Embedded System Application 4190.303C 2010 Spring Semester

DDR/DDR II/DDR III and DDRII controllers

Naehyuck Chang Dept. of EECS/CSE Seoul National University naehyuck@snu.ac.kr



High Speed Memory Design Considerations

- The signal integrity is a challenging issue in high speed design
- The following effects are more important in high speed design and can cause data corruption
 - Reflection
 - Crosstalk and interference
 - SSN (simultaneously switching noise)
- Following solutions are employed to improve signal integrity
 - Specialized voltage mode bus drivers
 - On die termination (ODT)
 - Off chip driver calibration (OCD)





SDRAM to DDR SDRAM

DRAM evolution summary

PM→	FPM	>EDO-	SDRAM	>DDR-	DDR2->	DDR3
-Simple RAS/CAS	-Fast CAS Access	-Latched Output	-Synchronous w/Clock -Multi-Bank -Programmable Burst & Latency -LVTTL Interface	-Data Read/Write on Both CLK Edges -Data Strobe -SSTL 2.5 Interface	-ODT -OCD -Posted CAS -SSTL 1.8 Interface	Faster Lower Power
				Name	Clock Freq.	Data Rate
				DDR200	100 MHz	200 MHz
				DDR266	133 MHz	266 MHz
				DDR333	167 MHz	333 MHz
				DDR400	200 MHz	400 MHz





SDRAM to DDR SDRAM

Gore time improvement







DDR, DDR II and DDR III Comparison

RAM	DDR SDRAM	DDR2 SDRAM	DDR3 SDRAM	
Clock frequency	100/133/166/200MHz	200/266/333/400MHz	533/667/800/933	
Effective Clock Speed	DDR200/266/333/400	DDR2-400/533/667/800	DDR3-1066/1333/1600/1866+	
Theoretical Bandwidth	PC1600/2100/2600/3200	PC2-3200/4200/5300/6400	PC3-8500/10667/12800/14900+	
Discreet Density	64Mb, 128Mb, 256Mb, 512Mb, 1Gb	256Mb, 512Mb, 1Gb, 2Gb	512Mb, 1Gb, 2Gb, 4Gb, 8Gb	
Module Density	32MB-1GB, 2GB	128MB-2GB, 4GB	256MB-4GB, 8GB, 16GB	
Supply voltage 2.5V		1.8V	1.5V	
CAS latency (CL) 2, 2.5, 3 clock		3, 4, 5, 6 clock	5, 6, 7, 8, 9, 10 clock	
Prefetch Buffer	Prefetch Buffer 2-bits		8-bits	
Burst length	2, 4, 8	4, 8	4 (Burst chop), 8	
On Die Termination	No	Yes	Yes (Dynamic ODT)	
Data Strobe	Single ended	Single ended / Differential	Differential Default	
Master Reset No		No	Yes	
CL/tRCD/TRP 15ns each		15 ns/each	12 ns /each	
Driver Calibration No		Off-Chip controller	On-Chip with ZQ pin	
Leveling	No	No	Yes	





DDR Architecture







Embedded Low-Power Laboratory

DDR Architecture

- Data input sampling
 - DIND: input data after DIN buffer
 - PE: pulse generated by the rising edge of DQS
 - PO: pulse generated by the falling edge of DQS
 - PCK: internal pulse generated by the rising edge of CLK







DDR Simplified State Machine



DDR Simplified State Machine



aboratory

DDR Bus Commands

- Solution CS*, RAS*, CAS*, and WE* are not no longer strobe signals
 - Rising and falling edges do not imply timing to latch data or address
 - Only level is important when the clock transition
 - Command scheme (encoded) is easier to explain

NAME (Function)	CS	RAS	CAS	WE	ADDR
DESELECT (NOP)	Н	Х	Х	Х	Х
NO OPERATION (NOP)	L	Н	Н	Н	Х
ACTIVE (Select bank and activate row)	L	L	Н	Н	Bank/Row
READ (Select bank and column, and start READ burst)	L	Н	L	Н	Bank/Col
WRITE (Select bank and column, and start WRITE burst)	L	Н	L	L	Bank/Col
BURST TERMINATE	L	Н	Н	L	Х
PRECHARGE (Deactivate row in bank or banks)	L	L	Н	L	Code
AUTO refresh or Self Refresh (Enter self refresh mode)	L	L	L	Н	Х
MODE REGISTER SET	L	L	L	L	Op-Code





