SigmaDDR-IIIe DQ ODT Control

Background

SigmaQuad-IIIe and SigmaDDR-IIIe ECCRAMs utilize differential amplifiers to receive all high-speed input signals, where one input of the diff-amp is connected to the signal itself and the other input is connected to a reference voltage (V_{REF}) typically equal to half the I/O voltage (V_{DDQ}); that is, $V_{REF} = V_{DDQ}/2$. These ECCRAMs also implement symmetric pull-up (to V_{DDQ}) and pull-down (to V_{SS}) on-die termination (ODT) on all high-speed input and I/O signals. Therefore, when the ODT is enabled it will pull an undriven/floating input to $V_{DDQ}/2 = V_{REF}$. This situation should be avoided, because if both inputs of the diff-amp are driven/ pulled to the same voltage (in this case, $V_{DDQ}/2$), the diff-amp can enter a metastable state and consume significantly more current than it would otherwise.

For terminated input-only (i.e. uni-directional) signals, this situation is easily avoided by having the ECCRAM Controller (i.e. ASIC, FPGA, etc.), henceforth referred to simply as the "Controller", keep its corresponding output drivers enabled and driving the signals Low or High at all times.

However, for terminated I/O (i.e. bi-directional) signals, this situation is not so easily avoided, because the Controller must disable its corresponding output drivers when it uses the signals to receive information from the ECCRAM.

The only terminated I/O signals implemented in SigmaQuad-IIIe and SigmaDDR-IIIe ECCRAMs are the data I/O signals (DQs) in SigmaDDR-IIIe devices. These signals function as inputs when the ECCRAM is receiving (write) data from the Controller, and they function as outputs when the ECCRAM is sending (read) data to the Controller.

For the purposes of this document, it is assumed that, like the ECCRAM, the Controller also implements symmetric pull-up and pull-down ODT on its DQs. Consequently:

- The ECCRAM must *enable* its DQ ODT and *disable* its DQ drivers, and the Controller must *disable* its DQ ODT and *enable* its DQ drivers, when the Controller is sending (write) data to the ECCRAM.
- The ECCRAM must *disable* its DQ ODT and *enable* its DQ drivers, and the Controller must *enable* its DQ ODT and *disable* its DQ drivers, when the Controller is receiving (read) data from the ECCRAM.

A methodology has been developed for these SigmaDDR-IIIe ECCRAMs (and for the Controllers designed to operate them) to ensure that the DQ bus is never pulled to $V_{DDQ}/2 = V_{REF}$, neither by the ECCRAM ODT nor by the Controller ODT, during Read-to-Write and Write-to-Read transitions.

The fundamental concept of the methodology is - both the ECCRAM and the Controller drive the DQ bus Low (with their respective DQ ODT disabled) at all times except:

1. When a particular device is driving the DQ bus with valid data, and
2. From shortly before to shortly after a particular device is receiving valid data on the DQ bus, during which time the receiving device enables its DQ ODT.

And, during Write-to-Read and Read-to-Write transitions, each device enables and disables its DQ ODT while the other device is driving DQ Low, thereby ensuring that the DQ bus is never pulled to $V_{DDQ}/2$.

In order for this methodology to work as described, the Controller must have the ability to:

1. Place the ECCRAM into “DQ Drive Low Mode” at the appropriate times (i.e. before and after the ECCRAM drives valid Read Data), and
2. Place the ECCRAM into “DQ ODT Mode” at the appropriate times (i.e. before, during, and after the ECCRAM receives valid Write Data).

That ability is provided via the existing R/$\overline{\text{W}}$ control pin.

When the ECCRAM samples R/$\overline{\text{W}}$ High (regardless of the state of the $\overline{\text{LD}}$ control pin), it disables its DQ termination, and drives the DQ bus Low except while driving valid Read Data in response to Read operations.

When the ECCRAM samples R/$\overline{\text{W}}$ Low (regardless of the state of $\overline{\text{LD}}$), it disables its DQ drivers, and enables its DQ termination.

Note that NOPs initiated with R/$\overline{\text{W}}$ High and $\overline{\text{LD}}$ High are referred to as “NOPr” operations.

Note that NOPs initiated with R/$\overline{\text{W}}$ Low and $\overline{\text{LD}}$ High are referred to as “NOPw” operations.

