EF_PSRAM_CTRL

A Controller for Quad I/O SPI PSRAM Pseudostatic RAM (PSRAM) is DRAM combined with a self-refresh circuit. It appears externally as slower SRAM, albeit with a density/cost advantage over true SRAM, and without the access complexity of DRAM. The controller was designed after:

- https://www.issi.com/WW/pdf/66-67WVS4M8ALL-BLL.pdf and
- https://www.microchip.com/en-us/parametric-search/514

utilizing SPI, QSPI and QPI modes.

The controller was verified against the Verilog model of Microchip M23LC1024 for SPI and QPI modes only.

Features:

- Small in size (800 cells)
- Supports PSRAM or serial SRAM memories up to 8MBytes.
- Programmable to configure: read/write commands, number of wait states, enter and exist quad i/o commands.

The wrapped IP

The IP comes with an AHBL Wrapper

Wrapped IP System Integration

```
EF_PSRAM_CTRL_AHBL INST (
        .HCLK(CLK),
        .HRESETn(RESETn),
        .HADDR(HADDR),
        .HWRITE(HWRITE),
        .HSEL(HSEL),
        .HTRANS(HTRANS),
        .HWDATA(HWDATA),
        .HRDATA(HRDATA),
        .HREADY(HREADY),
        .HREADYOUT(HREADYOUT),
        .sck(sck),
        .ce_n(ce_n),
        .din(din),
        .dout(dout),
        .douten(douten)
);
```

External IO interfaces

IO name	Direction	Width	Description
sck	output	1	SPI master output clock
ce_n	output	1	SPI Master slave select.
din	input	4	SPI Master data in , slave out
dout	output	4	SPI Master data out , slave in
douten	output	4	SPI Master data out enable

Implementation example

The following table is the result for implementing the EF_PSRAM_CTRL IP with different wrappers using Sky130 HD library and OpenLane2 flow.

Module	Number of cells	Max. freq
EF_PSRAM_CTRL	TBD	TBD
EF_PSRAM_CTRL_AHBL	TBD	TBD

The Programmer's Interface

Memory Map

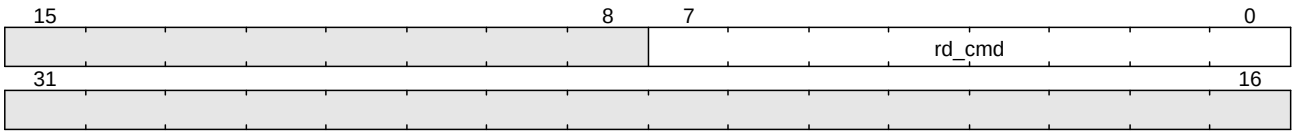
Region	Description
`0x07FF_FFFF - 0x0000_0000`	Data Access
`0x0FFF_FFFF - 0x0800_0000`	Configuration Registers

Registers

Name	Offset	Reset Value	Access Mode	Description
rd_cmd	800100	0x00000003	w	RD Command Register
wr_cmd	800200	0x00000002	w	WR Command Register
eqpi_cmd	800400	0x00000035	w	Enter QPI Command Register
xqpi_cmd	800800	0x000000FE	w	Exit QPI Command Register
wait_states	801000	0x00000000	w	Wait States Register
mode	802000	0x00000000	w	I/O Mode Register, {qpi, qspi}
enter_qpi	804000	0x00000000	w	Initiate Enter QPI (EQPI) Mode process Register
exit_qpi	808000	0x00000000	w	Initiate Exit QPI (XQPI) Mode process Register

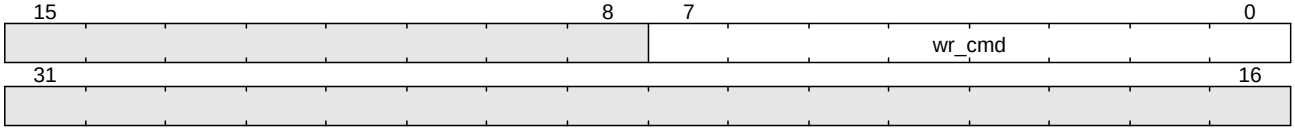
rd_cmd Register [Offset: 0x800100, mode: w]

RD Command Register



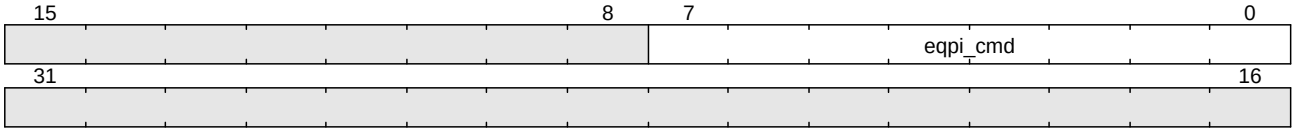
wr_cmd Register [Offset: 0x800200, mode: w]

