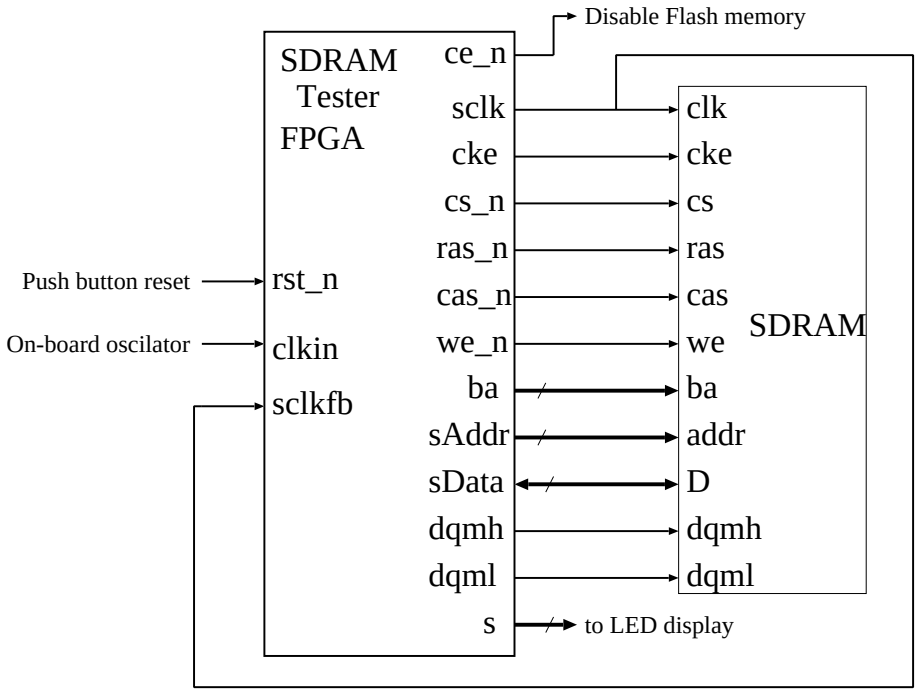
Design Example:
A Simple SDRAM Tester

Haibo Wang
ECE Department
Southern Illinois University
Carbondale, IL 62901

Design Objectives

- A simple testing procedure in which random numbers (generated by a LFSR) are written into the SDRAM, and then read back to compare.
- A simple SDRAM controller that provides a SRAM-like interface
- No pipeline, no burst operation mode, not programmable



- *The SDRAM controller is modified from XESS SDRAM controller application note*
- *The lecture also contain materials from Xilinx application notes XAPP174 and XAPP134*

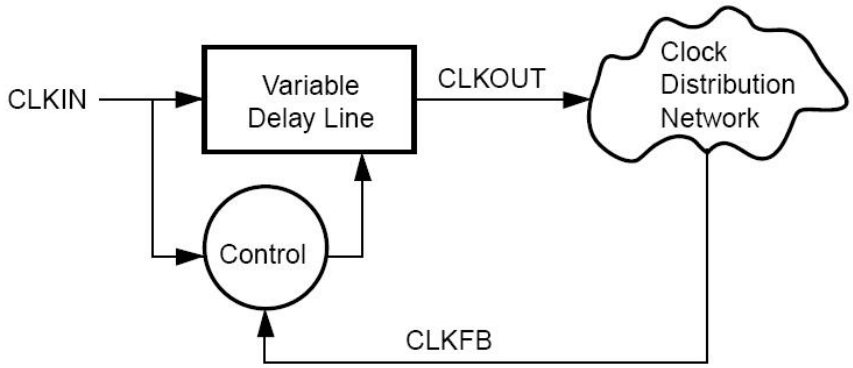
Introduction to Delay-Locked Loop (DLL)

- Delay-Locked Loop (DLL) and Phase-Locked Loop (PLL) are two types of components that used to remove clock skew.

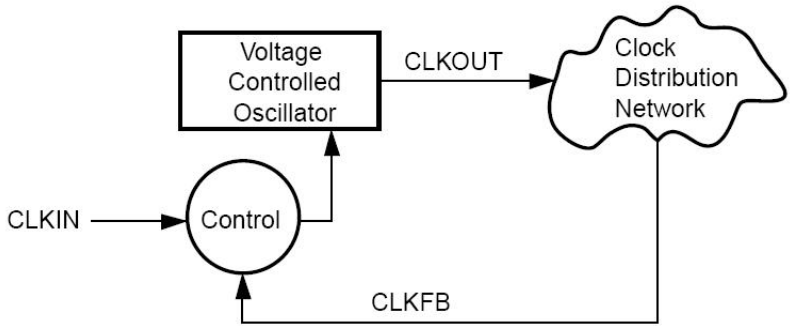
- DLL and PLL also provide additional functionality:
  - Frequency synthesis: clock multiplication & clock division
  - Clock conditioning: duty cycle correction and phase shifting

