0X0D0

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | R |  |  |
| FLAGS | 2:0 | R |  | Represent the items present in FIFO to read. |

I2CS\_INTERRUPT\_STATUS:

Offset = 0x100

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | R |  | Reserved |
| I2C\_APB\_FIFO\_WRITE\_STATUS | 2:2 | R |  | 1: Interrupt is generated for this field 0: Not genertated |
| APB\_I2C\_FIFO\_READ\_STATUS | 1:1 | R |  | 1: Interrupt is generated for this field 0: Not genertated |
| APB\_I2C\_MESSAGE\_AVAILABLE | 0:0 | R |  | 1: Interrupt is generated for this field 0: Not genertated |

I2CS\_INTERRUPT\_ENABLE:

Offset = 0x104

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | RW |  | Reserved |
| I2C\_APB\_FIFO\_WRITE\_STATUS\_INT\_ENABLE | 2:2 | RW |  | 1: enabled |
| APB\_12C\_FIFO\_READ\_STATUS\_INT\_ENABLE | 1:1 | RW |  | 1: enabled |
| APB\_I2C\_MESSAGE\_AVAILABLE\_INT\_ENABLE | 0:0 | RW |  | 1: enabled |

I2CS\_INTERRUPT\_I2C\_APB\_WRITE\_FLAGS\_SELECT:

Offset = 0x108

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| WRITE\_FLAG\_FULL | 7:7 | RW |  | 1:The write FIFO is full |
| WRITE\_FLAG\_1\_SPACE\_AVAIL | 6:6 | RW |  | 1: one space left |
| WRITE\_FLAG\_2\_3\_SPACE\_AVAIL | 5:5 | RW |  | 1: 2-3 spaces left |
| WRITE\_FLAG\_4\_7\_SPACE\_AVAIL | 4:4 | RW |  | 1: 4-7 spaces left |
| WRITE\_FLAG\_8\_31\_SPACE\_AVAIL | 3:3 | RW |  | 1: 8-31 spaces left |
| WRITE\_FLAG\_32\_63\_SPACE\_AVAIL | 2:2 | RW |  | 1: 32-63 spaces left |
| WRITE\_FLAG\_64\_127\_SPACE\_AVAIL | 1:1 | RW |  | 1: 64-127 spaces left |
| WRITE\_FLAG\_128\_\_SPACE\_AVAIL | 0:0 | RW |  | 1: 128+ spaces left |

I2CS\_INTERRUPT\_APB\_I2C\_READ\_FLAGS\_SELECT:

Offset = 0x10C

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| READ\_FLAG\_128\_SPACE\_AVAIL | 7:7 | RW |  | 1: 128 items present |
| READ\_FLAG\_64\_127\_SPACE\_AVAIL | 6:6 | RW |  | 1: 64-127 items to read |
| READ\_FLAAG\_32\_63\_SPACE\_AVAIL | 5:5 | RW |  | 1: 32-63 items present |
| READ\_FLAG\_8\_31\_SPACE\_AVAIL | 4:4 | RW |  | 1: 8-31 items |
| READ\_FLAG\_4\_7\_SPACE\_AVAIL | 3:3 | RW |  | 1: 4-7 items |
| READ\_FLAG\_2\_3\_SPACE\_AVAIL | 2:2 | RW |  | 1: 2-3 items |
| READ\_FLAG\_1\_SPACE\_AVAIL | 1:1 | RW |  | 1: 1 item |
| READ\_FLAG\_EMPTY | 0:0 | RW |  | 1: 0 items, empty |

I2CS\_INTERRUPT\_TO\_APB\_STATUS:

Offset = 0x140

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | RW |  | Reserved |
| APB\_I2C\_FIFO\_WRITE\_STATUS | 2:2 | RW |  | Interrupt status representing whether interrupt will generate or not. 1: Interrupt genereted |
| I2C\_APB\_FIFO\_READ\_STATUS | 1:1 | RW |  | Interrupt status representing whether interrupt will generate or not. 1: Interrupt generated |
| NEW\_I2C\_APB\_MSG\_AVAIL | 0:0 | RW |  | Interrupt status representing whether interrupt will generate or not. 1: Interrupt generated |

I2CS\_INTERRUPT\_TO\_APB\_ENABLE:

Offset = 0x0144

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| RESERVED | 7:3 | RW |  | Reserved |
| APB\_I2C\_FIFO\_WRTE\_STATUS\_ENABLE | 2:2 | RW |  | 1: enabled |
| I2C\_APB\_FIFO\_READ\_STATUS\_ENABLE | 1:1 | RW |  | 1: enabled |
| NEW\_I2C\_APB\_MSG\_AVAIL\_ENABLE | 0:0 | RW |  | 1: enabled |

I2CS\_INTERRUPT\_APB\_I2C\_WRITE\_FLAGS\_SELECT:

Offset = 0x148

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| WRITE\_FLAG\_FULL | 7:7 | RW |  | 1 : The Write FIFO is full |
| WRITE\_FLAG\_1\_SPACE\_AVAIL | 6:6 | RW |  | 1: one space left |
| WRITE\_FLAG\_2\_3\_SPACE\_AVAIL | 5:5 | RW |  | 1: 2-3 spaces left |
| WRITE\_FLAG\_4\_7\_SPACE\_AVAIL | 4:4 | RW |  | 1: 4-7 spaces left |
| WRITE\_FLAG\_8\_31\_SPACE\_AVAIL | 3:3 | RW |  | 1: 8-31 spaces left |
| WRITE\_FLAG\_32\_63\_SPACE\_AVAIL | 2:2 | RW |  | 1: 32-63 spaces left |
| WRITE\_FLAG\_64\_127\_SPACE\_AVAIL | 1:1 | RW |  | 1: 64-127 spaces left |
| WRITE\_FLAG\_128\_SPACE\_AVAIL | 0:0 | RW |  | 1: 128+ spaces left |

I2CS\_INTERRUPT\_I2C\_APB\_READ\_FLAGS\_SELECT:

Offset = 0x14C

| Field | Bits | Type | Default | Description |
| --- | --- | --- | --- | --- |
| READ\_FLAG\_128\_SPACE\_AVAIL | 7:7 | RW |  | 1: 128 items present |
| READ\_FLAG\_64\_127\_SPACE\_AVAIL | 6:6 | RW |  | 1: 64 - 127 items present |
| READ\_FLAG\_32\_63\_SPACE\_AVAIL | 5:5 | RW |  | 1: 32-63 items present |
| READ\_FLAG\_8\_31\_SPACE\_AVAIL | 4:4 | RW |  | 1: 8-31 items present |
| READ\_FLAG\_4\_7\_SPACE\_AVAIL | 3:3 | RW |  | 1: 4-7 items present |
| READ\_FLAG\_2\_3\_SPACE\_AVAIL | 2:2 | RW |  | 1: 2-3 items present |
| READ\_FLAG\_1\_SPACE\_AVAIL | 1:1 | RW |  | 1: 1 item present |
| READ\_FLAG\_EMPTY | 0:0 | RW |  | 1: 0 items, empty |