This extended definition of the R/$\overline{\text{W}}$ control pin enables the Controller to:

- Place the ECCRAM in DQ ODT Mode, via NOPw operations, before initiating Write operations.
- Keep the ECCRAM in DQ ODT Mode, via NOPw operations, after initiating Write operations.
- Place the ECCRAM in DQ Drive Low Mode, via NOPr operations, before initiating Read operations.
- Keep the ECCRAM in DQ Drive Low Mode, via NOPr operations, after initiating Read operations.

DQ Truth Table - Read Latency (RL) = 2 cycles

$\overline{\text{LD}}$	R/$\overline{\text{W}}$	Current Operation	DQ State
↑CK (t_n)	↑CK (t_n)	(t_n)	↑CK (t_{n+2})
1	0	NOPw	ODT Enabled
0	0	Write	ODT Enabled
1	1	NOPr	ODT Disabled, Drive Low
0	1	Read	ODT Disabled, Drive Read Data

DQ Truth Table - Read Latency (RL) = 3 cycles

$\overline{\text{LD}}$	R/$\overline{\text{W}}$	Current Operation	DQ State	
↑CK (t_n)	↑CK (t_n)	(t_n)	↑CK (t_{n+2})	↑CK (t_{n+3})
1	0	NOPw	ODT Enabled	---
0	0	Write	ODT Enabled	---
1	1	NOPr	ODT Disabled, Drive Low	---
0	1	Read	ODT Disabled, Drive Low	Drive Read Data

Note: For RL=3 case, when a Read operation is initiated in cycle “n”, R/$\overline{\text{W}}$ must be “High” at ↑CK of cycle “n+1” (i.e. a Read operation must always be followed by a Read or NOPr operation). In that case, the DQ state in cycle “n+3” is “Drive Read Data”, as indicated in the table above.

Controlling Write-to-Read and Read-to-Write Transitions

The Write-to-Read and Read-to-Write transition requirements outlined below are a function of several factors:

- ECCRAM Cycle Time (tCYC). tCYC varies by application.
- ECCRAM Write Latency (WL) - The number of cycles from Write Command input to (first) Write Data input.
  WL is fixed to 1 cycle in SigmaDDR-IIIe.
- ECCRAM Read Latency (RL) - The number of cycles from Read Command input to (first) Read Data output.
  RL is programmable to 2 or 3 cycles in SigmaDDR-IIIe.
- ECCRAM DQ ODT Enable/Disable Latency (ZTL) - The number of cycles from Write/NOPw Command input to DQ ODT enabled, and the number of cycles from Read/NOPr Command input to DQ ODT disabled.
  ZTL is fixed to 2 cycles in SigmaDDR-IIIe.
- ECCRAM Input Clock to DQ State Transition (tKQ) - In the cycle in which a DQ State Transition occurs, it is the amount of time from ↑CK to DQ Driver Enable/Disable/Low->High Transition/High->Low Transition, and from ↑CK to DQ ODT Enable/Disable.
  tKQ equals -0.4ns / 0.0ns / +0.4ns min/typ/max in SigmaDDR-IIIe.
- Propagation Delay between Controller and ECCRAM (tPD). tPD varies by application.

Write

1. The Controller initiates one or more Write operations.

   Prior to Step 1 (see Step 6 below for how this occurs):

   - The Controller is driving DQ Low.
   - The ECCRAM is in DQ ODT Mode.

   During Step 1:

   - The Controller stops driving DQ Low and begins driving valid Write Data 0.75 cycles after initiating the first Write operation, in order to meet the ECCRAM WL requirements.

Write-to-Read Transition

Immediately after initiating the Write operations, the Controller could initiate one or more NOPw operations to keep the ECCRAM in DQ ODT Mode for some amount of time after the Controller stops driving valid Write Data. However, it is not necessary, because ECCRAM ZTL (2 cycles) is always greater than WL (1 cycle). If the Controller initiates a NOPr or Read operation immediately after a Write operation, the ECCRAM will remain in DQ ODT Mode for at least “1.25*tCYC + tKQ.min” after the Controller stops driving valid Write Data (calculated at the ECCRAM pins). Consequently, no intervening NOPw operations are necessary.

2. The Controller initiates NOPr operations, to put the ECCRAM in DQ Drive Low Mode some amount of time before it begins driving valid Read Data.