DLL & PLL Block Diagram

- DLL

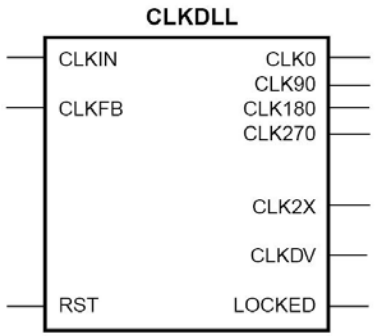


- PLL



Xilinx Spartan DLL Cell

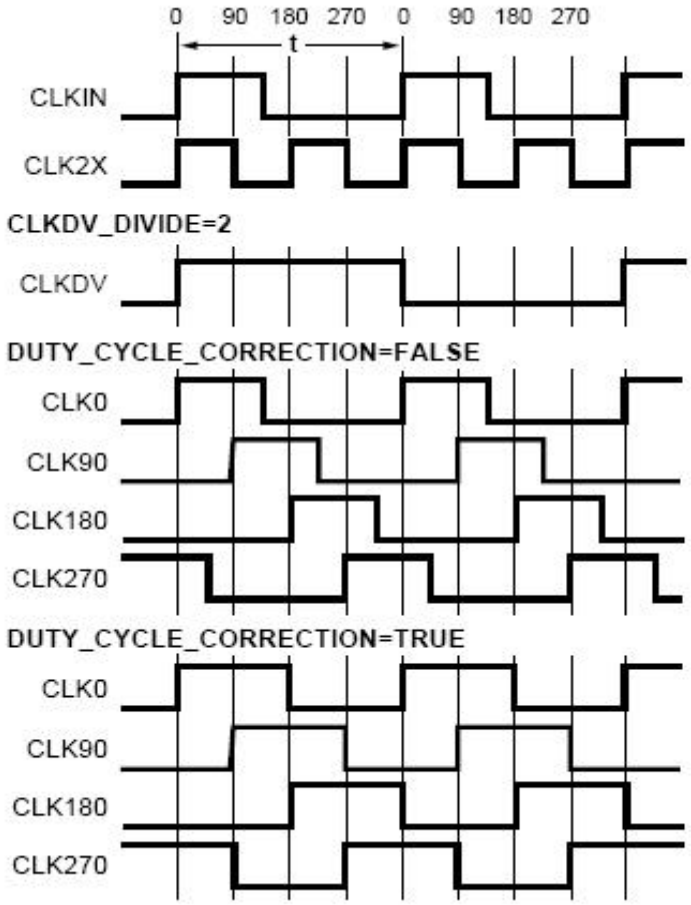
- Component symbol



- Relation of output clocks

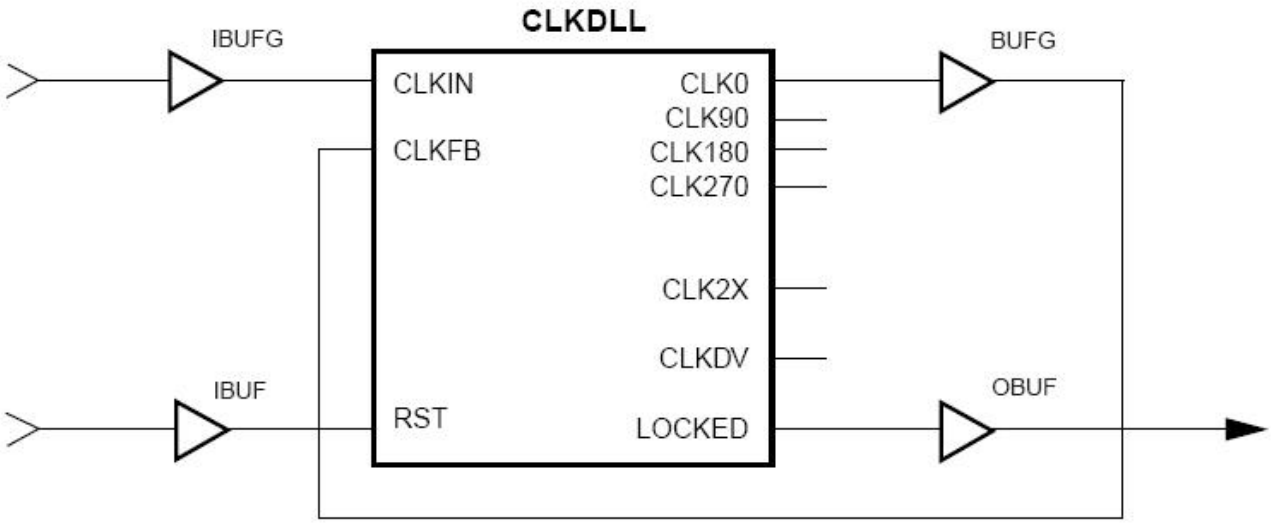
Output	Phase shift (degree)	Period shift (%)
clk0	0	0
clk90	90	25%
clk180	180	50%
clk270	270	75%

- DLL output characteristics



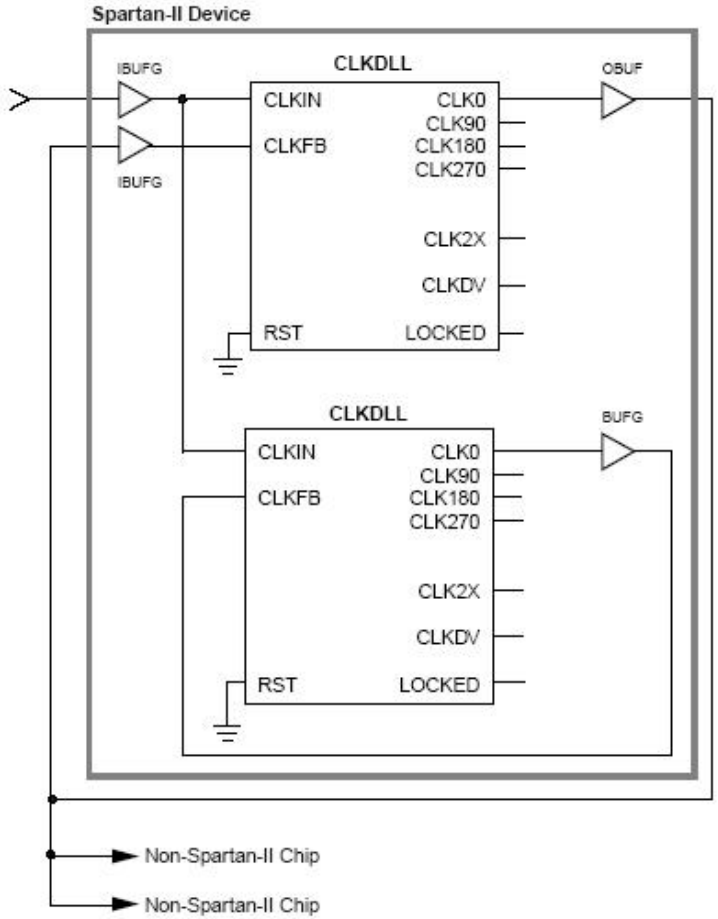
DLL Application Example

- Standard Usage



DLL Application Example

- Application for removing both internal and external clock skew



Introduction to SDRAM

- SDRAMs are multi-bank DRAM arrays with a command-driven synchronous interface; Read and write access to SDRAMs are burst oriented
- SDRAM pin description

Signal Name	Type	Description
CS	Input	Chip enable
CLK	Input	Clock
CKE	Input	Clock enable
RAS	Input	Row address strobe
CAS	Input	Column address strobe
WE	Input	Write enable
DQML, DQMH	Input	Data Mask for lower, upper Bytes
BA	Input	Bank address
A[10:0]	Input	address
DQ[15:0]	I/O	Data

SDRAM Mode Register

- Mode register is embedded in SDRAM. It defines SDRAM operation mode
- Mode register needs to be programmed via load mode command
- Mode register bit definition

Mode Register bits	# of bits	Description
M[2:0]	3	Burst length
M[3]	1	Burst type
M[6:4]	3	CAS latency
M[8:7]	2	Operation mode*
M9	1	Write burst mode
M[11:10]	2	Reserved for future use

* Normal operation mode is selected by setting M[8:7]=00. Other values are reserved for future use. Some SDRAMs do not support an operation mode choice.

