Hardware System Verification (HSV)
Vertical Solutions Engineering (VSE)

DDR/DDRII/DDRII+ SRAM
Palladium Memory Model
User Guide

Document Version: **2.1**

Document Date: **July 2018**

Contents

General Information

The Cadence Memory Model Portfolio provides memory device models for the Cadence Palladium XP, Palladium XP II and Palladium Z1 series systems. Optimizing the acceleration and/or emulation flow on these platforms for MMP memory models may require information outside the scope of the MMP user guides and related MMP documentation.

1.1 Related Publications

For basic information regarding emulation and acceleration, please refer to the following documents:

For Palladium XP and Palladium XP II:

UXE User Guide
UXE Library Developer's Guide
UXE Known Problems and Solutions
UXE Command Reference Manual
Palladium XP Planning and Installation Guide
Palladium Target System Developer's Guide
What's New in UXE

For Palladium Z1:

VXE User Guide
VXE Library Developer's Guide
VXE Known Problems and Solutions
VXE Command Reference Manual
Palladium Z1 Planning and Installation Guide
Palladium Target System Developer's Guide
What's New in VXE

DDR/DDRII/DDRII+ SRAM Memory Model

1. Introduction

The Cadence Palladium DDR/DDR2/DDR2+ SRAM Model is available in several configurations with model sizes to match real devices manufactured by Samsung. Different sizes from 18Mb up to 72Mb are available, please consult the memory model catalog for the current available list.

2. Model Release Levels

All models in the Memory Model Portfolio are graded with a release level. This release level informs users of the current maturity and status of the model. All families in the library are graded at one of these levels.

The different levels give an overall indication of the amount of testing, level of quality and feature availability in the model. For details on supported features check the User Guide for that particular model family.

There are three release levels for models in the MMP release.

Release Level		Model Status	Available in Release	Listed in Catalog	Requires Beta Agreement
Mainstream Release	MR	Fully released and available in the catalog for all customers to use.	Yes	Yes	No
Emerging Release	ER	Model has successfully completed Beta engagement(s). Most, but not all features have been tested. Documentation is available.	No	Yes	Yes
Initial Release	IR	Model has completed initial development and has been released to Beta customer(s). The model may have missing features, may not be fully tested, may not have documentation. Model may contain defects.	No	Yes	Yes

Access to Initial and Emerging Release versions of the models will require a Beta Agreement to be signed before the model can be delivered.

3. Configurations

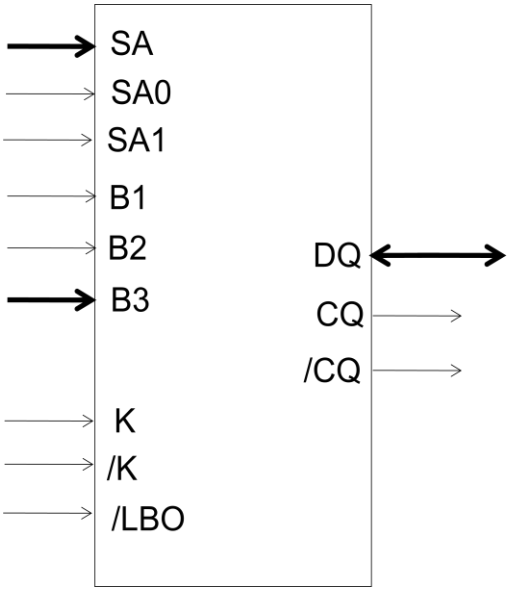
The following table lists the possible configurations. Not all configurations are available from all vendors. Please consult the appropriate vendor site for details on the parts they offer.

part number	Density	Organization
K7D323674C	32M bit	1Mx36
K7D321874C	32M bit	2Mx18
K7D163674B	16M bit	512Kx36
K7D161874B	16M bit	1Mx18
K7D803671B	8M bit	256Kx36
K7D801871B	8M bit	512Kx18
K7K3236U2C	36M bit	1Mx36
K7K3218U2C	36M bit	2Mx18
K7K3236T2C	36M bit	1Mx36
K7K3218T2C	36M bit	2Mx18
K7K1636U2C	18M bit	512Kx36
K7K1618U2C	18M bit	1Mx18
K7K1636T2C	18M bit	512Kx36
K7K1618T2C	18M bit	1Mx18
K7I643682M	72M bit	2Mx36
K7I641882M	72M bit	4Mx18
K7I643684M	72M bit	2Mx36
K7I641884M	72M bit	4Mx18
K7J643682M	72M bit	2Mx36
K7J641882M	72M bit	4Mx18
K7I323682C	36M bit	1Mx36
K7I321882C	36M bit	2Mx18
K7I323684C	36M bit	1Mx36
K7I321884C	36M bit	2Mx18
K7J323682C	36M bit	1Mx36
K7J321882C	36M bit	2Mx18
K7I163682B	18M bit	512Kx36
K7I161882B	18M bit	1Mx18
K7I163684B	18M bit	512Kx36
K7I161884B	18M bit	1Mx18
K7J163682B	18M bit	512Kx36
K7J161882B	18M bit	1Mx18