WR Command Register



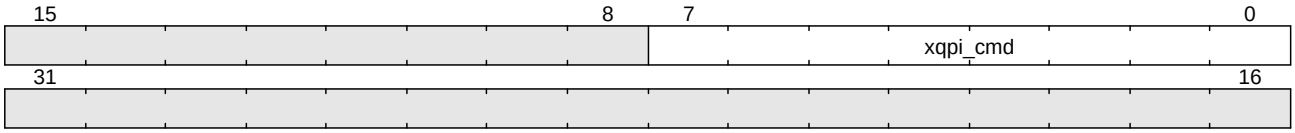
eqpi_cmd Register [Offset: 0x800400, mode: w]

Enter QPI Command Register



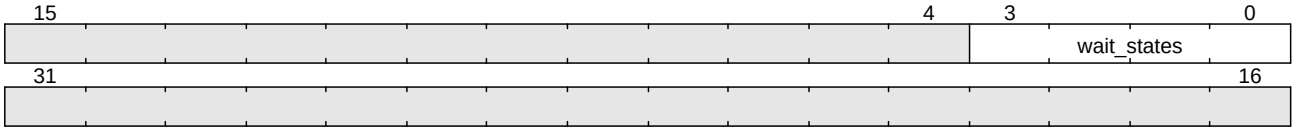
xqpi_cmd Register [Offset: 0x800800, mode: w]

Exit QPI Command Register



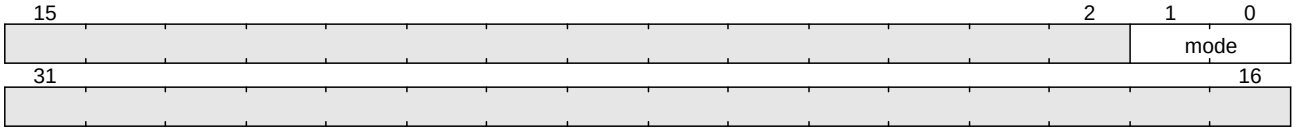
wait_states Register [Offset: 0x801000, mode: w]

Wait States Register



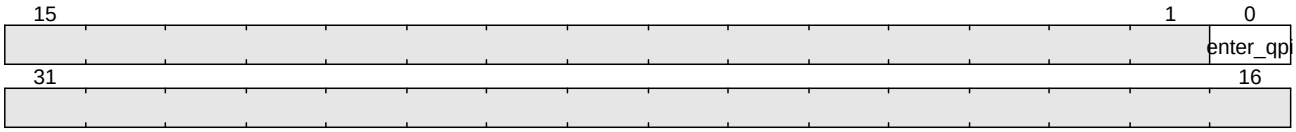
mode Register [Offset: 0x802000, mode: w]

I/O Mode Register, {qpi, qspi}



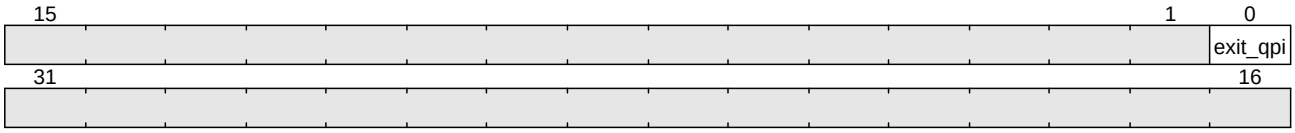
enter_qpi Register [Offset: 0x804000, mode: w]

Initiate Enter QPI (EQPI) Mode process Register



exit_qpi Register [Offset: 0x808000, mode: w]

Initiate Exit QPI (XQPI) Mode process Register



Firmware Drivers:

Firmware drivers for EF_PSRAM_CTRL can be found in the Drivers directory in the EFIS (Efabless Firmware Interface Standard) repo. EF_PSRAM_CTRL driver documentation is available here. You can also find an example C application using the EF_PSRAM_CTRL drivers here.

Installation:

You can install the IP either by cloning this repository or by using IPM.

1. Using IPM:

- [Optional] If you do not have IPM installed, follow the installation guide here
- After installing IPM, execute the following command `ipm install EF_PSRAM_CTRL .`

> **Note:** This method is recommended as it automatically installs EF_IP_UTIL as a dependency.

2. Cloning this repo:

- Clone EF_IP_UTIL repository, which includes the required modules from the common modules library, ef_util_lib.v.
  `git clone https://github.com/efabless/EF_IP_UTIL.git`
- Clone the IP repository `git clone github.com/efabless/EF_PSRAM_CTRL`

The Wrapped IP Interface

> ***NOTE:*** This section is intended for advanced users who wish to gain more information about the interface of the wrapped IP, in case they want to create their own wrappers.

Ports

Port	Direction	Width	Description
sck	output	1	SPI master output clock
ce_n	output	1	SPI Master slave select.
din	input	4	SPI Master data in , slave out
dout	output	4	SPI Master data out , slave in
douten	output	4	SPI Master data out enable