Standard SDRAM Command

- SDRAM commands are fed to SDRAM through pins:

CS, RAS, CAS, WE BA A10

- SDRAM command truth table

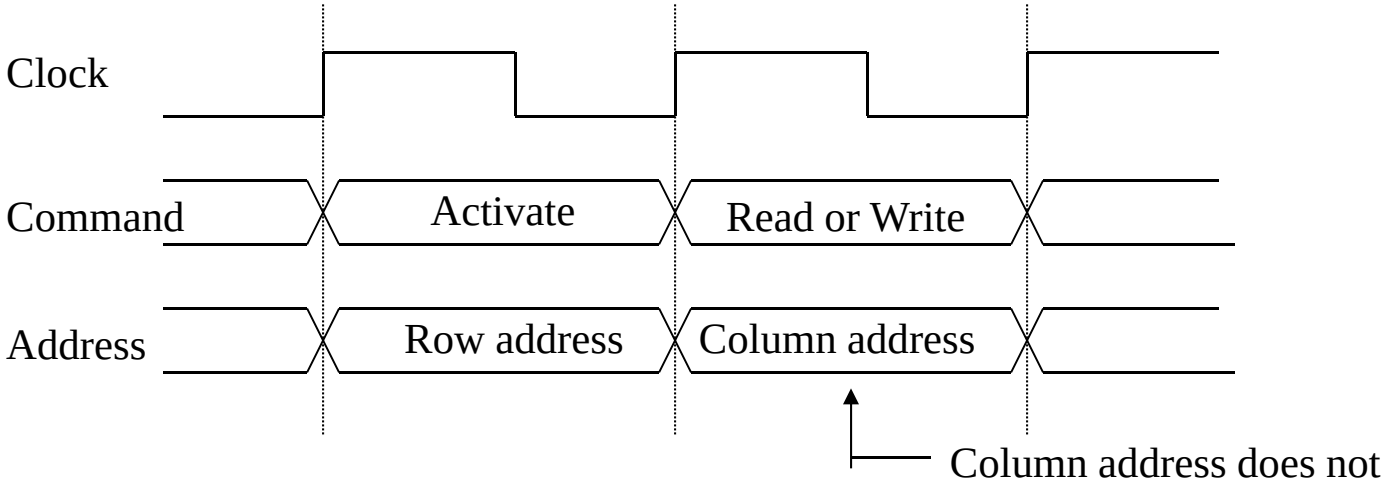
Function	Symbol	CS	RAS	CAS	WE	BA	A10	A[0:9]	Note
Device Deselect	DSEL	H	X	X	X	X	X	X	2
No Operation	NOP	L	H	H	H	X	X	X	2
Read	READ	L	H	L	H	V	L	V	2,3
Read w/ Auto Precharge	READAP	L	H	L	H	V	H	V	2,3
Write	WRITE	L	H	L	L	V	L	V	2,3
Write w/ Auto Precharge	WRITEAP	L	H	L	L	V	H	V	2,3
Bank Activate	ACT	L	L	H	H	V	V	V	2
Precharge Selected Bank	PRE	L	L	H	L	V	L	X	2
Precharge All Banks	PALL	L	L	H	L	X	H	X	2
Auto Refresh	CBR	L	L	L	H	X	X	X	2,4
Load Mode Register	MRS	L	L	L	L	V	V	V	2

Notes:

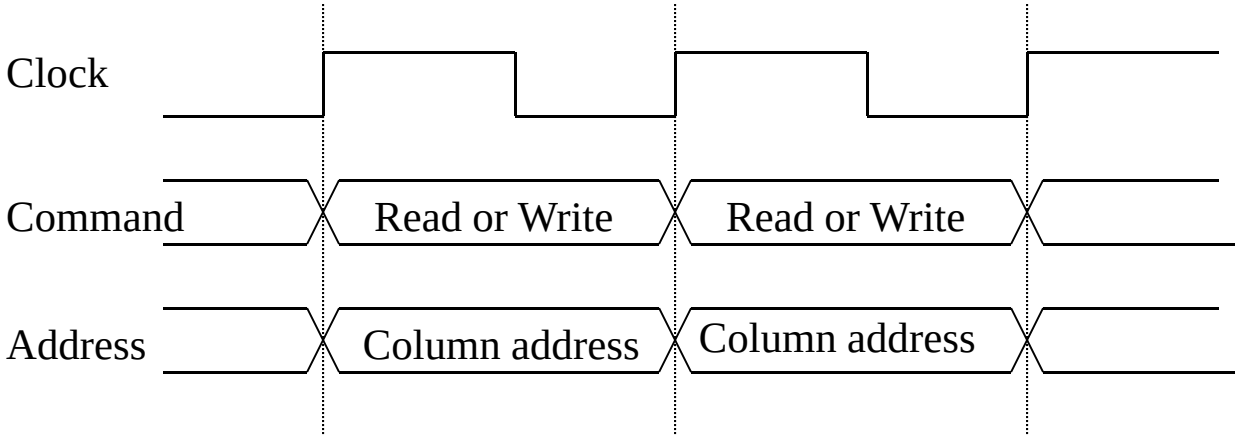
1. H: High level, L: Low level, X: don't care, V: Valid data input.
2. CKE is assumed to be High during all of these commands.
3. Only A[7:0] are needed to determine the Column address.
4. Self refresh uses the same command when CKE is Low.