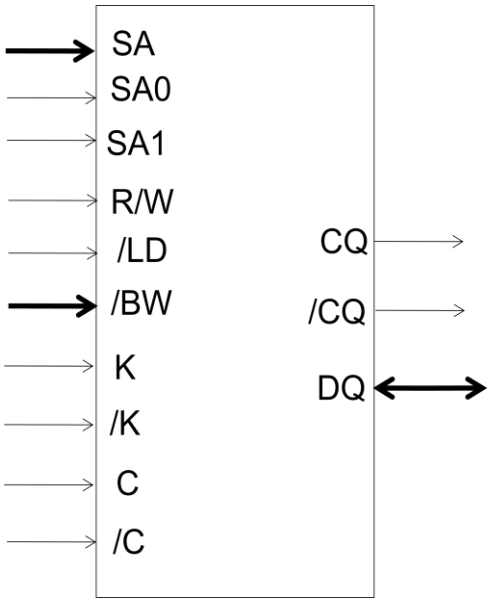
4. Model Block Diagram

The widths of the SA, D, BW, and Q buses are dependent on the density of the part being used.

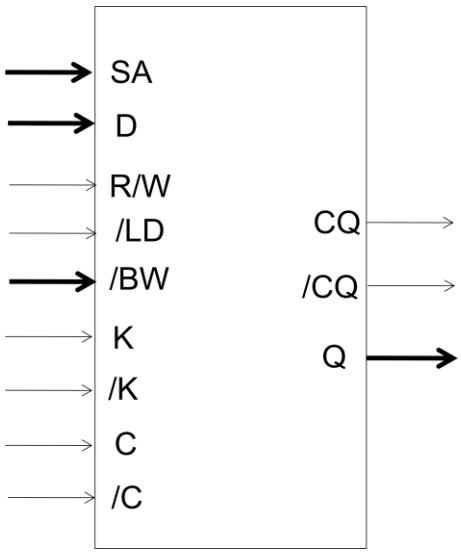
DDR SRAM:



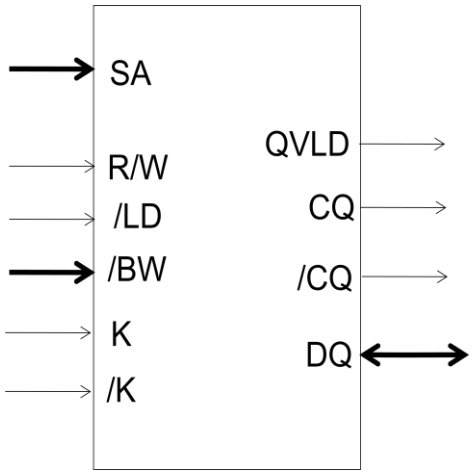
DDRII SRAM CIO:



DDRII SRAM SIO:



DDRII+ SRAM:



5. Pin names

SYMBOL	DESCRIPTION
K, /K	Input Clock
C, /C	Input Clock for Output Data
CQ, /CQ	Output Echo Clock
/Doff	DLL Disable when low
QVLD	Q valid Output
SA	Address Inputs
D	Data Inputs
Q	Data Outputs
DQ	Data Inputs Outputs
R/W	Read, Write Control Pin, Read active when high
/LD	Synchronous Load Pin, bus cycle sequence is to be defined when low
/BWx	Block Write Control Pin, active when low
SA0	Synchronous Burst Address Input, SA0=LSB
SA1	Synchronous Burst Address Input
B1	Load external address
B2	Burst Read/Write enable
B3	Single /Double Data Selection
/LBO	Linear Burst Order

6. Truth Table and Waveforms for Read/Write

6.1. DDR SRAM

K	B1	B2	B3	DQ	OPERATION
↑	H	L	X	Hi-Z	No operation, pipeline High-Z
↑	L	H	H	DOUT	Load Address, Signal Read
↑	L	H	L	DOUT	Load Address, Double Read
↑	L	L	H	DIN	Load Address, Signal Write
↑	L	L	L	DIN	Load Address, Double Write
↑	H	H	X	B	Increment Address, Continue

The DDR Model supports both single data rate and double data rate read and write, with max burst length of 4. There are two burst types, Linear and interleaved, and the mode is controlled by /LBO input.