   During Step 2:

   - The Controller stops driving valid Write Data and begins driving DQ Low 0.75 cycles after initiating the first NOPr operation (or after initiating the first Read operation, if no NOPr are initiated in Step 2). The Controller drives DQ Low in order to prevent the ECCRAM DQ ODT from pulling the DQ bus to $V_{DDQ}/2$.
   - The ECCRAM disables its DQ ODT and begins driving DQ Low ZTL = 2 cycles after detecting the first NOPr operation. The ECCRAM drives DQ Low in order to prevent the Controller DQ ODT (when it is enabled in Step 3) from pulling the DQ bus to $V_{DDQ}/2$.

   Typically, the ECCRAM should be put in DQ Drive Low Mode “m” = 1~3 cycles before it begins driving valid Read Data.

   In order to put the ECCRAM in DQ Drive Low Mode “m” cycles before it begins driving valid Read Data, the Controller must initiate “n” = “m + ZTL - RL” = “m + 2 - RL” NOPr operations in Step 2. For example, if m = 2 and RL = 3, then n = 1.

The integer value of “m”, and therefore the number “n” of NOPr operations needed in Step 2, is a function of tCYC, tKQ, and tPD. Specifically, “m” must be > 0.5 + ((tKQ.max + 2*tPD) / tCYC) (and therefore “m” must be ≥ 1).

This ensures that the Controller stops driving DQ Low and enables its DQ ODT (as described in Step 3 below) *after* the ECCRAM disables its DQ ODT and begins driving DQ Low, and *before* the ECCRAM stops driving DQ Low and begins driving valid Read Data. The former is required because if the Controller were to enable its DQ ODT *before* the ECCRAM disables its DQ ODT, both devices would be enabling their DQ ODT simultaneously, pulling the DQ bus to $V_{DDQ}/2$.

Note: SigmaDDR-IIIe devices *must* be controlled such that they drive DQ Low for at least one cycle before driving valid Read Data. When RL = 2, the equation “n” = “m + ZTL - RL” results in n ≥ 1 when (as required) m ≥ 1, and n ≥ 1 ensures that the ECCRAM drives DQ Low for at least one cycle before driving valid Read Data, meeting the requirement. When RL = 3, the equation “n” = “m + ZTL - RL” results in n ≥ 0 when (as required) m ≥ 1. But, the ECCRAM will always drive DQ Low for RL - ZTL = 3 - 2 = 1 cycle before driving valid Read Data, meeting the requirement even when n = 0.

Read

3. The Controller initiates one or more Read operations.

   During Step 3:

   - The Controller stops driving DQ Low and enables its DQ ODT “RL - 0.5” cycles after initiating the first Read operation. That equates to at least “(m - 0.5)*tCYC - (tKQ.max + 2*tPD)” after the ECCRAM enters DQ Drive Low Mode (calculated at the Controller pins), where “m” is the value calculated in Step 2. And, that equates to at least “0.5*tCYC + tKQ.min” before the ECCRAM begins driving valid Read Data (calculated at the ECCRAM pins).
   - When RL=2, the ECCRAM stops driving DQ Low and begins driving valid Read Data RL = 2 cycles after detecting the first Read operation.
   - When RL=3, the ECCRAM either continues to disable its DQ ODT and drive DQ Low (if one or more NOPr were initiated in Step 2), or disables its DQ ODT and begins driving DQ Low (if no NOPr were initiated in Step 2), ZTL = 2 cycles after detecting the first Read operation. Subsequently, the ECCRAM stops driving DQ Low and begins driving valid Read Data RL = 3 cycles after detecting the first Read operation.

Read-to-Write Transition

4. The Controller initiates “RL - ZTL” = “RL - 2” NOPr operations, to ensure that the ECCRAM drives valid Read Data properly in response to the last Read operation initiated in Step 3.
5. The Controller initiates more NOPr operations, to put the ECCRAM in DQ Drive Low Mode for some amount of time after it stops driving valid Read Data.

   During Step 5:

   - The ECCRAM stops driving valid Read Data and begins driving DQ Low ZTL = 2 cycles after detecting the first NOPr operation. The ECCRAM drives DQ Low in order to prevent the Controller DQ ODT from pulling the DQ bus to $V_{DDQ}/2$.

   Typically, the ECCRAM should be put in DQ Drive Low Mode for “p” = 1~3 cycles after it stops driving valid Read Data.

   In order to put the ECCRAM in DQ Drive Low Mode for “p” cycles after it stops driving valid Read Data, the Controller must initiate “q” = “p” NOPr operations in Step 5.