SDRAM Access

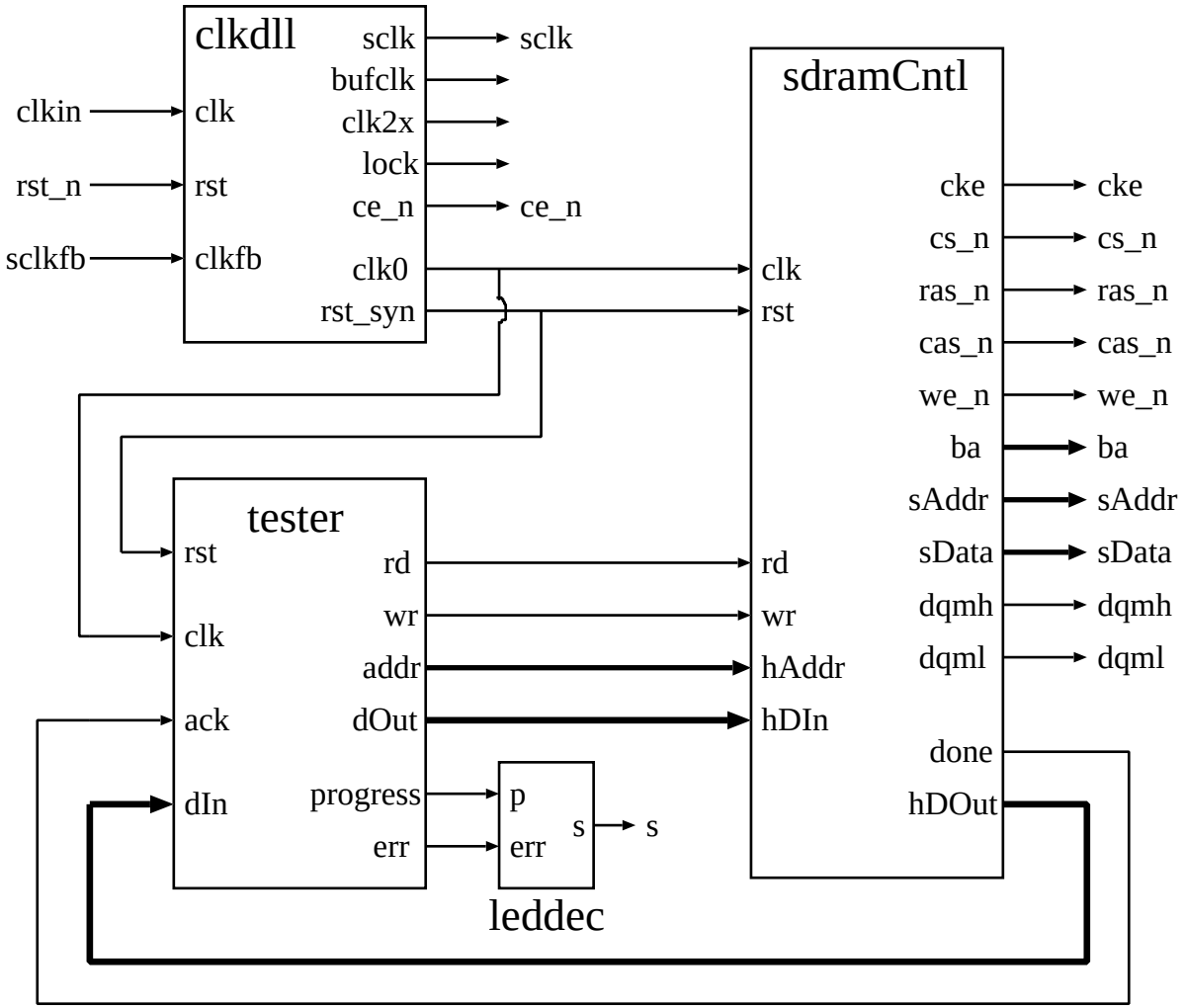
- Accessing an un-activated row



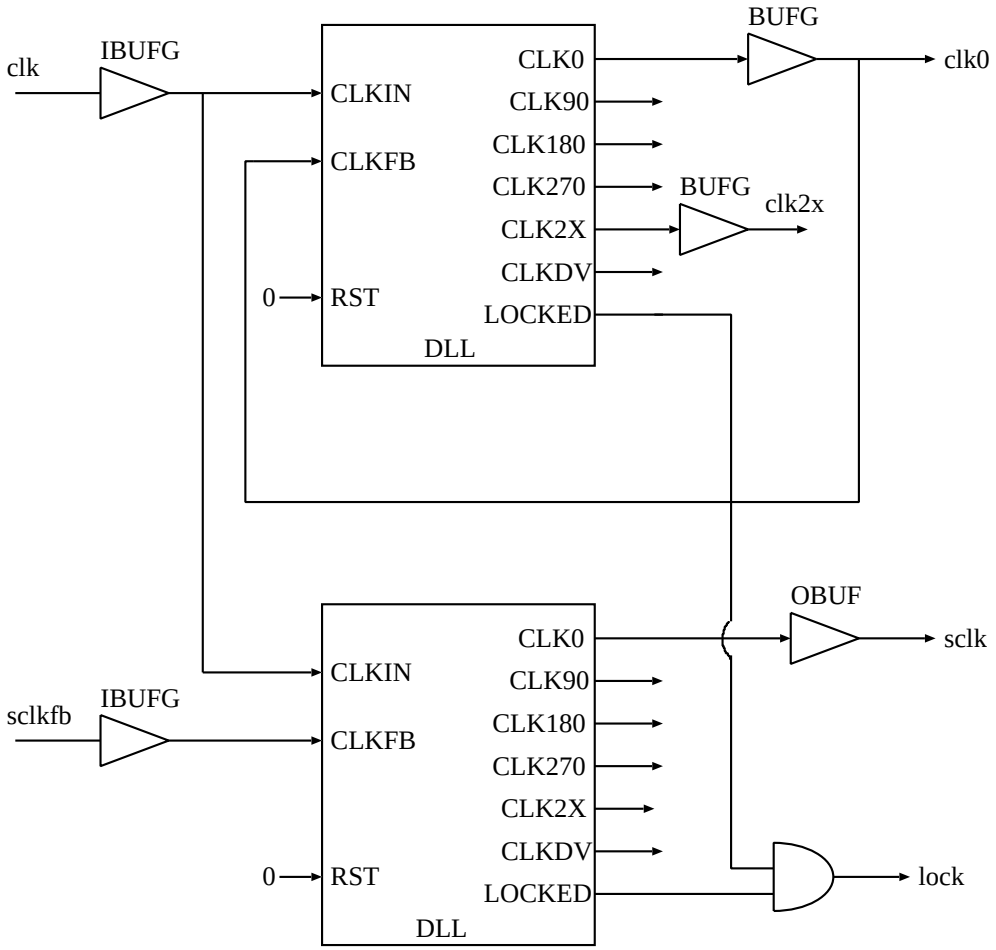
- Accessing an un-activated row



Block Diagram of the Design

