EE577B Final Project Phase II

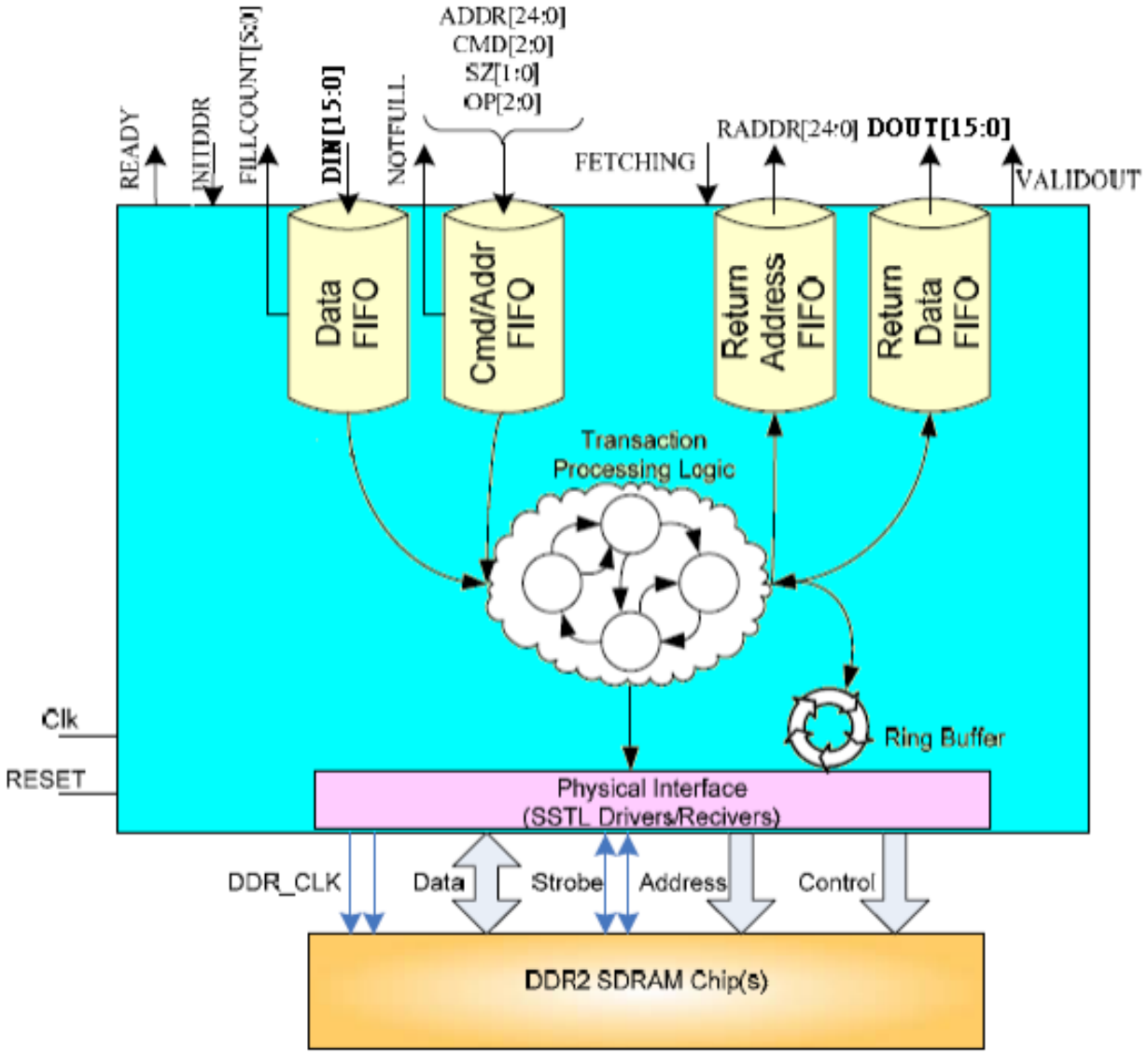
Fangzhou Wang

Mar 07, 2014

Introduction

- **Pre-requisite**
  - **DDR2 Initialization Engine**
  - **Confidence**
  - **Confidence**
- **Goal for Part I**
  - **Implement scalar read and write**
  - **Refresh logic**
- **Goal for Part II**
  - **Implement block read and write**