Interleaved burst mode	Case 1		Case 2		Case 3		Case 4	
	A1	A0	A1	A0	A1	A0	A1	A0
1st	0	0	0	1	1	0	1	1
2nd	0	1	0	0	1	1	1	0
3rd	1	0	1	1	0	0	0	1
4th	1	1	1	0	0	1	0	0

Linear burst mode	Case 1		Case 2		Case 3		Case 4	
	A1	A0	A1	A0	A1	A0	A1	A0
1st	0	0	0	1	1	0	1	1
2nd	0	1	1	0	1	1	0	0
3rd	1	0	1	1	0	0	0	1
4th	1	1	0	0	0	1	1	0

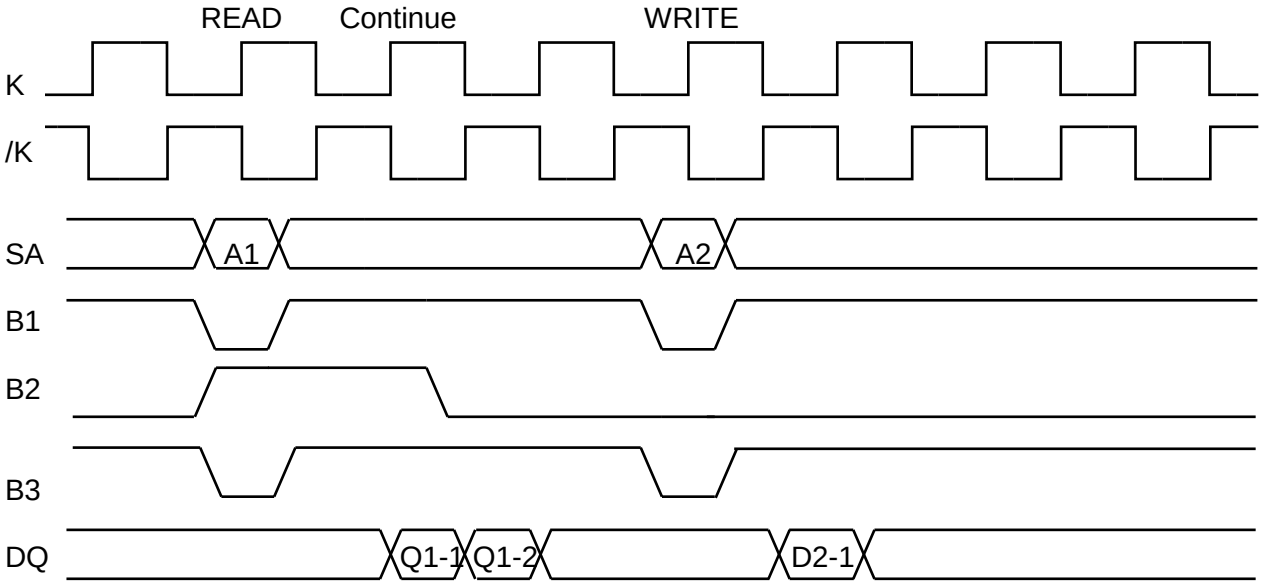
During SDR read operations, addresses and controls are registered at the first rising edge of K clock and then the internal array is read between first and second rising edges of K clock. Data outputs are updated from output registers off the second rising edge of K clock. During DDR read operations, addresses and controls are registered at the first rising edge of K clock, and then the internal array is read twice between first and second rising edges of K clock. Data outputs are updated from output registers sequentially by burst order off the second rising and falling edge of K clock.

To avoid data contention, at least two NOP operations are required between the last read and the first write operation.