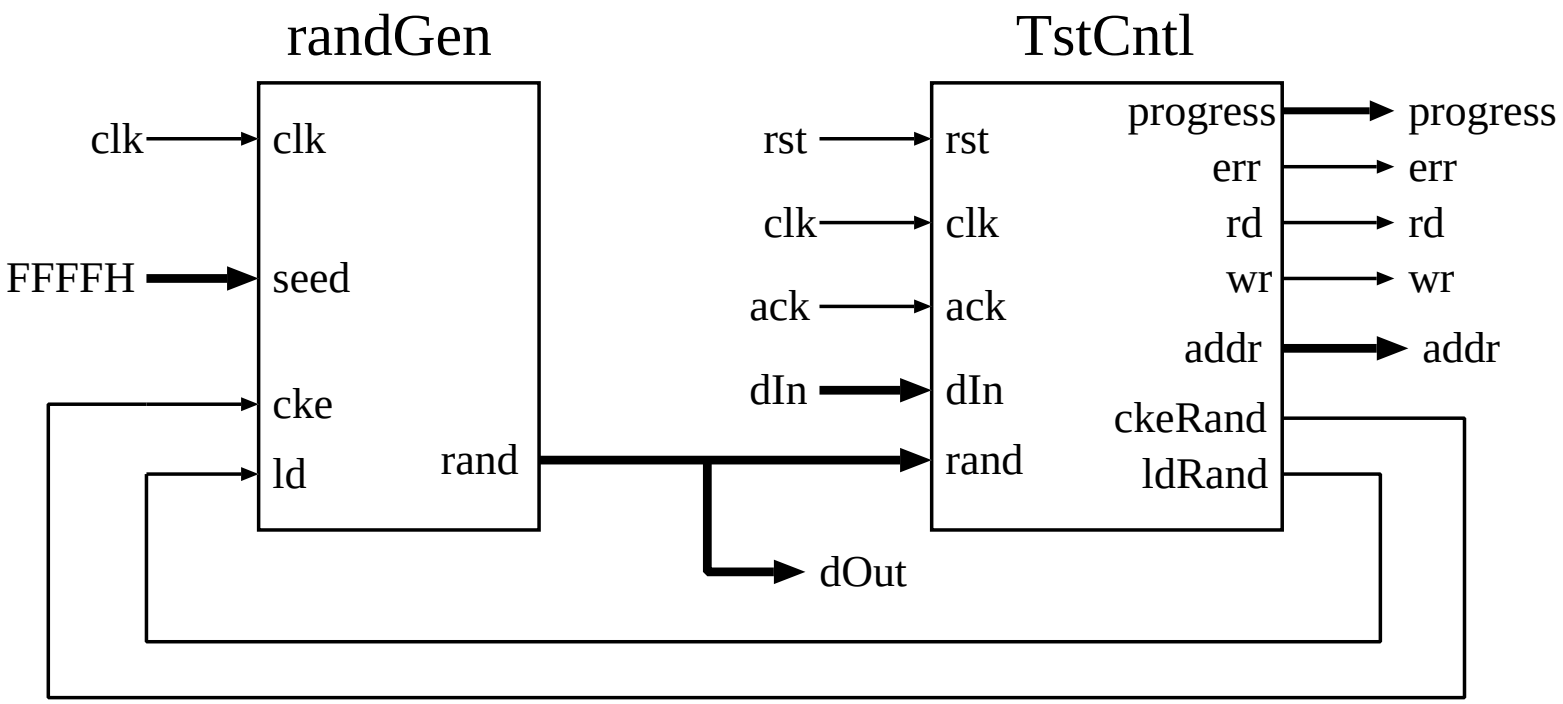


CLKDLL Block Diagram



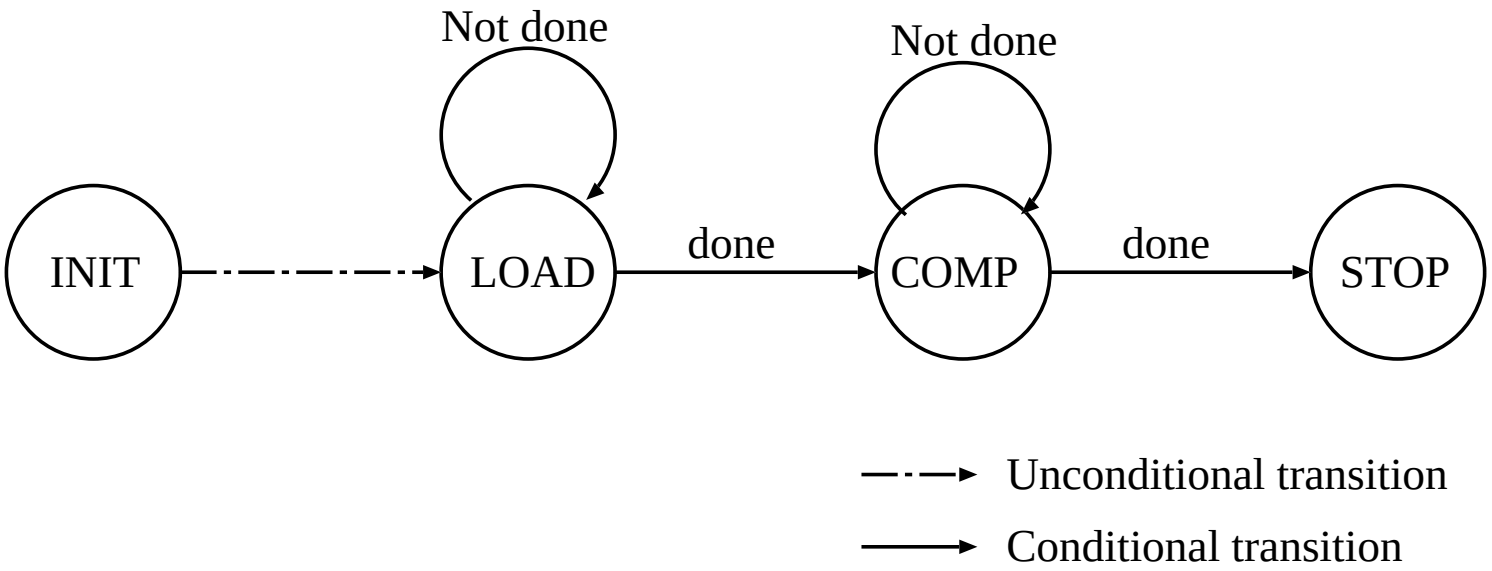
- It also contain circuits that generate synchronized reset signal.