DDR ACTIVE Commands

ACTIVE

- The ACTIVE command is used to open (or activate) a row in a particular bank for a subsequent access
- The value on the BA0, BA1 inputs selects the bank, and the address provided on inputs A0–A13 selects the row
- The row remains active (or open) for accesses until
 - A PRECHARGE issued to the bank
 - ♀ READ or WRITE with AUTOPRECHARGE issued

Row to Column Access delay (tRCD)

After ACTIVE a READ or WRITE command may be issued after tRCD

Row to Row Command (tRRD)

The minimum time interval between successive ACTIVE commands to different banks is defined by tRRD

NAME (Function)	CS	RAS	CAS	WE	ADDR
ACTIVE (Select bank and activate row)	L	L	н	н	Bank/Row





DDR ACTIVE Commands



t_{RRD} and t_{RCD} definition

ACTIVATE for Four-Bank DDR or DDR2 Devices







DDR Read Command and Timing

READ

- The READ command is used to initiate a burst read access to an active row
- The value on the BA0, BA1 inputs selects the bank
- Column address is provided on inputs A0—Ai
- ♀ The value on input A10 determines whether or not auto precharge is used
 - Determines to keep the raw active or not after a burst access

NAME (Function)	CS	RAS	CAS	WE	ADDR
READ (Select bank and column, and start READ burst)	L	н	L	н	Bank/Col







DDR Write Command and Timing

WRITE

- The WRITE command is used to initiate a burst write access to an active row
- The value on the BA0, BA1 inputs selects the bank
- Column address provided on inputs A0—Ai
- The value on input A10 determines whether or not **auto precharge** is used
- Data Mask (DM) controls writes of individual words into the memory
 - Selective write among the burst memory write operation
 - Data is written into the memory given the DM signal is registered LOW, the corresponding data will be written to memory

GAS to DQS delay t_{DQSS}

- DQS: strobe to latch data (source synchronous)
- \bigcirc The time between the WRITE command and the first corresponding rising edge of DQS is defined by a range of min and max t_{DQSS}, which ranges (75% to 125% of 1 clock cycle)

NAME (Function)	CS	RAS	CAS	WE	ADDR
WRITE (Select bank and column, and start WRITE burst)	L	н	L	L	Bank/Col





DDR Write Command and Timing







b

b

- Mode Register
 - The Mode Register is used to define the specific mode of operation of the DDR SDRAM
 - Burst length, burst type, CAS latency, **operating mode**
 - Programmed via the MODE REGISTER SET command
 - ♀ Retain the stored information until it is programmed again or the device loses power



Mode register definition





- Burst length
 - Read and write accesses to the DDR SDRAM are burst oriented
 - Determines the maximum number of column locations that can be accessed for a given READ or WRITE command

			Burst Length		
A2	A1	A0	A3 = 0	A3 = 1	
0	0	0	Reserved	Reserved	
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	Reserved	Reserved	

A3	Burst Type			
0	Sequential			
1	Interleaved			





DDR burst length and types

Burst	Starting			Order of Accesses Within a Burst		
Length	Address		nn SS	Type = Sequential	Type = Interleaved	
			A0			
2			0	0-1	0-1	
			1	1-0	1-0	
		A1	A0			
		0	0	0-1-2-3	0-1-2-3	
4		0	1	1-2-3-0	1-0-3-2	
		1	0	2-3-0-1	2-3-0-1	
		1	1	3-0-1-2	3-2-1-0	
	A2	A1	A0			
	0	0	0	0-1-2-3-4-5-6-7	0-1-2-3-4-5-6-7	
	0	0	1	1-2-3-4-5-6-7-0	1-0-3-2-5-4-7-6	
	0	1	0	2-3-4-5-6-7-0-1	2-3-0-1-6-7-4-5	
8	0	1	1	3-4-5-6-7-0-1-2	3-2-1-0-7-6-5-4	
	1	0	0	4-5-6-7-0-1-2-3	4-5-6-7-0-1-2-3	
	1	0	1	5-6-7-0-1-2-3-4	5-4-7-6-1-0-3-2	
	1	1	0	6-7-0-1-2-3-4-5	6-7-4-5-2-3-0-1	
	1	1	1	7-0-1-2-3-4-5-6	7-6-5-4-3-2-1-0	







- Read latency
 - The Read latency is the delay, in clock cycles, between the registration of a READ command the availability if the first piece of output data

A6	A5	A4	CAS Latency DDR 200 - 333	CAS Latency DDR 400
0	0	0	Reserved	Reserved
0	0	1	Reserved	Reserved
0	1	0	2	2
0	1	1	3 (Optional)	3
1	0	0	Reserved	Reserved
1	0	1	1.5 (optional)	1.5 (optional)
1	1	0	2.5	2.5
1	1	1	Reserved	Reserved





Solution Typical clock distribution



A Centrally located Clock Source will use matched trace delay (line length) to generate clock edges which arrive at all synchronous elements or cards at the exact same instant in time.