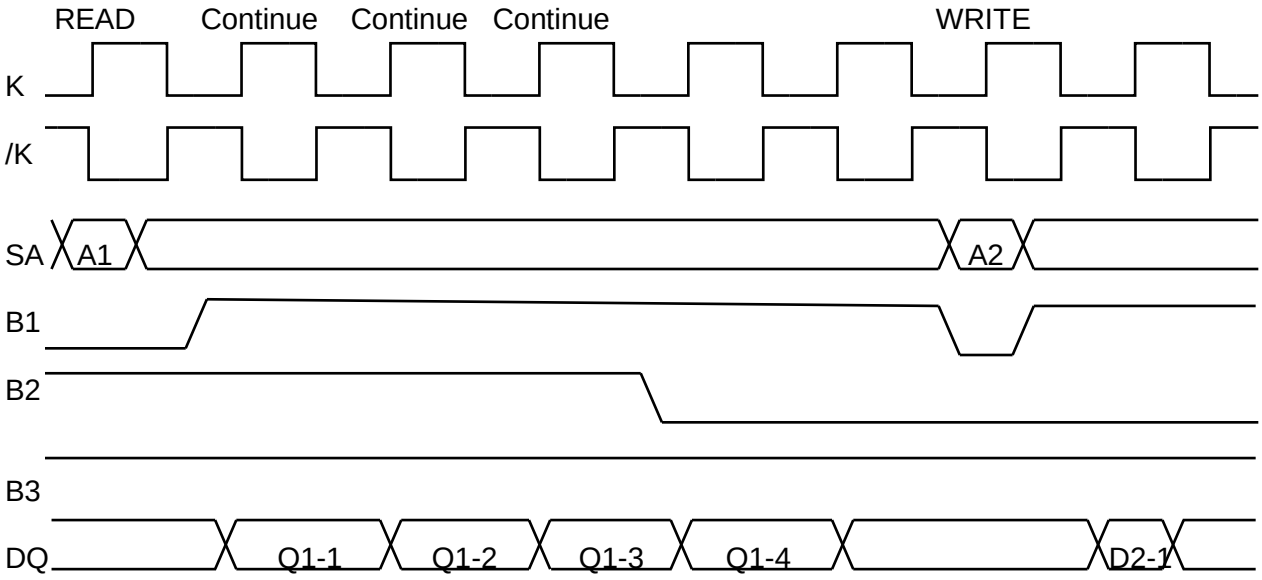
During SDR write operations, addresses and controls are registered at the first rising edge of K clock and data inputs are registered at the following rising edge of K clock. During DDR write operations, addresses and controls are registered at the first rising edge of K clock and data

inputs are registered twice at the following rising and falling edge of K clock. Write addresses and data inputs are stored in the data in registers until the next write operation, and only at the next write operation are data inputs fully written into SRAM array.

Timing Waveforms (Double Data Rate)



Timing Waveforms (Single Data Rate)



6.2. DDRII SRAM CIO (burst length = 2)

K	/LD	R/W	DQ		OPERATION
			DQ(A0)	DQ(A1)	
Stopped	X	X	Previous state	Previous state	Clock stop
↑	H	X	High-Z	High-Z	No operation
↑	L	H	QOUT at /C(t+1)	QOUT at C(t+2)	Read
↑	L	L	DIN at K(t+1)	DIN at /K(t+1)	Write

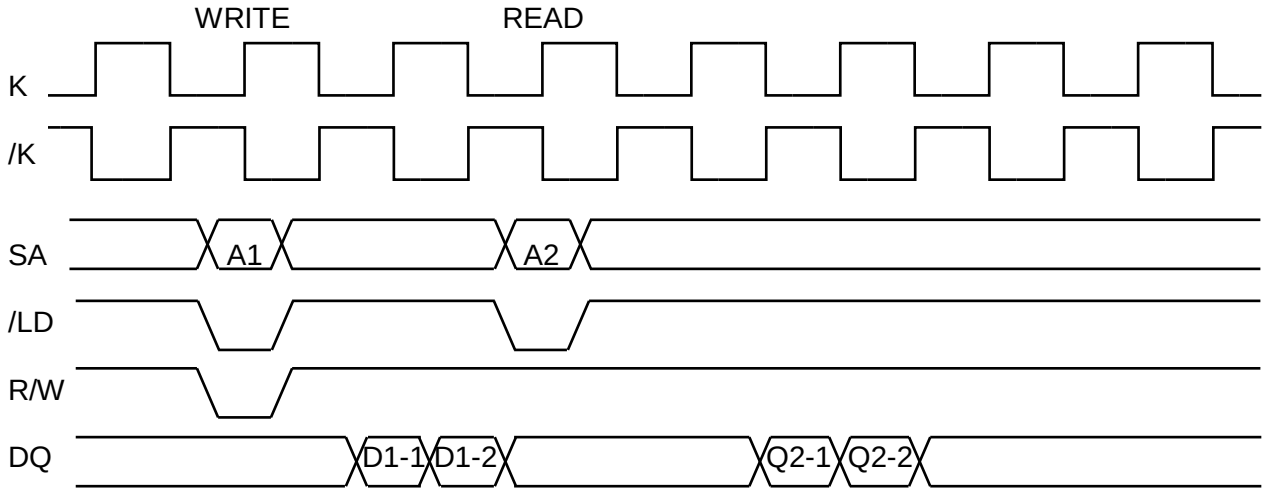
DDRII CIO SRAM has a common data input/output bus, and DDR interface on read and write ports.

Address, data inputs, and all control signals are synchronized to the input clock (K or /K). Normally data outputs are synchronized to output clocks (C and /C), but when C and /C are tied high, the data outputs are synchronized to the input clocks (K and /K). Read data are referenced to echo clock (CQ or /CQ) outputs, Read address and write address are registered on rising edges of the input K clocks. Common address bus is used to access address both for read and write operations.

The internal burst counter is fixed to 2-bit sequential for both read and write operations. Simple depth expansion is accomplished by using LD for port selection.