Overview – Part I



Implement RTL for scalar operations

I/O Name	Width	IN/OUT	Description
CMD[2:0]	3 bit	IN	000: No Operation (NOP) 001: Scalar Read (SCR) 010: Scalar Write (SCW) 011: Block Read (BLR) 100: Block Write (BLW) TBA

Scalar Read - SCR

- **Input:**
  - **A 25-bit address is given to the controller.**
- **Output:**
  - **A single 16-bit word data.**
- **Methods:**
  - **Read with auto pre-charge.**
  - **Read without pre-charge.**

Scalar Write - SCW

- **Input:**
  - **A 25-bit address is given to the controller.**
  - **A single 16-bit word data.**
- **Output:**
  - **None.**
- **Methods:**
  - **Write with auto pre-charge.**
  - **Write without pre-charge.**

Ring Buffer

```
// Capture data at the edges
always @(posedge fStrobe)
        case (count)
          0: r0 <= din;
          1: r2 <= din;
          2: r4 <= din;
          3: r6 <= din;
        endcase // case(counter)
always @(negedge fStrobe)
        case (count)
          0: r1 <= din;
          1: r3 <= din;
          2: r5 <= din;
          3: r7 <= din;
        endcase // case(counter)
```

Ring Buffer Cont.

- **Ports:**
  - **Dout**
  - **Listen**
  - **Strobe**
  - **readPtr**
  - **Din**
  - **reset**

RTL

- **What you are provided**
  - **ddr2_ring_buffer8.v (complete)**
  - **Processing_logic.v (incomplete)**
  - **tb.v (complete RTL test bench used for phase2-part1, your design should be 100% match this testbench)**
  - **ddr2_controller.tcl (complete tcl file used in synthesis step)**
  - **ddr2_test_pattern.dat (input test pattern for phase2-part1)**
  - **ddr2_out.dump (output pattern for you to check your result)**

RTL Cont.

- **What you need to modify:**
  - **Processing_logic.v**
    - ***(design the logic to implement Scalar Read/Write)***
  - **ddr2_initial_engine.v**
    - ***(change your previous settings, set BL=8, AL=1, CL=3)***
  - **ddr2_controller.v**
    - ***(change the FIFO size in order to match test bench, design the logic to implement FULL/EMPTY signal of each FIFO, design the interface of Processing_logic module)***
  - **FIFO.v**
    - ***(you may need to re-design your FIFO to meet the timing constraints)***