Basic structure



- Major examples
 - DDRSRAM, DDRSDRAM/DDRSGRAM, and RAMBUS DirectRAM
 - SCI, SGI CrayLink, and HIPPI-6400-PH
- Advantages
 - Remove the limit of the time of flight on wire between two ICs and do not require controlled clock skew between two ICs
 - Dramatically increased I/O frequencies





Traditional synchronous interfaces limit interconnect speed to less than 250 MHz and PCB interconnect length to approximately 5 inches





- In source-synchronous interfaces, the clock is sourced from the same device as the data, rather than another source, such as a common clock network
 - Clock is used within the receive interface to latch the accompanying data







- Tx sends data along with strobe (sort of a clock)
- Rx uses sent strobe to sample the data
- So clock or strobe skew issue as far as the delay is maintained







Source synchronous interface in an SoC with multiple clock domains







- Source synchronous concept example
 - Suppose that we transmit a data signal 1 ns prior to transmitting the strobe
 - You are given a 500 ps receiver setup requirement

 - You find that the flight time for the strobe signal also varies between 5.5 ns and 5.7 ns, but the two signals are not correlated





- Generation of the strobe
 - Must delay DQS to create data setup-and-hold time at the synchronization flip-flop
 - Possible delay techniques include using a digital-delay-locked loop (DLL) or PLL within the interface agent or using a pc-board etch-delay line







- Setup time condition
 - min(TF) > Tsetup + max (Tdata Tstrobe)
- Hold time condition
 - min(TB) > Thold + max (Tdata Tstrobe)
- Minimum clock period
 - min(TCLKH) = min(0.5 TCLK) > TB + min(TF) + min(TB)









- Margin for timing uncertainty
 - Although source synchronous system takes of most of the skew part but some type of jitter still remains
 - To tolerate the slight dynamic variation of time of flight of data and strobe timing margin is added to the timing equations
- Setup time condition considering jitter
 - Adding timing margin
 - min(TF) > Tsetup + max (Tdata Tstrobe) + Tmargin
- Hold time condition
 - min(TB) > Thold + max (Tdata Tstrobe) + Tmargin







Prefetch Buffers

- SDRAM
 - In most old DRAMs, the core and the I/O logic runs at the same frequency
 - In SDRAM each output buffer can release a single bit per clock cycle: prefetch of 1



- DDR 🖉
 - In DDR, every I/O buffer can output two bits per clock cycle
 - Each read command will transfer two bits from the array into the DQ
 - Use two separate data lines from the primary sense amps to the I/O buffers: prefetch of 2





DDR to DDR II

Modifications

- Increased bus speed
 - Targeting 667Mb/s/pin and operating at data rates of 400 MHz, 533 MHz, 667 MHz, and above
- Extended mode registers introduced to control advance features of DDR II
- Off chip driver (OCD) calibration introduced
- Galactic CAS latencies are increased to 3, 4, 5, and 6 cycles
- Posted CAS introduced to improves control bus bandwidth
 - ♀ An internal pipe line allows READ of WRITE command be issued in the cycle next to ACTIVE
- Prefetch of 4







Differential Strobe

Single-ended strobe



Reduced crosstalk and less SSN





aboratory

Differential Strobe

DDR II/III differential strobe







Differential Strobe

Imbalanced duty cycle





Laboratory

DDR II Simplified State Machine (1/2)