Byte write operation is supported with BW0 and BW1 (BW2 and BW3) pins for x18 (x36) device.

Timing Waveforms



6.3. DDRII SRAM CIO (burst length = 4)

K	/LD	R/W	DQ				OPERATION
			DQ(A0)	DQ(A1)	DQ(A2)	DQ(A3)	
Stopped	X	X	Previous state	Previous state	Previous state	Previous state	Clock stop
↑	H	X	High-Z	High-Z	High-Z	High-Z	No operation
↑	L	H	QOUT at /C(t+1)	QOUT at C(t+2)	QOUT at /C(t+2)	QOUT at C(t+3)	Read
↑	L	L	QIN at K(t+1)	QIN at /K(t+1)	QIN at K(t+2)	QIN at /K(t+2)	Write

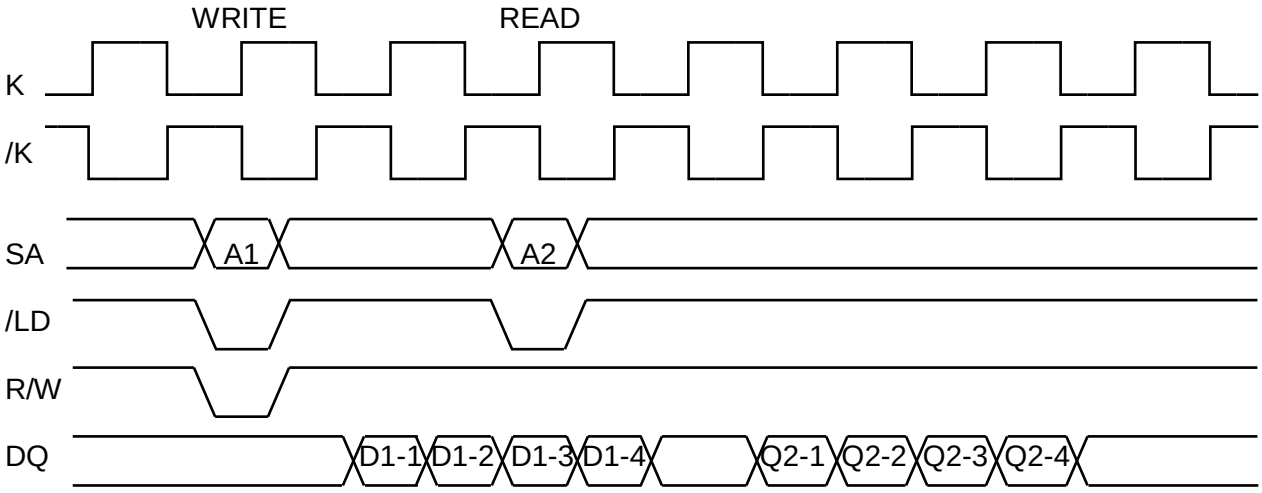
DDRII CIO SRAM has a common data input/output bus, and DDR interface on read and write ports.

Address, data inputs, and all control signals are synchronized to the input clock (K or /K). Normally data outputs are synchronized to output clocks (C and /C), but when C and /C are tied high, the data outputs are synchronized to the input clocks (K and /K). Read data are referenced to echo clock (CQ or /CQ) outputs, Read address and write address are registered on rising edges of the input K clocks. Common address bus is used to access address both for read and write operations.

The internal burst counter is fixed to 4-bit sequential for both read and write operations. Simple depth expansion is accomplished by using LD for port selection.

Byte write operation is supported with BW0 and BW1 (BW2 and BW3) pins for x18 (x36) device.

Timing Waveforms



6.4. DDRII SRAM SIO

K	/LD	R/W	D		Q		OPERATION
			D(A0)	D(A1)	Q(A0)	Q(A1)	
Stopped	X	X	Previous state	Previous state	Previous state	Previous state	Clock stop
↑	H	X	X	X	High-Z	High-Z	No operation
↑	L	H	X	X	QOUT at /C(t+1)	QOUT at C(t+2)	Read
↑	L	L	QIN at K(t+1)	QIN at /K(t+1)	High-Z	High-Z	Write

