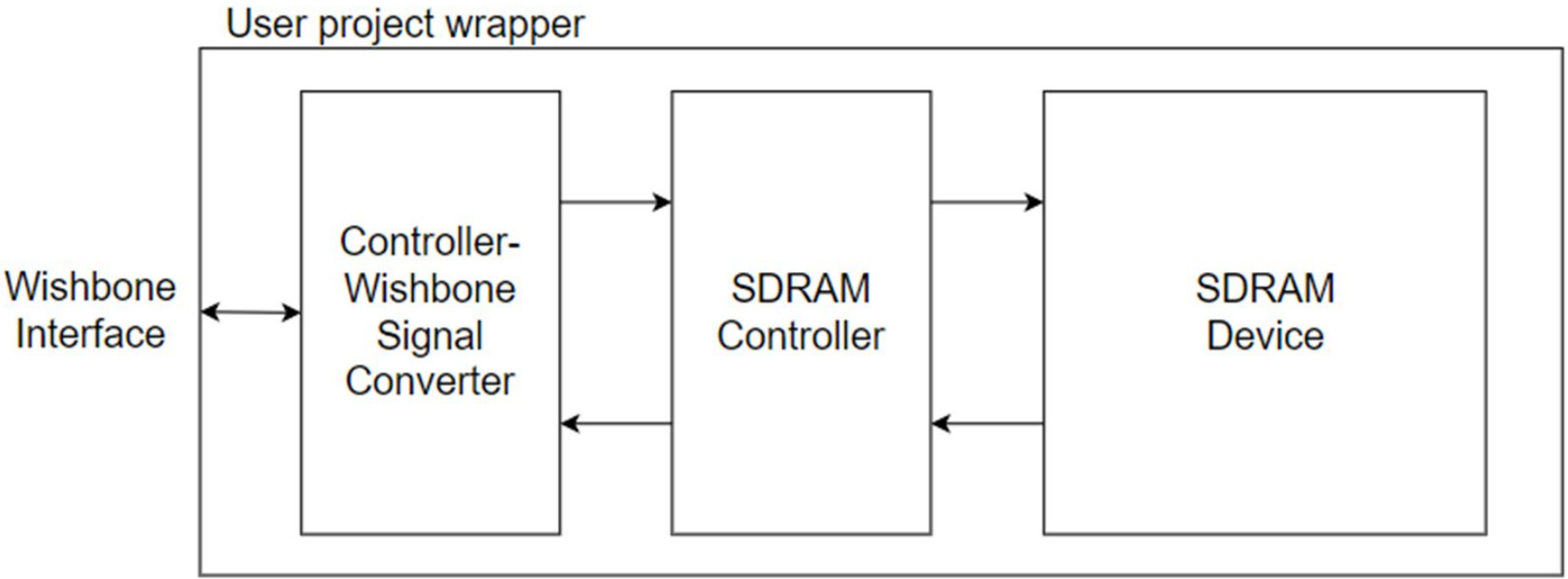
SDRAM Workbook (Lab #D)

SoC Design Laboratory

Lab #D – lab-sdram

- Refer to lab-exmem, but replace the BRAM with SDRAM (SDRAM controller + SDRAM)
  - SDRAM device - convert SDRAM model to hardware implement, memory array using BRAM
  - SDRAM controller - support page mode
- The combined SDRAM Controller + SDRAM device is to replace a Wishbone BRAM
- Reference:
  - https://github.com/bol-edu/caravel-soc_fpga-lab/tree/main/lab-sdram

SDRAM in Caravel User Project



SDRAM Controller

- SDRAM Device
  - sdram_cle, sdram_cs, sdram_cas, sdram_ras, sdram_we
  - sdram_dqm, sdram_ba, sdram_a
  - sdram_dqi, sdram_dqo
- User interface
  - user_addr
  - rw
  - data_in, data_out
  - busy
  - in_valid, out_valid

SDRAM Controller

- User interface
  - user_addr
    - address to read or write
  - rw
    - 1 for write command, 0 for read command
  - data_in
    - data from a read
  - data_out
    - data from a write