DDR II Simplified State Machine (2/2)



aboratory

DDR II Initialization

- Setup the MSCR register to use SSTL 1.8V I/O for DDR2 on all memory controller pins:
 - writemem.b 0xFC0A4074 0xAA ; MSCR_SDRAM
- Setup the memory controller chip selects for 128 Mbytes each
 - writemem.l 0xFC0B8110 0x4000001A ; SDCS0
 - writemem.l 0xFC0B8114 0x4800001A ; SDCS1
- Setup the required memory vendor's delays for various DDR commands
 - writemem.l 0xFC0B8008 0x65311810 ; SDCFG1
 - SRD2RWP=0x6=BurstLength/2+2
 - SWT2RWP=0x5=CAS+AdditiveLatency+twr −1=3+1+(15ns/7.5)−1
 - @ RD_LAT = 0x3 = CAS Latency in clock cycles
 - @ ACT2RW=0x1=(tRCD /tCLK)-1=(15ns/7.5ns)-1=1clockcycle
 - PRE2ACT=0x1=(tRP /tCLK)-1=(15ns/7.5ns)-1=1clockcycle
 - @ REF2ACT=0x8=(tRFC /(tCLK x2))+(1formathrounding)=(105ns/15ns)+1=8
 - WT_LAT=0x1=AdditiveLatency=(tRCD(min)/tCLK)-1=15ns/7.5ns-1=1
 - writemem.l 0xFC0B800C 0x59670000 ; SDCFG2
 - BRD2RP=0x5=BurstLength/2+AdditiveLatency=8/2+1=5
 - BWT2RWP = 0x9 = CAS Latency + Additive Latency + Burst Length / 2 + tWR/tCLK 1 = 3+1+4+2-1=9 - BRD2W=0x6=BurstLength/2+2=6 - BL=0x7=BurstLength-1=7





DDR II Initialization

- Delay (DDR2 memories have a delay requirement), typically 200 µs
 - writemem.l 0xFC0B8004 0xEA0F2002 ; SDCR
 - Set mode enable (1)

 - Disable automatic refresh (0)

 - ♀ Configure address mux to (10) = 512 Mbits configured as $14 \times 10 \times 4$ and 8-bit wide
 - Drive rule set to tri-state mode between reads and writes. Board uses parallel termination
 - ♀ Refresh count set to 0xF which means $(8k / (7.5 \text{ ns} \times 64)) 1 = 15$
 - Memory port size is set to 16 bits
 - DQS outputs are still disabled
 - Issue pre-charge all command
 - Deep power down is not used during initialization sequence
 - ♀ writemem.l 0xFC0B8000 0x40010408 ; SDMR
 - Write extended mode register command for non-mobile DDR
 - Set CMD bit to issue load extended mode command
 - Set extended mode: DLL is enabled, full strength output drive, internal parallel termination is disabled, posted CAS (additive latency) of 1, OCD not supported, differential DQS disabled, RDQS disabled, outputs enabled
 - writemem.l 0xFC0B8000 0x00010333 ; SDMR
 - Write mode register command
 - Set CMD bit to issue load mode register command
 - Set mode register contents: burst length of 8, sequential burst mode, CAS latency of 3, normal mode, DLL held in reset, write recovery set to 2, and power down set to fast exit mode





DDR II Initialization

- Delay 200 memory clock cycles before issuing the pre-charge all command in the next step
 - writemem.I 0xFC0B8004 0xEA0F2002 ; SDCR, issue PALL
 - Same as last SDCR write, which effectively issues another pre-charge all command
 - ♀ writemem.I 0xFC0B8004 0xEA0F2004 ; SDCR
 - ♀ Same as last SDCR write but now issue a refresh command











DDR II Extended Mode Register (1)









Posted CAS

- Normal write sequence
 - \bigcirc Write data → data buffer → memory writing → end of transaction
- Write posting
 - \bigcirc Write data → data buffer is latch → end of transaction → background memory writing
- Write posting may enhance the throughput if there is no consecutive writes







Posted CAS

- Posted CAS operation is supported to make command and data bus efficient
- DDR II SDRAM allows a CAS read or write command
 - To be issued immediately after the RAS bank activate command
 - \bigcirc Or any time during the RAS-CAS-delay time, t_{RCD}, period
 - The command is held for the time of the Additive Latency (AL) before it is issued inside the device
 - General Provide the second second







DDR II Write Timing

- \bigcirc Write latency = read latency 1 CLK
- AL does not affect write timing







ODT (On-die Termination)

- On-die termination (ODT) has been added to the DDR II data signals to improve signal integrity in the system
- $\label{eq:stars} $$ In the termination value of R_{TT}$ is the Thevinen equivalent of the resistors that terminate the DQ inputs to V_{SSQ} and V_{DDQ} $$$
- An ODT pin is added to the DRAM so the system can turn the termination on and off as needed







ODT (On-die Termination)

- On board termination resistance is integrated inside of DDR2 SDRAM
 - ODT turn on/off is controlled by ODT pin