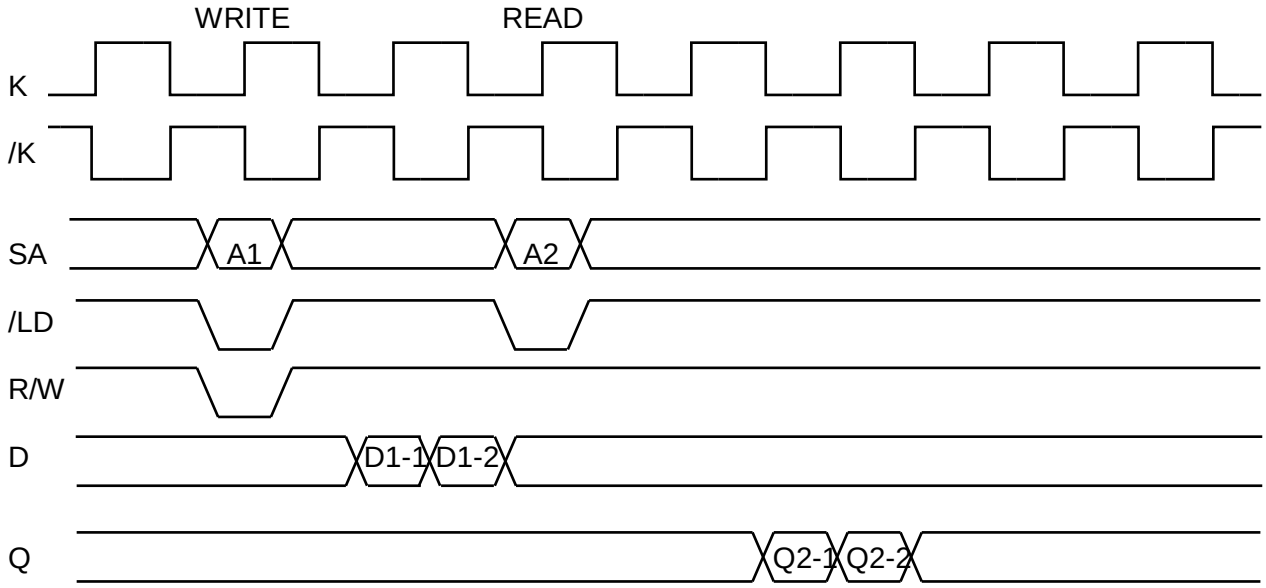
The DDRII SIO SRAM has independent read and write data ports, and DDR interface on read and write ports.

Memory bandwidth is higher than DDR SRAM without separate input output as separate read and write ports eliminate bus turn around cycle. Address, data inputs, and all control signals are synchronized to the input clock (K or /K). Normally data outputs are synchronized to output clocks (C and /C), but when C and /C are tied high, the data outputs are synchronized to the input clocks (K and /K). Read data are referenced to echo clock (CQ or /CQ) outputs, Read address and write address are registered on rising edges of the input K clocks.Common address bus is used to access address both for read and write operations.

The internal burst counter is fixed to 2-bit sequential for both read and write operations. Simple depth expansion is accomplished by using LD for port selection.

Byte write operation is supported with BW0 and BW1 (BW2 and BW3) pins for x18 (x36) device.

Timing Waveforms



6.5. DDRII+ SRAM

K	/LD	R/W	DQ		OPERATION
			DQ(A1)	DQ(A2)	
Stopped	X	X	Previous state	Previous state	Clock stop
↑	H	X	High-Z	High-Z	No operation
↑	L	H	QOUT at C(t+2)	QOUT at /C(t+2)	Read
↑	L	L	QIN at K(t+1)	QIN at /K(t+1)	Write

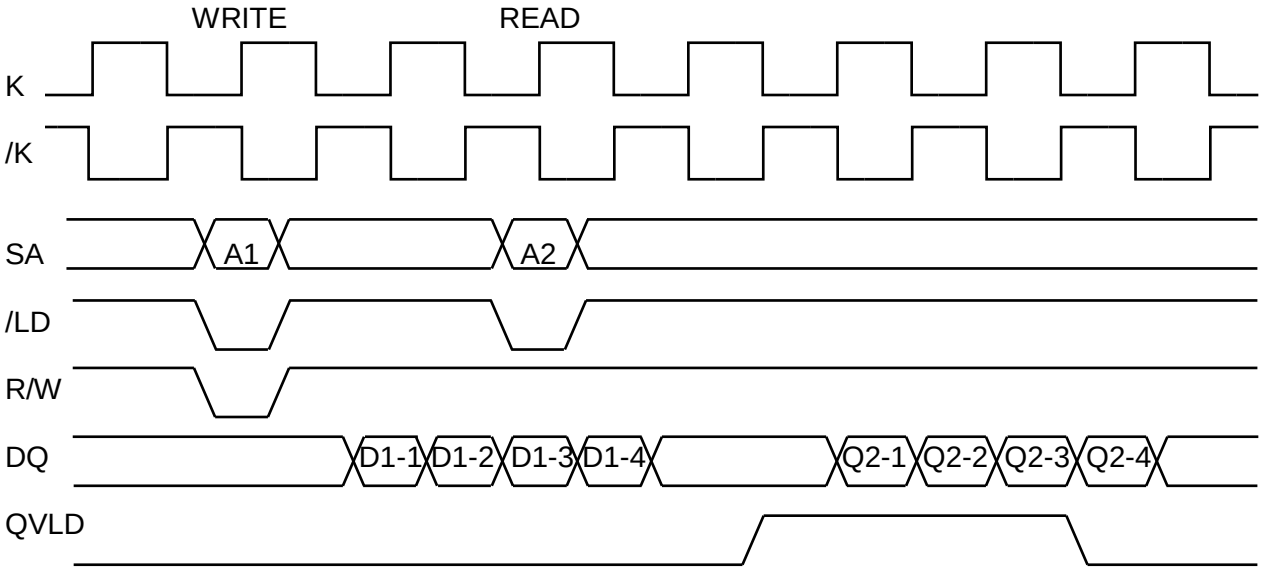
DDRII+ SRAM has a common data input/output bus, and DDR interface on read and write ports.