SDRAM Controller

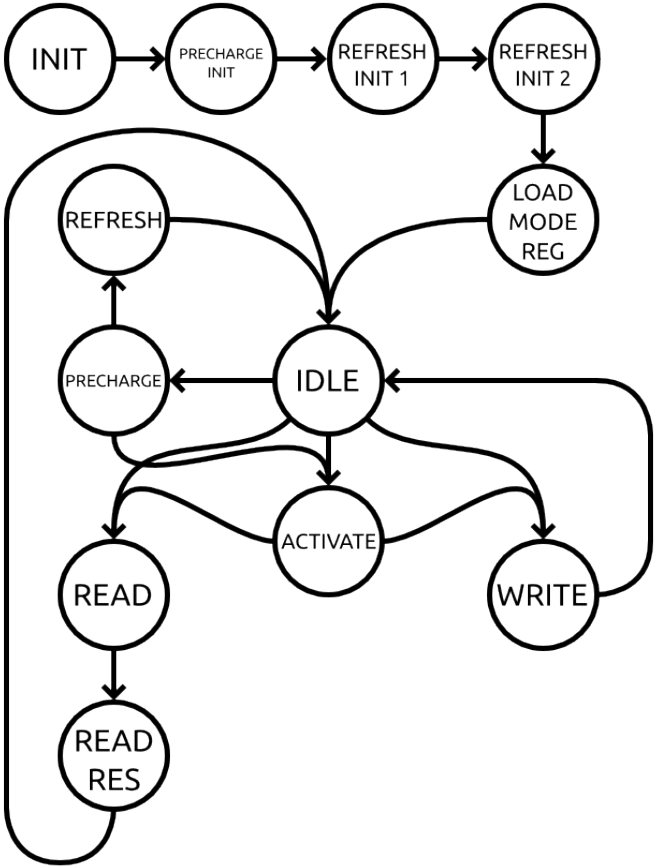
- User interface
  - busy
    - 1 for controller is busy, 0 for controller can get next command
  - in_valid
    - pulse high to initiate a read or write command
  - out_valid
    - pulse high when the data from read is valid

SDRAM Controller

- Controller state
  - INIT, WAIT, IDLE, REFRESH, ACTIVATE, READ, READ_RES, WRITE, PRECHARGE
- CAS Latency, Precharge Latency, Activate Latency
  - 3T
- Refresh Latency
  - 7T
- Refresh cycle
  - 750T

SDRAM Controller

- INIT→IDLE
- IDLE→ACTIVATE→WRITE→IDLE
- IDLE→ACTIVATE→READ→READ_RES →IDLE
- IDLE→WRITE→IDLE
- IDLE→READ→READ_RES→IDLE
- IDLE→PRECHARE→ACTIVATE→WRITE →IDLE
- IDLE→PRECHARE→ACTIVATE→READ →READ_RES→IDLE
- IDLE→PRECHARE→REFRESH→IDLE



Reference: https://alchitry.com/sdram-mojo

SDRAM Controller

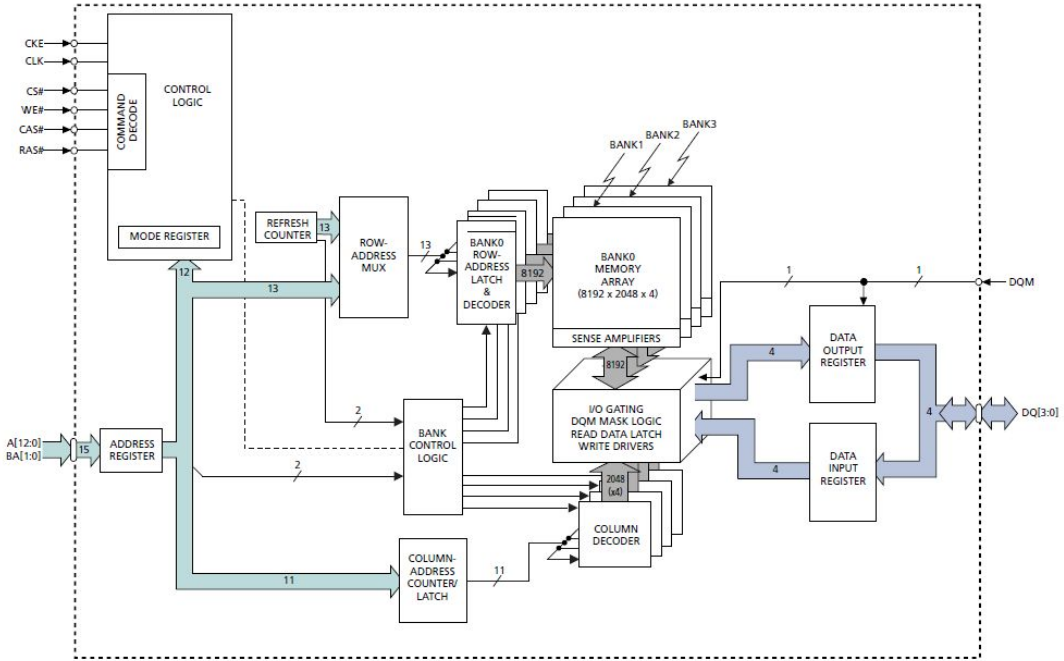
- User address remapping
- User address [22:0]
  - Bank address [9:8]
  - Row address [22:10]
  - Column address [7:0]
- Remap user address to create more off-page/on-page cases.