A6	A2	Rtt (Nominal)	
0	0	ODT Disabled	
0	1	75 ohm	SW1 on
1	0	150 ohm	+ ODT on
1	1	50 ohm —	SW2 on







ODT (On-die Termination)

ODT for active and idle devices









- Lower logic swing enables a higher frequency switching
 - Ramping of the voltages will show a significant skew
 - The skew can be reduced by increased drive strength
 - ♀ Side effects such as overshoot/undershoot
- High frequency signaling may cause asymmetric delay of differential lines
 - Use Off-Chip Driver calibration (OCD calibration)
 - Without OCD calibration, the DRAM has a nominal output driver strength of 18 ohms +30% and a pull-up and pulldown mismatch of up to 4 ohms
 - Using OCD calibration, a system can reduce the pull-up and pull-down mismatch and target the output driver at 18 ohms to optimize the signal integrity







DQS signal, /DQS signal, and drive performance







Setting OCD value







Adjust timing mode





Burst data operation

Burst Data				Operation		
DT0	DT1	DT2	DT3	Pull-up Driver strength	Pull-down Driver strength	
0	0	0	0	-	-	
0	0	0	1	Increased by 1 step	_	
0	0	1	0	Reduced by 1 step	_	
0	1	0	0	—	Increased by 1 step	
1	0	0	0	-	Reduced by 1 step	
0	1	0	1	Increased by 1 step	Increased by 1 step	
0	1	1	0	Reduced by 1 step	Increased by 1 step	
1	0	0	1	Increased by 1 step	Reduced by 1 step	
1	0	1	0	Reduced by 1 step	Reduced by 1 step	
Other	Other than above			Reserved		







Before OCD











- A general memory controller consists of two parts
- The front-end
 - Buffers requests and responses
 - Provides an interface to the rest of the system
- The back-end
 - Provides an interface towards the target memory





- 0 **Functional blocks**
 - A general memory controller consists of four functional blocks 9
 - Memory mapping 9
 - Arbiter 9
 - Command generator 9
 - Data path 9





Memory mapping

- The memory map decodes a memory address into (bank, row, column)
 - Decoding is done by slicing the address
- Different maps affect the memory access pattern
 - Bank sequential access
 - Bank interleaving
- Impacts bank conflict efficiency

Bank sequential access				Bank interleaving			
Bank Row Col Col				Row Bank Col			
Row 0 00 01 10 11 1 0 1 2 3 1 4 5 6 7 Bank 0	00 01 10 11 0 8 9 10 11 1 12 13 14 15 Bank 1	00 01 10 11 0 16 17 18 19 1 20 21 22 23 Bank 2	00 01 10 11 0 24 25 26 27 1 28 29 30 31 Bank 3	00 01 10 11 Row 0 0 1 8 9 1 16 17 24 25 Bank 0	00 01 10 11 0 2 3 10 11 1 18 19 26 27 Bank 1	00 01 10 11 0 4 5 12 13 1 20 21 28 29 Bank 2	00 01 10 11 0 6 7 14 15 1 22 23 30 31 Bank 3





Arbiter

- The arbiter chooses the order in which requests access memory
 - Potentially multiple layers of arbitration
- An arbiter can have many different properties
 - ♀ High memory efficiency
 - Predictable
 - Fast
 - 🥥 Fair
 - Flexible
- Some properties are contradictory to each other and are being traded in arbiter design





Command generator

- Generates the commands for the target memory

 - Parameterized to handle different timings









Controller Designs

- Two directions in controller design
 - Static memory controllers
 - Dynamic memory controllers









Static Memory Controllers

- Schedule is created at design-time
 - Traffic must be well-known and specified
 - A fixed schedule is not flexible
 - Computing schedules is not possible online for large systems
 - Allocated bandwidth, worst-case latency and memory efficiency can be derived from the schedule
 - Static controllers are predictable







Dynamic Memory Controller

- Dynamic memory controllers
 - Schedule requests in run-time
 - Are flexible
- Clever tricks
 - Schedules refresh when it does not interfere
 - Reorder bursts to minimize bank conflicts
 - Prefer read after read and write after write
- Dynamic arbitration
 - Allows diverse service to unpredictable traffic
 - Provides good average cases
 - Are very difficult to predict
- The schedule is not known in advance
 - Can provide statistical guarantees based on simulation
 - Memory efficiency is difficult to calculate







Memory Controller summary