Address, data inputs, and all control signals are synchronized to the input clock (K or /K). Read data are referenced to echo clock (CQ or /CQ) outputs. Read address and write address are registered on rising edges of the input K clocks. Common address bus is used to access address both for read and write operations.

The internal burst counter is fixed to 2-bit sequential for both read and write operations. Simple depth expansion is accomplished by using LD for port selection.

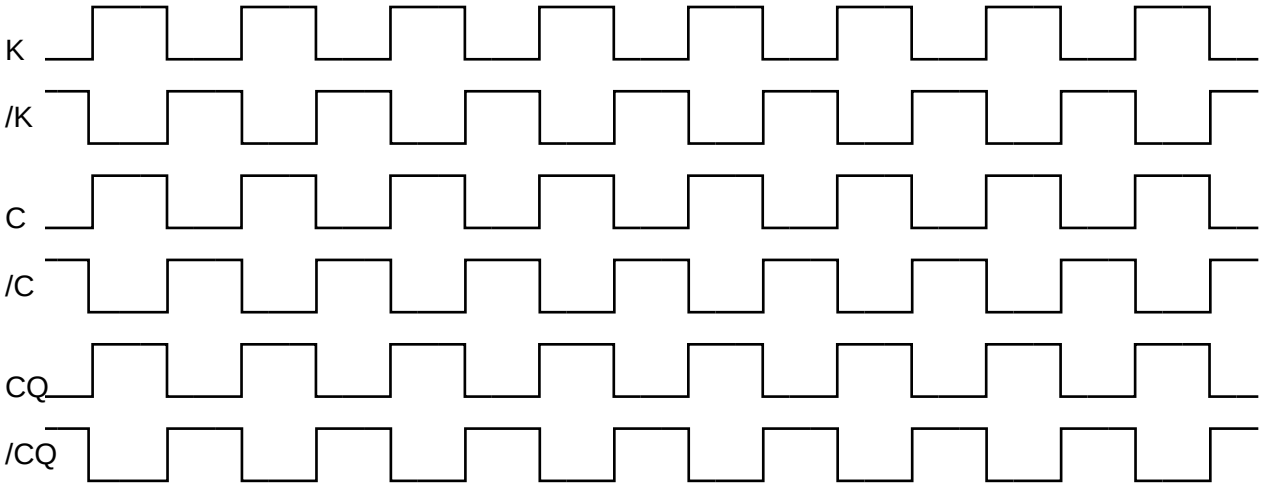
Byte write operation is supported with BW0 and BW1 (BW2 and BW3) pins for x18 (x36) device.

Timing Waveforms



7. Clocks

For DDR/DDRII/DDRII+ models provided by Cadence for Palladium, We usually don't care about the time delay and the internal DLL for phase adjustment, so the K, C and CQ signals are exactly the same, so do the /K , /C and /CQ.



8. MMP and ECC (Error Correcting Code)

MMP models do not support Error Correcting Code (ECC) functionality. ECC functions, if they are present in a memory device, are typically found in the NAND and DDRx families. The MMP product does not have any plans to provide such functions in the models. MMP models are provided as system level emulation models and not as verification IP. The below sections discuss work-arounds that enable the user to deal with some ECC scenarios. Note that ECC means different things to different device families.

8.1. DDRx and ECC (Error Correcting Code)

Error Correction Code (ECC) allows single bit errors to be corrected and other bit errors to be detected, thus improving high frequency operation reliability and data accuracy.
While the MMP DDRx models do not include any ECC functionality or handling, the user can arrange to "support" ECC on the DDR model first by using a model with a 72bit datapath and, additionally, by artificially injecting errors using some external mechanism, if such is needed. To word this differently, MMP models can often be found that interface with the memory controller data path, thus satisfying connectivity requirements. This arrangement might support a user who needs to test ECC related error conditions in emulation. The following paragraphs provide some details for consideration.