SDRAM Device

- Single Data Rate Synchrnous DRAM (SDR-SDRAM)
  - It need consider activate、precharge and refresh, when accessing device
- The behavior model is non-synthesable
  - parameter is defined in sdr_parameter.vh
  - FPGA doesn’t has inout port for interconnection
- Replace storage element by BRAM
  - Must fit the SDRAM behavior model
    - Column Address Strobe (CAS)
    - CAS Latency (CL)
    - Activate
    - Precharge
    - Refresh

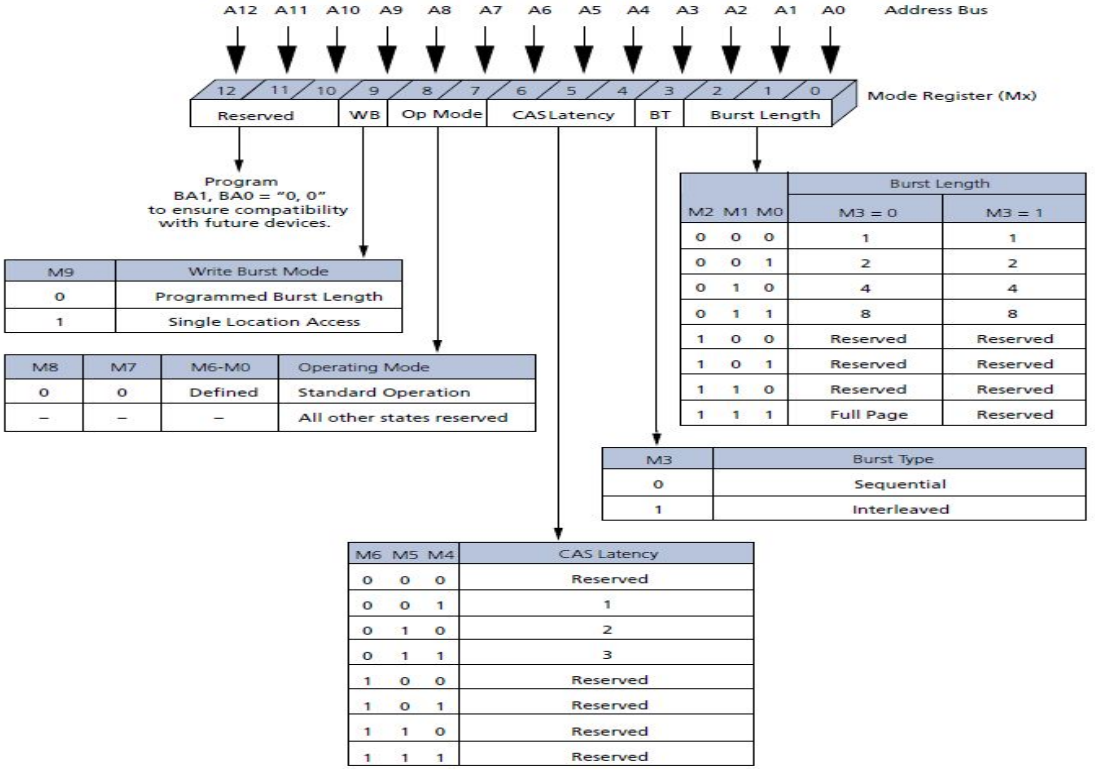
SDRAM Device - Behavior Model

- The behavior model refer to Micron MT48LC64M4A2
  - 16 Meg x 4 x 4 banks



SDRAM Device - Mode Register Definition

- Config mode register to specify operation of device



M9	Write Burst Mode
0	Programmed Burst Length
1	Single Location Access

M8	M7	M6-M0	Operating Mode
0	0	Defined	Standard Operation
–	–	–	All other states reserved