   The integer value of “p”, and therefore the number “q” of NOPr operations needed in Step 5, is a function of tCYC, tKQ, and tPD. Specifically, “p” must be > 0.5 + ((tKQ.max + 2*tPD) / tCYC) (and therefore “p” must be ≥ 1).

   This ensures that the Controller disables its DQ ODT and begins driving DQ low (as described in Step 6 below) *after* the ECCRAM stops driving valid Read Data and begins driving DQ Low, and *before* the ECCRAM stops driving DQ Low and enables its DQ ODT. The latter is required because if the Controller were to disable its DQ ODT *after* the ECCRAM enables its DQ ODT, both devices would be enabling their DQ ODT simultaneously, pulling the DQ bus to $V_{DDQ}/2$.

   Note: SigmaDDR-IIIe devices *must* be controlled such that they drive DQ Low for at least one cycle after driving valid Read Data. This requirement is met because q = p, and (as required) p ≥ 1.

6. The Controller initiates NOPw operations, to put the ECCRAM in DQ ODT Mode some amount of time before the Controller begins driving valid Write Data.

   During Step 6:

   - The Controller disables its DQ ODT and begins driving DQ Low 1.5 cycles after initiating the first NOPw operation. That equates to at least “(p - 0.5)*tCYC - (tKQ.max + 2*tPD)” after ECCRAM enters DQ Drive Low Mode (calculated at the Controller pins), where “p” is the value calculated in Step 5. And that equates to at least “0.5*tCYC + tKQ.min” before the ECCRAM enters DQ ODT Mode (calculated at the ECCRAM pins). The Controller drives DQ Low in order to prevent the ECCRAM DQ ODT from pulling the DQ bus to $V_{DDQ}/2$.
   - The ECCRAM stops driving DQ Low and enables its DQ ODT ZTL = 2 cycles after detecting the first NOPw operation.

   Typically, the ECCRAM should be put in DQ ODT Mode 0.75~2.75 cycles before the Controller begins driving valid Write Data.

   In order to put the ECCRAM in DQ ODT Mode “r - 0.25” cycles before the Controller begins driving valid Write Data, the Controller must initiate “s” = “r + 1” NOPw operations in Step 6. For example, if r = 2, s = 2 + 1 = 3.

   The integer value of “r”, and therefore the number “s” of NOPw operations needed in Step 6, is a function of tCYC, tKQ, and tPD. Specifically, “r” must be > 0.25 + ((tKQ.max + 2*tPD) / tCYC) (and therefore “r” must be ≥ 1).

   This ensures that the Controller stops driving DQ Low and begins driving valid Write Data (as described in Step 7 below) *after* the ECCRAM stops driving DQ Low and enables its DQ ODT.

Write

7. The Controller initiates one or more Write operations.

   During Step 7:

   - The Controller stops driving DQ Low and begins driving valid Write Data 0.75 cycles after initiating the first Write operation. That equates to at least “(r - 0.25)*tCYC - (tKQ.max + 2*tPD)” after the ECCRAM enters DQ ODT Mode (calculated at the Controller), where “r” is the value calculated in Step 6.

Advantages Over Conventional Implementations

In conventional SRAMs (specifically, in DDR-II+ devices) that implement ODT on bidirectional Data I/O signals, the SRAM DQ ODT is always enabled, except from 0.5 cycles before the SRAM begins driving valid Read Data to 0.5 cycles after the SRAM stops driving valid Read Data. During most of the time SRAM DQ ODT is enabled (i.e., during non-Read operations), the Controller can drive DQ Low or High in order to prevent the SRAM ODT from pulling the DQ bus to $V_{DDQ}/2 = V_{REF}$.

However, during non-Read to Read transitions, the Controller must stop driving DQ Low or High and enable its DQ ODT before receiving the Read Data. If the Controller does this *before* the SRAM disables its ODT, then the DQ bus will be pulled to $V_{DDQ}/2 = V_{REF}$ by both devices' ODT until the SRAM disables its ODT, and will continue to be pulled to $V_{DDQ}/2 = V_{REF}$ by the Controller's ODT until the SRAM begins driving valid Read Data. And if the Controller does this *after* the SRAM disables its ODT (difficult to implement, since the window in which this could occur is <0.5 cycles), then the DQ bus will be pulled to $V_{DDQ}/2 = V_{REF}$ by the Controller's ODT until the SRAM begins driving valid Read Data. Either way, the DQ bus will be pulled to $V_{DDQ}/2 = V_{REF}$ for some amount of time before the SRAM begins driving Read Data.