For DDRx models, ECC is performed by the controller. The controller calculates an ECC value and writes it to the upper 8 bits of a 72bit DIMM. When the controller writes data or when it reads data, it re-calculates the ECC based on current read data then checks the resulting value

against the read ECC stored in the upper 8 bits. If the re-calculated value is equivalent to the stored value, then there is no error.

So, to support ECC in an MMP model during acceleration or emulation, the user first needs a 72bit data path to provide the added ports and data path for handling the upper 8 bits of ECC above and beyond the standard 64bit data path. In other words, a normal DDR memory has a 64bit data bus where the ECC memory will have a 72bit data bus and 9 bits for DQS and DM. For most scenarios, ones where the user will not enable ECC function in their controller and will not inject errors, this operation is compatible for use with MMP models. The expanded ports and data path constitutes all of the support that MMP models can provide toward ECC handling.

There are two potential paths for achieving the 72bit data bus. The first is to select and use a 72bit DIMM. Please review the DIMM pages of the MMP catalogue for an appropriate 72bit DIMM. If an appropriate 72bit part cannot be found, the user may contact Cadence support to initiate a request for a 72bit data path version of the model of interest. A second approach to consider for some models is to expand the data path of a smaller width data bus to 72bits. The safest way to expand the data path to 72bits is to modify the data width parameter to 72 in the standard 64bit model. The user can set the parameter *data_bits* to 72 when instantiating the model. Not all models, however, have a *data_bits* parameter available for configuration.

Next, the user considers the error injection aspect. There is no capability in MMP models for error injection. MMP models are provided as system level emulation models and not as verification IP. For specifically testing error conditions, the user needs to work around the gap. One alternative that the user might consider is to corrupt memory data in order to mimic data corruption. In this scenario, the user can execute the xeDebug memory command to write some incorrect data to the array. For example, the following command will corrupt one location in an array:

```
memory -setvalue ddr3_inst -value 0x01234567 -start 0x10000100 -end 0x10000100
```

Another alternative is to consider using MMAV/Denali simulation models in IUS.

9. Compile and Emulation

The model is provided as a protected RTL file(s) (*.vp). The file(s) need to be synthesized prior to the back-end Palladium compile. An example of the command for compilation (including synthesis) of this model in the IXCOM flow is shown below.

```
ixcom -ua +dut+k7d161874b \
    ./ k7d161874b.vp \
    -incdir ../../../utils/cdn_mmp_utils/sv \
    ../../../utils/cdn_mmp_utils/sv/cdn_mmp_utils.sv \
    ……

xeDebug -64 --ncsim \
    -sv_lib ../../../utils/cdn_mmp_utils/lib/64bit/libMMP_utils.so -- \
    -input auto_xedebug.tcl
```

The script below shows two example for Palladium classic ICE synthesis:

```
1)
hdlInputFile k7d161874b.vp
hdlImport -full -2001 -l qtref
hdlOutputFile -add -f verilog k7d161874b.vg
hdlSynthesize -memory -keepVhdlCase -keepRtlSymbol -keepAllFlipFlop
k7d161874b
……

2)
vavlog k7d161874b.vp

vaelab -keepRtlSymbol -keepAllFlipFlop -outputVlog k7d161874b.vg
k7d161874b
……
```

NOTE: It is common for Palladium flows to require –keepallFlipFlop since it removes optimizations that are in place by default. For example, without –keepAllFlipFlop, HDL-ICE can remove flops with constant inputs and merge equivalent FF. The picture above is modified a bit when ICE ATB mode (–atb) is used since then a constant input FF is only optimized out when there is no initial value for it or the initial value is the same as the constant input value.

It is also common for Palladium flows to require –keepRtlSymbol. This option enables the HDL Compiler to keep original VHDL RTL symbols, such as ".", whenever possible. In other words, it maps VHDL RTL signal name a.b to the netlist entry, \a.b. Without this modifier, the signal name would otherwise be converted to a_b in the netlist.

If the recommended compile script includes the aforementioned options, the user must include them to avoid affecting functionality of the design.

Revision History

The following table shows the revision history for this document

Date	Version	Revision
August 2010	1.0	Initial release of document
August 2011	1.1	Updated doc with “Related Documents” Info
July 2014	1.2	Updated legal; added revision table.
September 2014	1.3	Remove version from UG file name. Update UXE / IXE documentation reference titles.
November 2014	1.4	Remove emulation capacity info.
January 2015	1.5	Add MMP and ECC section. Update related publications list.
July 2015	1.6	Update Cadence naming on front page
September 2015	1.7	Add compile and emulation section
January 2016	1.8	Update user guide for Palladium-Z1 and VXE
July 2016	1.9	Remove hyphen in Palladium naming
January 2018	2.0	Modify header and footer
July 2018	2.1	Update for new utility library