SSTL

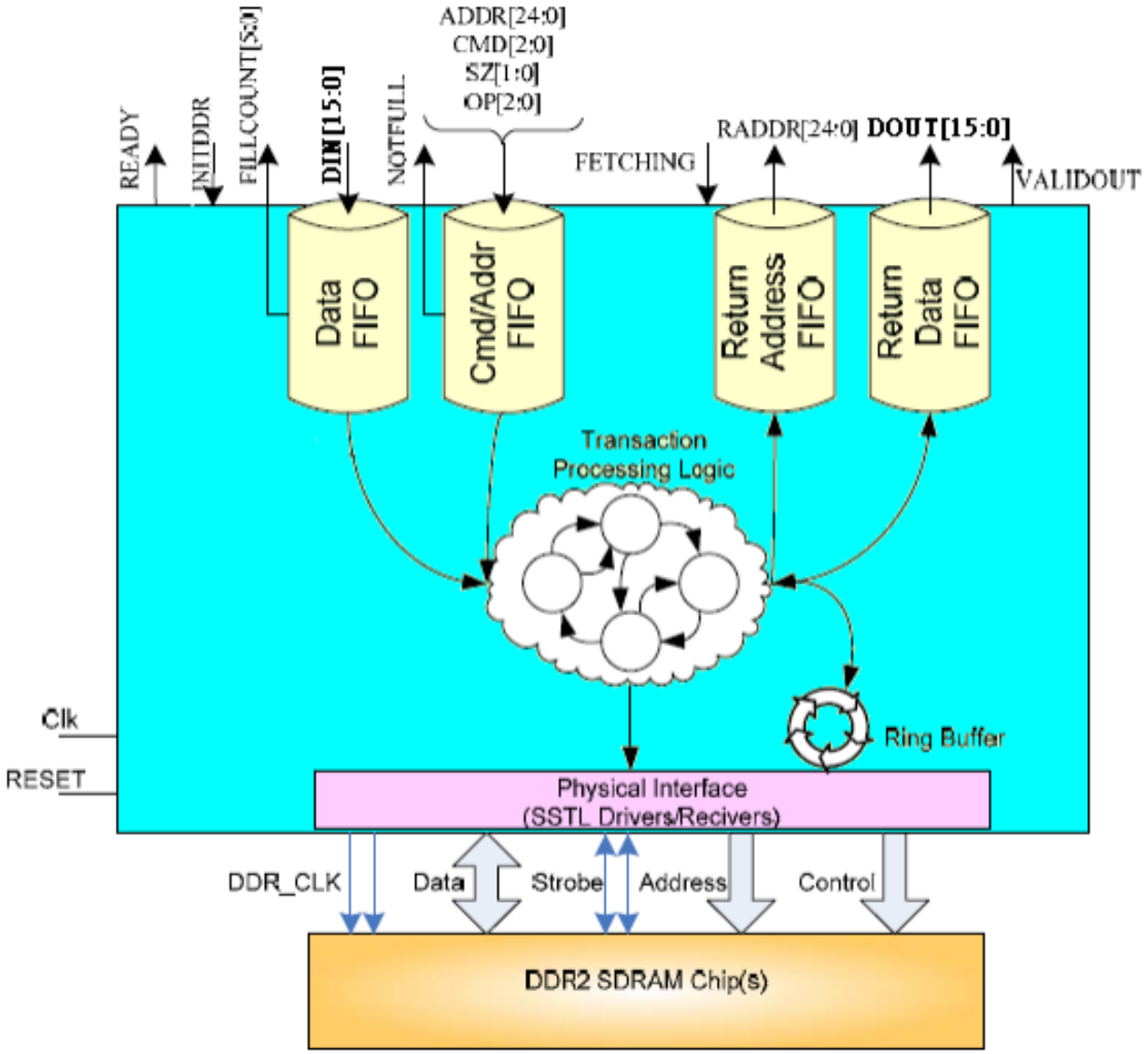
- module SSTL18DDR2 (PAD,Z,A,RI,TS);
  - inout PAD; // I/O PAD PIN
  - output Z; // Recieved data from PAD if RI is high
  - input A; // Data to PAD if TS is high
  - input RI; // Receiver Inhibit input
  - TS; // Driver Tristate Control
  - bufif1 b1 (PAD,A,TS);
  - and a4 (Z,PAD,RI);
- endmodule // SSTL18DDR2

- When sending data from the controller to Denali module, set TS=1 and RI = "don't care". When receiving data from Denali module to the controller, set TS= 0 and RI =1

Synthesize

- **What you need to synthesize**
  - **ddr2_controller.v**
  - **Ddr2_init_engine.v**
  - **Ddr2_ring_buffer8.v**
  - **FIFO.v**
  - **Processing_logic.v**
  - **SSTL18DDR2.v**
  - **SSTL18DDR2DIFF.v**
  - **SSTL18DDR2INTERFACE.v**

Overview – Part II



Important Note

- **Bank Interleaving**
  - **It improves the performance and is optional in this project. You are encouraged to do the bank interleaving.**

Implement RTL for block operations

I/O Name	Width	IN/OUT	Description
CMD[2:0]	3 bit	IN	000: No Operation (NOP) 001: Scalar Read (SCR) 010: Scalar Write (SCW) 011: Block Read (BLR) 100: Block Write (BLW) TBA
SZ[1:0]	2 bit	IN	00: Block size of 8 words 01: Block size of 16 words 10: Block size of 24 words 11: Block size of 32 words

Block Read - BLR

- **Input:**
  - **A 25-bit address is given to the controller.**
  - **Size is defined by sz[1:0].**
- **Output:**
  - **A block of 16-bit word data.**
- **Methods:**
  - **Use an internal counter.**
  - **Read with auto pre-charge.**
  - **Read without pre-charge.**

Block Write - BLW

- **Input:**
  - **A 25-bit address is given to the controller.**
  - **Size is defined by sz[1:0].**
  - **A block of 16-bit word data.**
- **Output:**
  - **None.**
- **Methods:**
  - **Use an internal counter.**
  - **Write with auto pre-charge.**
  - **Write without pre-charge.**

Refresh

- **DRAM needs to be refreshed after a certain amount of time in order to keep its data.**
  - **It can be implemented using a simple counter**
  - **It should block any further memory transaction before the refresh counter expires**
  - **To be on the safe side keep the refresh counter expiration such that the transactions in progress may not delay the refresh cycle**

Thank you

Fangzhou Wang

Mar 07, 2014