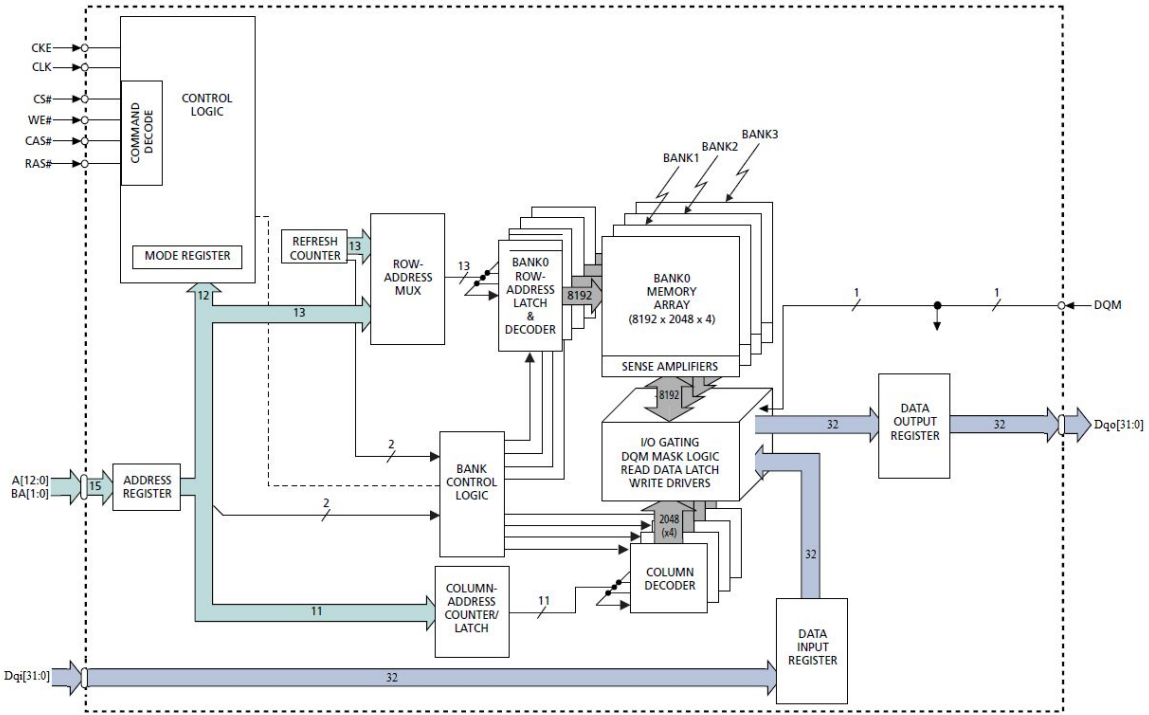
	Burst Length	
M2 M1 M0	M3 = 0	M3 = 1
0 0 0	1	1
0 0 1	2	2
0 1 0	4	4
0 1 1	8	8
1 0 0	Reserved	Reserved
1 0 1	Reserved	Reserved
1 1 0	Reserved	Reserved
1 1 1	Full Page	Reserved

M3	Burst Type
0	Sequential
1	Interleaved

M6 M5 M4	CAS Latency
0 0 0	Reserved
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	Reserved
1 0 1	Reserved
1 1 0	Reserved
1 1 1	Reserved

SDRAM Device - Block Diagram on FPGA

- Separate inout port with input and output port



Controller-Wishbone Signal Converter

```
assign valid = wbs_stb_i && wbs_cyc_i;
assign ctrl_in_valid = wbs_we_i ? valid : ~ctrl_in_valid_q && valid;
assign wbs_ack_o = (wbs_we_i) ? ~ctrl_busy && valid : ctrl_out_valid;
assign bram_mask = wbs_sel_i & {4{wbs_we_i}};
assign ctrl_addr = wbs_adr_i[22:0];

// IO
assign io_out = d2c_data;
assign io_oeb = {(`MPRJ_IO_PADS-1){rst}};

// IRQ
assign irq = 3'b000;    // Unused

// LA
assign la_data_out = {{(127-BITS){1'b0}}, d2c_data};
// Assuming LA probes [65:64] are for controlling the count clk & reset
assign clk = (~la_oenb[64]) ? la_data_in[64]: wb_clk_i;
assign rst = (~la_oenb[65]) ? la_data_in[65]: wb_rst_i;
assign rst_n = ~rst;

always @(posedge clk) begin
    if (rst) begin
        ctrl_in_valid_q <= 1'b0;
    end
    else begin
        if (~wbs_we_i && valid && ~ctrl_busy && ctrl_in_valid_q == 1'b0)
            ctrl_in_valid_q <= 1'b1;
        else if (ctrl_out_valid)
            ctrl_in_valid_q <= 1'b0;
    end
end
```