Tester Block Diagram

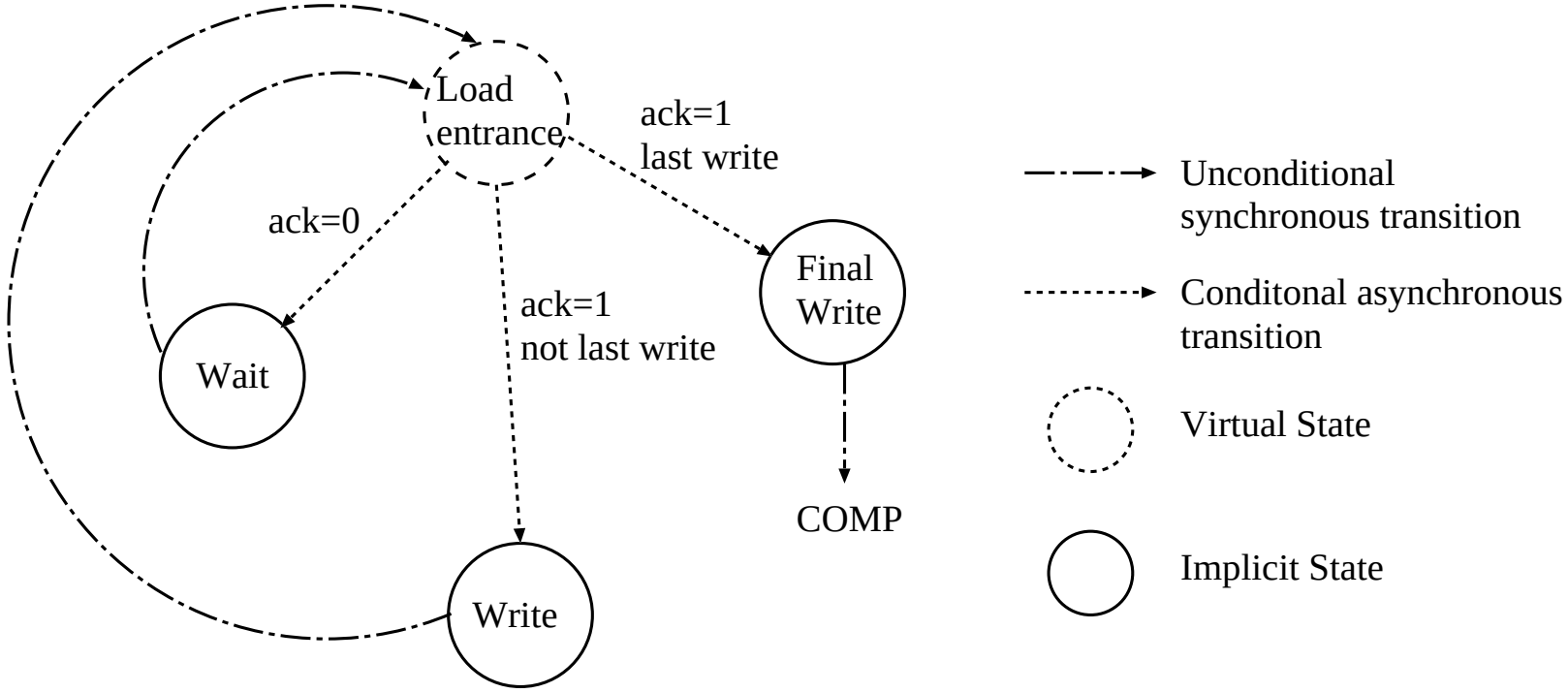


- randGen is an LFSR.

TstCntl FSM


Not done
Not done
INIT
LOAD
done
COMP
done
STOP
Unconditional transition
Conditional transition