For similar reasons, the DQ bus will be pulled to $V_{DDQ}/2 = V_{REF}$ for some amount of time during Read to non-Read transitions, after the SRAM stops driving valid Read Data, as well.

The primary advantage of the SigmaDDR-IIIe DQ ODT Control methodology (and the reason it was developed in the first place) is that it prevents the DQ bus from being pulled to $V_{DDQ}/2 = V_{REF}$ at ALL times, including during non-Read to Read, and Read to non-Read, transitions.

A secondary advantage of the SigmaDDR-IIIe DQ ODT Control methodology is that it reduces power consumption compared to the conventional SRAM methodology. The reason this is true is that, in the SigmaDDR-IIIe methodology, the "idle/NOP" state (i.e. when no Reads or Writes are occurring) of both the ECCRAM and the Controller is "DQ Drive Low". In which case there will be no DC current through either device's DQs during this time. Whereas in the conventional SRAM methodology, the "idle/NOP" state of the Controller is "DQ Drive Low or High" and the "idle/NOP" state of the SRAM is "DQ ODT". In which case there will be DC current through both device's DQs during this time.

Timing Diagrams

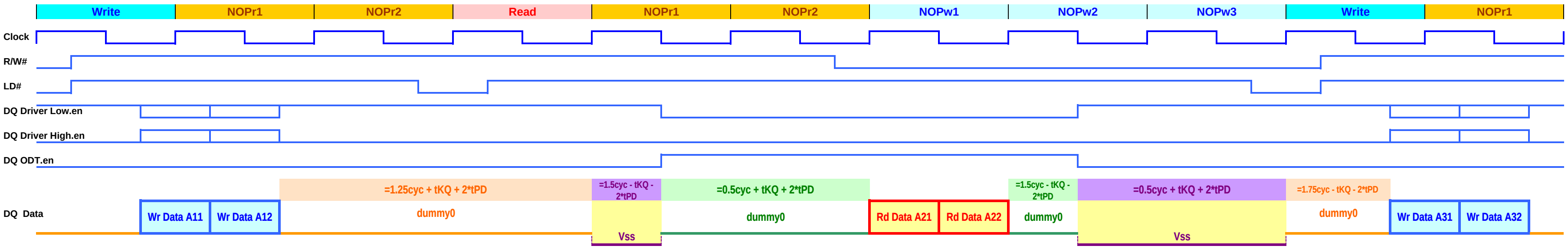
The following page shows timing diagrams for SigmaDDR-IIIe, for both RL = 2 and RL = 3. The diagrams depict signals at both the Controller and ECCRAM pins, to help illustrate why the methodology has been defined in the manner described.

In the timing diagrams:

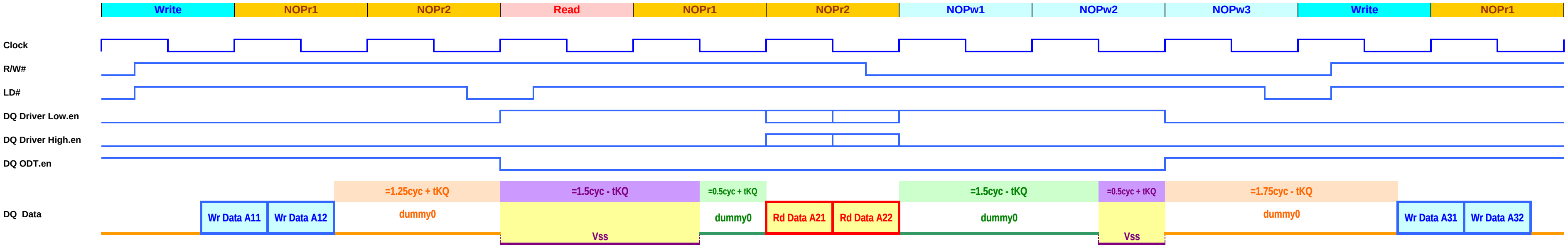
- "DQ Driver Low.en" is the internal signal in the Controller or ECCRAM that enables/disables its DQ Driver pull-down transistors ("1" = enable).
- "DQ Driver High.en" is the internal signal in the Controller or ECCRAM that enables/disables its DQ Driver pull-up transistors ("1" = enable).
- "DQ ODT.en" is the internal signal in the Controller or ECCRAM that enables/disables its DQ ODT pull-down and pull-up transistors ("1" = enable).
- The "dummy0" states shown on the DQ bus before and after Read Data are depicted as a green line at a Low level greater than Vss, and equal to that resulting from the ECCRAM DQ Drivers driving Low into the Controller DQ ODT.
- The "dummy0" states shown on the DQ bus before and after Write Data are depicted as an orange line at a Low level greater than Vss, and equal to that resulting from the Controller DQ Drivers driving Low into the ECCRAM DQ ODT.
- The "Vss" states shown on the DQ bus between "dummy0" states are depicted as a purple line at a Low level equal to Vss, and resulting from both the ECCRAM and Controller DQ Drivers driving Low simultaneously.