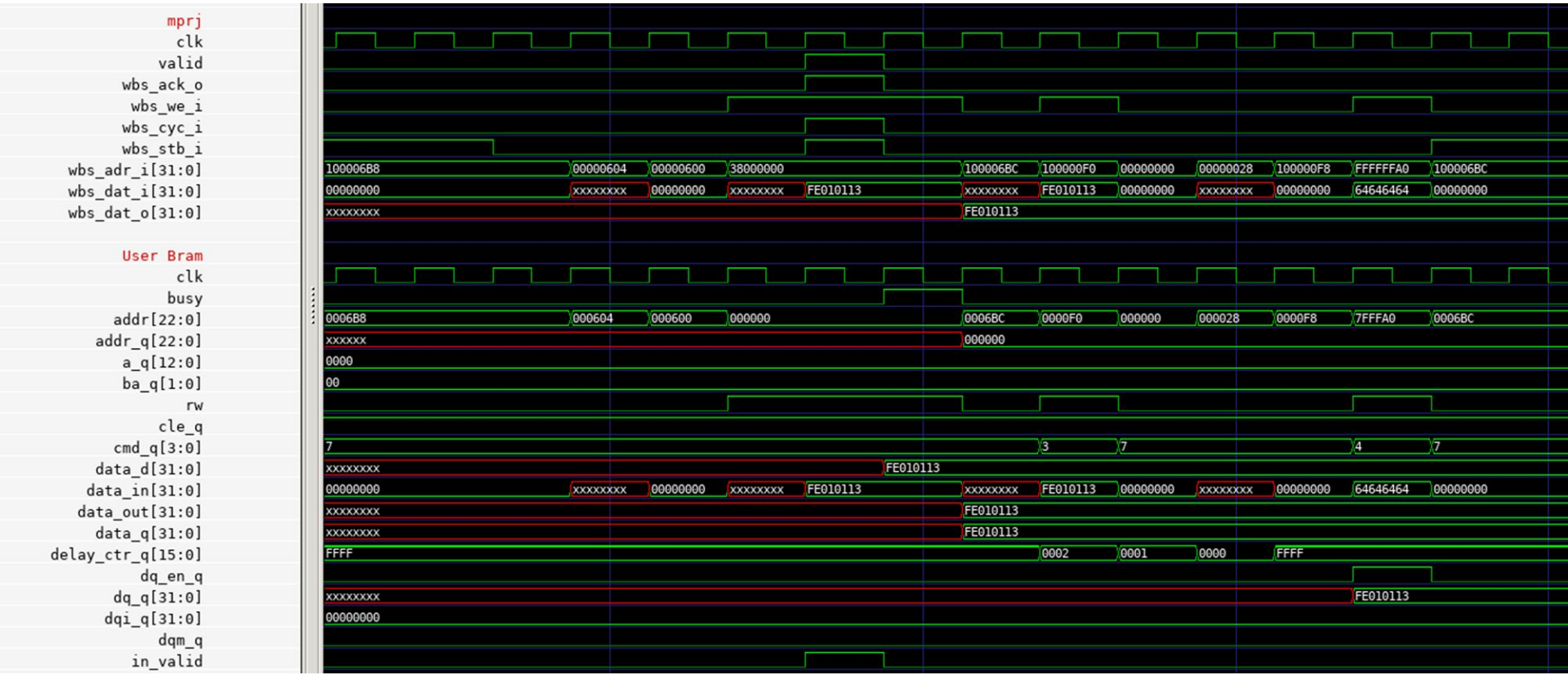
Controller-Wishbone Signal Converter

The controller has two output signals, **busy** and **out_valid**, for receiving a new command and telling the user interface that the output is ready from the read command.