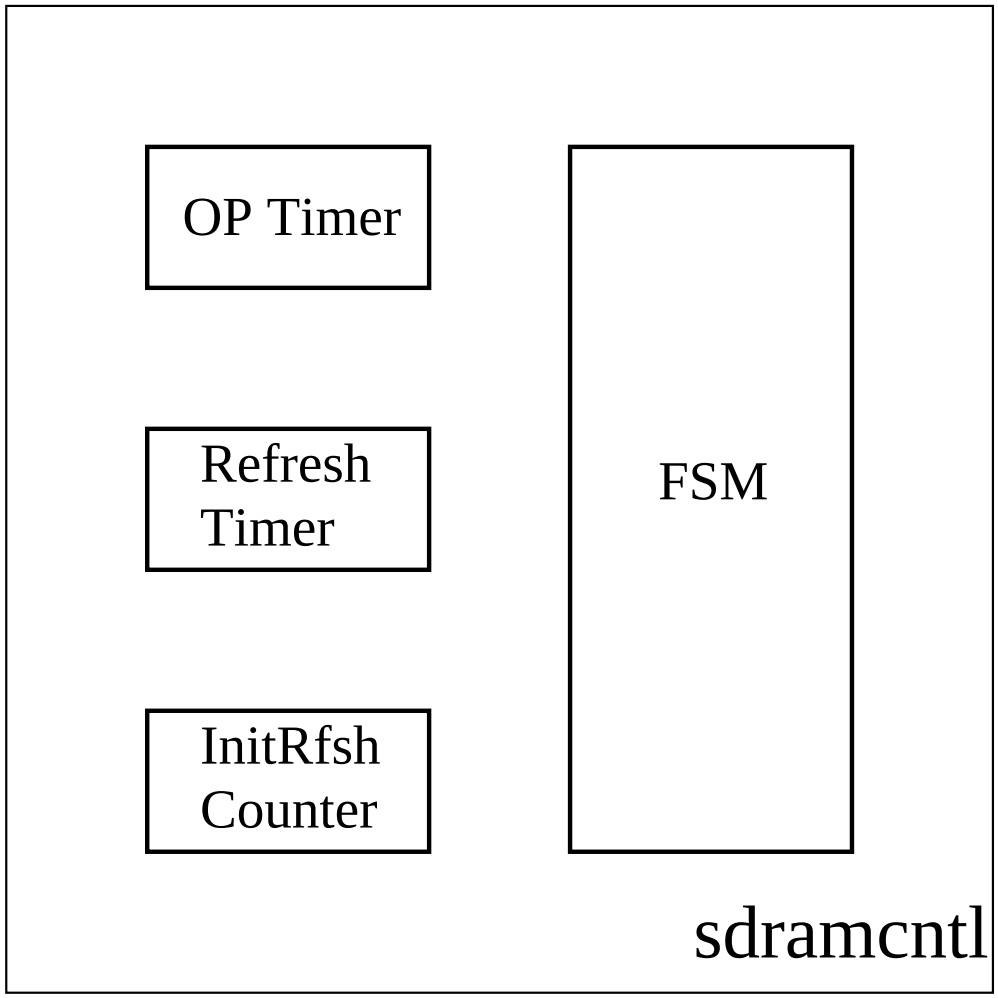
More About State LOAD



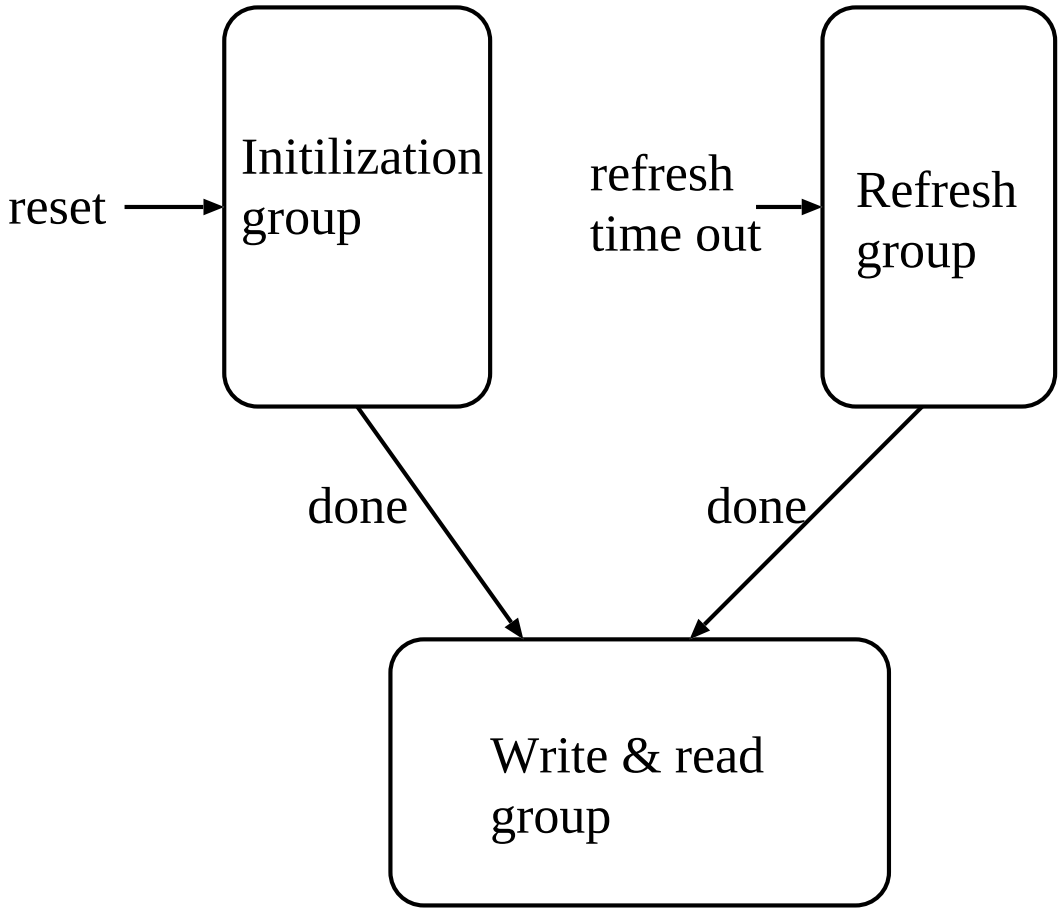
Verilog Coding for State Load

```
always @ (state)
   case state
     •••••••
     LOAD:                                                  // LOAD state
        if (~ack ) begin                  // wait for completing the previous write
           •••••••
          end else if (ack & not_last_write) begin        // previous write complete
            •••••••                                       // and not last write
            end else begin              // previous write complete and last write
                 •••••••
              end
      •••••••
```