SigmaDDR-IIIe. RL = 2. tKQ = 0ns. tPD = 0.5 cycles.

Signals at Controller

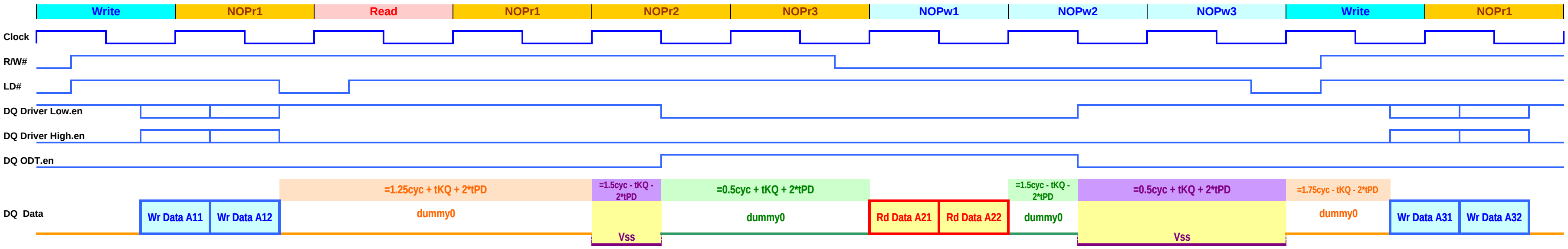


Signals at ECCRAM

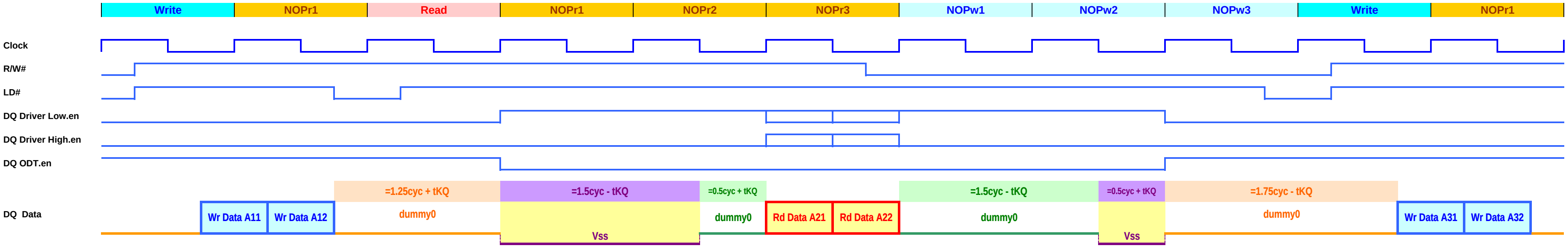


SigmaDDR-IIIe. RL = 3. tKQ = 0ns. tPD = 0.5 cycles.

Signals at Controller



Signals at ECCRAM



Write
NOPr1
Read
NOPr1
NOPr2
NOPr3
NOPw1
NOPw2
NOPw3
Write
NOPr1
Clock
R/W#
LD#
DQ Driver Low.en
DQ Driver High.en
DQ ODT.en
DQ Data
Wr Data A11
Wr Data A12
=1.25cyc + tKQ
dummy0
=1.5cyc - tKQ
Vss
=0.5cyc + tKQ
dummy0
Rd Data A21
Rd Data A22
=1.5cyc - tKQ
dummy0
=0.5cyc + tKQ
Vss
=1.75cyc - tKQ
dummy0
Wr Data A31
Wr Data A32