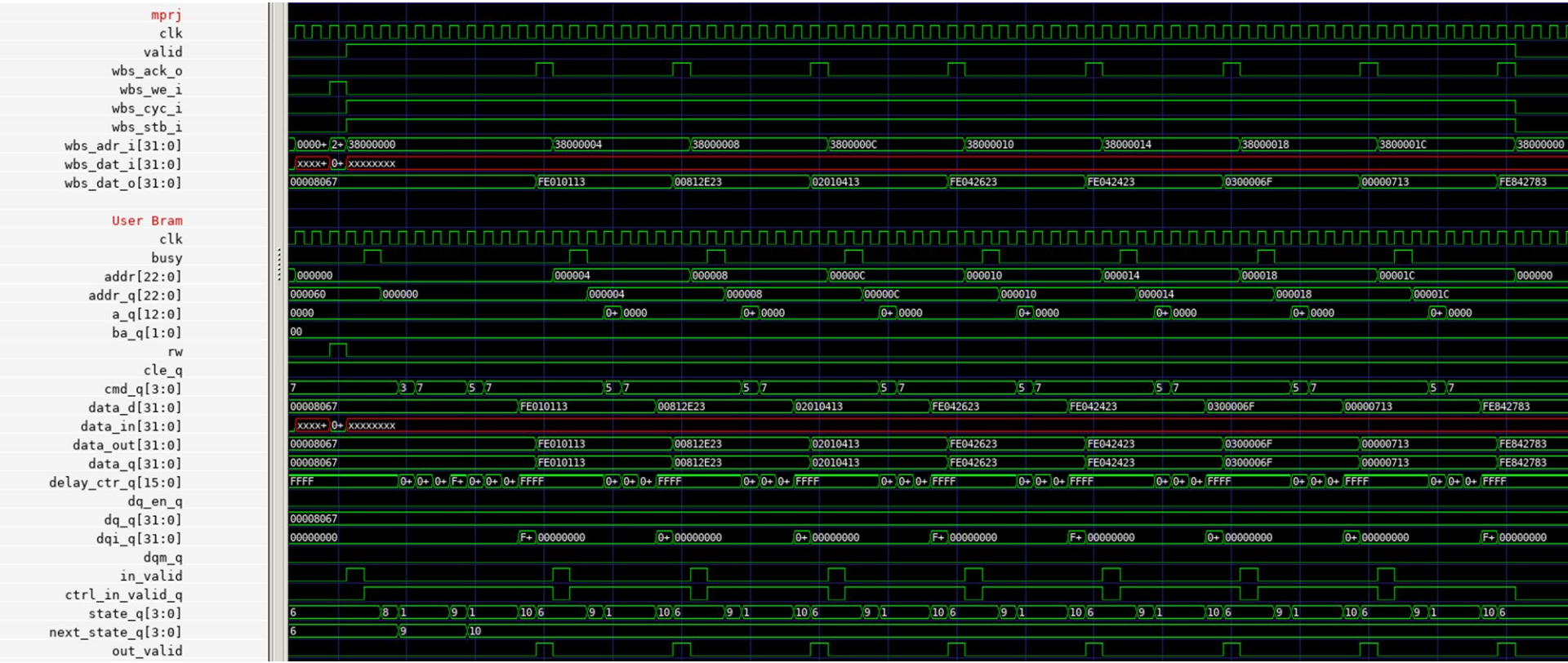
The Wishbone interface only has one output signal wbs_ack_o.

We need to convert two signals into one signal in this user project SDRAM.

Waveform - Write



Waveform - Read



Implementation

- Firmware matmul executed in SDRAM (provide matmul.hex)
- Add prefetch in SDRAM controller
  - Prefetch buffer serves as cache
  - Observe address access pattern, determine # of prefetch buffers, and prefetch scheme
- Adjust linear address map below, so that code execution and data fetch are from two different banks
  - Bank address [9:8]
  - Row address [22:10]
  - Column address [7:0]
- For the bank interleave, you may also change linker so that address of code, and data can take advantage of concurrent bank access. So is the prefetch.

Observe and write report on

- the SDRAM controller design, SDRAM bus protocol …
- Introduce the prefetch schme
- Introduce the bank interleave for code and data
- Introduce how to modify the linker to load address/data in two different bank
- Observe SDRAM access conflicts with SDRAM refresh (reduce the refresh period)
- others

Submission

- GitHub link
- Report