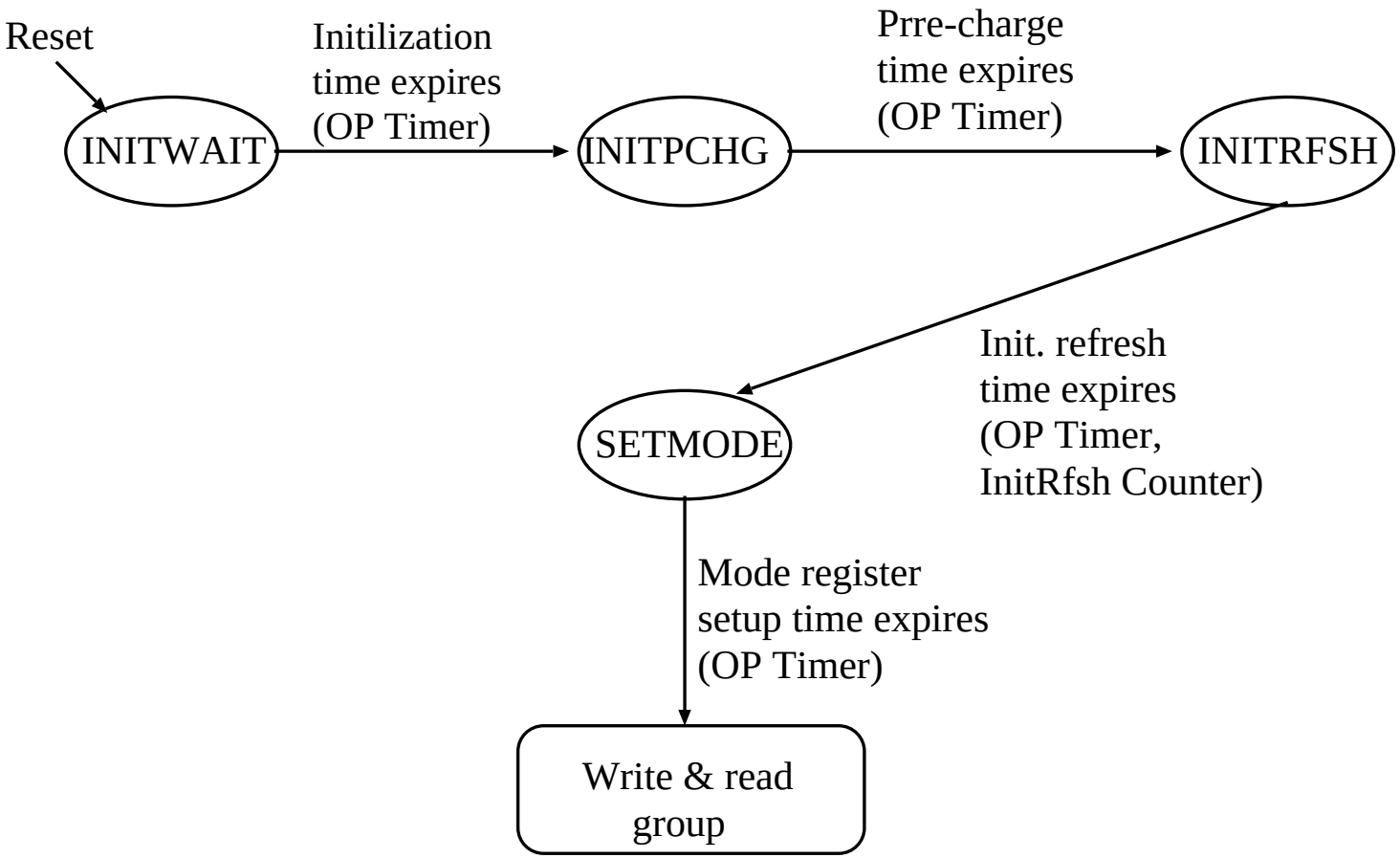
sdramcntl Block Diagram



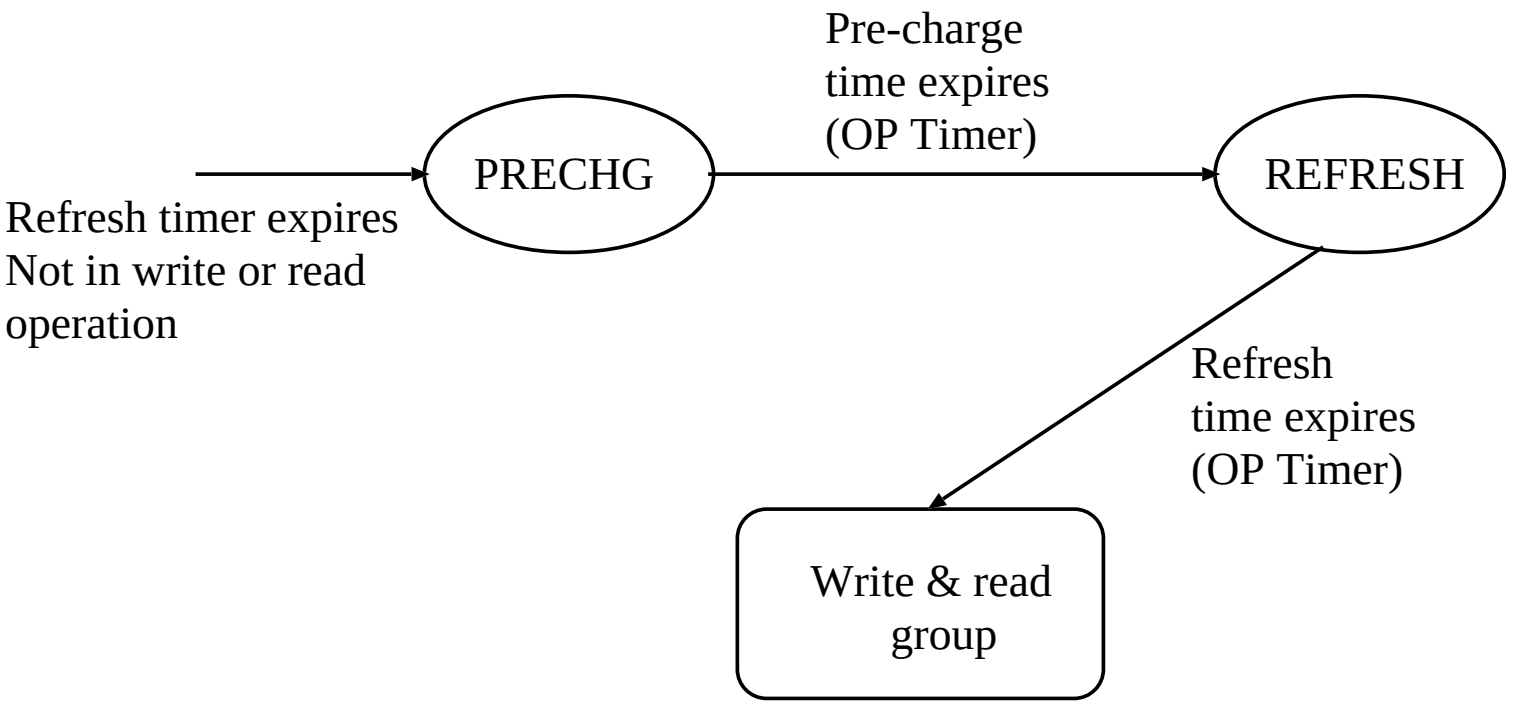
sdramcntl FSM



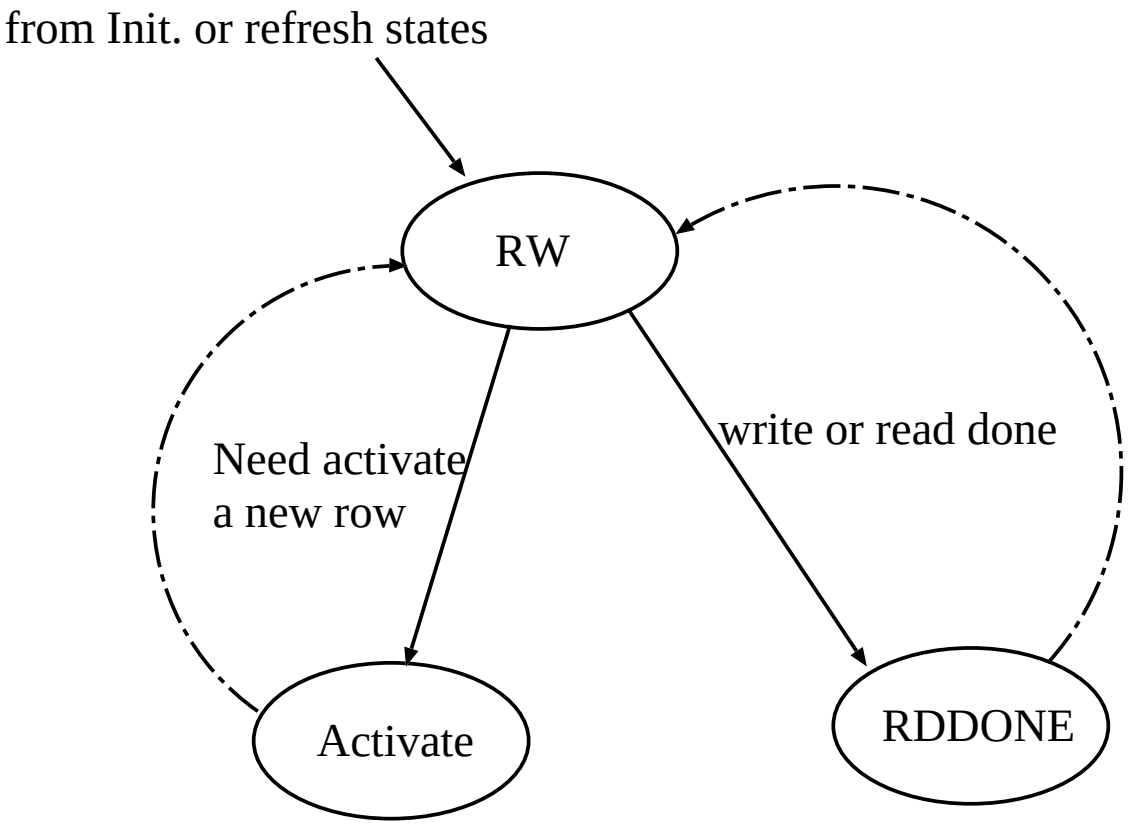
sdramcntl Initialization States



sdramcntl Refresh States



sdramcntl Read & Write States



More About State RW